CAIAC

INTERNATIONAL CUMHURIYET ARTIFICIAL INTELLIGENCE APPLICATIONS CONFERENCE 2025

Sep 25-26 | ONLINE

http://caiac.cumhuriyet.edu.tr

Proceedings Book

Edited by

- Dr. Emre Delibaş
- Dr. Abdulkadir Şeker
- Dr. Hakan Kekül

CUMHURİYET ÜNİVERSİTESİ 1974

SİVAS CUMHURİYET ÜNİVERSİTESİ YAPAY ZEKA VE VERİ BİLİMİ UYGULAMA VE ARAŞTIRMA MERKEZİ

SİVAS CUMHURİYET ÜNİVERSİTESİ

The Proceedings of the International Cumhuriyet Artificial Intelligence Applications Conference 2025"
CAIAC'25

ISBN
978-625-6497-69-6

Editörler
Dr. Öğr. Üyesi Emre Delibaş
Dr. Öğr. Üyesi Abdulkadir Şeker
Dr. Öğr. Üyesi Hakan Kekül

Kapak ve İç Düzen
Abdulkadir Kocatürk

Baskı
Sivas Cumhuriyet Üniversitesi Rektörlük Matbaa
Sertifika No: 40954

Dağıtım
Sivas Cumhuriyet Üniversitesi

Sivas / 2025

SİVAS CUMHURİYET ÜNİVERSİTESİ YAYINLARI No: 332

14/10/2025 Tarih ve 16 Numaralı Sivas Cumhuriyet Üniversitesi Yayın Kurulu Kararı ile 15/10/2025 Tarih ve 33 Numaralı Sivas Cumhuriyet Üniversitesi Yönetim Kurulu Kararına istinaden basımı uygun görülmüştür.

CONTENTS

CONFERENCE CHAIR

Dr. Emre Delibaş	Sivas Cumhuriyet University

ORGANIZATION COMMITTEE

Dr. Ahmet Gürkan Yüksek	Sivas Cumhuriyet University
Dr. Emre Delibaş	Sivas Cumhuriyet University
Dr. Serkan Akkoyun	Sivas Cumhuriyet University
Dr. Abdulkadir Şeker	Sivas Cumhuriyet University
Dr. Hakan Kekül	Sivas Cumhuriyet University

SCIENTIFIC COMMITTEE

Dr. Alpaslan Fığlalı	Kocaeli University
Dr. Andrew Kusiak	The University of Iowa
Dr. Bahriye Akay	Erciyes University
Dr. Banu Diri	Yıldız Technical University
Dr. Celal Öztürk	Erciyes University
Dr. Çetin Elmas	Azerbaijan Technical University
Dr. Çiğdem Erol	Istanbul University
Dr. Ecir Uğur Küçüksille	Süleyman Demirel University
Dr. Emre Dandıl	Bilecik Seyh Edebali University
Dr. Emre Delibaş	Sivas Cumhuriyet University
Dr. Emre Ünsal	Sivas Cumhuriyet University
Dr. Eyüp Çalık	Yalova University
Dr. Ferdi Sönmez	Fenerbahçe University
Dr. Ferhat Sayım	Yalova University
Dr. Feriştah Dalkılıç	Ege University
Dr. Güzin Ulutaş	Karadeniz Technical University
Dr. Halil Arslan	Sivas Cumhuriyet University
Dr. Hakan Kekül	Sivas Cumhuriyet University
Dr. Haldun Akpınar	Marmara University
Dr. Harun Uğuz	Konya Technical University
Dr. İhsan Hakan Selvi	Sakarya University
Dr. Konstantin P. Katin	National Research Nuclear University MEPhI
Dr. Kali Gürkahraman	Sivas Cumhuriyet University
Dr. Kemal Adem	Sivas Cumhuriyet University
Dr. Manafəddin Namazov	Baku Engineering University
Dr. Mehmet Ali Alan	Sivas Cumhuriyet University
Dr. Mehmet Göktürk	Gebze Technical University
Dr. Metin Zontul	Sivas University of Science and Technology
Dr. Metin Saygılı	Sakarya University of Applied Sciences
Dr. Muhammed Kürşad Uçar	Sakarya University

Dr. Murat Şeker	Gebze Technical University
Dr. Naima Amrani	Ferhat Abbas University of Setif
Dr. Naveed Muhammad	University of Tartu
Dr. Oğuz Kaynar	Sivas Cumhuriyet University
Dr. Osman Nuri Şahin	Alanya Alaaddin Keykubat University
Dr. Özlem Polat	Sivas Cumhuriyet University
Dr. Paul Stevenson	University of Surrey
Dr. Ramzi Maalej	Sfax University
Dr. Rihab Gargouri	Sfax University
Dr. Selçuk Ökdem	Erciyes University
Dr. Süleyman Eken	Kocaeli University
Dr. Şaban Gülcü	Necmettin Erbakan University
Dr. Şadi Evren Şeker	Antalya Bilim University
Dr. Vasif Nabiyev	Karadeniz Technical University
Dr. Volkan Göreke	Sivas Cumhuriyet University
Dr. Yasin Görmez	Sivas Cumhuriyet University
Dr. Yılmaz Atay	Gazi University
Dr. Yunus Doğan	Ege University
Dr. Yusuf Sinan Akgül	Gebze Technical University

KEYNOTE SPEAKER

Prof. Dr. Ramazan Katırcı — Sivas University of Science and Technology

"Quantum Machine Learning"

WELCOME TO CAIAC'25

We are delighted to present the Proceedings of the *International Cumhuriyet Artificial Intelligence Applications Conference 2025 (CAIAC'2025)*, held online between September 25–26, 2025.

Since its inaugural edition in 2021, CAIAC has evolved into a distinguished scientific platform for researchers, practitioners, and academicians working in the dynamic and rapidly expanding fields of Artificial Intelligence and Soft Computing. This year's conference continued that tradition by offering a venue for theoretical insights, practical innovations, and interdisciplinary dialogue.

Organized by the Artificial Intelligence and Data Science Application and Research Center at Sivas Cumhuriyet University, CAIAC'2025 aimed to foster collaboration between academia and industry, and to provide participants with opportunities to explore the latest research trends, methodologies, and applications in artificial intelligence. The enthusiastic participation of scholars from various institutions and countries enriched the scientific discourse and reinforced the relevance of AI research in addressing contemporary challenges.

We extend our sincere gratitude to all authors, reviewers, keynote speakers, and participants for their valuable contributions to this year's conference. We believe that the papers presented in this volume reflect the depth and breadth of current AI research and will serve as a valuable reference for future studies.

We hope that this collection inspires continued collaboration and innovation in the years to come.

Asst. Prof. Emre Delibaş
Conference Chair

Text Classification-Based Content Recommender System for Turkish News: A Multi-Level Experimental Study

Hakan Bulduk[1], Murat Işık[2], Muhammet Yasin Pak[3]

1 *Department of Computer Engineering, Gaziantep Islam Science and Technology University, Gaziantep, Türkiye, buldukhakan82@gmail.com*

2 *Department of Computer Engineering, Gaziantep Islam Science and Technology University, Gaziantep, Türkiye, muratisik079@gmail.com*

3 *Department of Computer Engineering, Gaziantep Islam Science and Technology University, Gaziantep, Türkiye, muhammetyasin.pak@gibtu.edu.tr*

ABSTRACT

With the widespread use of the internet and the advancement of digital platforms, users can now access online content much faster and more easily. Amidst this abundance of content, delivering information aligned with users' interests has increased the importance of personalized recommender systems. This study presents a text classification-based recommender system aimed at classifying Turkish news content as "relevant" or "irrelevant" from the user's perspective. The components of each news item at the title, summary, and full-text levels were analyzed separately. The collected data underwent preprocessing steps and was then classified using supervised machine learning algorithms. Each algorithm was evaluated in scenarios both with and without feature selection, and the impact of varying feature dimensions on classification performance was examined. Experimental results demonstrated that title-level content possesses a higher discriminative power compared to summaries and full texts. Comparisons between different users revealed the critical role of the data labeling process in determining the success of the recommender system. This study highlights the feasibility and applicability of content-based and classification-based recommender systems in the context of Turkish news, while also comprehensively assessing the effects of data diversity, stemming from different user profiles, on the performance of classification algorithms. The findings contribute to the academic literature and provide a solid foundation for the development of future commercial news recommender systems.

Keywords: *Content-Based Recommender Systems, Text Classification, Turkish News Recommendation*

INTRODUCTION

In the rapidly evolving digital age, advances in information technologies have made it easier for individuals to access information from various sources, yet the overwhelming abundance of data and content complexity have also made this process harder to manage. Especially in the case of news content, selecting and following news aligned with users' interests poses a significant challenge. With the transition of news dissemination from written, oral, and visual media to digital platforms, the interactive structures provided by internet technologies have increased the opportunities for personalization. In this context, personalized news delivery has emerged as a solution that enables users to access the most relevant content according to their individual preferences, while reducing information overload.

The tracking of user behavior, analysis of past preferences, and processing of textual content through artificial intelligence methods have laid the groundwork for the development of recommender systems. Recommender systems are information filtering systems designed to present users with content such as movies, music, products, or news [1]. They are generally based on two main approaches: collaborative filtering and content-based filtering. Collaborative filtering considers the behaviors of similar users, whereas content-based filtering generates recommendations by directly analyzing the features of the content.

Content-based recommender systems operate on the principle of recommending items that share similar characteristics with content previously rated positively by the user. In these systems, the recommendation process is carried out using the discrete features of the content (e.g., textual features). Typically, recommended items are selected from content assumed to

share common properties with those the user has liked before. In text-based content, features such as Term Frequency (TF) and Inverse Document Frequency (IDF) are commonly used for content representation. Based on these features, items can be represented in a multidimensional space using methods such as the vector space model or Latent Semantic Indexing [2]. For measuring similarities between items, metrics such as Cosine similarity or Jaccard similarity are frequently preferred [3, 4]. Another method, the text classification-based approach, involves classifying content using models trained with labeled data as "relevant" or "irrelevant," ensuring that only relevant content is recommended to the user.

Although there are many studies in the literature on text classification-based recommender systems developed for English news content [5], research on Turkish news in this context remains limited. Studies on Turkish news have generally focused on detecting similar news [6] or category classification [7]. However, no study in the literature has utilized Turkish news content labeled at the user level. This study aims to fill this gap by proposing an approach for a personalized news recommender system in the Turkish language. Furthermore, the news content is analyzed at multiple levels—not only the full text, but also the title and summary.

The goal of this study is to develop a content-based recommender system grounded in text classification that works on datasets of Turkish news content labeled by users. To achieve this, technology news content was collected and labeled by two different users, and after preprocessing and feature extraction steps, classification tasks were carried out using different machine learning algorithms.

MATERIALS AND METHODS

A. Dataset

The dataset was created by collecting news content from a technology news website in the form of titles, summaries, and full texts. The data was automatically retrieved using a custom-developed application. Based on the news titles, two different users individually labeled the news items as either "relevant" or "irrelevant." For the personalized recommender system, separate labeled datasets were prepared for each user. The dataset consists of the titles, summaries, and full-text sections of a total of 2,000 news articles.

B. Text Preprocessing

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

C. Feature Extraction and Selection

In this stage, features to be used as input for machine learning models were extracted from the text. The bag-of-words approach was employed, and word n-gram methods (unigram, bigram) were used. N-gram models are important tools for capturing language structure and context, making them highly effective for representing news texts [8, 9].
To select the most distinctive features from the high-dimensional feature space, the Information Gain method was applied. Information Gain measures each feature's contribution to classification accuracy [10, 11]. This approach aimed to obtain a low-dimensional yet effective feature set that could enhance system performance.

D. Classification

The selected features were transformed into numerical vectors using the TF-IDF method. This method calculates the weight of each word in the documents based on both its frequency within the document and its overall occurrence across all documents [12]. The resulting feature vectors were then used as input for different machine learning algorithms. The classification algorithms employed in this study are as follows:

Support Vector Machines (SVM): Aims to find the decision boundary (hyperplane) that best separates the classes in a high-dimensional feature space. To maximize the margin, only the support vectors located on the boundary are used. Thanks to kernel functions, data that is not linearly separable can be transformed into higher-dimensional spaces where it becomes linearly separable. This capability makes SVM highly effective for high-dimensional datasets such as text classification [13].

Random Forest (RF): An ensemble method composed of a large number of decision trees. Each tree is trained on a randomly selected subset of the training data and a random subset of the features. The final classification is determined by the majority vote of all trees. This random sub-sampling strategy reduces the risk of overfitting and provides strong generalization capabilities even with high-dimensional and noisy data [14].

Logistic Regression (LR): A widely used probabilistic linear classification method for binary problems. It estimates the probability of each instance belonging to a specific class by applying a logistic (sigmoid) function to the weighted sum of the input features. Due to its simple structure, interpretability, and fast training, it is often chosen as a strong baseline model for text classification tasks [15].

The classification process was conducted separately for the title, summary, and full-text levels of the news articles, and performance differences among these levels were analyzed comparatively.

E. Evaluation Metric

In this study, the accuracy metric was used to evaluate the performance of the classification models. Accuracy represents the proportion of total correct classifications (both true positives and true negatives) to all instances. It is mathematically defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP (True Positive): Correct classification of actual positive instances,
- TN (True Negative): Correct classification of actual negative instances,
- FP (False Positive): Negative instances incorrectly classified as positive,
- FN (False Negative): Positive instances incorrectly classified as negative.

EXPERIMENTAL WORKS

This section presents the experimental studies conducted to evaluate the performance of the developed text classification-based content-based recommender system. The experiments were carried out on datasets prepared at the title, summary, and full-text levels using different machine learning algorithms. In the classification process, Support Vector Machines (SVM), Random Forest (RF), and Logistic Regression (LR) algorithms were employed, each evaluated separately in scenarios with and without feature selection. In addition, the effect of different feature dimensions on classification performance was examined, and the models were compared using labeled data obtained from two different users.

Table 1 shows the accuracy values for User 1 obtained using the three different machine learning algorithms (SVM, RF, LR) without applying the feature selection step.

The title-level data produced the highest accuracy rates for all three algorithms. In this context, RF was the best-

TABLE I. Accuracy Results for User 1 Based on Title, Summary, and Content Without Feature Selection

	SVM	RF	LR
Title	0.838	0.851	0.849
Summary	0.743	0.746	0.707
Full-Text	0.752	0.713	0.735

TABLE II. Accuracy Results of SVM, RF, and LR Algorithms For User 1 With Feature Selection.

	SVM				
	250	500	1000	2500	5000
Title	0.756	0.764	0.780	0.818	0.827
Summary	0.683	0.703	0.707	0.727	0.734
Full-Text	0.692	0.690	0.694	0.709	0.720

			RF		
	250	500	1000	2500	5000
Title	0.763	0.771	0.772	0.817	0.843
Summary	0.677	0.687	0.670	0.723	0.734
Full-Text	0.669	0.669	0.693	0.713	0.718
			LR		
	250	500	1000	2500	5000
Title	0.764	0.777	0.786	0.822	0.819
Summary	0.689	0.703	0.692	0.692	0.685
Full-Text	0.688	0.639	0.680	0.672	0.682

performing model (85.1%). LR (84.9%) and SVM (83.8%) also achieved similar performances. For the summary-level data, the accuracy of the models decreased. Although RF again achieved the highest performance (74.6%), the difference compared to the other models was smaller. Logistic Regression produced the lowest accuracy rate at this level (70.7%). At the full-text level, the most successful model was SVM (75.2%).

RF performed relatively poorly here (71.3%), which may indicate that the model is less effective with long and detailed texts. When feature selection was not applied, title-based classifications in particular yielded more consistent and higher accuracy results. This can be explained by the fact that titles carry short yet dense information, providing more distinctive features for classification algorithms. While RF produced better results for short and summary-level content, SVM was able to generalize more effectively, especially for longer content.

Table 2 presents the accuracy values for User 1 obtained under conditions with feature selection, for different feature dimensions (250–5000) and different machine learning models (SVM, RF, LR).

When considering the results for the title data, it is observed that accuracy increases across all models as the feature dimension grows. The RF model achieved the best results on title data, with an accuracy of 84.3%. LR peaked at 82.2% with 2,500 features, followed by a slight decline. Title data appears to be the most effective level for classification due to

TABLE III. Accuracy Results for User 2 Based On Title, Summary, and Content Without Feature Selection

	SVM	**RF**	**LR**
Title	0.648	0.637	0.569
Summary	0.649	0.640	0.563
Full-Text	0.649	0.648	0.546

TABLE IV. Accuracy Results for User 2 Based on Title, Summary, and Content Without Feature Selection.

			SVM		
	250	500	1000	2500	5000
Title	0.641	0.646	0.640	0.641	0.649
Summary	0.642	0.644	0.640	0.647	0.650
Full-Text	0.647	0.648	0.653	0.648	0.648
			RF		
	250	500	1000	2500	5000
Title	0.635	0.628	0.595	0.589	0.625
Summary	0.630	0.613	0.585	0.607	0.633
Full-Text	0.616	0.603	0.547	0.637	0.649
			LR		
	250	500	1000	2500	5000
Title	0.641	0.637	0.615	0.581	0.567
Summary	0.636	0.545	0.595	0.545	0.539
Full-Text	0.639	0.634	0.612	0.560	0.526

its compact yet information-dense content. For summary data, the SVM and RF models both reached the highest accuracy of 73.4%. Since the content level consists of long texts, it maintains the lowest accuracy rates even after feature selection. The SVM model also stood out as the most successful on content data, while the accuracy of LR remained constant or declined as the feature dimension increased.

When comparing the results for User 1 in Table 1 with those for User 2 in Table 3, it is clear that User 1 achieved higher accuracy rates across all levels and models. SVM appeared to be the most stable model for both users. However, while SVM achieved 83.8% accuracy on titles for User 1, it only reached 64.8% for User 2. For User 2, accuracy rates across all models remained around 64–65%. At the content level, the SVM and RF models performed similarly (0.649 and 0.648, respectively). LR was particularly weak at the content level, producing the lowest result of 54.6% accuracy.

Differences in labeling by the users significantly affected the models' learning capacity. User 1's labeling style contained clearer patterns, allowing the models to generalize more accurately. In contrast, User 2's labels may have been more scattered or inconsistent, making it harder for the algorithms to distinguish between classes. Data from User 1 enabled the recommender system to produce more consistent and stronger results, whereas User 2's data was more challenging and likely less separable for classification. This demonstrates how critical personalized labeling is to the success of a recommender system.

Table 4 presents the accuracy values for User 2 under conditions with feature selection, for different feature dimensions (250–5000) and different machine learning models. The SVM model achieved its highest accuracy (0.653) on content data with 1,000 features. For summary data, the highest value (0.650) was obtained with 5,000 features. At the title level, accuracy improvement was minimal, with all feature dimensions producing values in the range of 0.64–0.65. The RF model produced its best results with 5,000 features on content (0.649) and summary (0.633) data, with its performance on content being close to that of SVM. The LR model showed a clear decline in accuracy as the number of features increased, suggesting that the model may suffer from overfitting or a loss of discriminative power with high-dimensional data.

User 1 consistently achieved higher accuracies across both models and content levels. While SVM was the most stable model for both users, LR performed poorly, especially for User 2. RF was a very strong model for title-level data with User 1, but failed to maintain that success for User 2.

In conclusion, user-based data labeling directly impacts the performance of a recommender system. The dataset created by User 2 appears to be more ambiguous and to have lower discriminative power. SVM offers the most reliable performance, especially for content- and summary-based recommender systems when supported by feature selection. Although performance generally increases with the number of features, certain models such as LR may experience performance degradation in such cases.

CONCLUSION

This study aimed to develop a content-based recommender system for Turkish news content and, in this context, proposed a machine learning-based framework capable of delivering news aligned with users' interests. The developed system was trained using labeled news data obtained from users and then evaluated at the title, summary, and full-text levels using different classification algorithms. Model performances were compared in both scenarios with and without feature selection, and the results were analyzed based on datasets from two different users (User 1 and User 2).

The findings indicate that title data generally yields higher accuracy compared to summary and full-text data. This can be attributed to the fact that titles contain concise, focused, and information-dense content, which is more distinctive for classification purposes. Although models without feature selection generally produced higher accuracy, scenarios with feature selection showed noticeable improvements, particularly for SVM and RF models. This demonstrates that an effective feature selection strategy can positively contribute to classification performance. Comparing user-based datasets revealed that the data labeled by User 1 was easier for classification models to learn from, resulting in higher accuracy rates. In contrast, the datasets from User 2 showed significantly lower accuracy across all content levels compared to User 1, highlighting the critical impact of the labeling process on the performance of the recommender

system. These results show that personalized systems depend not only on the algorithms but also heavily on the quality of user-provided data. Consistent, clear, and feature-rich data directly enhances system performance.

For future work, real-time user feedback on whether recommended news items are liked or not could be collected and integrated into the model online, enabling the system to become continuously learning and adaptive. To reduce the challenges of manual labeling by users, active learning or weakly supervised learning techniques could be applied to train effective models with fewer labeled data. Additionally, developing user-guided interfaces during the labeling process could improve data quality. Re-testing the system with data from users of different demographic groups would help reveal the model's generalizability and provide an opportunity to assess its performance in multi-user scenarios.

ACKNOWLEDGMENT

This study was supported by TUBITAK 2209-A - University Students Domestic Research Projects Support Program.

REFERENCES

[1] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, Recommender Systems: An Introduction. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[2] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," Knowl.-Based Syst., vol. 157, pp. 1–9, 2018.

[3] M. B. Magara, S. O. Ojo, and T. Zuva, "A comparative analysis of text similarity measures and algorithms in research paper recommender systems," in Proc. 2018 Conf. Inf. Commun. Technol. Soc. (ICTAS), 2018, pp. 1–5.

[4] Ö. Gelemet, H. Aydın, and A. Çetinkaya, "Netflix verileri üzerinde TF-IDF algoritması ve Kosinüs benzerliği ile bir İçerik Öneri Sistemi Uygulaması," AJIT-e: Acad. J. Inf. Technol., vol. 13, no. 48, pp. 31–52, 2022.

[5] W. L. Song, "News recommendation system based on collaborative filtering and SVM," in Proc. 2018 3rd Int. Conf. Autom., Mech. Electr. Eng. (AMEE), 2018.

[6] A. Karadağ and H. Takçı, "Metin madenciliği ile benzer haber tespiti," in Proc. Akademik Bilişim, 2010.

[7] F. Başkaya and İ. Aydın, "Haber metinlerinin farkli metin madenciliği yöntemleriyle siniflandirilmasi," in Proc. 2017 Int. Artif. Intell. Data Process. Symp. (IDAP), Malatya, Turkey, 2017, pp. 1–5, doi: 10.1109/IDAP.2017.8090310.

[8] M. Tutkan, M. C. Ganiz, and S. Akyokuş, "An Unsupervised Semantic Attribute Selection Method for Text Classification," in Proc. Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu, 2014, pp. 1–4.

[9] M. V. Mashak, "Feature Selection Using Co-occurrence of Terms for Binary Text Classification," Ph.D. dissertation, Eastern Mediterranean University, 2015.

[10] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in Proc. 14th Int. Conf. Mach. Learn., San Francisco, CA, USA, 1997, pp. 412–420.

[11] R. Cekik and A. K. Uysal, "A novel filter feature selection method using rough set for short text data," Expert Syst. Appl., vol. 160, p. 113691, 2020.

[12] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, pp. 721–735, 2009.

[13] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in Proc. Eur. Conf. Mach. Learn., Berlin, Heidelberg: Springer, 1998, pp. 137–142.

[14] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[15] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale Bayesian logistic regression for text categorization," Technometrics, vol. 49, no. 3, pp. 291–304, 2007.

Using HOG+SVM, VGG16, and ResNet50 Models for the Classification of Architectural Structures

Duygu Canbuldu[1], Emrah Aydemir[2], Suleyman Çelik[3]

[1] *Sakarya University, Management Information Systems, Sakarya, Turkiye, duygu.canbuldu@ogr.sakarya.edu.tr*

[2] *Sakarya University, Management Information Systems, Sakarya, Turkiye, emrahaydemir@sakarya.edu.tr*

[3] *Firat University, Banking and Insurance, Elazig, Turkiye, suleyman.celik@firat.edu.tr*

ABSTRACT

Accurate classification and preservation of architectural structures is very important for transferring knowledge to future generations. However, traditional classification methods are both time-consuming and subject to subjective evaluation due to the fact that architectural works are spread over a wide geography and integrated with other cultures, etc. Methods such as deep learning are needed for a more objective, fast, and reliable classification of architectural structures. The aim of this study is to go beyond traditional methods and comparatively evaluate the effectiveness of image-based machine learning and deep learning approaches in the classification of different architectural styles. The data was obtained from the Unsplash API application and Google Images, as a total of 940 architectural images in 3 different styles. These obtained images were analyzed with HOG+SVM, VGG16 and ResNet50 models via the Python programming language. According to the analysis findings, while the traditional machine learning HOG+SVM showed a success rate of 71%, the ResNet50 model showed the highest success rate with 88%. In the classification process, images of modern architecture were successfully classified with a high level of distinguishability; On the other hand, it has been observed that there is a significant mixing between the images of neoclassical architectural styles and Gothic images.

Keywords: *Machine Learning, HOG, SVM, VGG16, RESNET50*

INTRODUCTION

Architectural structures are valuable works that offer important information about the cultural, economic, religious, and legal fabric of the periods to which they belong and that provide later generations with a historical perspective. The recognition and preservation of these structures are of great importance not only for the sustainability of cultural heritage but also for their direct or indirect effects on a country's tourism, economy, and socio-cultural fabric. Therefore, taking strategic measures to accurately identify, classify, and preserve architectural structures is crucial for transmitting knowledge to future generations.

Knowing and understanding architectural structures requires the ability to describe and analyze the architectural styles of architects from different geographical regions through observation. However, traditional architectural culture studies mostly rely on historical documents and subjective expert opinions; as a result, these studies remain limited in explanatory power and may be affected by regional biases [1]. Yet, the accurate classification of architectural structures is of critical importance for properly analyzing the culture and civilization of a given period. The classification process becomes highly challenging due to factors such as regional characteristics, different construction materials, and historical eras [2,3]. In addition, the spread of architectural styles and structures over long historical periods and wide geographies, their substantial transformations, and cross-cultural integration make the classification process time-consuming and data-intensive, thus complicating classification by traditional methods [4].

Today, the rapid growth and widespread adoption of technological advancements such as artificial intelligence, machine learning, deep learning techniques, and large language models have also had significant impacts in the field of architecture; they have opened broad avenues of research and application in areas such as sustainability, visual enhancement, low-energy

design, diverse production, and performance optimization. Nevertheless, architectural style analyses remain relatively underexplored compared to other topics. Even so, developments in AI have steered architecture-style-focused research toward automated and data-driven approaches [1]. These approaches used in architectural style classification are regarded as optimal methods for analyzing data and achieving accurate classification [3].

The aim of this study is to accurately analyze and classify architectural structures based on their distinctive features using image processing and deep learning algorithms. To this end, images belonging to Gothic, Neoclassical, and Modern architectures were downloaded from the Unsplash API and Google Images, and the obtained images were classified using HOG+SVM, VGG16, and RESNET50 methods. This study is expected to contribute to work on deep learning and image processing in the field of architectural classification.

LITERATURE

With advances in artificial intelligence and the rapid increase in data volume due to technology use, architectural structure classification has shifted from traditional methods to more systematic, machine-learning-oriented approaches. Although the literature on image processing and deep learning is relatively limited, such research has begun to attract increasing attention in recent years.

Lomio, Farinha, Laasonen, and Huttunen [5] examined the construction sector and used both traditional and machine-learning methods to categorize buildings—based on Building Information Modeling (BIM) outputs—into apartment, industrial, and other classes. The images were BIM software renderings rather than real photographs. In their visual classification, HOG+SVM achieved 57% accuracy, while convolutional neural networks (CNNs) exceeded 89%, showing that deep learning outperformed the traditional approach despite a small dataset. Sun, Zhang, Duarte, and Ratti [4] used street- and facade-level images to classify building age and style with a deep learning model, focusing particularly on predicting building age and understanding temporal–spatial relationships between age and style. Their findings indicated accuracy above 80%. Diker and Erkan [3] centered not only on classification but also on how data diversity and quantity affect model performance in architectural style classification. Emphasizing the importance of choosing an adequate number of classes, they developed a CNN-based approach—tested on three separate datasets—to reduce the effective number of classes. Their results showed that as the number of samples increased, the classification accuracy for architectural styles decreased.

Xu, Tao, Zhang, Wu, and Tsoi [6] proposed a multinomial latent logistic regression (MLLR) model to capture morphological attributes of core architectural components and to address a multi-class setting. Using images gathered from Wikimedia, they built a large-scale dataset with 25 classes. MLLR yielded the best classification results among the compared algorithms and, through probabilistic predictions, also enabled interpretation of relationships between architectural styles and classes, including mixed-style compositions within a single building. Hong [7] aimed to categorize traditional architectural styles according to their distinctive attributes using a dataset of 1,200 images of Chinese architecture. An ASO-KNN approach achieved 98.79% accuracy and outperformed well-known deep models such as ResNet, DenseNet, and AlexNet. Rababaah and Rababah [8] developed a deep-learning-based intelligent machine-vision model for architectural style classification. Trained and validated on a public dataset comprising more than 5,000 images across eight styles, their model achieved a reliable performance with 95% accuracy.

The table below presents comparative data for these studies.

TABLE 1. Literature Comparison

Author	Method	Dataset	Findings
Lomio at all. [5]	HOG + SVM; Deep CNN	BIM model outputs (240 images)	HOG+SVM: %57; CNN: %89+
Sun at all. [4]	HOG + Iterative SVM; ResNet50 + CNN	Building facade images (22,000+ images)	HOG+SVM: %63; ResNet50: %80+
Diker and Erkan [3]	Deep CNN + data augmentation	Three different datasets, expanded to 4,776–9,552 images	As the number of data points increases, the success rate for architectural style classification decreases. Latent regression model: 85%

Xu at all.[6]	HOG + MLLR	A large architectural dataset in 25 styles	As the number of data points increases, the success rate for architectural style classification decreases. Latent regression model: 85%
Hong [7]	HOG + ASO-KNN + Deep Learning	1,200 images of Chinese architecture	ASO-KNN: %98
Rababaah and Rababah [8]	CNN	More than 5,000 images of 8 different architectural styles	CNN: %95

In summary, deep learning models achieved higher accuracy than classical machine learning. In addition, increasing the number of samples had a negative effect on style classification performance.

METHOD

In this study, three different methods were considered to classify images of architectural structures. The first of these methods is the classical machine learning approach that uses Histogram of Oriented Gradients (HOG) for feature extraction and Support Vector Machine (SVM) for classification. Then, the deep learning models VGG16 and ResNet50 were employed. Each of these methods aims to improve classification performance by extracting features at various levels of visual representation. Ten-fold cross-validation was performed for all models, and accuracy, precision, recall, and F1-score values were compared.

A. HOG + SVM METHOD

Histogram of Oriented Gradients (HOG) is a widely used feature extraction technique designed to detect edges and gradients in images. It was first proposed by Dalal and Triggs \[9] for human detection and found to be successful. Its greatest advantages are robustness in edge detection and resistance to illumination changes. However, its performance decreases in multi-object detection and in images with nested structures. In architectural visual data, it is effective in capturing local structural information for detecting building, facade, and detail elements [10].

In this study, features of architectural images were extracted using HOG, regional gradients were computed, and these were converted into histograms. The resulting HOG features were then processed with SVM as the classification algorithm.

B. VGG16 METHOD

VGG16 is a deep learning–based model trained on the ImageNet dataset. As its name suggests, it has a 16-layer architecture and is highly effective at learning from low-level edge features to high-level representations of images. Developed by Simonyan and Zisserman \[11], this model achieved notable success by comprehensively evaluating increasingly deep networks using very small (3×3) convolution filters, taking first and second place in the 2014 ILSVRC competition.

C. RESNET50 METHOD

ResNet50 is a model developed to train deeper neural networks. Thanks to shortcut (skip) connections between layers, ResNet enables the network to become deeper and more efficient. In 2015, it achieved excellent performance on the ImageNet classification dataset with an error rate of 3.57% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) using very deep residual networks [12].

With its 50-layer architecture, ResNet50 can learn increasingly complex and deep features at each layer of the image, and therefore can exhibit better performance in architectural style classification.

D. DATASET

The dataset of architectural images used in the study was downloaded from the Unsplash API and Google Images using Python. The images consist of photographs of architectural structures taken from different angles. The images in the dataset were labeled with three different names: Gothic, Neoclassical, and Modern architecture.

An example of each class is shown below:

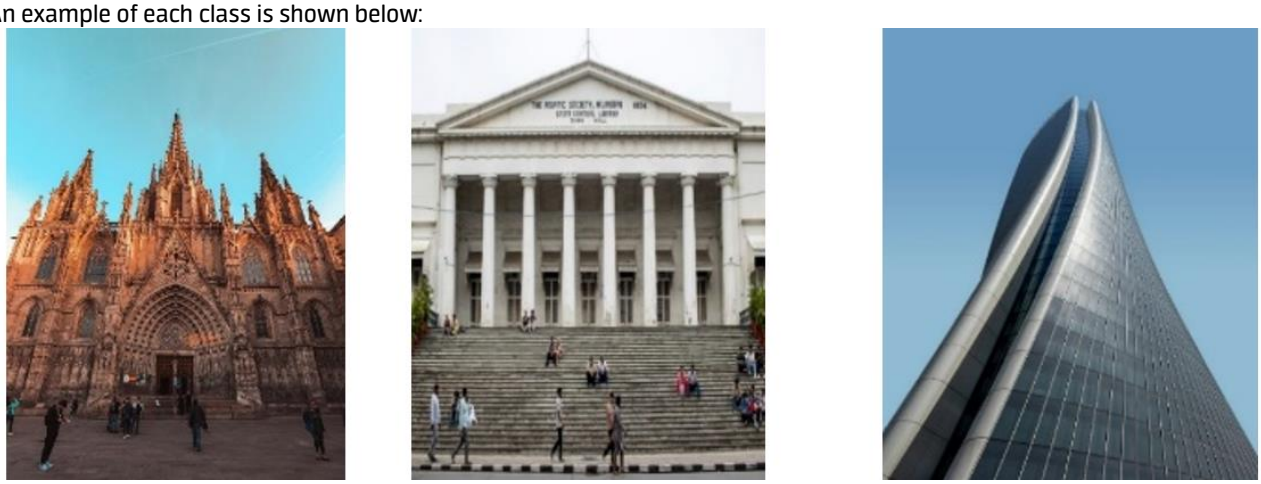


Fig. 1. Gotic, Neoclassical and Modern

The dataset consists of a total of 940 samples across Gothic, Neoclassical, and Modern styles. During labeling, discriminative characteristics of architectural styles were leveraged. These discriminative features are:

- Gothic Architecture: Emerging in France in the 12th century, Gothic architecture continued until the late 16th century. The three key features of Gothic architecture are pointed arches, ribbed vaults, and flying buttresses [13,14].
- Neoclassical Architecture: It remained influential from the mid-18th century to the early 19th century in Europe and America. It arose as a reaction to the excessively ornamental Baroque and Rococo styles. Its most important characteristics are the absence of ostentation, clear and clean lines that foreground simplicity and elegance, and the adoption of a minimalist approach [15].
- Modern Architecture: It emerged in the 19th century as a product of Western civilization. The aim is to create a more innovative and effective style of decoration. Clean, minimal lines; wide roof overhangs; large windows and glass walls; open and well-defined floor plans; and asymmetrical designs are among its prominent features [16].

RESULTS

The Python programming language was used to analyze the collected data. The data were examined with three different methods; the traditional classification method was compared with deep learning models. According to the analysis results, the method with the best performance was found to be ResNet50.

A. COMPARATIVE ANALYSES

The table below presents the comparative results of the HOG + SVM, VGG16, and ResNet50 models.

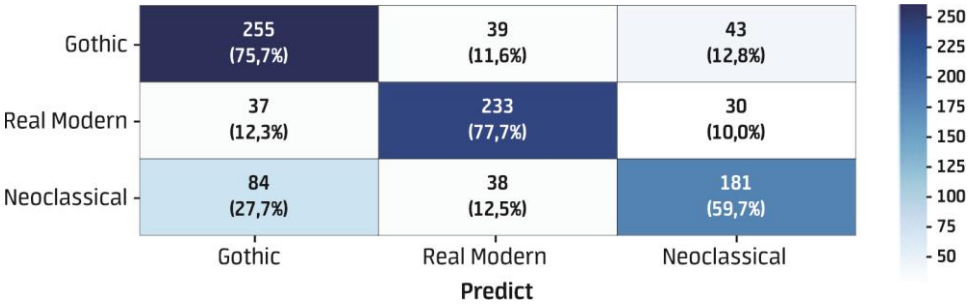
TABLE 1. Comparative Analysis

Model	Accuracy	Precision	Recall	F1-Score
HOG + SVM	0,71	0,71	0,71	0,71
VGG16	0,86	0,86	0,86	0,86
ResNet50	0,88	0,89	0,88	0,88

When Table 2 is examined, it is seen that deep learning-based models (VGG16 and ResNet50) exhibit significantly higher performance compared to the classical feature extraction method HOG + SVM. While the HOG + SVM method shows the lowest success with 71% accuracy, the VGG16 model achieves a significant improvement with an accuracy rate of 86% and demonstrates a balanced performance in Precision, Recall, and F1-Score values. The highest performance is obtained with the ResNet50 model, with an accuracy rate of 88%. The analysis results indicate that deep learning-based approaches provide a clear superiority over classical methods.

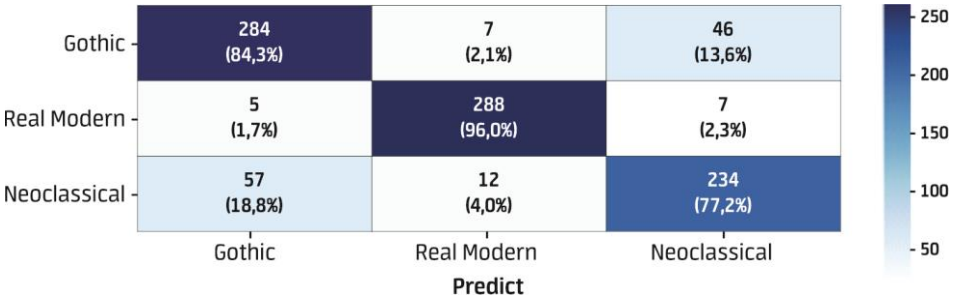
The confusion matrix tables for the models are given below in order.

TABLE 2. HOG+SVM Confusion Matrix



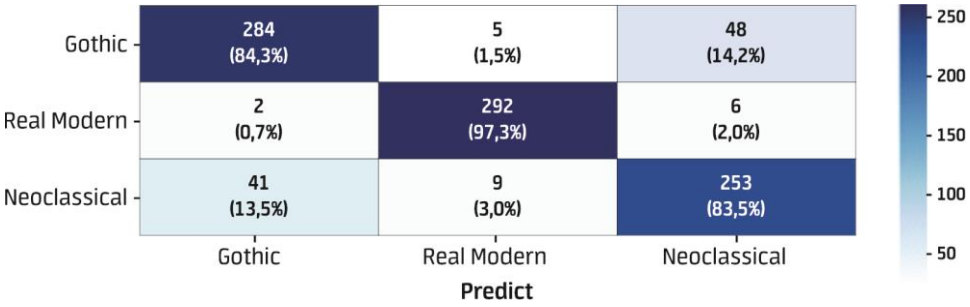
According to the confusion matrix values of the HOG+SVM model, Gothic architecture and Modern architecture are classified better than Neoclassical architecture. Accordingly, 255 out of 337 Gothic architectural images (76%) and 233 out of 300 Modern architectural images (78%) are correctly classified. Of the 303 Neoclassical architectural images, only 181 (60%) are correctly classified. Therefore, it can be said that the Gothic and Neoclassical architectural classes are more frequently confused with each other.

TABLE 3. VGG16 Confusion Matrix



According to the VGG16 confusion matrix values, the highest classification rate belongs to the Modern class with 96%, followed by the Gothic architectural class with 84%. The lowest classification occurs in the classification of Neoclassical architectural images with a rate of 77%. The best separation between classes occurs between Modern and Gothic architecture with approximately 2% error. The greatest confusion is identified between the Neoclassical and Gothic architectural classifications with 19%.

TABLE 4. RESNET50 Confusion Matrix



According to the confusion matrix values of ResNet50, the classification of Modern architectural images has achieved an excellent classification success with a rate of 97%. The classification rates of Gothic and Neoclassical images are both close to each other at approximately 84%, indicating that confusion still persists between these two classes.

Considering all the confusion matrix values, overall the best classification rate belongs to the classification of Modern architectural images, with Gothic architectural classification ranking second. Although Neoclassical image classification remains at a lower rate compared to the others, it is most often confused with Gothic architecture.

CONCLUSION

The preservation of architectural structures is highly important for reasons such as ensuring cultural transmission to future generations and contributing to tourism and the national economy. Accurate and rapid classification of architectural structures is crucial for clearly revealing intergenerational differences and ensuring accurate information transfer. In this study, different modeling approaches based on traditional and deep learning methods were comparatively evaluated for the purpose of automatically classifying images of architectural structures. A total of 940 labeled images were used, and the data were analyzed with 10-fold cross-validation. The aim is to determine the most effective vision-based model capable of distinguishing architectural styles. According to the research findings, deep learning models yield better results than classical machine learning methods. Among the applied models, ResNet50 shows the highest success. In addition, the highest success rate in image classification occurs in the Modern architecture class due to its distinctly geometric characteristics. In Neoclassical architectural classification, a pronounced confusion with Gothic architectural classification is identified. With this study, it is aimed to contribute to the work in this field by revealing the performances of both traditional and deep learning–based methods in architectural classification.

The limited number of datasets and platforms from which the data were obtained has led to constrained results. In future research, enlarging the datasets and considering the integration of different classification algorithms and artificial intelligence techniques can be pursued to increase the generalizability of the obtained results and improve success rates.

REFERENCES

[1] Zhong, J., Yin, J., Li, P., Zeng, P., Zhang, M., Lu, S., & Luo, R. (2025). ArchiLense: A Framework for Quantitative Analysis of Architectural Styles Based on Vision Large Language Models. arXiv preprint arXiv:2506.07739.

[2] Wang, B., Zhang, S., Zhang, J., & Cai, Z. (2023). Architectural style classification based on CNN and channel–spatial attention. Signal, Image and Video Processing, 17(1), 99-107.

[3] Diker, F., & Erkan, İ. (2024). An Approach With Deep Convolutional Neural Networks For Accurate Architectural Style Classification. New Design Ideas Vol.8, No.3, 2024, pp.615-640 https://doi.org/10.62476/ndi83615.

[4] Sun, M., Zhang, F., Duarte, F., & Ratti, C. (2022). Understanding architecture age and style through deep learning. Cities, 128, 103787.

[5] Lomio, F., Farinha, R., Laasonen, M., & Huttunen, H. (2018, November). Classification of building information model (BIM) structures with deep learning. In 2018 7th European Workshop on Visual Information Processing (EUVIP) (pp. 1-6). IEEE.

[6] Xu, Z., Tao, D., Zhang, Y., Wu, J., & Tsoi, A. C. (2014). Architectural style classification using multinomial latent logistic regression. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13 (pp. 600-615). Springer International Publishing.

[7] Hong, S. (2025). Architectural Heritage Style Identification Using Avian Swarm Optimized K-Nearest Neighbours and Deep Learning. Informatica, 49(19).

[8] Rababaah, A. R., & Rababah, A. M. (2022). Intelligent machine vision model for building architectural style classification based on deep learning. International Journal of Computer Applications in Technology, 70(1), 11-21.

[9] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.

[10] Tekol, A. Y., Elmacı, M., & Aslantaş, V. (2024). Görüntü İşleme ve Derin Öğrenme ile Yüz Tanıma Tabanlı Akıllı Kapı Kilit Sistemi. Electronic Letters on Science and Engineering, 20(1), 11-36.

[11] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[13] Arargüç, M.F. (2016). Mimari Tarzdan Edebi bir Türe :Gotik. Güzel Sanatlar Enstitüsü Dergisi 36:245-257.

[14] https://decombo.com/gotik-mimari-nedir/

[15] https://mimariterim.com/neoklasik-mimari/

[16] https://www.therapinterior.com/post/modern-mimari-nedir

Cell Type Annotation of scRNA-seq Data Using Machine Learning Approaches

Elif İzci[1], Berat Doğan[2]

[1] *Department of Electronics and Automation, Biomedical Device Technology Program, Van Yüzüncü Yıl University, Van, Turkiye, elifizci@yyu.edu.tr*

[2] *Department of Biomedical Engineering, Inönü University, Malatya, Turkiye, berat.dogan@inonu.edu.tr*

ABSTRACT

Single-cell RNA sequencing (scRNA-seq) enables high-resolution characterization of cellular heterogeneity, offering unprecedented insights into complex biological systems. A key step in translating these data into biological knowledge is accurate cell-type annotation, which defines the unique transcriptomic profiles of individual cells. However, manual annotation is often impractical in large-scale single-cell studies due to its reliance on expert knowledge, time-intensive nature, and inherent subjectivity. Machine learning-based annotation provides a powerful alternative, allowing the automated extraction of meaningful biological insights from scRNA-seq data. This study provides a comparative evaluation of supervised machine learning methods for cell-type annotation in scRNA-seq data. We assessed the performance of widely used algorithms across multiple datasets representing diverse tissue types, sample sizes, and cellular compositions. Our results show that the k-Nearest Neighbors (kNN) and Random Forest (RF) classifiers achieved the most balanced trade-off between accuracy and sensitivity. These findings underscore the potential of automated scRNA-seq analysis to enhance the accuracy, reproducibility, and efficiency of cell-type annotation in large-scale single-cell studies.

Keywords: *scRNA-seq, cell annotation, machine learning, classification*

INTRODUCTION

Cells, the fundamental units of organisms, play a crucial role in determining the function and diversity of biological systems. Investigating cells at the molecular level helps us to understand the complex mechanisms of life and illuminate many unresolved biological conditions [1]. Each cell exhibits unique functions through the expression of genetic material, and these functions play a great importance in development and disease processes [2]. Cell identities are defined by the expression of specific genes that are not found in other cell types. The heterogeneity of cells, especially in complex tissues, necessitates molecular-level research, leading to the development of modern biotechnological methods. Analyzing cellular gene expression profiles has become one of the most powerful strategies for understanding disease mechanisms. Sequencing refers to the determination of the nucleotide order in nucleic acids such as DNA or RNA [3]. In particular, cell sequencing enables the exploration of molecular mechanisms in biological systems through comprehensive profiling of gene expression. Over recent years, various technologies and methods have been developed to investigate the heterogeneity and genetic diversity of cell populations [4].

Next-generation sequencing (NGS) technology, one of the sequencing methods, is based on the parallel processing of DNA or RNA molecules [5]. Developed in 2009 using NGS technology, scRNA-seq has transformed biological research by enabling the identification of the unique transcriptomic profile of individual cells [6]. This approach provides sequencing data for each cell, allowing analysis at single-cell resolution. The capabilities of scRNA-seq make it possible to investigate both similarities and differences in gene expression across cell populations. Owing to these strengths, scRNA-seq has become an important tool in biomedical research, with applications ranging from elucidating the molecular mechanisms of diseases to identifying treatment-resistant cell populations and diagnosing genetic disorders [7]. It also plays a critical role in identifying new drug targets and new biomarkers for disease diagnosis and treatment [8]. The scRNA-seq method not

only focuses on identifying known transcripts but also aims to discover new transcripts. The discovery of new transcripts facilitates to identification of rare cells. The analysis of rare cells can enable early diagnosis of diseases such as cancer and simplify the treatment process [9].

Complex and high-dimensional datasets are generated with scRNA technology to identify the gene expression profiles of individual cells [10]. High-dimensional datasets contain thousands of genes (more than 20,000) and cells. This high dimensionality results in variations in gene expression levels between cells, and these variations contain critical information for identifying cell types and subpopulations. However, some challenges may be encountered in scRNA-seq datasets. Zero expression values are frequently observed in scRNA-seq datasets due to low expression levels of certain genes (dropout) [11]. Furthermore, the data also contain technical variation (batch effect) and biological noise.

Recently, different machine learning methods have been developed to analyze high-dimensional, sparse, and noisy scRNA datasets. These methods can be used for dimensionality reduction, clustering, or classification in scRNA datasets. In the dimensionality reduction process, Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Multilevel Approximation and Projection (UMAP) methods are used to transform high-dimensional datasets into low-dimensional space [12]. This process allows for better visualization of the scRNA dataset and the efficient selection of specific features. Clustering is performed using unsupervised machine learning methods such as k-means or hierarchical clustering algorithms to discover cell subpopulations [13]. Classification of scRNA datasets involves classifying cellular phenotypes, such as cell annotation, using supervised machine learning methods such as Support Vector Machines (SVM) or RF [14].

Cell annotation typically involves determining the cell type or subtype of individual cells. This process is either automated using predefined reference datasets or performed manually based on specific gene markers. Manual annotation is based on the use of marker genes for the biological interpretation of each cell cluster. Such markers are obtained from reference databases such as CellMarker [15], MSigDB [16], PanglaoDB [17], Human Protein Atlas. However, manual annotation is a time-consuming, expert-dependent process with limited reproducibility. Recently, various computational methods have been developed for the automatic annotation of cell clusters [18]. In general, these methods combine reference datasets and machine learning algorithms and are categorized as unsupervised, supervised, and hybrid learning. For example, Single R uses reference transcriptome datasets (e.g., the Human Cell Atlas) to determine the gene expression profiles of unknown cells using a correlation-based approach and predictively assign cell types [19].

Supervised learning-based cell annotation employs machine learning models trained on labeled reference datasets to classify unlabeled scRNA-seq data. In this approach, a classifier (e.g., Random Forest, Support Vector Machine, Logistic Regression) is trained using preprocessed and feature-selected scRNA-seq data. During the testing phase, the trained model assigns cell-type labels to previously unseen data. Following this framework, scPred [20] utilizes an SVM classifier to perform supervised cell-type annotation. scPred has demonstrated high accuracy in identifying cell types across diverse datasets, including pancreatic tissue, peripheral blood mononuclear cells, and colorectal tumor biopsies. Another method, SingleCellNet [21], constructs a model from gene pairs selected in reference datasets and applies a Random Forest classifier to categorize unlabeled scRNA-seq data. scAnnotate [22], developed by Ji et al., specifically addresses the dropout problem commonly observed in scRNA-seq data. This approach employs a mixture model that assigns different distributions to each gene depending on whether its expression level is zero (dropout) or greater than zero. The mixture models generated for individual genes are then integrated using an ensemble machine learning strategy to perform cell-type annotation.

In recent years, deep learning methods have also been developed for cell annotation. Transfer learning and variational autoencoder-based models have become particularly prominent in these studies. The scMRA [23] and scVI [24] offer different approaches to exploring complex structures in scRNA-seq data using deep learning techniques. scDeeepSort [25] uses a weighted graph neural network (GNN) model for cell annotation of scRNA-seq data. Using a pre-trained GNN model, the method successfully annotates cell types without requiring additional information such as marker genes or RNA-seq profiles.

Studies on the automated analysis of scRNA-seq data highlight the significance and potential of machine learning methods in this domain. They demonstrate that valuable biological insights can be automatically extracted from high-dimensional sequencing datasets and applied to a wide range of analyses. Such approaches play a pivotal role in biomedical research,

including elucidating tissue composition, identifying disease-specific cell subtypes, and modeling immune responses. Based on this, our study conducts a comparative evaluation of widely used machine learning methods for cell-type annotation in scRNA-seq data. Multiple datasets of varying sizes, cell types, and structural characteristics were employed. The classification performance of each method was systematically assessed for every dataset. This comprehensive comparison provides detailed insights into the strengths and limitations of different classification approaches. Our findings demonstrate the broad applicability of machine learning–based annotation methods across scRNA-seq datasets with diverse properties and cellular heterogeneity.

METHOD

Five publicly available scRNA-seq datasets were analyzed. Data preprocessing was performed in R using the Seurat package, and classification was carried out with the caret package. For each dataset, cell-type annotation was conducted using Support Vector Machine (SVM), k-Nearest Neighbors (kNN), Linear Discriminant Analysis (LDA), Naive Bayes (NB), and Random Forest (RF) classifiers. The performance of these methods was then compared across datasets.

A. Dataset

The five datasets represent different organisms and tissue types and also vary in terms of cell-gene count and cell type. Detailed information about the datasets is given in Table I. The datasets were obtained from the NCBI-GEO (Gene Expression Omnibus) public database [26]. The Zeisel dataset [27] contains 3,005 cells and approximately 19,972 genes isolated from the mouse hippocampus, designed to investigate the diversity of nervous system cell types. The Li dataset [28] contains 561 cells and approximately 55,000 genes from tumor samples, aiming to identify differential expression patterns between tumor and normal cells. The Xin dataset [29] includes 1,600 cells and 39,851 genes from the human pancreas, enabling detailed profiling of cells from non-diabetic and type 2 diabetic individuals. Darmanis dataset [30] contains 466 cells and 22,085 genes obtained from human brain tissue (prefrontal cortex), used for brain cell-type characterization. Finally, the Treutlein dataset [31] contains 80 cells and approximately 23,271 genes from lung tissue, developed to study cell differentiation during lung development.

TABLE 1. Detail information about Datasets

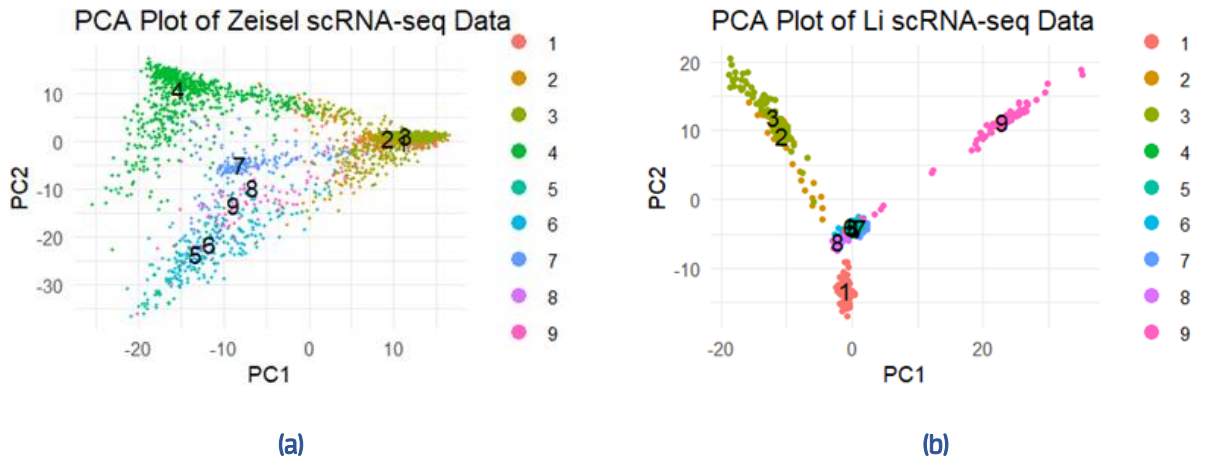
Dataset	Number of Cells	Number of Genes	Number of Cell Types
Zeisel	3005	19972	9
Li	561	55186	9
Xin	1600	39851	8
Darmanis	466	22088	9
Treutlein	80	23271	5

B. Preprocessing

All datasets exhibit a high-dimensional, sparse, and noisy cell-by-gene count matrix structure, reflecting the fundamental characteristics of scRNA-seq data. Therefore, preprocessing was applied before classification to improve data quality.

In the preprocessing step, the count matrix was subjected to normalization, feature selection, scaling, and dimensionality reduction using the Seurat package in R. For normalization, log-normalization was applied to minimize technical variability in gene expression. Despite the abundance of zeros in the count matrix, this transformation also balanced the data distribution, enabling more reliable analysis. Following normalization, genes with biologically significant variability were identified using the Highly Variable Gene (HVG) filter, serving as a feature selection step to retain genes most informative for distinguishing cell types. Dimensionality reduction was then performed using Principal Component Analysis (PCA), which projects the data into a lower-dimensional space by capturing correlations between genes. This process reduces noise

and improves the computational efficiency of classification algorithms. PCA plots of the first two principal components (PCs) for the Zeisel and Li datasets are presented in Fig. 1.



(a) (b)

Fig. 1. PCA plot of the first two PCs in (a) Zeisel and (b) Li datasets.

C. *Classification*

The PCA-reduced datasets were used as input for classification. Cell-type annotation was performed using five different classifiers, each offering a distinct approach to accurately distinguishing cell types within the complex structure of scRNA-seq data. All models were implemented using the caret package in R. The input data were split into 70% for training and 30% for testing. Classifiers were trained with a 5-fold cross-validation procedure, and performance was assessed using accuracy, sensitivity, specificity, and F1 score. Additionally, a confusion matrix was generated for each model.

RESULTS and DISCUSSION

The performance metrics for each classifier and each dataset are presented in Table II. As representative examples, confusion matrices of the models for cell annotation on the Zeisel and Li datasets are shown in Fig. 2 and Fig. 3, respectively.

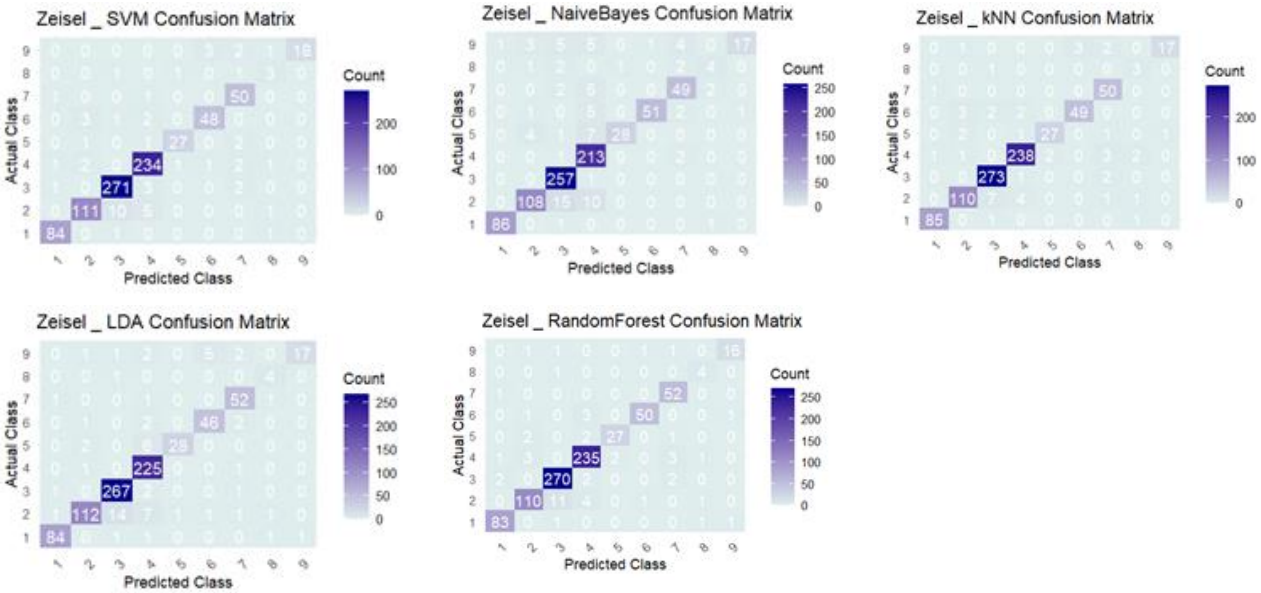


Fig 2. Confusion matrices of the models for Zeisel dataset

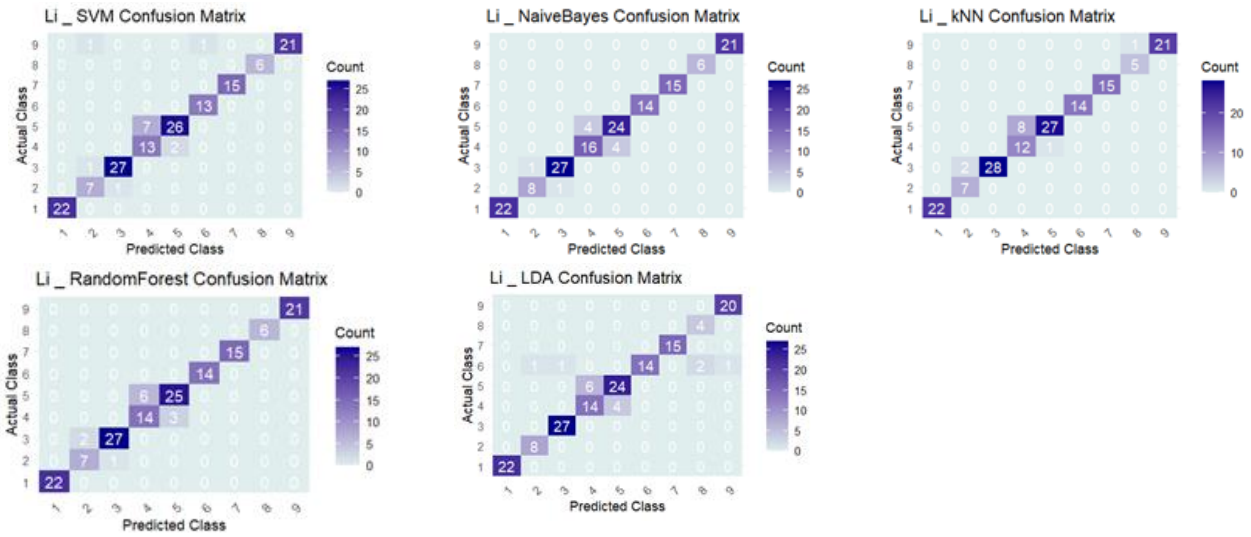


Fig 3. Confusion matrices of the models for Li dataset

For the Zeisel and Darmanis datasets, both containing nine cell types, the kNN model outperformed the other classifiers, achieving accuracies of 0.95 and 0.87, respectively. The RF and LDA models yielded the best results for the Xin and Treutlein datasets, which have fewer classes, likely due to their ability to capture well-separated decision boundaries in lower-complexity classification tasks. Across all five classifiers, the lowest performance was observed for the Darmanis dataset, with accuracies ranging from 0.82 to 0.87.

TABLE 2. The Performance Metrics For Each Classifier And Each Dataset

Dataset	Model	Accuracy	Sensitivity	Specificity	F1 Score
Zeisel	*SVM*	0.94	0.88	0.99	0.87
	NB	0.90	0.89	0.99	0.82
	kNN	0.95	0.88	0.99	0.88
	RF	0.94	0.89	0.99	0.89
	LDA	0.93	0.89	0.99	0.87
Li	*SVM*	0.92	0.92	0.99	0.92
	NB	0.94	0.95	0.99	0.95
	kNN	0.93	0.91	0.99	0.92
	RF	0.93	0.93	0.99	0.93
	LDA	0.91	0.89	0.99	0.90
Xin	*SVM*	0.95	0.56	0.99	0.87
	NB	0.96	0.82	1.00	0.87
	kNN	0.96	0.59	0.99	0.79
	RF	0.98	0.77	1.00	0.87
	LDA	0.98	0.87	0.99	0.87
Darmanis	*SVM*	0.82	0.74	0.98	0.80
	NB	0.82	0.78	0.98	0.75
	kNN	0.87	0.81	0.98	0.81

	RF	0.84	0.79	0.98	0.80
	LDA	0.83	0.75	0.98	0.74
Treutlein	***SVM***	0.90	0.83	0.85	0.85
	NB	0.86	0.75	0.75	0.75
	kNN	0.90	0.90	0.90	0.90
	RF	0.95	0.92	0.94	0.94
	LDA	0.95	0.92	0.94	0.94

This reduced performance may be attributed to the higher transcriptomic similarity among certain cell types in the human prefrontal cortex, as well as potential batch effects and higher technical noise, which increase classification difficulty. Despite the Li dataset being high-dimensional and containing a relatively large number of cell types, all classifiers achieved accuracies exceeding 90%. The NB model achieved the best performance on this dataset with 94% accuracy, possibly due to the presence of highly distinctive gene expression patterns that align well with the model's probabilistic assumptions. In contrast, the NB model exhibited the lowest performance on the Zeisel, Darmanis, and Treutlein datasets, suggesting that its independence assumption between features is less suited to datasets with more correlated gene expression profiles.

Overall, the kNN and RF classifiers demonstrated the most balanced performance in terms of both accuracy and sensitivity across datasets. Notably, in the Xin dataset, SVM and kNN achieved high accuracy (0.95 and 0.96) but relatively low sensitivity (0.56 and 0.59), indicating difficulty in correctly identifying certain minority cell types despite overall classification success. These observations highlight the substantial influence of dataset characteristics, such as the number of classes, inter-class similarity, sparsity, and technical variability, on classifier performance, underscoring the importance of tailoring method selection to the specific properties of scRNA-seq data in future studies.

CONCLUSION

Cell annotation plays a critical role in solving key biological questions such as understanding cell heterogeneity, discovering rare cell populations, and elucidating disease mechanisms. In contrast to manual marker gene analysis, machine learning algorithms accelerate the process by enabling automatic annotation, making it scalable and providing objective results. Automated cell annotation has revolutionized fields such as cancer research, immunology, and regenerative medicine, and allows for efficient mapping of complex biological systems. Our study emphasizes the importance of selecting robust machine learning models for the automatic analysis of single cells. The results show that data preprocessing plays an important role in the success of classification. The study findings indicate that a single classifier does not yield the best results for all datasets, and performance can vary depending on the structure of the dataset.

REFERENCES

[1] Wani, S. A., Quadri, S. M. K., Mir, M. S., & Gulzar, Y. "A Comparative Study of Machine Learning Techniques for Cell Annotation of scRNA-Seq Data." Algorithms 18.4 (2025): 232.

[2] Lucken, M.D.; Burkhardt, D.B.; Cannoodt, R.; Lance, C.; Agrawal, A.; Aliee, H.; Chen, A.T.; Deconinck, L.; Detweiler, A.M.; Granados, A.A.; et al. A sandbox for prediction and integration of DNA, RNA, and protein data in single cells. In Proceedings of the NeurIPS 2021 Track Datasets and Benchmarks, Virtual, 6–14 December 2021.

[3] Pareek, Chandra Shekhar, Rafal Smoczynski, and Andrzej Tretyn. "Sequencing technologies and genome sequencing." Journal of applied genetics 52.4 (2011): 413-435.

[4] Papalexi, Efthymia, and Rahul Satija. "Single-cell RNA sequencing to explore immune cell heterogeneity." Nature Reviews Immunology 18.1 (2018): 35-45.

[5] Hu, T., Chitnis, N., Monos, D., & Dinh, A. "Next-generation sequencing technologies: An overview." Human immunology 82.11 (2021): 801-811.

[6] Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C., & Teichmann, S. A. "The technology and biology of single-cell RNA sequencing." Molecular cell 58.4 (2015): 610-620.

[7] Zhao, Qiuchen, Tong Zhang, and Hao Yang. "ScRNA-seq identified the metabolic reprogramming of human colonic immune cells in different locations and disease states." Biochemical and Biophysical Research Communications 604 (2022): 96-103.

[8] Huang, Y., Cai, L., Liu, X., Wu, Y., Xiang, Q., & Yu, R. "Exploring biomarkers and transcriptional factors in type 2 diabetes by comprehensive bioinformatics analysis on RNA-Seq and scRNA-Seq data." Annals of Translational Medicine 10.18 (2022): 1017.

[9] Torre, Eduardo, et al. "Rare cell detection by single-cell RNA sequencing as guided by single-molecule RNA FISH." Cell systems 6.2 (2018): 171-179.

[10] Wu, Yan, and Kun Zhang. "Tools for the analysis of high-dimensional single-cell RNA sequencing data." Nature Reviews Nephrology 16.7 (2020): 408-421.

[11] Kharchenko, Peter V., Lev Silberstein, and David T. Scadden. "Bayesian approach to single-cell differential expression analysis." Nature methods 11.7 (2014): 740-742.

[12] Sun, S., Zhu, J., Ma, Y., & Zhou, X. "Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis." Genome biology 20.1 (2019): 269.

[13] Petegrosso, Raphael, Zhuliu Li, and Rui Kuang. "Machine learning and statistical methods for clustering single-cell RNA-sequencing data." Briefings in bioinformatics 21.4 (2020): 1209-1223.

[14] Qi, R., Ma, A., Ma, Q., & Zou, Q. "Clustering and classification methods for single-cell RNA-sequencing data." Briefings in bioinformatics 21.4 (2020): 1196-1208.

[15] Zhang, Xinxin, et al. "CellMarker: a manually curated resource of cell markers in human and mouse." Nucleic acids research 47.D1 (2019): D721-D728.

[16] Liberzon, Arthur, et al. "Molecular signatures database (MSigDB) 3.0." Bioinformatics 27.12 (2011): 1739-1740.

[17] Franzén, Oscar, Li-Ming Gan, and Johan LM Björkegren. "PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data." Database 2019 (2019): baz046.

[18] Pasquini, G., Arias, J. E. R., Schäfer, P., & Busskamp, V. "Automated methods for cell type annotation on scRNA-seq data." Computational and Structural Biotechnology Journal 19 (2021): 961-969.

[19] Aran, Dvir, et al. "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage." Nature immunology 20.2 (2019): 163-172.

[20] Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q., & Powell, J. E. "scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data." Genome biology 20.1 (2019): 264.

[21] Tan, Yuqi, and Patrick Cahan. "SingleCellNet: a computational tool to classify single cell RNA-Seq data across platforms and across species." Cell systems 9.2 (2019): 207-213.

[22] Ji, Xiangling, et al. "scAnnotate: an automated cell-type annotation tool for single-cell RNA-sequencing data." Bioinformatics Advances 3.1 (2023): vbad030.

[23] Yuan, Musu, Liang Chen, and Minghua Deng. "scMRA: a robust deep learning method to annotate scRNA-seq data with multiple reference datasets." Bioinformatics 38.3 (2022): 738-745.

[24] Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., & Yosef, N. "Deep generative modeling for single-cell transcriptomics." Nature methods 15.12 (2018): 1053-1058.

[25] Shao, Xin, et al. "scDeepSort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network." Nucleic acids research 49.21 (2021): e122-e122.

[26] Barrett, Tanya, et al. "NCBI GEO: archive for functional genomics data sets—update." Nucleic acids research 41.D1 (2012): D991-D995.

[27] Zeisel, Amit, et al. "Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq." Science 347.6226 (2015): 1138-1142.

[28] Li, Huipeng, et al. "Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors." Nature genetics 49.5 (2017): 708-718.

[29] Xin, Yurong, et al. "RNA sequencing of single human islet cells reveals type 2 diabetes genes." Cell metabolism 24.4 (2016): 608-615.

[30] Darmanis, Spyros, et al. "A survey of human brain transcriptome diversity at the single cell level." Proceedings of the National Academy of Sciences 112.23 (2015): 7285-7290.

[31] Treutlein, Barbara, et al. "Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq." Nature 509.7500 (2014): 371-375.

Anomaly Detection on Industrial Processes with Convolutional Neural Networks

Mehmet Uğur Türkdamar[1], Celal Öztürk[2]

[1] *Computer Engineering, Niğde Ömer Halisdemir & Erciyes University, Niğde & Kayseri, Türkiye, mturkdamar@erciyes.edu.tr*

[2] *Software Engineering, Erciyes University, Kayseri, Türkiye, celal@erciyes.edu.tr*

ABSTRACT

In this study, fault detection systems were developed by running CNN on two well-known data sets obtained from industrial processes which are Tennessee Eastman Process and Skoltech Anomaly Benchmark. The sliding window approach was applied to both data sets to be given to CNN networks and the temporal dependencies in the data were captured. Accordingly, the same CNN network achieved high test accuracies with 99.98% on the TEP dataset and 99.39% on the SKAB dataset. These results show the power and generalization ability ability of the proposed network in different industrial environments.

Keywords: Fault Detection, CNN-based Deep Learning, TEP and SKAB Benchmarks, Time Series, Anomaly Detection, Sliding Window.

INTRODUCTION

Industrial processes require a robust and reliable monitoring mechanism to ensure operational safety, maintain product quality and enhance overall efficiency. Classical model-based fault detection approaches which rely on physics-based or mathematical models often struggle to cope with the increasing complexity of modern industrial processes, the availability of high-dimensional data and the low tolerance for measurement errors. These limitations make it challenging to adapt or scale traditional methods to data rich, dynamic industrial environments.

In recent years, deep learning has emerged as a powerful and flexible alternative for data-driven anomaly detection in industrial processes. Among the various deep learning architectures, Convolutional Neural Networks (CNNs) have demonstrated remarkable success particularly in capturing local dependencies and learning hierarchical feature representations from time series data which are common in industrial process monitoring.

This study investigates the effectiveness of CNN-based approaches for anomaly detection on industrial processes focusing on both simulated and real-world scenarios. Specifically fault detection and diagnosis tasks are performed using CNNs combined with sliding time window techniques applied to two well-known industrial benchmark datasets.

This study focuses on measuring the effectiveness of CNN in both simulation and real-world scenarios including the application of fault detection and diagnosis in industrial systems. In particular two well-known industrial datasets with integrated CNN and sliding time windows are studied:

1) Tennessee Eastman Process (TEP) [1] – Standard dataset for evaluating process monitoring techniques with simulation model of complex chemical processes.
2) Skoltech Anomaly Benchmark (SKAB) [2] – A wide range of sensor readings from valves and pumps as a real-world dataset designed for anomaly detection tasks.

By evaluating the proposed approach across both synthetic and real-world datasets this research not only measures the accuracy and anomaly detection capability of CNNs but also examines their generalization capacity and robustness under varying operational conditions and data distributions. The results highlight the practical potential of integrating CNN-based

anomaly detection into intelligent monitoring systems contributing to more reliable, adaptive and efficient industrial process management.

Our contributions are as follows:

- We achieved high accuracy by designing an 8-layer CNN architecture.
- We performed comparative analysis on synthetic and real-world data to evaluate the model performance.
- We showed that these comparative results can be obtained with a low number of iterations.

RELATED WORK

In the recent past CNNs, especially one-dimensional CNN architectures have been frequently applied to fault detection and anomaly detection in heat exchangers using sensor time series data. These methods have increased the power of CNN to extract local temporal features which is critical for distinguishing healthy and faulty operational states.

The architecture of the 1D CNN based heat exchanger fault detection approach proposed in the study consists of multiple Conv1D layers which are fully connected layers along with batch normalization and pooling [3]. In another similar work the developed deep learning framework detects anomalies in heat exchanger sensor data by combining Conv1D layers with dropout regularizer [4] and transfer learning [5].

Moreover, in another study they presented batch normalization and max pooling layers along with improved 1D CNN model on a well-known heat exchanger fault dataset. In their work they highlighted the importance of convolution layers in temporal feature extraction with robust anomaly detection [6].

In these studies the effectiveness of 1D CNN architecture consisting of Conv1D layers with kernel size varying between 3-5, batch normalization, max pooling and dropout dense layers in heat exchanger anomaly classification tasks was mentioned.

The authors of the study introduced the LOF algorithm based on local data point densities [7]. They proposed a new unsupervised anomaly detection - time series evaluation framework for sustainable maintenance work in industry [8]. Recent developments in multivariate time series anomaly detection include ensemble and deep learning based methods. They achieved improved performance on well-known datasets such as SKAB and SMD by augmenting discrete feature subsets with Nested Rotational Feature Bagging technique and detecting localized anomalies with PCA-based transformations [9]. They integrated a hybrid anomaly detection method, multiple time windows and a voting mechanism to detect sudden and cumulative anomalies in software processes [10]. In contrast they enabled online learning capability to observe water distribution systems in real time by deploying Deep Echo State Network on resource-limited edge devices for anomaly detection task [11].

Overall, these studies prove the increasing interest in robust, adaptable and effective anomaly detection frameworks in various application domains and computational environments.

DATASETS

In this study, two widely known industry datasets are used: TEP (Tennessee Eastman Process) and SKAB (Skoltech Anomaly Benchmark) datasets. Both datasets are frequently used in the literature for anomaly detection and fault classification for cyber-physical (integrated computational and physical) systems.

The TEP dataset simulates the behavior of chemical processes under various fault conditions; it includes single and multiple faults. It is a mixture of multivariate time series data with continuous process variables and discrete control signals. For this study, both healthy and faulty samples were extracted. Each sample was labeled as “Anomaly” or “Normal”.

The SKAB dataset is real normal and faulty sensor data collected from industrial actuators and valves. It is designed for real-time anomaly detection scenarios. Data obtained from different sensors are synchronized and pre-processed to ensure consistency between files.

TABLE 1. Summary of Benchmark Datasets for Fault Detection

Dataset	Type	Source	Total Samples	Features	Classes
TEP	Simulated	Chemical Process	15.330.000	56	Healthy / Faulty
SKAB	Real-World	Actuator Sensors	93.612	9	Healthy / Faulty

Familiar datasets play a major and critical role in the development of artificial intelligence and deep learning in industrial fault detection. The reliability and generalization ability of such networks heavily depend on the quality, complexity and diversity of the datasets used in training and testing. Comparative analyses are invaluable for both synthetic and real-world datasets because they encourage researchers to evaluate network performance under various conditions (both controlled and real operational scenarios).

From Table I TEP dataset is obtained from chemical process control system. It has approximately 15.330.000 samples and 56 features. It captures complex and multivariate time series data and reflects various operational situations and failure scenarios. With its high dimensionality and various failure types TEP dataset is highly suitable for capturing discrete patterns and distinguishing normal from faulty in highly dynamic systems.

In contrast SKAB dataset is real-world actuator sensor data collected from industrial environment. It contains 93.612 samples and 9 features covering real anomaly cases. Although it has lower dimensionality than TEP, SKAB shows real sensor behaviors and error patterns in real operations in terms of practical relevance. Therefore it constitutes real environments for robustness as a basic dataset and machine learning as a real-time application.

By using TEP and SKAB datasets together a comprehensive fault detection system will be activated. While TEP dataset is ideal for evaluating network performance for complex, high-dimensional synthetic data; SKAB provides insight into how networks generalize in the real world with fewer variables. This complementary approach contributes to the development of fault detection methods by combining theoretical validation with industry application with both technical rigour and practical deployment.

PROPOSED METHOD

In this paper we propose a 1D Convolutional Neural Network architecture with an integrated deep learning based fault detection framework. The method is specifically designed to extract temporal patterns from multivariate time series data from industrial systems. We apply the same architecture and training settings to both simulated TEP and real-world SKAB datasets to enable consistency and direct performance comparison across different datasets.

We convert raw time series data into supervised learning samples by applying a sliding window approach. Each window is a fixed-length segment of sequential sensor readings that allows the network to learn local temporal dependencies associated with the fault pattern.

A window size of 1.000 and 250-step windows were used. Each window was labeled as Healthy or Faulty at each of its last time steps. This technique aimed to ensure that the network maintains high sample diversity and dataset balance while processing temporal content.

A. CNN Architecture

CNNs are a deep learning architecture. They process time or image data with a grid-like topology. They were originally developed for image recognition tasks. Various applications including visual recognition and language processing have benefited from the capabilities of CNNs and most recently fault detection in industrial systems by processing time series.

Key architectural components: Conv1D layers: Extract local temporal features from sensor streams. Batch Normalization: Stabilizes learning and accelerates convergence. MaxPooling: Reduces dimensionality by undersampling features and captures dominant patterns. Dense layers with Dropout: Prevents over-fitting by activating non-linear decision boundaries. Soft-max output: Generates a probability distribution across the two target classes; Healthy and Faulty.

In summary, CNNs provide robust and scalable methods for automatic feature extraction and classification particularly effective when the input data contains spatial or temporal patterns.

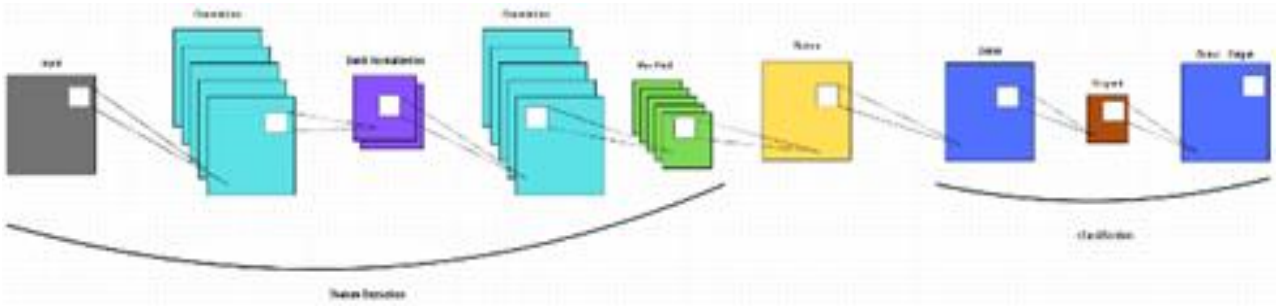


Fig. 1. Structure of the 1D CNN-Based Classification Model

Figure 1 illustrates the structure of the 1D Convolutional Neural Network employed in this research. This network is designed to perform feature extraction and classification tasks on time series data. The network consists of the following structures:

First Convolution Layer (Conv1D):

- It uses 64 filters with 5 kernel sizes with ReLU activation function.
- This layer is responsible for capturing local patterns from input sequences.

Batch Normalization:

- It is implemented to speed up the training process and reduce the risk of overfitting by normalizing the activations.

Second Convolutional Layer (Conv1D):

- Focuses on detailed feature extraction with 32 filters with 3 kernel sizes.
- Also uses ReLU activation function.

Max Pooling Layer (MaxPooling1D):

- Reduces noise in feature maps by reducing dimensionality with a pooling window of size 4.

Flatten Layer:

- It comes last after the feature extractor section and prepares the data for classification.

Dense Layer (Fully Connected Layer):

- It consists of 128 neurons with ReLU activation and learns high-level abstract features from the extracted representations.

Dropout Layer (Rate: 0.5):

- During training, it prevents overfitting and increases generalization by randomly deactivating half of the neurons.

Output Layer (Dense):

- A fully connected layer containing two neurons with Softmax activation is employed for binary classification

This architecture is particularly effective in classifying time series or sequential data. Stacked convolutional layers are good at capturing both short-running and long-running dependencies in the sequential input. The presence of Batch

Normalization and Dropout contributes to the generalization ability of the network and the balanced performance between training and testing datasets.

Powerful real-world industry fault detection, biomedical signal classification and financial time series analysis can be achieved with the proposed network.

EXPERIMENTAL STUDIES

Table II summarizes the key training hyperparameters used in the proposed method. The Adam optimizer [12] which is frequently preferred in the literature and has proven its effectiveness with sparse gradients and adaptive learning rate was preferred. For typical deep learning applications including time series data a learning rate of 0.0001 was selected to approach stability and convergence in training. For the loss value function standard categorical crossentropy was applied in multiple classification tasks so the model was effectively able to separate healthy and faulty.

Using a batch size of 64, the stable gradient updates in computational efficiency were balanced. In addition the model performed a consistent initial search by processing the data using a window length of 1.000 and a step size of 250 thus ensuring that the network effectively captured temporal dependencies in fixed-length sequences between overlapping samples. With these choices anomaly detection and time series modeling were performed in a practical way, a secure, accurate and effective training source was created.

Below are the hyper parameters frequently used in the literature, selected according to the same proposed method used for both data sets:

TABLE 2. Hyper Parameters

Parameter	Value
Optimizer	Adam
Learning rate	0.0001
Loss function	Categorical Crossentropy
Batch size	64
Window size	1.000
Step size	250

The network was trained on stratified data to preserve class ratio distributions and ensure balanced learning. Each data set was divided into 80% training and 20% validation sets. Performance was evaluated using accuracy.

B. Experiments on TEP Data

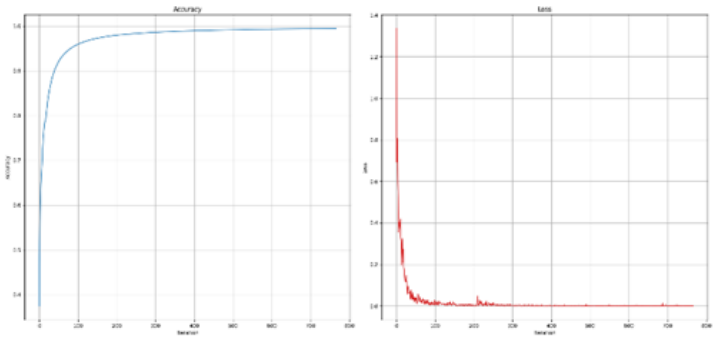


Fig. 2. Training Accuracy and Loss Graphs Obtained From the Training Process on the TEP Dataset

Figure 2 shows the training accuracy and training loss of the network running on the TEP dataset over 767 iteration. When we look at the left side of the Figure 2 the training accuracy showed a increase approaching 1.0 in the 100th iteration and increased this level in the remaining iterations. When we look at the right side of the same graph we see that the loss has reached from 1.3 to 0.02 before even 100 iterations have passed. In the remaining 667 iterations the loss value is around 0.

TABLE 3. TEP Data Results

Authors	Accuracy Score
Imanov et al.	97.84%
Alam et al.	96%
Guo et al.	98%
Proposed Method	99.98%

Table III presents a comparative analysis of recent studies that apply 1D CNN for anomaly detection and classification on the Tennessee Eastman Process (TEP) dataset. All methods listed in the table enhance the time modeling capability of 1D CNN enabling local patterns to be extracted from time series sensor data.

Imanov et al. (2025) achieved 97.84% accuracy over 20 epochs with 1D CNN architecture. The approach they used for the TEP dataset is a baseline performance for CNN models.

Alam et al. (2025) also used a more specific data segmentation strategy utilizing a window size of 1.000 and a stride of 5.000 which affected the granularity and stepping of the data given to the model. They achieved 96% accuracy which is lower than Imanov et al. despite training for 50 epochs probably due to architectural adjustments and differences in preprocessing.

Guo et al. (2025) 1D CNN achieved 98% accuracy in TEP classification. Although specific training parameters such as epoch or windowing strategies were not specified, depth or optimization improvements in the architecture were mentioned as improvements. The proposed method also used 1D CNN architecture but provided noticeable performance improvements with key innovations. It provided more sensitive anomaly detections by obtaining fine-grained time resolution using a sliding window of 1.000 samples with an overlap of 750 samples. With only 767 iterations the network achieved almost perfect 99.98% accuracy and 1.0 F1 Score achieving high scores in all classes in both precision and recall.

When these results are evaluated collectively the robustness and suitability of 1D CNNs for industrial anomaly detection tasks are demonstrated. Consistent architectures in various studies 1D CNNs effectively capture temporal dependencies. Moreover the performance increases seen in the proposed method, It depends on architectural choices, advanced hyper-parameter tuning and effective training strategies to reduce computational time without compromising accuracy.

C. Experiments on SKAB Data

The proposed 1D CNN network was also tested on the SKAB dataset and its status in different fault detection industrial systems was evaluated. The SKAB dataset required the presentation of a compelling anomaly detection algorithm as a multivariate time series obtained from sensor readings under normal and abnormal operating conditions.

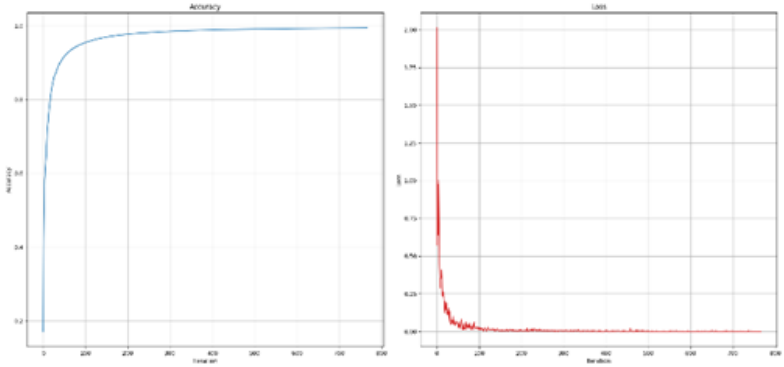


Fig. 3. Training Accuracy and Loss Graphs Obtained From the Training Process on the SKAB Dataset

Figure 3 shows the stable progress of the network running on SKAB data between iterations. When we look at the left side of Figure 3 the accuracy value has increased from 0.15 to 0.9 in 50 iterations. When we look at the right side of a same

graphic the loss value has decreased from 2 to 0. The smooth trajectories for both metrics indicate that the network learned incrementally effectively.

During the experiments the network was trained with data segmented into windows of 1.000 points advancing 250 samples at a time. The model was trained with an 767 iterations reflecting the relatively small but diverse set of training segments in SKAB. The evaluation covers the faulty and healthy cases which are binary classification.

The results show that the network achieved strong performance on this dataset: 100% F1 score and an impressive 99.39% accuracy score. These results show that the network not only achieved high levels of precision and recall but also detected different anomaly patterns in unseen data.

The high F1 score exceeds the critical point in real-world broadcasts indicating a balanced performance in both anomaly detection and false alarm prevention. The excellent accuracy demonstrates the effectiveness of the 1D CNN architecture in capturing complex temporal dependencies in SKAB.

CONCLUSION and FUTURE WORK

This work presents an integrated CNN based framework that is evaluated on two well-known TEP and SKAB datasets. The results demonstrate the high detection performance, efficiency and generalization ability of the proposed network on both synthetic and real-world datasets. The success of the approach demonstrates the applicability of the approach in industry environments such as TEP and SKAB.

Despite its impressive performance the current framework is limited to offline learning and is only dependent on the convolution structure. To overcome these limitations and further network improvements several directions will be explored. First we will investigate the ability of hybrid architectures such as CNN-LSTM to learn both spatial relationships and temporal dependencies on time series data. Second we will focus on scenarios where error signals evolve over time by integrating transformer-based networks with self-attention mechanisms and long-running sequential modeling. Finally we will experiment with online and incremental learning strategies in real-time continuous-operation industrial systems especially in data-constrained environments. With these improvements we will enable more intelligent, adaptive and autonomous error detection systems in complex and dynamic environments.

REFERENCES

[1] Averkij. (n.d.). Tennessee Eastman Process Simulation Dataset [Dataset]. Kaggle. https://www.kaggle.com/datasets/averkij/tennesseeeastman-process simulation-dataset

[2] Waico. (n.d.). SKAB: Skoltech Anomaly Benchmark [Data set]. GitHub. https://github.com/waico/SKAB/tree/master/data

[3] Imanov, T., Yildiz, M., Teimourian, H., Matijosius, J., Kale, U., & Kilikevicius, A. (2025). 1D convolutional neural networks application on aircraft engine thermal performance parameters. Journal of Thermal Analysis and Calorimetry, 1-13.

[4] Alam, TE., Ahsan, M. M., & Raman, S. (2025). Multimodal Bearing Fault Classification Under Variable Conditions: A 1D CNN with Transfer Learning. arXiv preprint arXiv:2502.17524.

[5] Öztürk C, Taşyürek M, Türkdamar MU. Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the CNN-based deep learning models. Concurrency Computat Pract Exper. 2023; 35(4):e7542. doi:10.1002/cpe.7542

[6] Guo, J., Sun, L., Kawaguchi, T., & Hashimoto, S. (2025). Fault Diagnosis of Wire Disconnection in Heater Control System Using One-Dimensional Convolutional Neural Network. Processes, 13(2), 402.

[7] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-based Local Outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (pp. 93-104).

[8] Carrasco, J. (2021). Anomaly detection in predictive maintenance: A new evaluation framework for temporal unsupervised anomaly detection algorithms. arXiv:2105.12818v2

[9] Iliopoulos, A., Violos, J., Diou, C., & Varlamis, I. (2025). Feature Bagging with Nested Rotations (FBNR) for anomaly detection in multivariate time series. Future Generation Computer Systems, 163, 107545.

[10] Shi, T., Zou, Z., & Ai, J. (2023). Software Operation Anomalies Diagnosis Method Based on a Multiple Time Windows Mixed Model. Applied Sciences, 13(20), 11349.

[11] Pau, D., Khiari, A., & Denaro, D. (2021, November). Online learning on tiny micro-controllers for anomaly detection in water distribution systems. In 2021 IEEE 11th International Conference on Consumer Electronics (ICCE-Berlin) (pp. 1-6). IEEE.

[12] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980

Innovative Approaches to High-Energy Gamma Particle Classification: A Coevolutionary Artificial Neural Network for Cherenkov Telescopes

Ali Deveci[1], Mehmet Ali Erkan[2], İhsan Tolga Medeni[3], Tunç Durmuş Medeni[4]

[1] *Department of Computer Engineering, Hacettepe University, Ankara, Türkiye, ORCID ID: 0000-0002-4990-0785*

[2] *Department of Computer, Engineering, METU, Ankara, Türkiye, ORCID ID: 0009-0007-5760-1914*

[3] *Department of Information, Management, METU, Ankara, Türkiye, ORCID ID: 0000-0002-0642-7908*

[4] *Department of Information, Management, Yıldırım Beyazit Unv., Ankara, Türkiye, ORCID ID: 0000-0002-2964-3320*

ABSTRACT

This study addresses the challenges in analyzing data from a ground-based atmospheric Cherenkov gamma telescope, aiming to simulate and observe high-energy gamma particles. Notable challenges include differentiating signals from high-energy gamma rays and background noise induced by cosmic-ray-initiated hadronic showers. Robust methodologies, especially for statistical significance amidst varying energy levels, are essential. The study underscores the need for nuanced solutions in effective data analysis, contributing significantly to our understanding of high-energy gamma phenomena.

A cooperative coevolution-based artificial neural network model, developed in response to these challenges, achieves a classification accuracy of over 91%. This success highlights the model's efficacy in addressing scientific problems, effectively separating gamma rays from background noise, and contributing to future research on atmospheric Cherenkov gamma telescopes.

Keywords: ANN, Cooperative Co-evoluation, Deep Learning

INTRODUCTION

Machine learning algorithms and optimization models have been widely used across various domains, ranging from law [1] to numerical fields. This study addresses the scientific intricacies associated with the analysis and discrimination of data generated by a ground-based atmospheric Cherenkov gamma telescope. This research endeavors to present a comprehensive methodology for simulating and observing high-energy gamma particles. In the pursuit of this endeavor, various scientific challenges and inquiries have surfaced, shaping the landscape of inquiry in this field.

a) *Background Separation:* One of the persistent challenges lies in the nuanced differentiation between signals originating from high-energy gamma rays and the ambient background noise induced by hadronic showers initiated by cosmic rays. This perpetual struggle necessitates an exploration into methodologies that can augment the telescope's discernment capabilities to accurately segregate gamma ray signals from the intricate tapestry of background noise.

b) *Statistical Significance:* Another ongoing challenge involves the attainment of statistical significance when discriminating between signals and background events. The intricate interplay of varying energy levels further complicates this endeavor, prompting researchers to devise and implement robust statistical methodologies. This exploration becomes particularly crucial in light of the wide spectrum of energy levels exhibited by primary gamma rays.

These challenges, which constitute focal points of inquiry, underscore the need for nuanced solutions in the effective analysis of data generated by the gamma telescope. The successful resolution of these intricacies not only advances our understanding of high-energy gamma phenomena but also contributes substantively to the broader scientific community.

In this study, a cooperative coevolution-based artificial neural network model has been developed to address the challenges mentioned. The purpose of this model is to provide an effective solution for the analysis and discrimination of

high-energy gamma particles. The developed model achieves a classification accuracy of over 91%, indicating its successful ability to handle the scientific problems and effectively separate gamma rays from background noise. This success underscores the model's capability to analyze data more effectively and provide solutions to complex discrimination issues. The cooperative coevolution-based artificial neural network model may contribute significantly to future research, enabling more precise examination of data generated by atmospheric Cherenkov gamma telescopes and enhancing the detection of high-energy gamma particles.

BACKROUND

1) A. Artificial Neural Network

A neural network is a computational framework inspired by the complex operations of biological neural networks within the human brain. It serves as a foundational component in the extensive realms of machine learning and artificial intelligence. Neural networks excel at pattern recognition, decision-making, and, notably, learning from data, mirroring the adaptive capabilities inherent in the human brain[2].

The fundamental building block of a neural network is the neuron, also referred to as a node. These networks are organized into layers, where neurons are systematically arranged. The connections between neurons are characterized by weights, and the network processes input data through these interconnected layers to generate an output. As seen in the Figure1 ,outlined below are key components and principles associated with neural networks are,

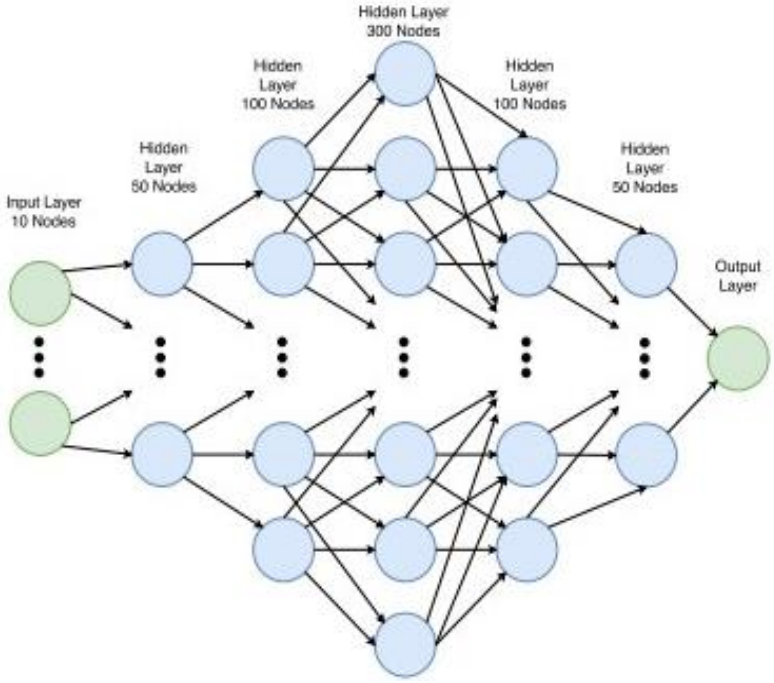


Fig. 1. An ANN model

a) Neuron (Node): Neurons act as the fundamental computational units within a neural network. Each neuron receives one or more input signals, conducts a computation (often a weighted sum), applies an activation function, and produces an output.

b) Layer: Neural networks are structured into layers, typically including[2]:

- Input Layer: Receives input signals.
- Hidden Layer(s): Process input through weighted connections.
- Output Layer: Generates the final output of the network.

c) Connection (Weight): Neurons are interconnected, and the connections carry weights that determine the strength of the linkage. During training, these weights are adjusted to facilitate the network in learning from the provided data.

d) Activation Function: The activation function of a neuron determines its output based on the received input. Common activation functions comprise sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU). The equations for these activation functions are[3]:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (2)$$

ReLU(*x*) = max(0,*x*) (3) *e) Feedforward and Backpropagation:* Neural networks undergo a feedforward process wherein input data is processed to produce an output. In the training phase, backpropagation adjusts weights based on the disparity between the predicted and target outputs [4].

e) Training Data and Labels: Neural networks require a training dataset, encompassing input examples coupled with corresponding labels or target outputs. Throughout the training process, the network learns to map input patterns to the correct output.

The training process involves iteratively adjusting parameters (weights and biases) based on provided data, empowering the network to generalize and make accurate predictions on new, unseen data[4].

2) B. Gamma and Hadron Rays

The term "gamma" commonly denotes a gamma ray, an energetic photon with high energy levels[5]. Gamma rays manifest as a type of electromagnetic radiation, characterized by having the shortest wavelengths and the highest frequencies within the electromagnetic spectrum. While gamma rays are often associated with nuclear reactions and radioactive decay, the specific context in the provided text revolves around Cherenkov gamma telescopes, focusing on high-energy gamma rays originating from celestial sources.

Conversely, a "hadron" is a complex particle composed of quarks bound together by the strong force[5]. Hadrons encompass fundamental particles like protons and neutrons, serving as the foundational constituents of atomic nuclei. There exist two primary classifications of hadrons: baryons, consisting of three quarks (e.g., protons and neutrons), and mesons, formed by the combination of one quark and one antiquark. The term "hadronic showers" pertains to cascades of particles initiated by high-energy cosmic rays in the upper atmosphere. These showers involve various types of hadrons generated through the interactions between cosmic rays and atmospheric particles. The ability to discern between gamma rays and hadronic showers stands out as a pivotal aspect of the analysis carried out by Cherenkov gamma telescopes. These telescopes play a crucial role in distinguishing between the electromagnetic showers induced by high-energy gamma rays and the hadronic showers produced by cosmic rays, thereby contributing to our understanding of celestial phenomena[6].

3) C. Genetic Programming (GP)

GP is a subset of evolutionary algorithms, falling under the umbrella of evolutionary computation[7]. Its objective is the automatic evolution of computer programs for problemsolving. GP applies the principles of natural selection and genetic crossover to evolve populations. The process initiates with the generation of a set of computer programs created randomly. These programs typically consist of fundamental programming elements, such as mathematical expressions, functions, or control structures. This population comprises programs striving to perform a specific task. Evaluation occurs through a fitness function, gauging the proficiency of each program in the designated task. Programs demonstrating superior performance are chosen for the subsequent generation and undergo reproduction through genetic operators. Genetic programming is commonly applied in addressing issues such as symbolic regression, automatic program generation, and the discovery of symbolic expressions.

4) D. Genetic Operators

Genetic operators are fundamental components of genetic algorithms (GAs), which are optimization algorithms inspired by the process of natural selection and genetics. There are three main genetic operators in a typical genetic algorithm:

- Selection:The selection operator simulates the process of natural selection by favoring the reproduction of individuals with better fitness (i.e., individuals that have higher values for the objective function being optimized).
- Crossover (Recombination):The crossover operator involves taking two parent individuals and creating one or more offspring by combining their genetic material. The goal is to exchange information between good solutions to generate potentially better solutions.
- Mutation:This operator introduces random changes in the genetic material of an individual. This helps to maintain diversity in the population and prevents the algorithm from converging too quickly to a suboptimal solution.

In the context of GAs, individuals are often represented as strings of binary digits, but the concept of genetic operators can be adapted to other types of representation depending on the complex problem being solved. The combination of these genetic operators allows GA to explore the solution space efficiently, gradually converging toward better solutions over successive generations. The effectiveness of GAs lies in their ability to harness the principles of evolution to find optimal or near-optimal solutions in complex search spaces.

5) E. Coevolution

Coevolution is a biological and ecological concept that describes the reciprocal evolutionary influence between two or more interacting species [8]. It occurs when the genetic changes in one species result in selective pressures that lead to adaptations in another species, and vice versa. In other words, the evolution of one species is intertwined with and influenced by the evolution of another species with which it has a close ecological relationship.

The process of coevolution can lead to a dynamic and intricate interplay of adaptations and counter-adaptations over time. It contributes to the diversity and complexity of ecosystems, as species continuously shape and respond to each other's evolutionary trajectories. Coevolutionary processes can be observed not only in biological systems but also in artificial systems, such as in the context of coevolutionary algorithms in computer science,physical science and optimization.

6) F. Competitive Coevolution

In competitive coevolution, multiple populations of evolving entities (e.g., individuals, strategies, or solutions) are pitted against each other in a competitive environment [9]. These entities engage in a competitive struggle, and the dynamics of the competition lead to the concurrent evolution and adaptation of all participating populations.

7) G. Cooperative Co-evolution

Cooperative coevolution is an evolutionary computation paradigm that underscores the collaborative evolution of subcomponents to address complex problems [10]. This strategy is particularly effective in scenarios where the problem structure allows for decomposition into modular subcomponents, thereby facilitating a more efficient and scalable evolutionary process. The working process is:

- Identification of Subcomponents: The initial step involves breaking down the larger problem into smaller, specialized subcomponents. Each subcomponent is assigned a specific task.
- Independent Evolution: Each subcomponent undergoes its evolutionary process independently. This allows each subcomponent to focus solely on its designated task, leading to more specific solutions.
- Communication and Collaboration: At certain stages or periodically during the evolutionary process, there is an exchange of information or solutions between subcomponents. This collaborative exchange allows subcomponents to work together more effectively.
- Continuation of Evolution: Collaboration among subcomponents may guide the evolution of each subcomponent further. This fosters compatibility among subcomponents and contributes to the creation of a global solution.

Cooperative coevolution leverages the advantages of modular evolution, making it a valuable strategy for solving largescale and complex problems. Its interdisciplinary applications underscore its adaptability and effectiveness. As research on cooperative coevolution progresses, further refinements and innovative applications are anticipated, solidifying its standing as a powerful approach within the broad landscape of evolutionary computation.

RELATED WORK

In recent years, artificial intelligence and machine learning applications have become indispensable components of contemporary life. Their rapid integration into various domains reflects not only the transformative power of these technologies but also their adaptability to diverse contexts. Today, AI and ML are no longer confined to the boundaries of computer science; instead, they extend across an expansive spectrum ranging from the social sciences (e.g. [1]) to quantitative (e.g. [11]) disciplines. Within the social sciences, they are employed for tasks such as analyzing human behavior, modeling decision-making processes, and improving policy development through data-driven insights. On the other hand, in quantitative and natural sciences, AI and ML contribute significantly to areas such as predictive modeling, optimization, pattern recognition, and large-scale data analysis. This broad applicability demonstrates that AI and ML function as cross-disciplinary enablers, bridging theoretical foundations with real-world practice and offering innovative solutions to complex problems. As a result, they are not merely technological tools but essential catalysts for advancing knowledge and shaping the future of both scientific research and everyday human activities.

HERWIG is a versatile event generator in the field of particle physics, encompassing the simulation of various scattering scenarios such as hard lepton-lepton, lepton-hadron, and hadron-hadron interactions, along with soft hadron-hadron collisions, all integrated into a unified framework. It adopts the parton-shower approach to model initial-state and final-state Quantum Chromodynamics (QCD) radiation, accounting for color coherence effects and azimuthal correlations within and between jets. This paper[12] presents a concise overview of the theoretical foundations of HERWIG, followed by an in-depth exploration of the program itself. The discussion includes specifics about the input parameters and control mechanisms employed by the program, as well as the output data it generates. Additionally, a comprehensive examination of sample output from a typical simulation is provided, complete with annotations for clarity.

In[13], the primary objective within lattice Quantum Chromodynamics (QCD) is to conduct first-principles calculations of multi-hadron dynamics, representing a pivotal endeavor. Substantial advancements have been made in formulating, integrating, and utilizing theoretical methodologies that establish connections between finite-volume parameters and their counterparts in infinite volume. This review encompasses recent progress in both theoretical frameworks and numerical findings pertaining to multi-particle characteristics within a confined volume. Noteworthy results discussed here encompass N scattering, configurations involving two and three mesons with maximal isospin, exploration of three-body resonances in a simplified model, and the development of effective theories tailored for multi-nucleon systems within future works.

In [14], the utilization of an Artificial Neural Network (ANN) method has been implemented in the analysis of yray images derived from observations of the Crab Nebula and Markarian 421 through the Whipple Observatory TeV imaging telescope. In the context of parameterized data, this technique has demonstrated superior performance compared to alternative methods, falling short only to Supercuts in effectively discriminating against undesired hadronic background. Nevertheless, it is noteworthy that the effectiveness of the ANN technique diminishes when applied to unparameterized shower images. The Cherenkov Telescope Array (CTA) is poised to become the preeminent ground-based gamma-ray observatory globally, facilitating the exploration of very highenergy phenomena in the Universe. The impending influx of data from CTA, on the order of petabytes, necessitates the exploration of improved data analysis methods beyond existing ones. Machine learning algorithms, particularly deep learning techniques, emerge as promising avenues for addressing this challenge. Notably, convolutional neural network (CNN) methods applied to images have exhibited efficacy in pattern recognition, generating data representations capable of yielding satisfactory predictions.

In[15], they assess the utility of convolutional neural networks in discriminating signal from background images with high rejection factors and in providing reconstruction parameters for gamma-ray events. The networks are trained and

assessed using artificial datasets of images. The findings reveal that neural networks, when trained with simulated data, prove instrumental in extracting gamma-ray information. This promising outcome positions such networks as valuable tools for optimizing the analysis of vast amounts of real data anticipated in the coming decades.

Although coevolution and artificial neural network (ANN) studies are prevalent across diverse scientific domains, this study conducted on the Cherenkov Telescope stands out as the first one where evolutionary processes and ANN have been synergistically integrated. This groundbreaking study not only marks a unique convergence of these two methodologies but also showcases their application specifically within the context of Cherenkov telescope data analysis. This innovative approach holds the potential to open new avenues for enhanced understanding and interpretation of gamma/hadro-ray observations, setting a precedent for future research endeavors in the intersection of evolutionary algorithms and artificial neural networks within the realm of astrophysics.

DATASET DESCRIPTION

The dataset has taken from the uciMachineLearningRepository [16] is generated through Monte Carlo simulations to emulate the recording of high-energy gamma particles in an atmospheric Cherenkov gamma telescope located on the ground, utilizing imaging techniques. The Cherenkov gamma telescope observes high-energy gamma rays by harnessing the radiation emitted from charged particles generated within electromagnetic showers initiated by gamma rays and progressing through the atmosphere. This Cherenkov radiation, spanning visible to UV wavelengths, penetrates the atmosphere and is captured by the detector, facilitating the reconstruction of shower parameters. The provided data comprises pulses recorded on the photomultiplier tubes in a plane known as the camera, originating from incoming Cherenkov photons. Depending on the primary gamma's energy, several hundred to around 10,000 Cherenkov photons are collected in distinct patterns, referred to as the shower image. This allows for statistical discrimination between those caused by primary gammas (signal) and images of hadronic showers initiated by cosmic rays in the upper atmosphere (background). The dataset has 19020 instances with 11 features (class-included). Table I shows the information of the dataset and Fig. 2 shows the distribution of gamma and hadron classes.

TABLE I. Dataset Description.

Attribute	Type	Description
fLength:	cont.	major axis of ellipse [mm]
fWidth:	cont.	minor axis of ellipse [mm]
fSize:	cont.	10-log of sum of content of all pixels [in phot]
fConc:	cont.	ratio of sum of two highest pixels over fSize [ratio]
fConc1:	cont.	ratio of highest pixel over fSize [ratio]
fAsym:	cont.	distance from highest pixel to center [mm]
fM3Long:	cont.	3rd root of third moment along major axis [mm]
fM3Trans:	cont.	3rd root of third moment along minor axis [mm]
fAlpha:	cont.	angle of major axis with vector to origin [deg]
fDist:	cont.	distance from origin to center of ellipse [mm]
class:	g,h	gamma (signal), hadron (background)

The thorough examination of attributes, including detailed analysis, preprocessing when necessary, and the foundation for applying other statistical methods, involved studying the distributions and mean distributions of all attributes.

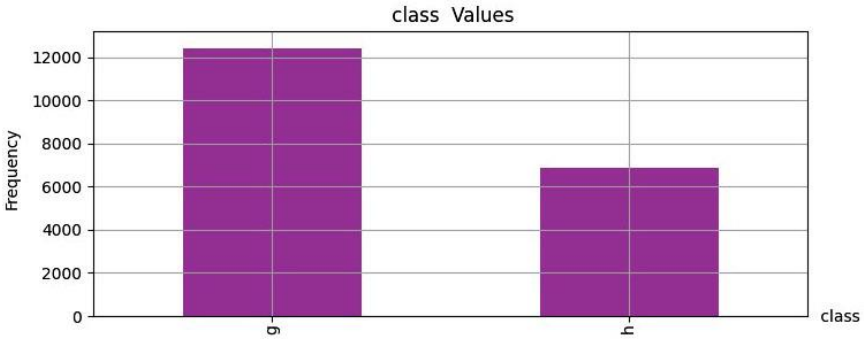


Fig. 2. Class Distribution. (g = gamma, h = hadron)

Fig.3 illustrates the average values for attributes, while Fig. 4 displays the distribution of attributes. Here, the values of attributes with the highest and lowest density may not be considered to prevent overfitting and underfitting. However, in our conducted experiments, we did not observe overfitting or underfitting.

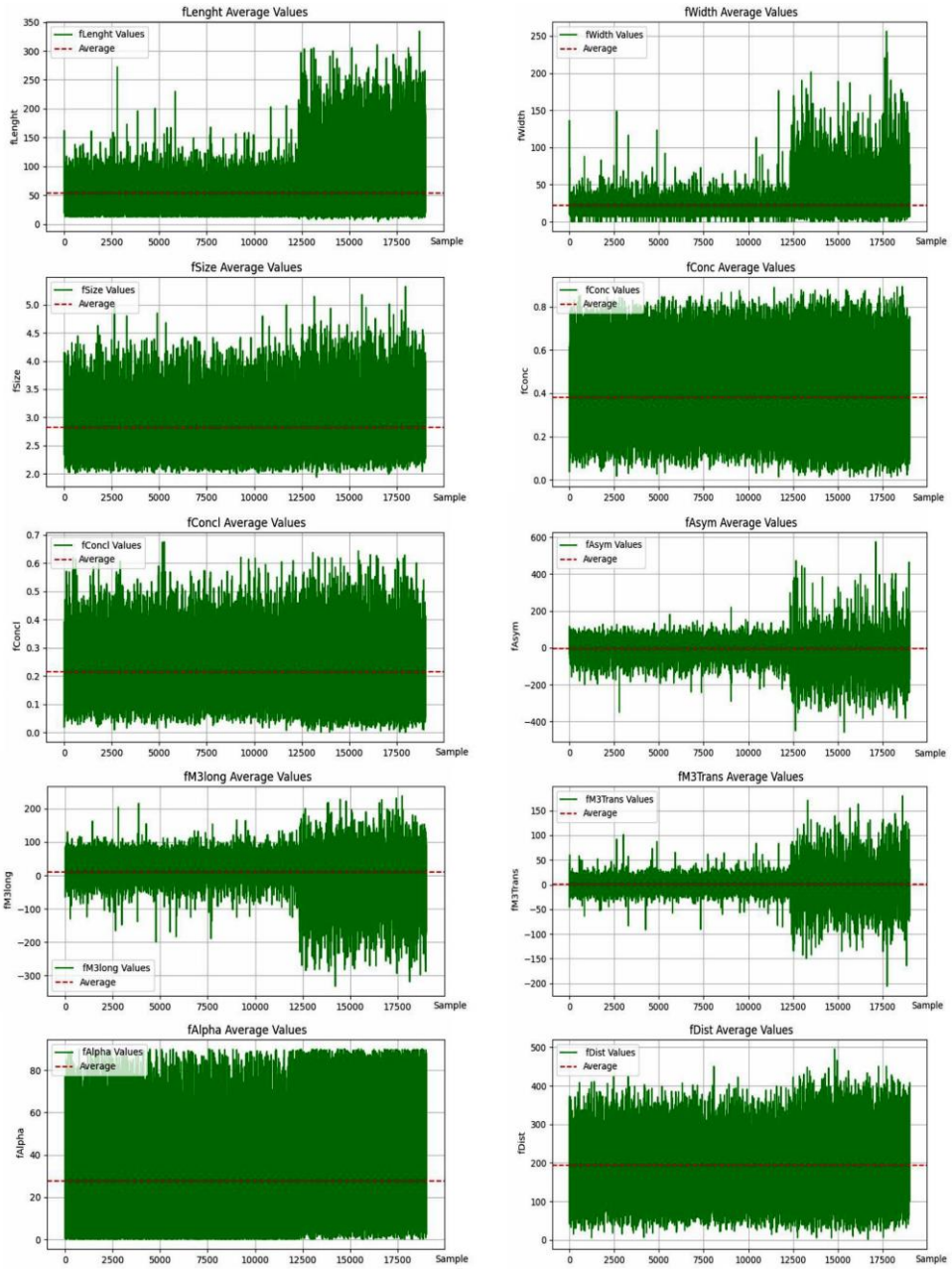


Fig. 3. The average values for attributes.

PROPOSED METHOD

In order to accurately classify gamma and hadron rays, we employed a cooperative coevolution-based Artificial Neural Network (ANN) approach. Fig.5 shows the flow chart of proposed model. Here, a genetic algorithm was used to generate a random initial population of 100 individuals. Since we have two classes (g and h), the initial population was divided into two subpopulations (population1 and population2). The fitness value of each individual is calculated based on accuracy, and individuals with the maximum accuracy, or ideally the maximum, are considered promising candidates and are provided as input to the artificial neural network. The primary goal is to utilize genetic agents to optimize the input values for the network, thereby reducing computational costs.

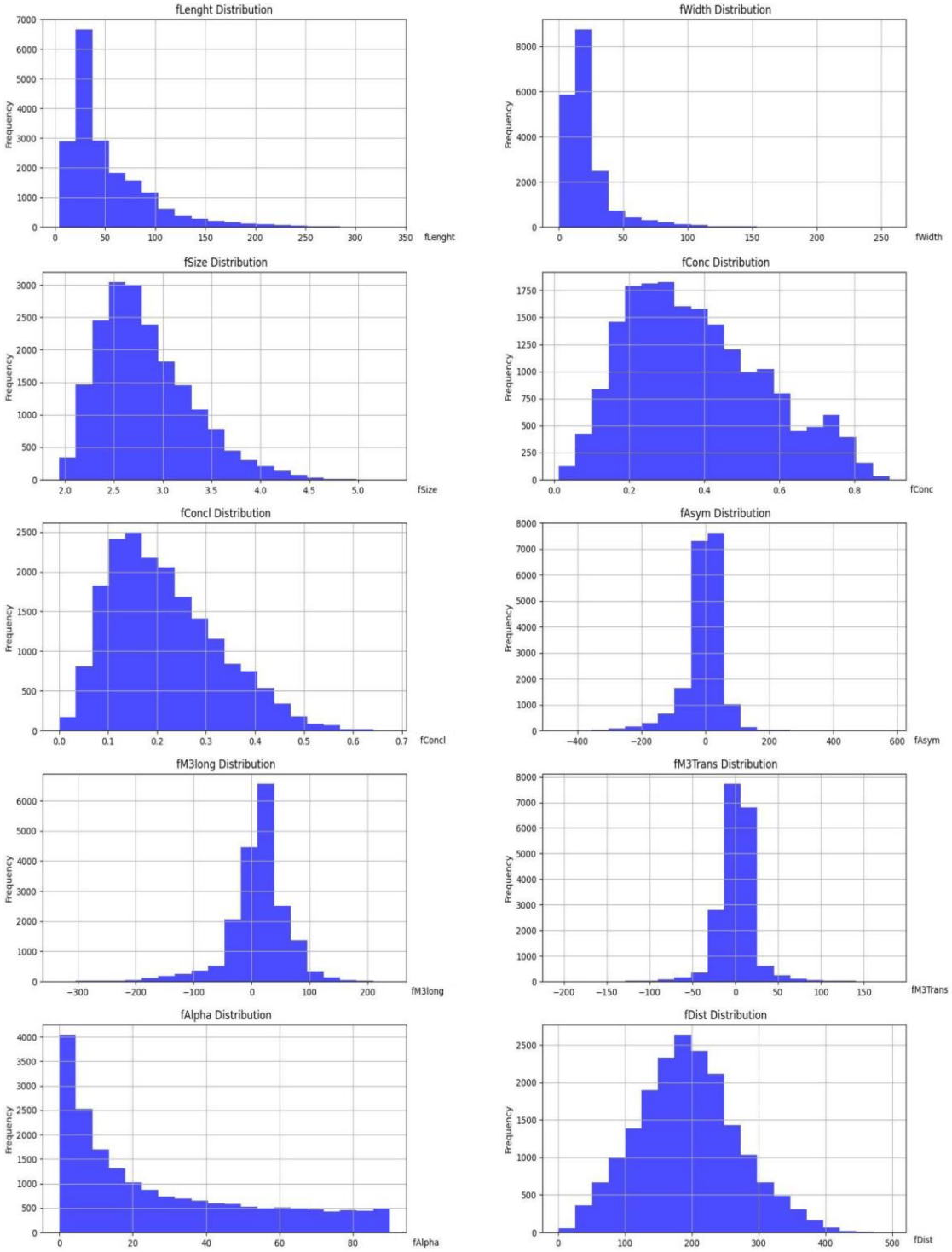


Fig. 4. Distributions of attributes.

Our artificial neural network outputs are utilized in two ways to implement a feedback mechanism. The first involves communication between artificial neural networks, where the output of one network serves as an additional input to another. The second method involves sending the outputs to a pool (similar to a temporary dataset pool). The data in this pool, derived from elite individuals with the best or close-to-best values via genetic agents, is termed the "Elite pool." In our neural networks, we used three hidden layers, each containing 10 neurons. ReLU (Rectified Linear Unit) activation function was employed for all neurons.

In the final step of our model, the data within the Elite pool, originating from elite individuals with maximum accuracy values for both classes, allows for the segregation of data into their respective class labels.

EXPERIMENTAL RESULTS

In this study, the creation of genetic agents, processing of genetic operators, and optimization were accomplished using the JAVA programming language. Specifically, to implement

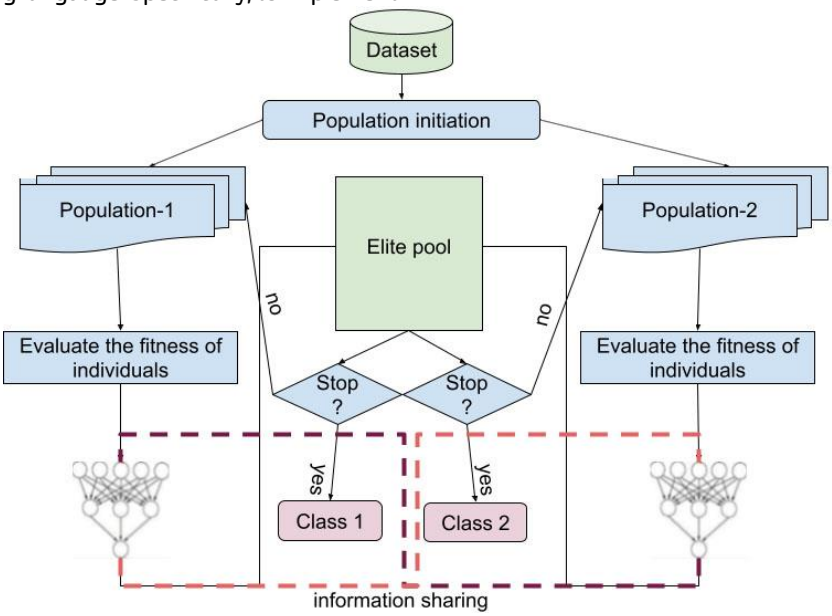


Fig. 5. Flow chart of the Proposed method.

the elitist strategy (ensuring the survival of the best individuals), an implementation of the Java-based Evolutionary Computation Research System (ECJ) was implemented. Firstly, as a preliminary experiment, we examined the performance of neural networks with simple and multi-layer features. In the first model, our perceptron has 10 neurons. For the other models, a design was implemented with 3 hidden layers, each containing a different number of neurons. TableII and Fig.6 illustrates the classification results of these models.

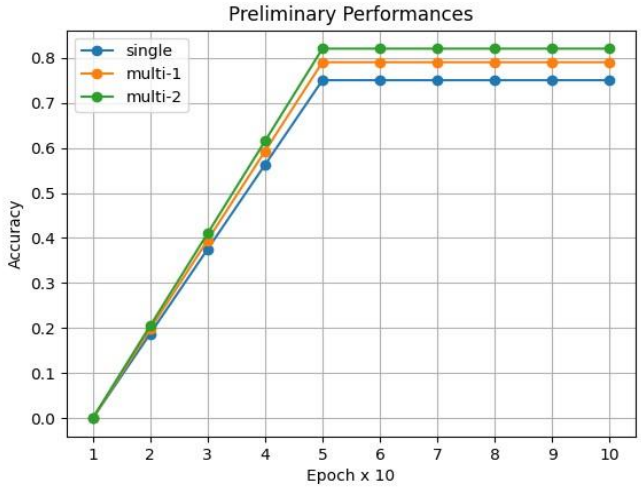


Fig. 6. Performances of preliminary experiments.

TABLE II. Preliminary Performances

Function	Single	Multi-1	Multi-2
Accuracy	0.75	0.79	0.82

While accuracy values above 80% were achieved for this complex problem, these accuracy rates may not always be sufficient and satisfactory. This consideration motivated us to create a model based on cooperative coevolution. In this model, we adopted a new approach by extending the traditional neural network architecture and applying a strategy based on evolutionary computation. In this context, we added three different hidden layers to make the neural network architecture more complex. These layers were designed to increase the learning capacity of the model and capture more intricate relationships. The decision to apply evolutionary theory outputs to the network as an input is a unique strategy aimed at enhancing the model's performance. This approach allows the outputs generated by genetic algorithms to be more effectively integrated into the learning process of the neural network. Consequently, it is anticipated that the features obtained during the evolutionary process will contribute to the neural network gaining better generalization capabilities. Fig.7 shows the classification performance and Fig.8 shows the error rates of proposed model.

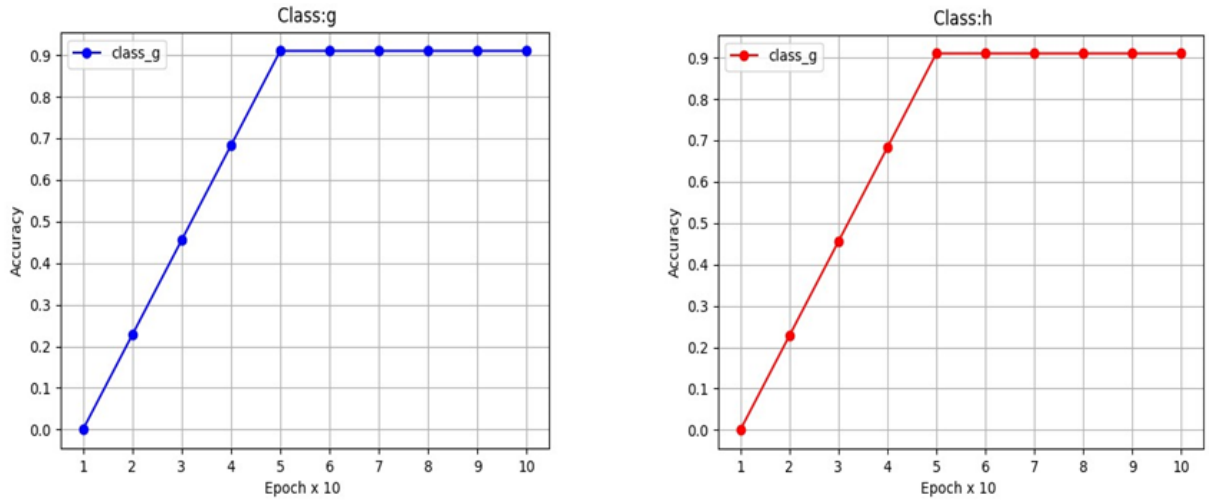


Fig. 7. Model performances.

This model was designed to achieve better success, particularly in solving complex problems. The incorporation of different hidden layers and the evolutionary computation strategy aim to enhance the model's ability to unravel deeper and more hidden relationships in the dataset, ultimately striving for higher accuracy. This innovative approach provides a new perspective on optimizing the performance of neural network-based models.

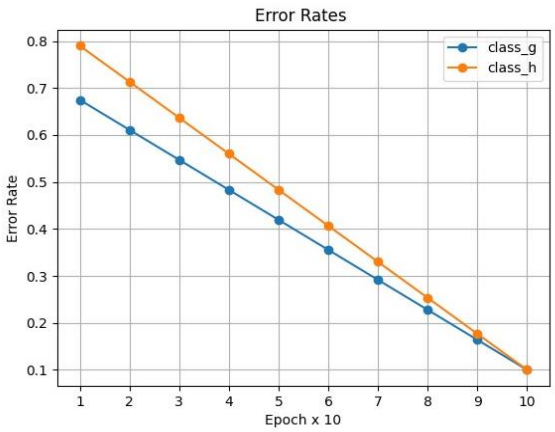


Fig. 8. Error rates

CONCLUSION

This study tackles the intricate challenges associated with analyzing data from a ground-based atmospheric Cherenkov gamma telescope, focusing on simulating and observing highenergy gamma particles. The identified challenges, including the differentiation of signals from high-energy gamma rays and handling background noise induced by cosmic-rayinitiated hadronic showers, emphasize the need for sophisticated methodologies, particularly for statistical significance in the presence of varying energy levels.

The study's outcomes contribute significantly to advancing our comprehension of high-energy gamma phenomena and offer nuanced solutions for effective data analysis. The developed cooperative coevolution-based artificial neural network model, boasting a classification accuracy exceeding 91%, stands out as a successful solution to the scientific problems at hand. This model not only effectively separates gamma rays from background noise but also holds promise for future research in refining the analysis of data generated by atmospheric Cherenkov gamma telescopes, ultimately enhancing the detection of high-energy gamma particles.

REFERENCES

[1] E. Yücesan, M. A. Erkan, A. Deveci, and ̇I. T. Medeni, "Bekenbey ai: Innovative solutions at the intersection of deep learning and law," *Sivas Cumhuriyet Üniversitesi Mühendislik Fakültesi Dergisi*, vol. 2, no. 2, pp. 185–192, 2024.

[2] J. Zupan, "Introduction to artificial neural network (ann) methods: what they are and how to use them," *Acta Chimica Slovenica*, vol. 41, pp. 327– 327, 1994.

[3] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[4] P. Chinatamby and J. Jewaratnam, "A performance comparison study on pm2. 5 prediction at industrial areas using different training algorithms of feedforward-backpropagation neural network (fbnn)," *Chemosphere*, vol. 317, p. 137788, 2023.

[5] P. R. Das and K. Boruah, "Gamma hadron separation method in groundbased gamma ray astronomy using simulated data," *Indian Journal of Physics*, vol. 97, no. 2, pp. 347–357, 2023.

[6] R. Urbanowicz, R. Zhang, Y. Cui, and P. Suri, "Streamline: A simple, transparent, end-to-end automated machine learning pipeline facilitating data analysis and algorithm comparison," in *Genetic Programming Theory and Practice XIX*, pp. 201–231, Springer, 2023.

[7] L. Vanneschi and S. Silva, "Genetic programming," in *Lectures on Intelligent Systems*, pp. 205–257, Springer, 2023.

[8] L. Rodriguez-Coayahuitl, A. Y. Rodríguez-González, D. Fajardo-Delgado, and M. G. S. Cervantes, "Problem decomposition strategies and credit distribution mechanisms in modular genetic programming for supervised learning," *IEEE Transactions on Evolutionary Computation*, 2025.

[9] X. Yao and S. Y. Chong, "Principal approaches of coevolution: Competitive and cooperative," in *Coevolutionary Computation and Its Applications*, pp. 69–91, Springer, 2025.

[10] T. C. John, Q. Ul Ain, H. Al-Sahaf, and M. Zhang, "Genetic programming with co-operative co-evolution for feature manipulation in basal cell carcinoma identification," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pp. 254–269, Springer, 2025.

[11] Ü. Karabıyık, "Matematik egitiminde yenilikçi bir yaklas¸ım: Chatgpt'nin˘ rolü," *Us¸ak Üniversitesi Eg˘itim Aras¸tırmaları Dergisi*, vol. 10, no. 1, pp. 26–46, 2024.

[12] G. Marchesini, B. R. Webber, G. Abbiendi, I. Knowles, M. H. Seymour, and L. Stanco, "Herwig 5.1-a monte carlo event generator for simulating hadron emission reactions with interfering gluons," *Computer Physics Communications*, vol. 67, no. 3, pp. 465–508, 1992.

[13] F. Romero-López, "Multi-hadron interactions from lattice qcd," *arXiv preprint arXiv:2212.13793*, 2022.

[14] P. Reynolds and D. Fegan, "Neural network classification of tev gammaray images," *Astroparticle Physics*, vol. 3, no. 2, pp. 137–150, 1995.

[15] S. Mangano, C. Delgado, M. I. Bernardos, M. Lallena, J. J. Rodríguez Vázquez, and C. Consortium, "Extracting gamma-ray information from images with convolutional neural network methods on simulated cherenkov telescope array data," in *Artificial Neural Networks in Pattern Recognition: 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September 19–21, 2018, Proceedings 8*, pp. 243–254, Springer, 2018.

[16] uciMachineLearning, "UCI Machine Learning Repository — archive.ics.uci.edu." https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope. [Accessed 08-01-2024].

A Systematic Evaluation of Patch-Based 3D ResUNet and TransUNet Architectures for Multi-Stage Pancreas Segmentation

Hasan Basri Öksüz[1], Rahime Ceylan[2]

[1] *Department of Electronics and Automation, Konya Technical University, Vocational School of Technical Sciences, Institute of Graduate Studies, Konya Technical University*, Konya, Turkey, hboksuz@ktun.edu.tr

[2] *Department of Electrical and Electronics Engineering, Konya Technical University*, Konya, Turkey, rceylan@ktun.edu.tr

ABSTRACT

In this study, a multi-stage patch-based deep learning approach is proposed for the pancreas segmentation problem. In the first stage, a 3D ResUNet is employed on 256×256 CT images to localize the coarse region of the pancreas. In the second stage, fine segmentation is performed within this region using ResUNet and TransUNet models, and their results are compared. Experimental findings demonstrate that the proposed method achieves high accuracy and robust performance on both the MSD Task07_Pancreas and FLARE22 datasets. Furthermore, the study provides a computationally efficient alternative for systems with limited hardware resources and highlights its advantages over existing methods in the literature.

Keywords: *Pancreas Segmentation, Deep Learning, 3D ResUNet, TransUNet*

INTRODUCTION

Computed tomography (CT) plays a critical role as an auxiliary technique in the diagnosis of pancreatic diseases by enabling the characterization of lesion locations and their relationships with surrounding tissues. Consequently, accurate segmentation of the pancreas from CT images is of great importance for preoperative diagnosis, treatment planning, and postoperative follow-up of pancreatic diseases. The challenges in pancreas segmentation can be attributed to the following factors: [1] :

- A time-consuming and subjective process: Radiologists are required to manually annotate abdominal CT scans slice by slice, a task that demands advanced spatial reasoning and extensive clinical expertise. This inherently introduces subjectivity into lesion diagnosis and contributes to inter-observer variability in the results.
- Complex tissue structure: Since the pancreas occupies less than 0.5% of the abdominal CT volume, it is difficult to delineate from adjacent organs and surrounding tissues, making accurate identification particularly challenging.
- Ambiguous boundaries and individual morphological variations: Unlike larger organs such as the liver or lungs, the pancreas exhibits less distinct boundaries and a high degree of inter-individual morphological variability. This uniqueness further complicates the accurate and effective segmentation of the pancreas and its lesions.

With advances in modern medicine and imaging technologies, Deep Convolutional Neural Networks (DCNNs) have emerged as a reliable and effective approach for pancreatic segmentation. DCNNs excel at identifying subtle pixel-level differences that may elude the human eye and enable the modeling of complex feature representations. Nevertheless, meeting the expectations of clinicians remains a significant challenge for these technologies. DCNNs are anticipated to provide precise and robust segmentation of the pancreas.

Deep convolutional neural networks require substantial computational power to ensure accurate pixel classification. This demand not only prolongs processing time but also increases hardware resource utilization, thereby imposing a significant computational overhead [1]. Moreover, the large volume of data generated during the training and deployment of these models becomes a limiting factor, particularly restricting their applicability in real-time and clinical settings.

In the segmentation of small and morphologically complex organs such as the pancreas, it is recommended to employ more efficient strategies rather than performing direct segmentation at full resolution. In this context, the widely adopted *coarse-to-fine* paradigm has gained particular attention in the literature. This approach first identifies the approximate location of the organ (coarse localization), followed by high-resolution and detailed segmentation within the defined region (fine segmentation). Such a two-stage framework not only reduces computational costs but also enhances segmentation accuracy, thereby improving overall performance [2].

In this study, patch-based deep learning approaches were investigated for pancreatic segmentation. To mitigate the high computational cost of full-volume 3D models, instead of processing the entire volumetric data directly, volumetric patch blocks generated from consecutive slices were used as model inputs, with the corresponding segmentation labels produced as outputs. This strategy enables effective utilization of 3D contextual information while improving computational efficiency.

The proposed segmentation pipeline was designed as a two-stage framework. In the first stage, the coarse localization of the pancreas was obtained using a patch-based ResUNet architecture on region-of-interest (ROI) images extracted at a resolution of 256×256. In the second stage, detailed segmentation was performed within the localized region using 96×176 patch blocks, employing both the ResUNet and the Transformer-based TransUNet architectures.

Experimental results were evaluated through comparative analysis both among the proposed methods and against state-of-the-art approaches reported in the literature. The primary aim of this study is to systematically investigate the impact of deep learning architectures on pancreatic segmentation, to identify the strengths and limitations of the models, and to discuss their clinical applicability, particularly for challenging organs such as the pancreas.

The main contributions of this study can be summarized as follows:

- A multi-stage (coarse-to-fine), patch-based deep learning framework is proposed for the pancreatic segmentation problem.
- In the first stage, the coarse localization of the pancreas was effectively determined using a patch-based ResUNet architecture on 256×256 CT images.
- In the second stage, fine segmentation was carried out within the localized region using patch-based ResUNet and TransUNet models, and their performances were comparatively evaluated.
- The proposed systematic framework provides a computationally efficient yet accurate alternative for segmentation systems operating under limited hardware resources.
- The obtained results were compared against state-of-the-art methods in the literature, highlighting the advantages of the proposed approach.

LITERATURE REVIEW

Abdominal organ segmentation can generally be divided into two main categories: model-based and learning-based approaches [3]. Model-based methods include atlas registration, statistical shape models, and energy function minimization algorithms [4]. These approaches aim to capture anatomical variations from training data and adapt them to new images [5]. However, they often exhibit limited performance for organs such as the pancreas, which present ambiguous boundaries and low contrast [3]. Key challenges arise from the pancreas's small and irregular structure, high anatomical variability, and close spatial relationships with neighboring organs such as the duodenum and gallbladder [6].

To overcome these limitations, learning-based approaches have been developed, which perform segmentation by directly learning meaningful features from labeled computed tomography (CT) images [3]. Depending on the availability of labels, learning-based methods can be categorized into supervised, semi-supervised, and unsupervised learning. In recent years, numerous studies have demonstrated that deep learning (DL)-based semantic segmentation networks outperform traditional methods such as intensity-based thresholding, morphological operations, and geometric analysis in medical image segmentation [7], [8]. With the recent advancements in deep learning, convolutional neural networks (CNNs) have emerged as effective learning-based approaches for various medical imaging tasks, including classification, detection, and

segmentation. In segmentation applications, CNN-based methods are generally classified into 2D, 2.5D, and 3D models. 2D models process CT volumes slice by slice in the axial, sagittal, or coronal planes and use these slices as input to the network [9]. Moreover, there are networks trained using only axial plane slices [10]. While these approaches are computationally efficient, they may not sufficiently capture inter-slice contextual information [11]. On the other hand, 3D models directly process volumetric data, thereby providing richer spatial context; however, their high computational cost and demand for large amounts of annotated data remain significant limitations [6]. Patch-based approaches, developed to strike a balance between these two extremes, aim to capture 3D contextual information while offering reduced memory consumption [12].

Although 3D networks demonstrate strong performance in learning volumetric features, their heavy computational load and reliance on large annotated datasets pose limitations in clinical applications. In contrast, 2D models attempt to achieve volumetric segmentation by combining outputs derived from separately processed images in the three anatomical planes. However, for organs such as the pancreas, which are small and have ambiguous boundaries, these models can be adversely affected by background interference [9].

Deep learning approaches for pancreatic segmentation are generally categorized into two groups: direct methods and two-stage methods [13]. In direct methods, the segmentation is performed in a single step directly on the image, whereas two-stage methods first conduct a coarse localization of the pancreas, followed by fine segmentation using a more precise model within the localized region [14]. Such multi-stage frameworks not only facilitate accurate delineation of the pancreas—an organ characterized by its similarity in intensity and structure to surrounding tissues—but also offer the advantage of reducing computational overhead.

Several two-stage strategies have been proposed in the literature to address this challenge. For example, Roth et al. [15], developed a two-stage 3D U-Net model for multi-organ abdominal segmentation. In this approach, coarse localization of the organ regions was first performed, followed by detailed segmentation in the second stage, achieving a Dice Similarity Coefficient (DSC) of 82.2% for the pancreas. However, performing both coarse and fine segmentation within the same network architecture resulted in nearly twice the memory requirements compared to conventional architectures [16]. imilarly, Yang et al. [17] introduced a cascaded neural network for pancreatic segmentation that exploits both intra-sectional and inter-sectional information. In this framework, a Fully Convolutional Network (FCN) was employed to extract intra-sectional features, while a Recurrent Neural Network (RNN) was used to model inter-sectional relationships. Liu et al. [18] proposed a method in which candidate pancreatic regions were first identified during the coarse segmentation stage using superpixel-based patch classification. Subsequently, five separate 2.5D U-Net architectures—each trained with different loss functions—were employed for segmentation, and their outputs were combined through an ensemble model to obtain the final result. In another study, Hu et al. [19], applied multi-atlas registration for coarse segmentation, while fine segmentation was performed by integrating a patch-based 3D CNN with three slice-based 2D CNNs. Probability maps generated by the 3D CNN were used to define a pancreas bounding box, which was then fused with the original CT image and fed into three consecutive 2D U-Nets. To further refine boundary accuracy, a third stage incorporating a 3D level-set method was applied to the segmentation outputs.

In this study, a two-stage patch-based approach is proposed, utilizing two different datasets and two distinct deep learning architectures. To comprehensively evaluate the effectiveness of the proposed method, not only quantitative metrics but also qualitative results were analyzed. In this context, segmentation outputs of representative cases with the highest and lowest performance in the dataset were presented, thereby highlighting the strengths and weaknesses of the model from a visual perspective.

MATERIALS AND METHODS

The datasets used, preprocessing steps, architectural details, and the training and validation strategies are presented in detail in this section.

A. Datasets

1) MSD Task07 Pancreas

The Task07 Pancreas dataset from the Medical Segmentation Decathlon (MSD) has emerged as a widely used benchmark reference in the field of pancreatic segmentation. The dataset consists of a total of 281 contrast-enhanced abdominal CT volumes, of which 240 were used for training and 41 for testing in this study. The training set includes manually annotated reference segmentation masks for the pancreas. Since the images were collected from multiple clinical centers, they exhibit substantial variability in resolution, contrast, noise levels, and scanner parameters. This diversity makes the dataset particularly well-suited for developing and evaluating generalizable deep learning models.

2) FLARE22

The FLARE22 (Fast and Low Resource Abdominal Organ Segmentation 2022) dataset is a recent benchmark designed for the automatic segmentation of the pancreas and other abdominal organs. It comprises a large number of abdominal CT volumes with both high- and low-contrast quality, collected from multiple clinical centers. Consequently, the dataset exhibits substantial variability in terms of resolution, contrast, noise levels, and scanner parameters. Owing to these characteristics, the FLARE22 dataset is widely employed to evaluate the generalizability of deep learning models, computational efficiency, and the effectiveness of different preprocessing strategies. The properties of the datasets used in this study are summarized in Table I.

TABLE I. DATASET CHARACTERISTICS

Characteristics	MSD Task07_Pancreas	FLARE22
Number of patients	281	50
In-plane spatial resolution (x, y)	512 x 512	512 x 512
Number of slices in depth (z)	[37 - 751]	[71 - 113]
Mean z-dimension (number of slices)	95	96
Pixel spacing along x-axis (mm/voxel), Left–Right [min–max]	[0.61 – 0.98]	[0.64 – 0.98]
Pixel spacing along y-axis (mm/voxel), Anterior–Posterior [min–max]	[0.61 – 0.98]	[0. 64 – 0.98]
Slice thickness along z-axis (mm)	[0.7- 7.5]	[2.5 - 5]

B. 3B ResUNet

The 3D ResUNet is a deep learning architecture developed to achieve highly accurate segmentation of volumetric medical images [20]. The model employs residual blocks and skip connections within an encoder–decoder framework to effectively process 3D features. The encoder consists of layers with 64, 128, 256, and 512 filters, where max-pooling is applied only in the x–y plane to prevent information loss along the z-axis. The bottleneck contains a residual block with 1024 filters. In the decoder stage, upsampling is performed in the x–y plane using Conv3DTranspose and combined with the corresponding skip connections from the encoder; the filter numbers are 512, 256, 128, and 64, respectively. The final layer applies either a sigmoid or softmax activation function, depending on the number of classes. This architecture preserves 3D contextual information while enabling detailed and efficient segmentation.

C. 3B TransUNET

The 3D TransUNet architecture is a deep learning model that integrates the encoder–decoder structure of the conventional 3D U-Net with Transformer-based attention mechanisms [21]. In the encoder, hierarchical feature representations are extracted from the input volume through sequential Conv3D or residual blocks. The Transformer module, positioned between the encoder and decoder, captures global contextual information and models spatial relationships. In the decoder stage, Conv3DTranspose layers and residual blocks are employed to upsample the feature maps, which are then

fused with the corresponding skip connections from the encoder. This design enables the effective integration of both local and global information, making it particularly suitable for the segmentation of small and irregular organs such as the pancreas. In the final layer, a sigmoid activation function is applied for binary segmentation.

D. Evaluation Metrics for Segmentation

In this study, overlap-based metrics, namely the Dice Similarity Coefficient (DSC) and the Jaccard Index (JI), were employed for evaluation. The DSC measures the degree of overlap between the automatic segmentation and the reference mask, and is one of the most widely used metrics for assessing segmentation accuracy in the literature. The JI is defined as the ratio of the intersection volume to the union volume between the segmentation result and the reference data.

$$DSC = \frac{2|A \cap B|}{|A| + |B|} = \frac{2TP}{2TP + FP + FN} \quad (1)$$

$$JI = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \quad (2)$$

In (1) and (2), True Positive is denoted as TP, True Negative as TN, False Positive as FP, and False Negative as FN.

E. Hyperparameter Configuration in Segmentation Models

The experiments were conducted on a computational system equipped with an Intel® Core™ i7-12700K CPU operating at 2.10 GHz, 32 GB of RAM, and an NVIDIA GeForce RTX 3060 GPU with 12 GB of dedicated VRAM. All processes were carried out on the Windows 10 operating system using TensorFlow-Keras version 2.5.0 and the Adam optimization algorithm. To address class imbalance and improve performance in challenging regions during network training, the Focal Dice Loss (FDL) function was employed. FDL combines the regional overlap sensitivity of Dice Loss (DL) with the capability of Focal Loss (FL) to focus on hard-to-classify examples.

$$\mathrm{FL} = -[(1 - p)^{\gamma} \cdot y \cdot log(p) + p^{\gamma} \cdot (1 - y) \cdot log(1 - p)] \quad (3)$$

$$\mathrm{FDL} = 0.5(FL) + 0.5(DSC) \quad (4)$$

In the loss function defined in (3), p denotes the predicted probability, while y represents the ground-truth label. The balanced integration of DL and FL reduces class imbalance, facilitates the learning of fine structural details, and enhances the focus on difficult regions, thereby improving overall performance. This approach allows the model to preserve global segmentation accuracy while achieving more precise delineation of ambiguous and small structures.

To reduce computational cost and improve training reliability, a batch size of 1 was employed. This choice is consistent with previous studies reporting successful training with small batch sizes [22]. Furthermore, a 5-fold cross-validation scheme was adopted, and training was performed for 10 epochs. The learning rate was set to 0.0001, in line with optimization strategies reported in related work [23]–[25]. This low learning rate was chosen to ensure both stable convergence and effective learning. In addition, for models trained on both coarse and fine segmentation tasks, a depth of 8 slices along the z-axis was utilized to enable more efficient processing of three-dimensional volumes.

RESULTS

F. Coarse Segmentation Results

In the coarse segmentation stage, two separate experimental studies were conducted. In the first study, a 3D ResUNet network was trained using only the first 240 samples selected from the MSD Task07_Pancreas dataset. In the second study, training was performed on the same 240 MSD samples supplemented with 10 additional cases from the FLARE22 dataset. In both studies, testing was carried out on the last 41 samples of the MSD Task07_Pancreas dataset and the last 40 samples of the FLARE22 dataset.

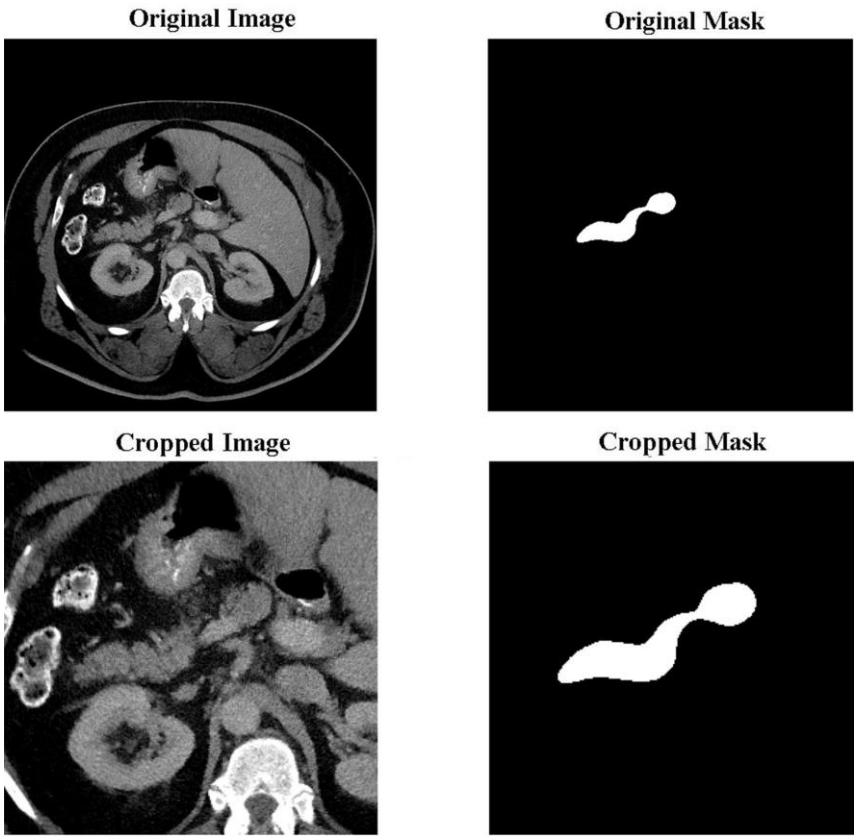


Fig. 1. An original CT image, its corresponding ground-truth mask, a cropped 256×256 image, and the associated cropped mask sample

In the coarse segmentation stage, 256×256 windows covering the pancreatic region were first extracted, following the approach proposed by Kurnaz et al. [10]. Fig. 1 presents, for a representative slice, the original image, the corresponding ground-truth mask, the cropped image, and the cropped mask. Subsequently, the image intensities were normalized within the range of –240 to 100 HU. The resulting preprocessed data were then used for network training, and the outcomes were reported in a comparative manner. The results corresponding to the 256×256 window size are provided in Table II, while those obtained by reinserting the cropped patches into a 512×512 window are presented in Table III.

TABLE II. 256x256xd COARSE SEGMENTATION RESULTS

Training Data	Test Data	Model	DSC %	JI %
MSD Task07_Pancreas	MSD Task07_Pancreas	ResUNet	66.27±23.61	53.44±22.31
MSD Task07_Pancreas	FLARE22	ResUNet	85.36±9.89	75.56±12.94
MSD Task07_Pancreas ve FLARE22	MSD Task07_Pancreas	ResUNet	62.73±25.24	49.95±23.40
MSD Task07_Pancreas ve FLARE22	FLARE22	ResUNet	82.91±11.31	72.15±14.00

TABLE III. 512x512 COARSE SEGMENTATİON RESULTS

Training Data	Test Data	Model	DSC %	JI %
MSD Task07_Pancreas	MSD Task07_Pancreas	ResUNet	67.25±24.06	54.78±23.11
MSD Task07_Pancreas	FLARE22	ResUNet	86.41±9.05	77.04±12.22
MSD Task07_Pancreas ve FLARE22	MSD Task07_Pancreas	ResUNet	63.76±25.80	51.35±24.38
MSD Task07_Pancreas ve FLARE22	FLARE22	ResUNet	83.92±10.52	73.51±13.42

For the model trained solely on the MSD dataset, the DSC increased from 66.27% with 256×256 windows to 67.25% with full 512×512 windows. Similarly, the JI improved from 53.44% to 54.78%. This indicates that reinserting the cropped results into the full window provides a slight performance gain. On the FLARE22 test set, the same model achieved an improvement in DSC from 85.36% to 86.41%, and in JI from 75.56% to 77.04%.

When 10 FLARE22 cases were added to the MSD training data, the performance on the MSD test set increased from 62.73% to 63.76% in terms of DSC, and from 49.95% to 51.35% in terms of JI when moving from 256×256 to 512×512 windows. A similar trend was observed on the FLARE22 test set, where the DSC rose from 82.91% to 83.92% and the JI from 72.15% to 73.51%.

These findings indicate that adding a limited amount of heterogeneous data does not necessarily improve generalization performance and, in some cases, may even cause fluctuations in test performance due to distributional differences in the training data.

Following coarse segmentation, 3D Connected Component Analysis was applied to the predicted masks to extract the pancreas region of interest (ROI). The statistical properties of the ROIs obtained from the MSD Task07_Pancreas dataset are presented in Table IV. Based on these characteristics, the window size for fine segmentation was determined.

TABLE IV. THE STATISTICAL PROPERTIES OF THE CROPPED WINDOWS

Data	Min -Max h	Min -Max w	Min -Max d	Mean - Std h	Mean - Std w	Mean- Std d
Training	44-156	74-257	16-80	91.07±21.05	157.74±31.34	34.25±9.13
Test	44-172	100-229	14-152	90.44±23.67	166.90±28.70	38.59±21.61

G. Fine Segmentation Results

For fine segmentation, a 3D pancreas region of interest was delineated for each patient and resampled to a window size of 96×176. A representative slice is illustrated in Fig. 2.

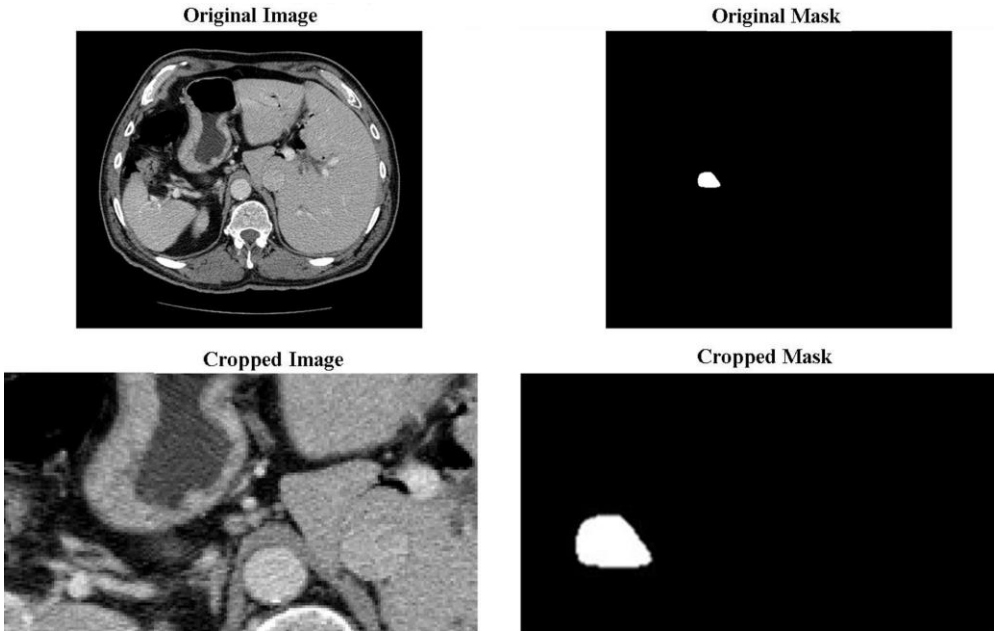


Fig. 2. Example of the original CT image, the corresponding reference mask, the cropped region of interest, and the cropped mask.

During the evaluation of the training and test datasets, the average height and width of the regions of interest were found to be approximately 90 and 162 pixels, respectively. Since the employed network architectures consist of four encoder–decoder layers, the input dimensions needed to be multiples of 16 to ensure compatibility with the downsampling and upsampling operations. Accordingly, a window size of 96 × 176 pixels—the closest dimensions to the observed averages—was selected. The extracted regions of interest and their corresponding masks were rescaled, either upsampled or

downsampled, to match this size. Following this preprocessing step, the training data were normalized within the range of –240 to 100 HU, and the 3D TransUNet and 3D ResUNet architectures were trained. For evaluation, both patch-level scores and whole-image scores—obtained by reassembling the rescaled patches into their original positions—were computed. The corresponding results are reported in Tables V and VI.

TABLE V. 176x96xd FINE SEGMENTATION RESULTS

Training Data	Test Data	Model	DSC %	JI %
MSD Task07_Pancreas	MSD Task07_Pancreas	ResUNet	67.35±20.14	53.84±20.44
	FLARE22	ResUNet	83.43±12.94	73.29±15.58
	MSD Task07_Pancreas	TransUNet	60.90±22.04	47.07±21.06
	FLARE22	TransUNet	74.25±17.12	61.63±19.17

TABLE VI. 512x512 FINE SEGMENTATION RESULTS

Training Data	Test Data	Model	DSC %	JI %
MSD Task07_Pancreas	MSD Task07_Pancreas	ResUNet	68.22±20.57	55.05±21.20
	FLARE22	ResUNet	84.23±12.51	74.37±15.15
	MSD Task07_Pancreas	TransUNet	61.95±22.64	48.44±22.02
	FLARE22	TransUNet	75.09±16.81	62.63±18.95

Table V presents the fine segmentation results obtained with ROIs of size 176 × 96 × d. The ResUNet model achieved a Dice score of 67.35% and a Jaccard index of 53.84% on the MSD test set, and 83.43% and 73.29%, respectively, on the FLARE22 test set. In comparison, the TransUNet model reached a Dice score of 60.90% on the MSD test set and 74.25% on the FLARE22 test set, exhibiting lower performance than ResUNet across both datasets.

Table VI reports the results obtained with ROIs of size 512 × 512. For the ResUNet model, the Dice score reached 68.22% with a Jaccard index of 55.05% on the MSD test set, and 84.23% with 74.37%, respectively, on the FLARE22 test set, reflecting a slight performance improvement compared to Table 5. A similar trend was observed for the TransUNet model, which achieved a Dice score of 61.95% on the MSD test set and 75.09% on the FLARE22 test set.

Overall, the fine segmentation results yielded small but consistent improvements compared to coarse segmentation. For instance, in the case of ResUNet, the Dice score on the MSD test set increased from 67.25% to 68.22%. Similarly, modest gains were observed in the Jaccard index. These findings suggest that fine segmentation on high-resolution ROIs enables the capture of more detailed anatomical information, offering an advantage over coarse segmentation, albeit with only limited performance gains. Notably, on the FLARE22 test set, the coarse segmentation results outperformed the fine segmentation outcomes.

TABLE VII. SEGMENTATION PERFORMANCE REPORTED IN PREVIOUS STUDIES ON THE MSD TASK07_PANCREAS DATASET.

Study	Dataset	DSC%	JI%
Xie et al.[26]	MSD	73.6	59.1
Chen et al.[27]	MSD	76.6	62.6
Cao et al.[28]	MSD	75.53	-
Liang et al. [29]	MSD	82.7	71.25
This study – ResUNet	MSD	68.22	55.05
This study – ResUNet	MSD training / FLARE22 test	84.23	74.37

Table VII presents segmentation results reported in the literature on the MSD Task07_Pancreas dataset. Reported Dice scores in previous studies generally range between 73% and 83%. Specifically, Xie et al. [26] achieved 73.6%, Chen et al. [27] reported 76.6%, Cao et al. [28] obtained 75.53%, and Liang et al. [29] reached 82.7%.

In the present study, the ResUNet model trained solely on the MSD dataset achieved a Dice score of 68.22% and a Jaccard index of 55.05% on the same dataset. Although these results fall below the highest values reported in the literature, it is noteworthy that the method was implemented in a patch-based manner under hardware constraints, which may have influenced performance.

In contrast, the model trained on the MSD dataset achieved a Dice score of 84.23% and a Jaccard index of 74.37% on the FLARE22 test set, demonstrating the generalizability of the proposed method across different datasets. This outcome indicates that the approach can remain competitive with many state-of-the-art methods in the literature, particularly under domain shift scenarios.

The results of the 3D ResUNet model in the fine segmentation stage are visualized on the best- and worst-performing patient cases selected from the MSD and FLARE22 datasets. For each patient, three representative axial slices are shown, organized into four columns: (1) input CT image, (2) reference mask, (3) model-predicted mask, and (4) error map (red = over-segmentation, green = under-segmentation). In addition, three-dimensional renderings of the reference and predicted masks are provided for each case. Fig. 3 illustrates the best case from the MSD dataset (Patient #389, Dice = 91.42%), Fig. 4 the worst case (Patient #409, Dice = 9.11%), Fig. 5 the best case from the FLARE22 dataset (Patient #40, Dice = 96.13%), and Fig. 6 the worst case (Patient #14, Dice = 28.75%).

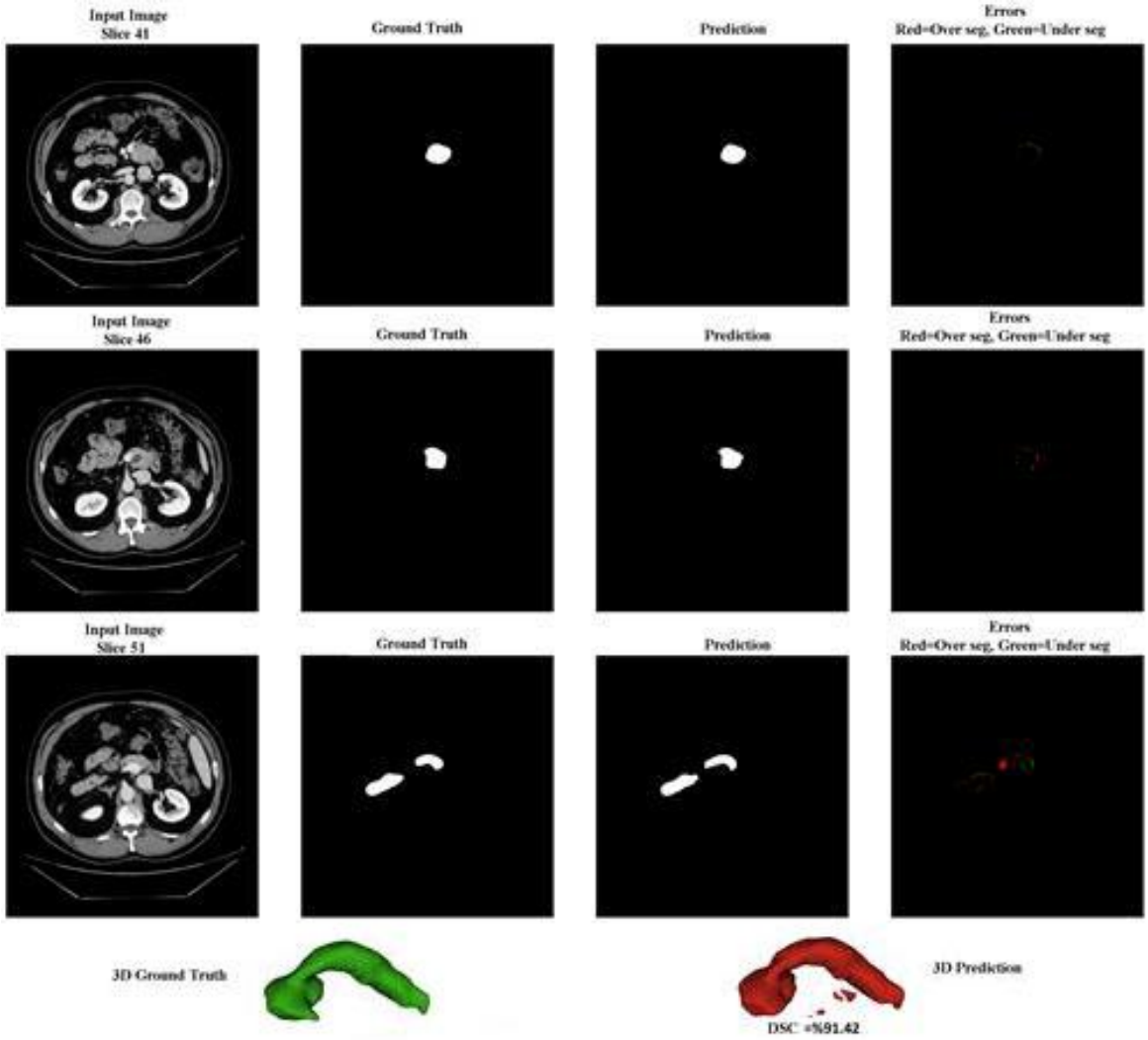


Fig. 3. The best patient images in fine segmentation were achieved in the MSD test set

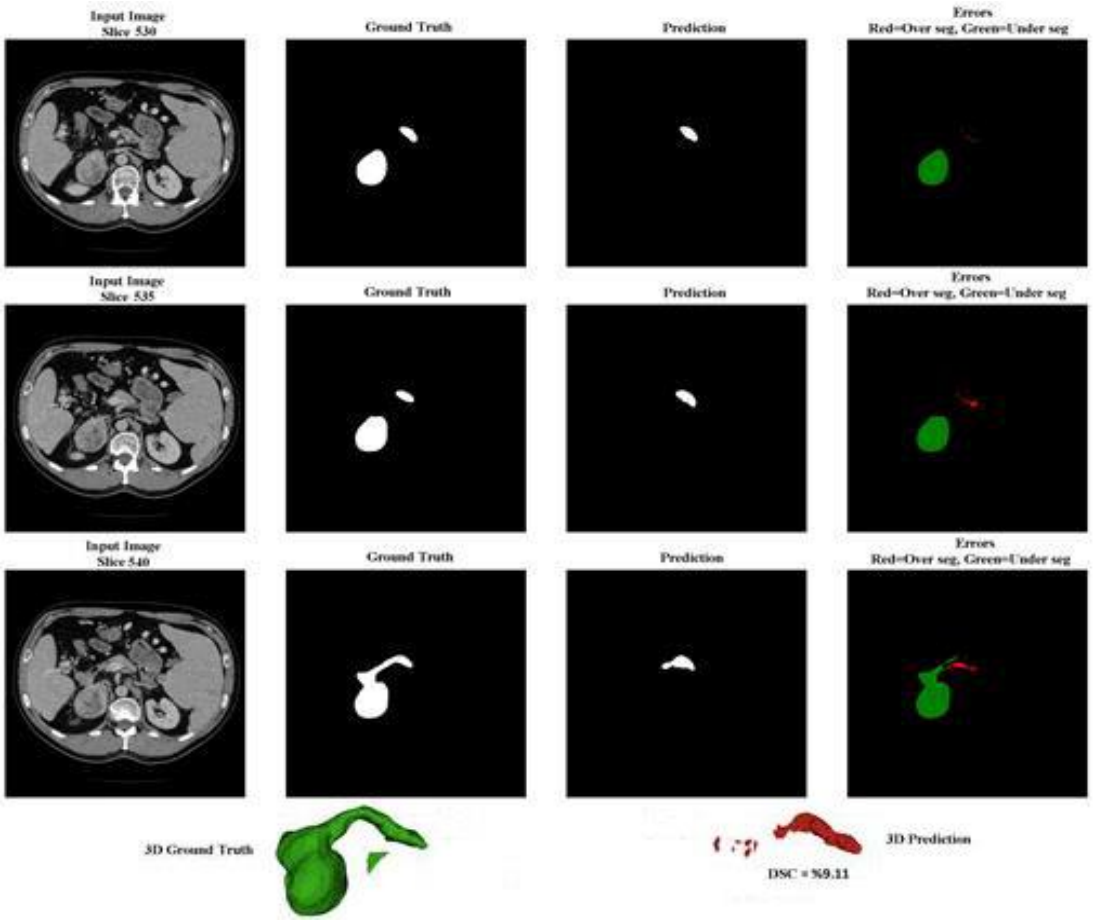


Fig. 4. The worst patient images in fine segmentation were achieved in the MSD test set

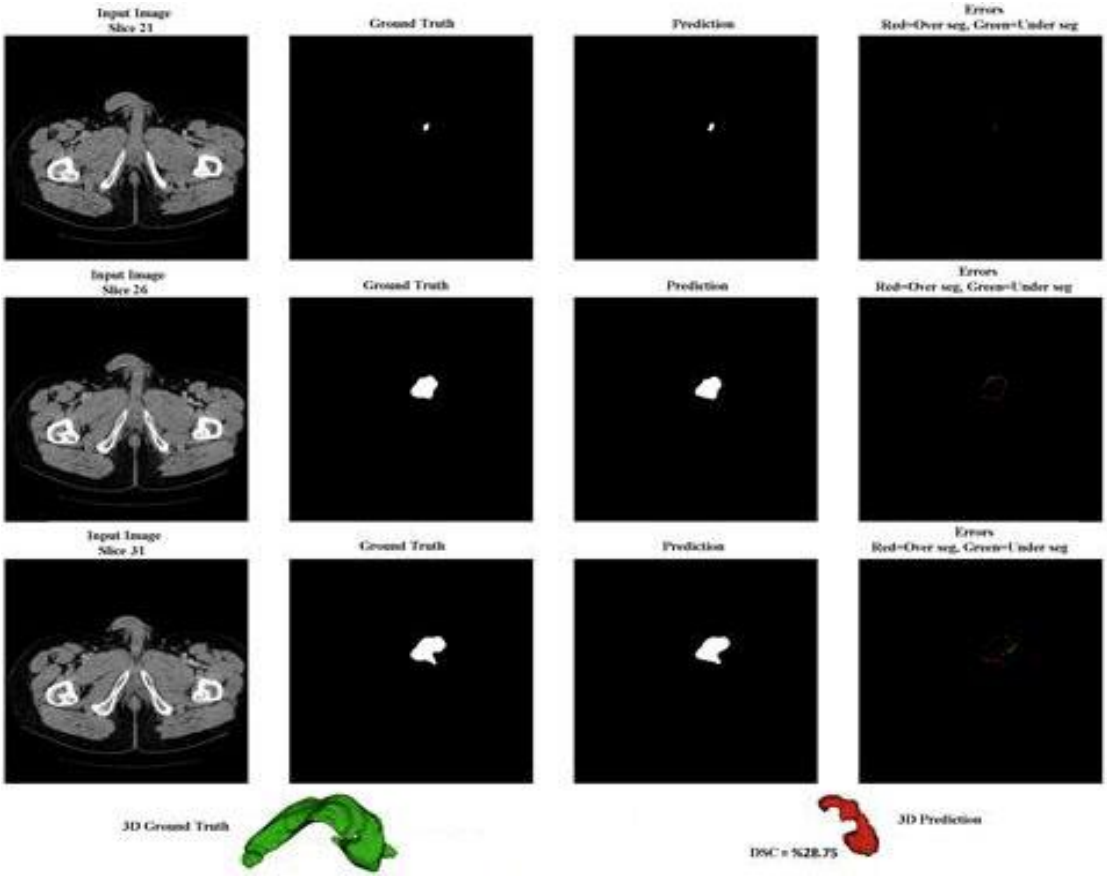


Fig. 5. The best patient images were achieved in fine segmentation in the FLARE22 dataset

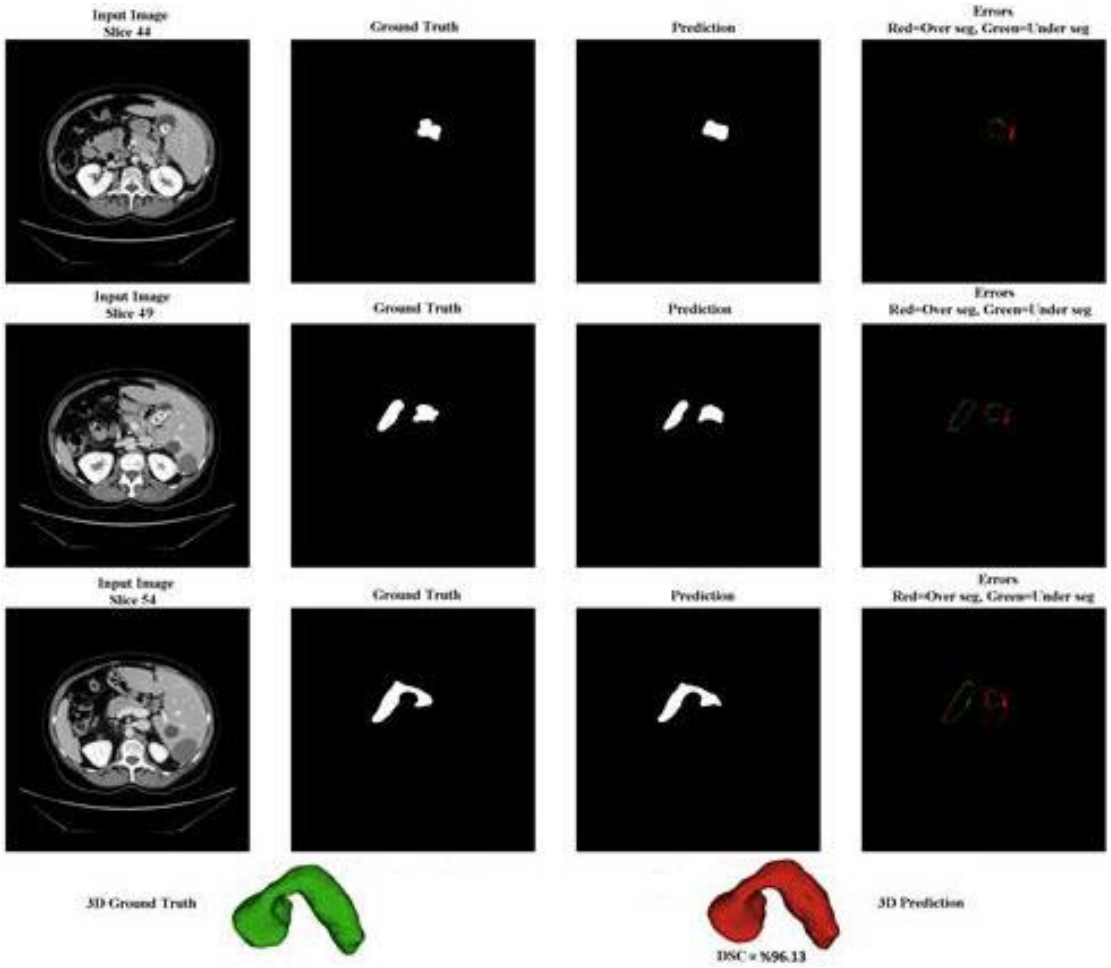


Fig. 6. The worst patient images in fine segmentation were achieved in the FLARE22 dataset

REFERENCES

[1] L. Cao and J. Li, "Strongly representative semantic-guided segmentation network for pancreatic and pancreatic tumors," Biomedical Signal Processing and Control, vol. 87, no. October 2023, 2024, doi: 10.1016/j.bspc.2023.105562.

[2] Z. Fatemeh, S. Nicola, K. Satheesh, and U. Eranga, "Ensemble U-net-based method for fully automated detection and segmentation of renal masses on computed tomography images," Medical physics, vol. 47, no. 9, pp. 4032–4044, 2020.

[3] J. Ma et al., "Abdomenct-1k: Is abdominal organ segmentation a solved problem," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[4] A. Moglia, M. Cavicchioli, L. Mainardi, and P. Cerveri, "Deep Learning for Pancreas Segmentation: a Systematic Review," 2024.

[5] E. Gibson et al., "Automatic Multi-Organ Segmentation on Abdominal CT With Dense V-Networks," IEEE Transactions on Medical Imaging, vol. PP, p. 1, Feb. 2018, doi: 10.1109/TMI.2018.2806309.

[6] Y. Yan and D. Zhang, "Multi-scale U-like network with attention mechanism for automatic pancreas segmentation," PLoS One, vol. 16, no. 5, p. e0252287, 2021.

[7] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "A review of deep learning based methods for medical image multi-organ segmentation," Physica Medica, vol. 85, pp. 107–122, 2021.

[8] S.-H. Lim, Y. J. Kim, Y.-H. Park, D. Kim, K. G. Kim, and D.-H. Lee, "Automated pancreas segmentation and volumetry using deep neural network on computed tomography," Scientific Reports, vol. 12, no. 1, p. 4075, 2022.

[9] Y. Zhang et al., "A deep learning framework for pancreas segmentation with multi-atlas registration and 3D level-set," Medical Image Analysis, vol. 68, p. 101884, 2021.

[10] E. Kurnaz, R. Ceylan, M. A. Bozkurt, H. Cebeci, and M. Koplay, "A Novel Deep Learning Model for Pancreas Segmentation: Pascal U-Net," Inteligencia Artificial, vol. 27, no. 74, pp. 22–36, 2024, doi: 10.4114/intartif.vol27iss74pp22-36.

[11] Y. Wang, J. Zhang, H. Cui, Y. Zhang, and Y. Xia, "View adaptive learning for pancreas segmentation," Biomedical Signal Processing and Control, vol. 66, p. 102347, 2021.

[12] V. Asadpour, R. A. Parker, P. R. Mayock, S. E. Sampson, W. Chen, and B. Wu, "Pancreatic cancer tumor analysis in CT images using patch-based multi-resolution convolutional neural network," Biomedical Signal Processing and Control, vol. 68, p. 102652, 2021, doi: https://doi.org/10.1016/j.bspc.2021.102652.

[13] Y. Chen, C. Xu, W. Ding, S. Sun, X. Yue, and H. Fujita, "Target-aware U-Net with fuzzy skip connections for refined pancreas segmentation," Applied Soft Computing, vol. 131, p. 109818, 2022.

[14] T. A. Qureshi et al., "Morphology-guided deep learning framework for segmentation of pancreas in computed tomography images," Journal of Medical Imaging, vol. 9, no. 2, p. 24002, 2022.

[15] H. R. Roth et al., "An application of cascaded 3D fully convolutional networks for medical image segmentation," Computerized Medical Imaging and Graphics, vol. 66, pp. 90–99, 2018, doi: https://doi.org/10.1016/j.compmedimag.2018.03.001.

[16] X. Zhao et al., "Prior Attention Network for Multi-Lesion Segmentation in Medical Images," IEEE Transactions on Medical Imaging, vol. 41, no. 12, pp. 3812–3823, 2022.

[17] Z. Yang et al., "Pancreas segmentation in abdominal CT scans using inter-/intra-slice contextual information with a cascade neural network," in 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019, pp. 5937–5940.

[18] S. Liu et al., "Automatic Pancreas Segmentation via Coarse Location and Ensemble Learning," IEEE Access, vol. 8, pp. 2906–2914, 2020, doi: 10.1109/ACCESS.2019.2961125.

[19] P. Hu et al., "Automatic pancreas segmentation in CT images with distance-based saliency-aware DenseASPP network," IEEE journal of biomedical and health informatics, vol. 25, no. 5, pp. 1601–1611, 2020.

[20] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 162, pp. 94–114, 2020.

[21] J. Chen et al., "Transunet: Transformers make strong encoders for medical image segmentation," arXiv preprint arXiv:2102.04306, 2021.

[22] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in International conference on medical image computing and computer-assisted intervention, 2016, pp. 424–432.

[23] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," Nature methods, vol. 18, no. 2, pp. 203–211, 2021.

[24] X. Liu et al., "Development and validation of the 3D U-Net algorithm for segmentation of pelvic lymph nodes on diffusion-weighted images," BMC Medical Imaging, vol. 21, no. 1, p. 170, 2021, doi: 10.1186/s12880-021-00703-3.

[25] H. B. Öksüz and R. Ceylan, "A preprocessing method based on 3D U-Net for abdomen segmentation," Computers in Biology and Medicine, vol. 196, p. 110709, 2025, doi: https://doi.org/10.1016/j.compbiomed.2025.110709.

[26] L. Xie, Q. Yu, Y. Wang, Y. Zhou, E. Fishman, and A. Yuille, "Recurrent Saliency Transformation Network for Tiny Target Segmentation in Abdominal CT Scans," IEEE Transactions on Medical Imaging, vol. PP, p. 1, Jul. 2019, doi: 10.1109/TMI.2019.2930679.

[27] H. Chen, Y. Liu, Z. Shi, and Y. Lyu, "Pancreas segmentation by two-view feature learning and multi-scale supervision," Biomedical Signal Processing and Control, vol. 74, p. 103519, Apr. 2022, doi: 10.1016/j.bspc.2022.103519.

[28] L. Cao, J. Li, and S. Chen, "Multi-target segmentation of pancreas and pancreatic tumor based on fusion of attention mechanism," Biomedical Signal Processing and Control, vol. 79, p. 104170, Jan. 2023, doi: 10.1016/j.bspc.2022.104170.

[29] P. Liang, G. Xin, X. Yi, H. Liang, and C. Ding, "Automatic Pancreas Segmentation in CT Images Using EfficientNetV2 and Multi-Branch Structure," Computers, Materials & Continua, vol. 83, pp. 2481–2504, Apr. 2025, doi: 10.32604/cmc.2025.060961.

Comparative Analysis Of Open-Source Libraries In Emotoin Recognition

Miray Ataş[1], Ahmet Gürkan Yüksek[2]

[1] *Department of Computer Engineering Faculty of Engineering, Sivas Cumhuriyet University, Sivas, Türkiye, mirayatasss@gmail.com, ORCID ID: 0009-0005-7513-6054*

[2] *Department of Computer Engineering Faculty of Engineering, Sivas Cumhuriyet University, Sivas, Türkiye, agyuksek@cumhuriyet.edu.tr, ORCID ID: 0000-0001-7709-6360*

ABSTRACT

Emotion detection is an important area of research in artificial intelligence to establish more natural, effective, and trustworthy human-machine interaction. Emotion analysis, carried out through facial expressions, tone of voice, text content, and multimodal approaches, has practical applications in many fields, spanning healthcare to educational technology, driver safety to customer experience management. In this study, the widely used open-source libraries in emotion recognition research were examined comprehensively. Review was done under four wide-ranging dimensions. These are face expression-based libraries (DeepFace, OpenFace, FER, Py-Feat, Face-api, Affectiva SDK, EmoReact, EmoPy), speech-based libraries (openSMILE, PyAudioAnalysis, SpeechBrain, Essentia, DeepSpectrum, Librosa), text-based libraries (VADER, TextBlob, Transformers, NRC Emotion Lexicon, SentiwordNet, DistilBERT), and multimodal support tools (CMU Multimodal SDK, Py-Feat, Deep Multimodal Emotion Recognition, Multimodal Emotion Recognition Toolkit, EmoReact Multimodal).

All libraries were compared in terms of the emotion labels supported, features, strengths, and weaknesses offered. The findings show that single modality specialist libraries achieve high success for specific tasks, but multimodal systems offer more comprehensive treatment to emotion recognition. This paper attempts to assist researchers in selecting appropriate libraries for different modalities and provide methodological recommendations for future research.

Keywords: *Emotion recognition, facial expressions, voice analysis, text mining, multimodal emotion recognition, open-source libraries, artificial intelligence, human-machine interaction.*

INTRODUCTION

Accurate perception and understanding of human emotions are one of the vital aspects of social interaction. With advancements in artificial intelligence and machine learning processes in recent years, computers are also going to obtain the ability to perceive and interpret human emotions. Emotion recognition is hence an interdisciplinary field of study that attempts to make human-machine interaction more natural and efficient. Emotion detection is achieved through processing of collected information from different modalities such as facial expressions, tone of voice, linguistic expressions, and biophysiological signals. These technologies are applied intensively in the healthcare sector for stress and mental well-being detection, in education to monitor students' emotional states, in driver safety systems to monitor attention and fatigue, and in marketing and customer experience monitoring. However, emotion recognition is faced with some methodological and technical challenges. Inter-cultural differences, variability of human facial expressions, noise during voice recording, linguistic ambiguity during text-based analysis, and multimodal data fusion are among the most important among these challenges. As a result, it can be seen that researchers need robust software tools and libraries for

tackling different categories of data. Open-source libraries offer researchers affordable but flexible options, making way for development in this field. They are composed of a modular system. Open-source libraries pave the way for future development in this field by offering researchers both affordable and flexible options. The libraries are crucial to emotion recognition research because they are modular, have a wide community base, are updated regularly, and can be integrated smoothly. However, the performance level and functionality of different libraries vary across modalities. The aim of the proposed study is to review systematically open-source libraries utilized in emotion recognition tasks and comparatively examine their significant features, supported emotion classes, ease of use, and limitations. The study aims to help researchers and practitioners select an appropriate library based on modality and contribute methodologically to follow-up research.

METHOD and THEORETICAL EXPLANATION

A. Method

The research in this instance evaluates open-source libraries used in emotion recognition within literature reviews. Literature review was attained through scientific publications, conference papers, and software manual, ranging from 2015 to 2025. IEEE Xplore, SpringerLink, ACM Digital Library, and Google Scholar databases were tapped while executing the search process.

Three fundamental criteria were used when making the decision on which libraries to evaluate. These were that they are open source and provide access for academic/applied research, that they are used actively and have associated communities, and that they can support face, voice, text, or multimodal data in the course of emotion recognition.

The evaluation factors were the categories of emotions addressed by the libraries, the technical aspects they provided (preprocessing, modeling, labeling), level of ease of integration, performance reports, and documentation quality.

B. Theorical Explanations

Methods used in emotion detection research are usually grouped into four basic modalities:

- Facial expression-based emotion detection: Facial micro-expressions and facial muscle activity provide key indicators for the detection of basic emotions. Algorithms in this field are based on computer vision and deep learning. Convolutional neural networks (CNN) are widely used, and some libraries include emotion classification based on pre-trained models.
- Voice-based emotion detection: In speech, changes in tone, pitch, rhythm, and energy reflect an individual's mood. Feature extraction is crucial in voice-based analysis, where Mel-Frequency Cepstral Coefficients (MFCC), prosodic features, and spectral parameters are employed frequently. Open-source libraries extract these features programmatically and pass them along to classification models.
- Text-based affect detection: Affect detection from written or transcribed text is accomplished by natural language processing (NLP) methods. While older techniques employed sentiment dictionaries and statistical methods, currently, transformer-based deep learning architectures (BERT, RoBERTa, T5) are preferred. These models offer high accuracy with the ability to perform contextual sentiment analysis of text.
- Multimodal emotion recognition: As single-modality systems are limited, multimodal approaches that leverage several data sources are attracting increasing interest. According to these frameworks, facial expressions, voice cues, and text data are evaluated together so as to provide more accurate and reliable emotion estimates. Multimodal libraries manage different data types to specify a common emotion classification model.

In accordance with this theoretical framework, the next part of the study critically examines the Open-source libraries identified in detail and discusses them comparatively.

RESEARCH AND FINDINGS

A. Libraries Based on Facial Expressions

Libraries used in emotion recognition studies can be categorized into four main categories: facial expression-based, voice-based, text-based, and multimodal. The following section illustrates popular open-source as well as commercial libraries of each category in tabular form and further discusses their strengths and limitations.

Single-modality-based libraries are normally fast with high accuracy but may lack certain variations of emotion when not used in combination with more complex or multimodal systems.

Facial expressions are the most commonly used modality for emotion detection. Single-modality-based systems are generally fast with high accuracy. But when not used in conjunction with more sophisticated or multimodal systems, they become limiting, especially in detecting finer variations of emotion [1][5].

While summarized in Table 1 as being good in accuracy with pre-trained CNN models, libraries such as DeepFace [1] and Fer [2] are supplemented by tools such as OpenFace [3] and Py-Feat [4], which provide more detailed analysis features through action units and multimodal support. While commercial solutions are offered by Affectiva SDK [14], more specialized or small community-supported libraries such as EmoReact [18] and EmoPy [19] are intended for specific uses.

B. Voice-Based Libraries

Voice-based emotion recognition systems classify emotional states by the acoustic properties of speech. OpenSMILE [6] is presently the most popular library for academic research and provides a broad spectrum of features. PyAudioAnalysis [7] and Essentia [15] offer Python libraries with easy access for researchers. SpeechBrain [8], which is a recent approach, accommodates incorporation into multimodal systems through PyTorch-based deep learning models. Spectrogram-based methods such as DeepSpectrum [20] and low-level audio processing libraries such as Librosa [21] have frequently been employed in prototyping and deep learning-based research in recent years. However, most of these libraries are limited by the requirement of high accuracy GPUs and complex setups.

C. Text-Based Libraries

Text-based sentiment analysis has widespread usage in applications such as social media, customer reviews, and dialogue systems. VADER [9] and TextBlob [10] are lightweight options for fast, lexicon-based analysis. Transformer-based models (BERT, RoBERTa, T5) [11] provide much higher performance with contextual sentiment analysis. Lexicon-based tools such as NRC Emotion Lexicon [16] and SentiWordNet [22] are language-independent options frequently used in research studies. DistilBERT [23] is a lighter but efficient alternative with very low resource needs. The contrast between these libraries is provided in Table 3.

D. Multimodal Libraries

In recent years, multimodal emotion recognition systems have become more accurate through the incorporation of facial expressions, voice, and text information in combinations [12], [13]. The CMU Multimodal SDK (MMSDK) [12] is among the most used academic tools, with the capability of multi-data integration. Py-Feat [4] also offers large-scale datasets for multimodal use. Next-generation libraries such as MERT [17] and Deep Multimodal ER [13] are used in research because they possess novel modularity. Additionally, user-specific libraries of tools such as EmoReact Multimodal [24] are being developed. They contain more complex structures, thus possessing a high learning curve as well as computation costs.

TABLE 1. Libraries Based on Facial Expressions

Library	License	Supported Emotions	Key Features	Advantages	Disadvantages
DeepFace [1]	MIT	Happy, Sad, Angry, Fearful, Surprised, Disgusted, Neutral	CNN-based, pre-trained models, Python	High accuracy, easy to use	Large model size, GPU requirement
Fer [2]	MIT	Happy, Sad, Angry, Fear, Surprised, Disgust, Neutral	Python, Keras, datasets for training and testing	Simple API, rapid prototyping	Limited customization
OpenFace [3]	Apache 2.0	Basic emotion + Action units	C++/Python, real-time facial analysis	Detailed action unit measurement	Installation complexity
Py-Feat [4]	MIT	Basic and complex emotions	Python, face + behavior analysis, multimodal support	Multimodal capabilities, extensive dataset support	Learning curve for beginners
Face-api [5]	MIT	Happy, Sad, Angry, Neutral, etc.	JavaScript, browser-based	Suitable for web applications	Limited performance on large datasets
Affectiva SDK [14]		Happy, Sad, Angry, Fear, Surprise, Disgust, Neutral	Facial expressions + head movement analysis	Powerful API for commercial and research use	Not fully open source, limited access
EmoReact [18]		Basic emotions	PyTorch-based, focused on children's facial expressions	Special domain support	Limited in general use
EmoPy [19]	Apache 2.0	Happy, Sad, Angry, Fear, Surprised, Disgust, Neutral	Python + Keras-based	Easy to use, open source	Small community support

TABLE 2. Voice-Based Libraries

Library	License	Supported Emotions	Key Features	Advantages	Disadvantages
OpenSMILE [6]	Non-commerical open source	Happy, Sad, Angry, Neutral, etc.	Voice feature extraction, C++-based	High flexibility, widely used in academic settings	Complex configuration
PyAudioAnalysis [7]	Apache 2.0	Happy, Sad, Angry, Neutral, etc.	Python, feature extraction and classification	Easy to use, training materials available	More limited model variety
SpeechBrain [8]	Apache 2.0	Happy, Sad, Angry, Neutral, etc.	PyTorch-based deep learning models	Modern architecture, multimodal integration	GPU requirement
Essentia [15]	AGPLv3	Happy, Sad, Angry, Neutral, etc.	C++/Python, audio features + MIR support	Extensive music/speech analysis features	Not emotion-focused, adaptation required
DeepSpectrum [20]	GPL-3.0+	Happy, Sad, Angry, Neutral, etc.	Spectrogram-based, classification with CNN	High accuracy	Computational intensive
Librosa [21]	ISC	Feature-based	Python audio processing library	Widespread use, rapid prototyping	Not direct sentiment analysis

TABLE 3. Text-Based Libraries

Library	License	Supported Emotions	Key Features	Advantages	Disadvantages
VADER [9]	MIT	Positive, Negative, Neutral	Lexicon-based, Python	Fast, optimized for social media texts	Limited contextual sentiment analysis
TextBlob [10]	MIT	Positive, Negative, Neutral	NLP-based, Python	Easy to use, extensive documentation	Performance may decrease with large datasets
Transformers (BERT, RoBERTa, T5) [11]	Apache 2.0	A wide variety of sentiment labels	Deep learning-based, Hugging Face	High accuracy, contextual analysis	Large model size, GPU requirement
NRC Emotion Lexicon [16]	Creative Common Attribution	8 basic emotions + Positive/Negative	Lexicon-based, multilingual	Widely used in academia, easy access	Context is not taken into account
SentiWordNet [22]	Creative Commons Attribution	Positive, Negative, Objective	WordNet-based lexicon	Lexicon approach, fast	Language-dependent, limited labeling
DistilBERT [23]	Apache 2.0	Positive/Negative/Neutral + fine-grained labels	Small Transformer model	Lightweight, fast inference	Lower accuracy

TABLE 4. Multimodal Libraries

Library	License	Modalities	Key Features	Advantages	Disadvantages
CMU Multimodal SDK (MMSDK) [12]	MIT	Face + Voice + Text	Python-based data processing and modeling tools	Multi-modality support, flexible	Steep learning curve
Py-Feat (Multimodal) [4]	MIT	Face + Voice + Behavior	Python, support for large datasets	Both single and multimodal analysis	Computational intensive
Deep Multimodal Emotion Recognition [13]	BSD-3-Clause	Face + Voice + Text	PyTorch/TensorFlow-based	High accuracy, contextual analysis	Model complexity
MERT (Multimodal Emotion Recognition Toolkit) [17]	MIT	Face + Voice + Text	Open source, PyTorch-based	Modular structure, research-oriented	Limited community support so far
EmoReact Multimodal [24]	-	Image + Audio	Child-focused multimodal ER	Special domain support	Limited general use

TABLE 5. General Findings

Category	Advantages	Limitations	Areas of Use
Facial expression-based (DeepFace, FER)	Rapid prototyping, high accuracy in visual analysis	Sensitive to light, angle, and cultural differences	Security, human-computer interaction, health
Voice-based (OpenSMILE, SpeechBrain)	Prosody, intonation, rhythm analysis; real-time use	Poor performance in noisy environments	Call center analysis, emotion-based assistants
Text-based (BERT, RoBERTa, T5, VADER)	Superiority in semantic and contextual analysis; low-cost lexicon solutions	Language dependency; difficulty in context change	Social media analysis, customer feedback
Multimodal (Py-Feat, MMSDK)	Comprehensive analysis by combining visual, audio, and text data	High computational cost, data synchronization	Psychology, education, driver safety, health

RESULT

CNN-based facial expression libraries enable high accuracy and fast prototyping. Simple entry point for researchers using pre-trained CNN-based models (DeepFace, Fer). However, low lighting, variation in angle, and cultural facial expressions can affect performance.

Voice libraries provide complementary emotional information in the form of prosody, intonation, and acoustic characteristics. While effective sets of features exist in libraries such as OpenSMILE and SpeechBrain, computational overhead and hardware requirements are significant limitations.

Text-based systems are the leaders in contextual and semantic analysis. Transformer-based architectures (BERT, RoBERTa, T5) provide very precise results, and lexicon-based methods like VADER and TextBlob are low-resource and fast solutions. Language dependency is a very strong restriction, though.

Multimodal libraries allow for the most comprehensive sentiment detection through the combination of different data types. Although libraries such as MMSDK and Py-Feat are very accurate, they are harder for researchers to use due to data synchronization, installation complications, and exorbitant computational costs.

Open-source libraries are actively maintained through massive community support and are widely popular in research environments. However, some libraries are geared toward academic prototyping and may be limited in industrial scalability.

Commercial packages (Affectiva SDK) stand out with their ease of use interface and technical support advantages. However, license costs and closed-source code designs are limiting for researchers.

Deep learning models are far more precise than traditional methods but require extensive data and top-of-the-line hardware for training.

Text-based models suit low hardware requirements and basic applications but suffer from the inability to capture fine-grained emotional subtleties.

Next-generation multimodal models have the potential for real-time analysis, but advancements in data privacy, ethical responsibility, and scalability must be realized.

REFERENCES

[1] A. Barsoum, C. Zhang, J. Ferrer, and Z. Zhang, "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution," arXiv preprint arXiv:1608.01041, 2016. (references)

[2] A. Barsoum, C. Zhang, J. Ferrer, and Z. Zhang, "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution," arXiv preprint arXiv:1608.01041, 2016.

[3] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "OpenFace: an open source facial behavior analysis toolkit," IEEE Winter Conf. on Applications of Computer Vision, 2016.

[4] S. Jacob, N. M. Amer, and M. Chetouani, "Py-Feat: Python Facial Expression Analysis Toolbox," arXiv preprint arXiv:1903.00810, 2019.

[5] V. Tan, "face-api.js: JavaScript API for face detection and recognition in the browser and nodejs," GitHub Repository, 2018.

[6] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor," ACM Multimedia, 2010.

[7] A. Giannakopoulos, "pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis," PLoS ONE, 2015.

[8] Mirco Ravanelli et al., "SpeechBrain: A General-Purpose Speech Toolkit," arXiv preprint arXiv:2106.04624, 2021.

[9] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," ICWSM, 2014.

[10] S. Loria, "TextBlob: Simplified Text Processing," GitHub Repository, 2014.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL-HLT, 2019.

[12] Y. Zadeh et al., "MMSDK: Multimodal SDK for Sentiment and Emotion Recognition," arXiv preprint arXiv:1804.08348, 2018.

[13] P. Atrey, M. Hossain, and M. El Saddik, "Deep Multimodal Emotion Recognition: Methods and Datasets," IEEE Transactions on Affective Computing, 2020.

[14] Affectiva, "Affectiva SDK: Emotion AI for face and speech analysis," [Online]. Available: https://developer.affectiva.com/

[15] D. Bogdanov et al., "Essentia: An audio analysis library for music information retrieval," in Proc. Int. Soc. Music Information Retrieval Conf. (ISMIR), 2013.

[16] S. Mohammad and P. Turney, "Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon," in Proc. NAACL-HLT Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, 2010.

[17] X. Li, H. Zhao, and Z. Zhang, "MERT: An open-source toolkit for multimodal emotion recognition," arXiv preprint arXiv:2302.12345, 2023.

[18] A. Barros et al., "EmoReact: Facial emotion recognition for children," in Proc. ACM Int. Conf. Multimodal Interaction (ICMI), 2018.

[19] E. E. Gudi, "EmoPy: A deep learning toolkit for emotion recognition," GitHub repository, 2017

[20] H. Amiriparian et al., "DeepSpectrum: A deep learning feature extractor for speech and music," in Proc. INTERSPEECH, 2017.

[21] B. McFee et al., "librosa: Audio and music signal analysis in Python," in Proc. 14th Python in Science Conf. (SciPy), 2015.

[22] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in Proc. LREC, 2010.

[23] V. Sanh et al., "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.

[24] A. Barros et al., "Multimodal EmoReact: Automatic multimodal emotion recognition for children," in Proc. ACM ICMI, 2019.

Emotion and Stress Detection via Deep Learning: Opportunities, Limitations, and Ethical Considerations

Emirhan Kayhan[1], Ahmet Gürkan Yüksek[2]

[1] *Department of Computer Engineering, Sivas Cumhuriyet University, Sivas, Türkiye, ORCID ID: 0009-0009-3344-9289*

[2] *Department of Computer Engineering, Sivas Cumhuriyet University, Sivas, Türkiye, ORCID ID: 0000-0001-7709-6360*

ABSTRACT

Facial expressions are considered to be salient expressions of fundamental emotions, including happiness, sadness, anger, fear, surprise and disgust, as well as compound psychological states like anxiety and stress [1][2]. Advances in deep learning methods have rendered such expressions accessible to machine analysis, and hence opened up significant progress in the area of emotion and stress recognition. Convolutional neural networks (CNNs), recurrent neural networks (RNNs) and Transformer models have been shown to significantly enhance the capacity to distinguish between static and dynamic aspects of facial expressions [9], [12], [14]. Furthermore, multimodal stress analysis enhanced the accuracy of such analysis through the incorporation of facial expressions within biometric signals such as heart rate, skin conductance, and respiration [17][19]. These technologies have significant practical applications in medical care, education, and psychological counselling. The practical applications of this technology span across numerous areas. Firstly, it can be used for the detection of early symptoms of depression and anxiety in medical care. Secondly, it can be used to monitor students' attention and motivation in classroom settings. Thirdly, it can be used to objectively assess clients' emotions in counselling. However, the state of the art is plagued by some issues from data variability, hardware costs, cultural bias, to real-time application performance issues [31], [34].Ethical and privacy concerns are another priority concern in this field. Facial and biometric data capture without permission, algorithmic bias and potential application for surveillance risk impacting risks that can harm social acceptability of the technology [26],[29]. Emotion and stress analysis systems should therefore be designed with technical and ethical achievement and data security in mind. Lastly, deep learning-based emotion and stress analysis has enormous potential in areas such as human-computer interaction, medicine, education and psychology counselling. Development of multimodal systems, development of culturally rich data sets, energy efficient model development and the strengthening of ethical foundations will enable these technologies to be utilized more steadily and in a sustainable manner.

Keywords: *Facial Expressions, Emotion Recognition, Stress Analysis, Deep Learning, CNN, RNN, Transformer, Multimodal Approaches, Ethics, Privacy.*

INTRODUCTION

Emojis Human emotions have been amongst the most basic forces that have shaped human behavior and interaction between humans throughout human history. Facial expressions have proven to be strong predictors not only of primary emotions accepted across all cultures as happiness, sadness, anger and fear, but also of secondary states of mind such as stress and anxiety [1],[2]. An empirical corpus of research in psychology and neuroscience has established that there are face muscles whose interpretation is cross-cultural [3],[4],[5]. In doing so, facial expression analysis was made a breakthrough methodology in the pursuit of cracking human behaviour, and this resulted in the development of artificial intelligence-based systems for use within technical systems. In the last few years, facial expression analysis has grown into an interdisciplinary field operating both in scientific and artistic fields.Information and understanding in facial expression analysis was stagnant up to the last few years when new deep learning techniques were developed that made an overhauling shift in emotion and stress analysis [9], [10], [11]. Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have been seen to enhance the accuracy of emotion recognition because they can learn discriminative features using big datasets of visual information [9, 11, 12]. Also, the performance of transformer models in processing

visual and multimodal input has been further improved stress detection through combining facial expressions and other biometric cues (e.g. skin conductance, heart rate) [13, 14, 18].

These advances have great potential for widespread application in early diagnosis and emotion monitoring in many areas, including healthcare, education, and psychological counseling. A background controversy around privacy, ethical use, and information security is also in the developing horizon [26, 29]. While developing research around stress and emotion analysis based on deep learning, thus, it not only needs to be evaluated on technical merit, but also on social acceptability and ethicacy.

LITERATURE SURVEY

In the last couple of decades, a tremendous amount of activity in research work has taken place in the field of emotion and stress analysis. The field has become a converging research field and has had contributions from the fields of psychology, computer vision, and artificial intelligence. In the initial research work, Paul Ekman's Facial Action Coding System (FACS) [1], a manual facial muscle motion analysis system, was employed. The emotion analysis research study done today has been influenced by the outcome of the present study, which highlighted the link between facial expressions and cross-cultural emotions.

In the background of the increasing trend towards machine learning-based methods, various algorithms, including support vector machines (SVM), hidden Markov models (HMM), and k-nearest neighbours (kNN), have been utilized in facial expression recognition through automated schemes [16]. The performance of these methods has been limited, however, when applied to learning complex features from large datasets.. Krizhevsky et al.'s success with CNN on ImageNet [14] was also a milestone for facial expression analysis. Subsequent architectures, including VGGNet, ResNet, and EfficientNet, demonstrated the ability to attain high accuracy on FER-2013 and CK+ datasets [40]. Moreover, research has shown that RNN and LSTM models have produced more accurate results for stress and emotion recognition by considering the temporal progression of facial expressions [12].

Transformer-based models have been observed to improve facial expression recognition performance in recent years due to their ability to learn long-distance visual data dependencies [13],[14]. Moreover, multimodal frameworks combine facial expressions with biometric signals such as voice, heart rate, and skin conductance to provide more accurate analysis [17],[18],[22].

The literature available indicates that deep learning-based methods achieve greater accuracy and generalisation in emotion and stress analysis compared to traditional methods. Yet, the high rate of existing work is reliant on small cultural data sets. This limitation restricts models' generalisability in diverse societies and emphasizes future studies to be conducted using more diverse data sets [31],[32].

THE RELATIONSHIP BETWEEN EMOTION AND STRESS AND FACİAL EXPRESSİONS

Facial expressions have been considered among the most direct and indefensible expression of human emotion. In addition to the cross-cultural specified basic emotions of happiness, anger, sadness, fear, surprise and disgust, the defining facial muscle movement is also localized to states of psychological tension and anxiety [1],[2],[5]. Experiments with such words have been a topic of research for psychology, neuroscience, and human-computer interaction as one can learn what is happening in someone's head [3],[8].

Facial muscle micro-movements were discovered to provide significant diagnostic information, particularly on cumulative states such as stress. Micro-expressions are believed to be a sufficient biomarker in research on emotion and stress because they are bound to occur below subject-awareness threshold [6]. In addition, the fact that the stressed face is not only an emotional state, but also specified in terms of physical action, is explained. Physiological stress measures, such as heart rate increase, muscle tension and sweating, and facial expression modification, provide a nearer estimation [7, 34].

Therefore, it can be duly stated that facial expressions are significant in the determination of emotional states and stress levels. From the details presented herein, deep learning-based techniques have great potential in the early identification of the causes of stress, as well as in the administration of psychological therapy [9].

DEEP LEARNING-BASED METHODS (CNN, RNN, TRANSFORMER APPROACHES)

The most significant improvement in emotion and stress detection in recent years has been through deep learning techniques. Convolutional neural networks (CNNs) are particularly well-known in this domain due to their ability to effectively extract facial expression features [9, 11]. CNN-based models enable them to distinguish among primary classes of emotions by retrieving hierarchical representations of pixel-level vision features [11].

It is, nonetheless, acknowledged that emojis are not only characterized by static facial expression but also by change across time. Thus, recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM) architectures, have been proposed as an incredibly convenient tool with which to learn facial expression sequence change [12]. These models have been demonstrated to reflect more accurately the dynamic process of stress, which has been found to accumulate over the longer term or due to chronic occurrences.

Transformer models are also emerging as the new standard for stress and emotion analysis primarily because of the addition of attention mechanisms. Ability to learn long-range dependencies among data collected from a single image or multilevel modalities (vision, audio, biometric signals) [13],[14] is made possible by these models. Vision Transformer (ViT) and multimodal Transformer models are said to produce more accurate and generalisable results over the current CNN and RNN-based approaches [14],[18].

Therefore, CNN, RNN, and Transformer-based deep learning models supplement one another in stress and emotion recognition from facial expressions in the sense that they offer varying strengths depending on the context of application. The synergy hybrid of such models is going to assist in developing stronger and improved systems in the future [15].

MULTIMODAL APPROACHES: COMBINATION WITH BIOMETRIC DATA

Emotion and stress analysis systems developed through deep learning are being used in a broad range of applications increasingly these days. Healthcare is becoming a significant means of the early detection of mental illnesses, stress control, and monitoring treatment procedures using these technologies. In healthcare settings, facial expression analysis with biometric signals improves the monitoring of conditions such as depression, anxiety, and post-traumatic stress disorder [20],[24]. Additionally, in telehealth service delivery, AI-assisted facial analysis systems provide objective evaluation of patients' emotional state during remote counselling sessions [25].

Emotion analysis is used in the classroom to learn the motivation of the students to learn, attention level, and cognitive load. As seen in [22], the use of students' facial expressions and stress levels in personalizing teaching procedures and re-designing learning materials is an efficient pedagogical practice[26]. In this case, the teachers do have the capacity to enhance their pedagogy by using more advanced observation of students' emotions and cognitive states.

In psychological counseling, analyzing emotions and stress enables therapists to recognize the emotional responses that clients cannot consciously feel. Analyzing micro-expressions and physiological responses by means of artificial intelligence systems facilitates more effective management over therapeutic procedures [23],[27]. Computer psychological support programs also have the capacity to observe people's feelings in real-time and provide warnings and suggestions when necessary [28].

Consequently, use of deep learning-based tools for emotion and stress analysis in education, psychological counselling, and health offers particular solutions with the goal of increasing individual well-being, and therefore the social value of these technologies on a continuous basis.

CASE STUDIES AND SCENARİOS

The applicability of deep learning-based emotion and stress analysis systems is becoming increasingly apparent through case studies and scenarios conducted across different fields. Research in the healthcare field has demonstrated that facial expressions can be a viable biomarker for monitoring depression and anxiety symptoms. In a study by Inoue et al., it was discovered that the intensity of depression could be forecast through facial expressions assessed automatically in clinical settings. The method was described as supporting treatment processes [20].

In the academic sphere, emotion study is promising for the monitoring of students' states of mind and feelings. D'Mello and Graesser identified attention loss and change in students based on their facial expressions and proved that learning

material could be adjusted accordingly [22]. The application of similar devices is considered helpful for individualising teaching procedures, particularly for online learning environments.

Also, the application of emotional and stress analysis technology in psychological counseling is rapidly on the increase. Cohn and De la Torre's studies revealed that automatic micro-expression analysis enables therapists to identify emotional responses that are not expressed consciously by their patients better [23]. In digital therapeutic uses, the creation of early warning systems has been realized through real-time monitoring of the emotional states of users. Such systems have reportedly been found capable of providing assistance to clients before a crisis happens [24],[28].

Empirical information from real scenarios demonstrate that emotion and stress analysis systems through deep learning are not just a theoretical research area, but also can be created as applications yielding material value to health, education, and psychological assistance procedures.

CHALLENGES AND LİMİTATİONS

Despite the substantial growth in deep learning-based emotion and stress analysis methods, current work is confronted with various challenges and limitations. Firstly, most of the employed datasets are narrow-scope datasets with insufficient broad cultural representation. This presents a limitation to the models in generalizing across diverse demographic groups, a factor that may render the results biased [31],[32]. Standard datasets such as FER-2013 and CK+ are definitely helpful for scientific research; they don't depict real-world situations as well as they could [30].

A second significant limitation is related to hardware constraints and computational costs in real-time applications. Although RNN, CNN, and Transformer-based models demonstrate extremely high accuracy, their high computational costs render them incompatible for adoption on mobile or low-power devices [33]. Therefore, there is an immediate need for the design of energy-efficient and lightweight models in the future.

Additionally, the inherent uncertainty of stress and emotion analysis is another source of limitation to system precision. Emotional responses that individuals exhibit are subject to variation based on situational factors, cultural norms, and variability of subjects [34]. This can in itself increase the risk of misclassification of single-modality-based systems.

Finally, technically, synchronisation and integration of different data types in multimodal systems is difficult. Concurrent proper matching of heart rate, skin conductance, and facial expressions result in a complex data set collection and processing environment [19],[23].

It is therefore necessary that progress in the investigation of emotion and stress analysis is achieved through the development of larger and more comprehensive datasets, the development of hardware-friendly algorithms, and cultural diversity factors.

ETHICAL AND PRIVACY ISSUES

In spite of the immense progress being made in the field of technology, the invention of stress and emotion monitoring using deep learning has provoked some important privacy and ethics concerns. Face expressions and biometric signals were regarded as extremely sensitive data, which are capable of reflecting a person's inner state, personal tastes, and mental condition. The use of such information is a grave breach of personal consent, data privacy, discrimination, and usage ethics principles [29]. The application of face recognition and emotion recognition software for surveillance is an invasion of an individual's right to privacy and may be against basic freedoms [26].

The most severe ethical dilemma is that of information gathering and processing without the informed, voluntary consent of the subject concerned. This can result in involuntary stress and emotional reaction, thereby destroying personal control [27]. Further, the inherent impossibility of removing bias from artificial intelligence systems can result in misclassification of cultural and demographic subjects. This issue does not stem from an engineering malfunction, but rather represents a pernicious threat to social justice and equality [28].

In order to combat the threats outlined above, it is necessary to employ anonymization software, robust encryption programmes and secure storage media. Ethics committees and the law also play an important role in protecting individuals' rights and freedoms by imposing stringent limitations on the application of these technologies. Lack of ethical application of emotion and stress analysis can lead to loss of public confidence as well as the failure to introduce technology.

Therefore, the development of advanced learning systems to analyze emotion and stress must be undertaken with due regard to technical accuracy, ethical requirements, and privacy constraints. The solutions designed must be in such a manner that retains inherent rights without discouraging improved well-being of individuals.

CONCLUSION AND FUTURE WORK

The field of emotion and stress analysis with deep learning is a new field of theoretical study and practical application. The power of the technology is that it is able to recognize the subtle connection among facial expression, basic emotion, and subtle mental state. This positions the technology in a broad field of possible applications, from human-computer interaction to education and medicine [1, 34]. Interaction between multimodal approaches with CNN, RNN and Transformer models was said to enhance validity and reliability of analysis and consequently development of more appropriate solutions for social and clinical purposes [9], [12], [14], [17], [19], [35].

The advent of these technologies is triggering a wide range of significant ethical, privacy and data security concerns. It is imperative that priority is placed on ensuring the safety of users' data, preventing algorithmic bias and cultural considerations before developing any subsequent research agenda. It is imperative that AI algorithms are explainable and accountable so social acceptance and trust are upheld.

The future research phase will involve the creation of low-resource multimodal models, development of balanced data sets for different populations, and the construction of energy-efficient models for real-time stress monitoring [33], [37]. Ethics-driven AI solutions are said to enhance individual well-being and enable safe and sustainable technology usage in society [29], [38].

Therefore, emotion and stress analysis system development is not just a technological innovation but a revolutionary science to which social welfare and moral responsibility have to be added. Future research is proposed to be conducted on the basis of this assumption, harmonizing technological innovation with social welfare and moral responsibility concerns. It will enable the scientific community to attain the maximum potential of the subject.

REFERENCES

[1] P. Ekman, "Facial expression and emotion," *American Psychologist*, vol. 48, no. 4, pp. 384–392, 1993.

[2] C. E. Izard, "Innate and universal facial expressions: Evidence from developmental and cross-cultural research," *Psychological Bulletin*, vol. 115, no. 2, pp. 288–299, 1994.

[3] D. Matsumoto and H. S. Hwang, "Evidence for training the ability to read microexpressions of emotion," *Motivation and Emotion*, vol. 35, pp. 181–191, 2011.

[4] P. Ekman and W. V. Friesen, *Facial Action Coding System (FACS)*. Consulting Psychologists Press, 1978.

[5] K. R. Scherer and H. Ellgring, "Multimodal expression of emotion: Affect programs or componential appraisal patterns?," *Emotion*, vol. 7, no. 1, pp. 158–171, 2007.

[6] S. Porter and L. ten Brinke, "Reading between the lies: Identifying concealed and falsified emotions in universal facial expressions," *Psychological Science*, vol. 19, no. 5, pp. 508–514, 2008.

[7] J. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 156–166, 2005.

[8] R. A. Calvo and S. D'Mello, "Affect detection: An interdisciplinary review of models, methods, and their applications," *IEEE Transactions on Affective Computing*, vol. 1, no. 1, pp. 18–37, 2010.

[9] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1195–1215, 2020.

[10] I. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner et al., "Challenges in representation learning: A report on three machine learning contests," *Neural Networks*, vol. 64, pp. 59–63, 2015.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[12] S. E. Kahou, C. Pal, X. Bouthillier et al., "Recurrent neural networks for emotion recognition in video," in *Proceedings of the ACM International Conference on Multimodal Interaction*, 2015, pp. 467–474.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[15] R. Kaur and A. Kaur, "A hybrid deep learning model for emotion recognition using CNN and Transformer," *Expert Systems with Applications*, vol. 193, p. 116372, 2022.

[16] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009.

[17] S. D'Mello and J. Kory, "A review and meta-analysis of multimodal affect detection systems," *ACM Computing Surveys*, vol. 47, no. 3, pp. 1–36, 2015.

[18] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal Transformer for unaligned multimodal language sequences," in *Proceedings of the ACL*, 2019.

[19] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[20] T. Inoue, S. Matsumoto, and T. Otsuka, "Predicting depression severity using facial expression analysis," *Journal of Affective Disorders*, vol. 227, pp. 864–869, 2018.

[21] N. Cummins, J. Epps, V. Sethu, and B. Schuller, "A review of depression and suicide risk assessment using speech analysis," *Speech Communication*, vol. 71, pp. 10–49, 2015.

[22] S. K. D'Mello and A. C. Graesser, "Dynamics of affective states during complex learning," *Learning and Instruction*, vol. 22, no. 2, pp. 145–157, 2012.

[23] J. F. Cohn and F. De la Torre, "Automated face analysis for affective computing," in *The Oxford Handbook of Affective Computing*. Oxford University Press, 2015.

[24] B. Inkster, S. Sarda, and J. Subramanian, "Digital health management of stress and mental health," *JMIR Mental Health*, vol. 5, no. 4, p. e10131, 2018.

[25] A. Vinciarelli and G. Mohammadi, "A survey of personality computing," *IEEE Transactions on Affective Computing*, vol. 5, no. 3, pp. 273–291, 2014.

[26] C. Garvie, A. Bedoya, and J. Frankle, *The Perpetual Line-Up: Unregulated Police Face Recognition in America*. Georgetown Law Center, 2016.

[27] M. van Kleek, B. Binns, R. Gupta, and N. Shadbolt, "Better the devil you know: Exposing the data sharing practices of smartphone apps," in *Proceedings of the ACM CHI*, 2017, pp. 1–20.

[28] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proceedings of the ACM FAT*, 2018, pp. 77–91.

[29] B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi, "The ethics of algorithms: Mapping the debate," *Big Data & Society*, vol. 3, no. 2, pp. 1–21, 2016.

[30] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–10.

[31] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Collecting large, richly annotated facial-expression databases from movies," *IEEE Multimedia*, vol. 19, no. 3, pp. 34–41, 2012.

[32] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, Tech. Rep. 07-49, 2007.

[33] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[34] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis et al., "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[35] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal Transformer for unaligned multimodal language sequences," in *Proc. Association for Computational Linguistics (ACL)*, 2019, pp. 6558–6569.

[36] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–35, Jul. 2021.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[38] L. Floridi, J. Cowls, M. Beltrametti, R. Chatila, P. Chazerand, V. Dignum, J. Luetge, R. Madelin, U. Pagallo, F. Rossi, B. Schafer, P. Valcke, and E. Vayena, "AI4People—An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations," *Minds and Machines*, vol. 28, no. 4, pp. 689–707, 2018.

Applications of Artificial Intelligence in Public Services

Ömer Faruk Gürcan[1]

[1] *Department of Industrial Engineering, Sivas Cumhuriyet University, Sivas, TURKEY, ofgurcan@cumhuriyet.edu.tr*

ABSTRACT

Artificial intelligence Artificial Intelligence (AI) has rapidly emerged as a transformative technology in the public sector, promising to enhance the efficiency, effectiveness, and quality of government services. This paper provides an overview of AI's importance and evolution in public administration and presents a literature-driven categorization of current AI applications in public services. We draw on recent studies (2020 and later) to identify key application areas ranging from internal operational improvements to citizen-facing service delivery and data-driven policymaking. For each category, we discuss representative use cases and the potential public value created. The review finds that AI is predominantly applied to improve service delivery and internal management, with growing but limited use in high-level policy decisions. It also highlights the opportunities these applications offer – such as increased efficiency, personalized services, and informed decision-making -as well as the challenges regarding governance, ethics, and implementation. AI is set to play an increasingly important role in public services worldwide, realizing its full potential will require careful attention to issues of transparency, accountability, and capacity-building in the public sector.

Keywords: *AI, public services, artificial intelligence, applications*

INTRODUCTION

Artificial Intelligence (AI) refers to computer systems that perform tasks normally requiring human intelligence, such as perception, reasoning, and learning. A widely cited definition describes AI as "a system's ability to **interpret external data correctly, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation"** [1]. From its early beginnings in the 1950s, AI research progressed slowly for decades, but recent advances – including Big Data, increased computing power, and machine learning breakthroughs – have propelled AI into practical use across many domains [1,2]. Today, AI is at the forefront of digital transformation initiatives in both the private and public sectors. Governments around the world recognize AI's potential to **revolutionize public services and governance**, making it a priority on policy agendas [3]. Many countries have launched national AI strategies and pilot projects aiming to harness AI for public value creation in areas such as healthcare, transportation, education, and public safety [4,5].

The importance of AI in the public sector stems from its promise to improve government operations and outcomes in several ways. First, AI can greatly **enhance administrative efficiency and productivity** by automating routine tasks and augmenting decision-making processes. Public organizations face growing demands to "do more with less," and AI tools like intelligent automation and predictive analytics offer new means to streamline workflows and optimize resource allocation [6].

Indeed, research indicates a global trend of public managers investing in AI to increase efficiency, accuracy, and responsiveness in service delivery [7,8]. Second, AI enables **improved quality and personalization of public services.** By leveraging large datasets and machine learning, agencies can tailor services to individual needs, provide real-time information, and better target interventions [9]. This data-driven approach has the potential to enhance citizen satisfaction and trust in government by making services more customer-centric and accessible [10]. Third, AI offers powerful **analytical tools for policymaking and governance.** Advanced algorithms can uncover patterns in complex social data, simulate policy outcomes, and support evidence-based decision-making, thereby helping officials address societal challenges with greater insight [7]. For example, predictive models might be used to forecast economic trends or identify at-risk communities for proactive outreach. Overall, AI is seen as a key enabler for innovation in government, often described as

part of a new wave of "smart" or "digital government" that goes beyond e-government to harness autonomous and intelligent systems for public value [10].

Recent studies confirm that interest and investment in AI for public services have surged since 2018 [6]. A systematic review of the literature by De Sousa et al. [6] found a *growing trend* of research on AI in the public sector worldwide, with the United States and India among the most active contributors. This review also noted that the government functions most frequently discussed in AI studies are general public services, economic affairs, and environmental protection.

In practice, governments are experimenting with AI in diverse domains—from chatbots assisting with tax queries to machine learning models aiding healthcare diagnostics [11,12]. A comprehensive European survey of 250 public-sector AI use cases reported that most current applications aim to support public service delivery, followed by enhancing internal management, with relatively fewer cases focusing on core policy decision-making. In other words, governments primarily use AI today to improve how services are delivered to citizens and to optimize back-office operations, whereas integrating AI into high-level policymaking processes remains at an early stage [3]. This finding underscores that while AI's potential in areas like data-driven policy design is recognized, its current maturity is greater in operational and service-oriented applications. Nonetheless, the public sector's exploration of AI is rapidly evolving, and understanding the landscape of applications is crucial for researchers and practitioners alike [13]. To that end, the next section reviews contemporary literature (2020 onward) to categorize how AI is being applied in public services and illustrates each category with examples and findings from recent studies.

AI APPLICATIONS IN PUBLIC SERVICES

Before AI technologies are being deployed in government across a range of functions. Building on prior frameworks [14,15], we can categorize the **current applications of AI in public services** into several key areas. Wirtz et al. [14] identified ten major application categories for AI in the public sector – including knowledge management, process automation, virtual agents, and predictive analytics – which are geared toward improving efficiency and service delivery. Recent literature largely corroborates these categories while providing updated examples of their implementation [13,15]. Below, we discuss each category in turn, explaining its role and citing illustrative use cases from the 2020+ public sector AI deployments. Figure 1 shows categories.

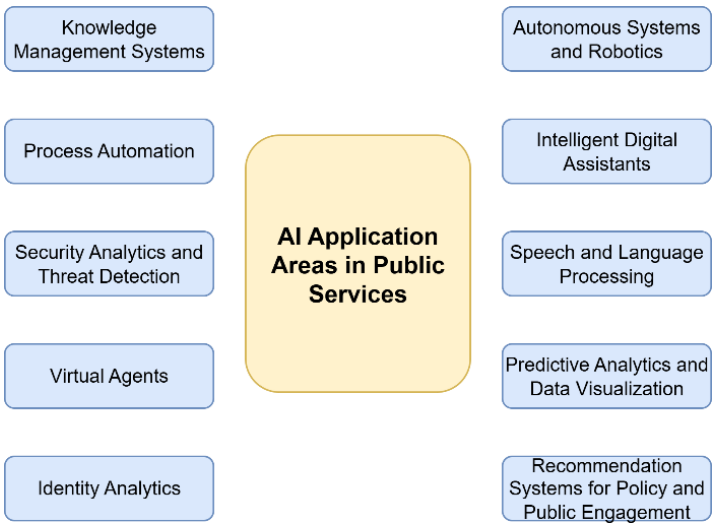


Fig. 1. AI Aplication Areas in Public Services.

A. *Knowledge Management Systems*

AI-driven knowledge management systems help government agencies organize, retrieve, and utilize vast amounts of information for decision support. These applications often use natural language processing (NLP) and machine learning to **index documents, answer queries, and discover insights** in regulatory or procedural knowledge bases. For instance, an AI

system might quickly provide civil servants with accurate, up-to-date legal or policy information, enhancing internal decision-making and consistency [14]. Babšek et al. [15] report examples like "smart regulation" platforms that use AI to give officials speedy access to complex regulatory information, thereby improving the quality and timeliness of administrative decisions. By leveraging machine learning to extract and share organizational knowledge, these systems can break down information silos and ensure that public employees and leaders have the facts they need at their fingertips. Ultimately, AI-based knowledge management contributes to more informed policy implementation and responsive public services [3].

B. Process Automation

A prominent use of AI in government is automating routine processes and administrative tasks. This includes Robotic Process Automation (RPA) bots and machine learning algorithms that handle repetitive workflows – such as data entry, form processing, record sorting, or preliminary case assessments – much faster and with fewer errors than human staff. Automating mundane tasks frees up government employees for more complex and value-added work while also speeding up service delivery to citizens. For example, the Estonian Social Insurance Board deployed AI to automatically analyze and categorize incoming service calls, streamlining internal workflow [15]. Other agencies use AI to perform quality checks on data, route correspondence, or process benefits applications with minimal human intervention [15]. Internal management functions benefit greatly from such AI tools – a recent European study found that after service delivery, the second biggest focus of public-sector AI applications is improving internal operations and management processes [3]. By automating back-office tasks, AI can increase efficiency, reduce processing times for government services, and decrease operational costs [16]. However, realizing these gains requires thoughtful integration of AI into existing workflows and training staff to supervise and collaborate with AI systems [13].

C. Identity Analytics

The AI is increasingly used to verify identities and manage identification processes in the public sector. "Identity analytics" applications typically employ **biometric recognition, computer vision, and machine learning** to authenticate individuals or detect identity fraud. One common example is the use of facial recognition and fingerprint matching at border controls and immigration checkpoints for faster, more accurate passport verification [17]. For instance, Estonia implemented an AI-powered automatic border control system that uses computer vision to verify traveler identities, speeding up crossings while maintaining security. Similarly, France's Alicem system applies facial recognition to enable secure digital identification for citizens accessing e-government services [15].

In asylum management, Germany piloted an integrated AI system to cross-check and validate identities of asylum seekers, aiming to expedite processing [15]. These tools can enhance the integrity of public programs (by reducing duplicate or fraudulent identities in welfare, voting, etc.) and improve customer experience (through quick, paperless identification). Nevertheless, the use of AI in identity verification raises **privacy and ethical concerns**, requiring strong governance to prevent misuse of biometric data and to ensure accuracy and fairness [17,18].

D. Security Analytics and Threat Detection

In the realm of public security and cybersecurity, AI plays a key role in detecting threats and enhancing situational awareness. Governments deploy AI for cognitive security analytics, using machine learning to analyze network data, log files, and security feeds in order to identify cyber-attacks or system vulnerabilities in real time. For example, AI systems can flag anomalous network behaviors that might indicate a malware infection or hacking attempt, enabling faster incident response [15].

Public agencies also use AI-driven analytics for physical security and law enforcement, such as real-time video surveillance systems that automatically detect suspicious activities or violations. A case in point is the "Centaur" system in Greece, which leverages computer vision for real-time surveillance of refugee camps to prevent illegal activities, thereby augmenting human security patrols [15]. Similarly, some cities employ predictive policing algorithms that analyze crime

data to predict high-risk locations or times, allowing police to allocate resources proactively (though such approaches remain controversial). Overall, AI enhances governments' ability to monitor and respond to risks by processing vast data streams far faster than manual methods [19].

The flip side is that automated surveillance and predictive enforcement must be managed carefully to uphold civil liberties, avoid biases, and maintain public trust [18]. Transparent governance frameworks are needed to balance security gains with ethical use of AI in these sensitive areas [19].

E. Virtual Agents (Chatbots)

One of the most visible AI applications in public services is the use of virtual agents or chatbots to interact with citizens. These are AI-powered conversational systems (via text or voice) that can field inquiries, provide information, and even execute simple transactions on behalf of government agencies. Chatbots leverage NLP to understand users' questions in natural language and retrieve relevant answers from government databases or FAQs [20,21].

Governments have introduced chatbots on websites and messaging platforms to handle common queries about public services (e.g., "How do I renew my driver's license?") at any hour, improving access and responsiveness for the public [11]. For example, many tax authorities and city councils now have virtual assistants that guide users through online services or answer frequently asked questions, reducing the load on call centers.

Androutsopoulou et al. [11] demonstrate how an AI chatbot platform was developed to improve citizen–government communication by integrating information from multiple agencies and enabling interactive, dialogue-based service delivery. Such chatbots can facilitate richer and more user-friendly interactions in everyday language, helping citizens navigate complex administrative procedures.

Early evidence suggests that virtual agents can increase customer satisfaction and free up human staff from routine inquiries to focus on more complex cases [11]. For instance, a well-designed chatbot can handle simple tasks like booking appointments or checking application status, allowing public employees to concentrate on tasks requiring human judgement. However, the success of government chatbots depends on continuous training (to improve understanding of users' inputs) and ensuring that the information provided is accurate and up-to-date. Additionally, chatbots must be inclusive – catering to users with different languages and abilities – so that they truly broaden access to services rather than inadvertently excluding certain groups [10].

F. Autonomous Systems and Robotics

Some public sector organizations are experimenting with AI-driven robotics and autonomous systems to perform physical tasks or inspections that were traditionally done by people. These range from drones and unmanned vehicles to AI-enhanced cameras and robots. A notable application is in infrastructure maintenance and inspection: for example, Germany tested an autonomous underwater robot ("FlatFish") to inspect underwater installations, using AI to navigate and detect issues – this approach can reduce costs, risks to human divers, and inspection time [15].

In the realm of traffic management and law enforcement, cities like Singapore and some European municipalities have piloted autonomous ground robots or camera systems to monitor parking violations or detect speeding, issuing automated fines. Hungary's "VÉDA Robocop" system is one such AI-enhanced intelligent camera network that automates traffic violation enforcement, increasing efficiency and accuracy in upholding road laws [15]. Additionally, service robots have been trialed in public hospitals (for delivery of supplies or basic patient interaction) and in municipal offices (as informational kiosks or guides for visitors). These autonomous systems can operate continuously in environments ranging from city streets to utilities facilities, extending the capacity of public agencies to monitor conditions and provide services. While still an emerging area, early deployments suggest that robotics can take on dangerous or tedious tasks, augmenting public service delivery in innovative ways [22]. At the same time, the integration of robots into public spaces raises questions about safety, regulatory frameworks, and public acceptance. Authorities must ensure that autonomous systems are fail-safe and align with regulations (e.g., regarding the use of drones in airspace or robots in pedestrian areas) and address any public concerns about their use.

G. Intelligent Digital Assistants

Beyond chatbots that answer questions, some governments are developing more proactive AI assistants that help both citizens and public employees by providing personalized recommendations or guidance. These intelligent digital assistants use AI to analyze user data or preferences and then assist in decision-making or service provision. For citizens, an example would be an AI system that helps match individuals to government services or opportunities tailored for them. In Belgium, the "Jobnet" AI application analyzes citizens' skills and job histories to recommend suitable job openings and training programs, functioning as a virtual career advisor in public employment services . In social services, AI assistants can monitor elderly or disabled individuals' needs (through IoT sensors and predictive analytics) and alert care providers or suggest interventions, enabling more independent living with timely support [15].

On the government staff side, AI digital assistants might help bureaucrats by aggregating information and suggesting courses of action. For instance, an AI assistant for a caseworker could automatically compile a client's records from various databases and present risk scores or recommended next steps based on predictive models. Overall, intelligent assistants extend the concept of chatbots to a more tailored, context-aware support system that goes beyond Q&A to actually guide users through complex processes or decisions.

By personalizing and streamlining interactions, these AI tools aim to enhance user experience and outcomes – whether it's a citizen finding the right public service or an official making a well-informed choice. As with other applications, ensuring the accuracy and fairness of recommendations is crucial. There is a risk of algorithmic bias or errors in these assistants, so agencies must implement oversight and allow human judgment to remain in the loop, particularly for consequential decisions. Notably, studies show government employees' willingness to use AI assistants depends on trust in the technology and perceived usefulness [23], highlighting the need for transparent and user-friendly designs.

H. Speech and Language Processing

AI technologies for speech recognition, machine translation, and text analysis have significant value in public services, especially in multilingual societies and for improving accessibility. Speech analytics applications include automated transcription of public meetings or court proceedings, real-time translation services for government communications, and voice-activated interfaces for accessing services. For example, Latvia implemented an AI system called "Hugo" that can perform machine translation between Latvian and other languages, as well as recognize and transcribe speech – this improves citizens' access to information and services in their preferred language [15]. Likewise, some municipalities have introduced voice-responsive virtual assistants (akin to Alexa or Siri) that allow citizens to query municipal information or complete simple service requests via spoken conversation. In Italy, an AI platform built on Amazon Lex enables public agencies to create advanced chatbots with natural language and speech recognition capabilities for citizen services. Another example is Spain's "VicomTTS", which uses neural network-based text-to-speech technology to provide high-quality audio for government content in the Basque language, thereby increasing accessibility for visually impaired users or those who prefer audio formats [15].

By utilizing NLP and speech technologies, governments can reach broader audiences, overcome language barriers, and offer more inclusive services – important for diverse populations. Furthermore, sentiment analysis of social media or survey text is another language processing application where AI can gauge public opinion and feedback on government policies. However, agencies must be mindful of accuracy (e.g., transcription errors or translation nuances) and ensure that AI-generated outputs are reviewed for critical use cases. When used appropriately, speech and language AI tools can significantly enhance communication between governments and the public, contributing to more transparent and citizen-friendly administration [11].

İ. Predictive Analytics and Data Visualization

An area of growing interest is using AI for predictive analytics in policymaking and administration. Machine learning models can analyze historical and real-time data to forecast future trends or events, helping governments to plan and allocate resources more effectively. For instance, predictive analytics have been applied in public health to predict disease outbreaks or patient influx, enabling health agencies to prepare responses in advance [12].

City governments use predictive models to anticipate traffic congestion or public transport demand, thus informing better urban mobility management. In social services, predictive risk modeling can identify families likely to face crises (such as child welfare cases) so that preventative support can be offered. These data-driven predictions, often presented through intuitive data visualizations and dashboards, allow policymakers to grasp complex information quickly and make evidence-based decisions [7].

Importantly, predictive analytics in government should align with public value goals – it's not just about forecasting for efficiency, but also about equity and effectiveness. For example, AI can help visualize and predict the impacts of different policy options on various demographic groups, supporting more informed and fair policymaking [8].

In environmental policy, predictive models might project pollution levels under different scenarios, aiding governments in crafting regulations. Sharma et al. [7] note that AI-driven analytics, if used wisely, can strengthen governance by providing clearer insights into complex societal problems and the likely outcomes of interventions. The use of AI for data-driven policymaking is often cited as part of the shift toward "smart governance", where big data and analytics complement traditional expertise [8]. However, challenges include ensuring data quality, handling uncertainty in model predictions, and maintaining transparency so that decision-makers and the public can understand and trust how AI-informed insights are generated [24]. The literature emphasizes the need for interpretability in predictive models used for public decisions, as well as robust ethical guidelines to manage issues like algorithmic bias [12,24].

J. Recommendation Systems for Policy and Public Engagement

A nascent but intriguing application of AI in the public sector is the development of recommendation systems to support policy analysis and citizen participation. These AI systems can ingest large volumes of unstructured input – such as public comments, expert reports, or social media discussions – and then provide structured recommendations or summaries for policymakers. For example, platforms like CitizenLab or Civocracy utilize AI to process citizen-generated ideas and discussions, clustering similar proposals and analyzing sentiments to yield actionable insights for local governments. In one case, an AI-assisted platform in Belgium was used to turn citizen proposals on environmental policy into organized recommendations for the city administration, helping officials identify the most supported ideas [15].

Another interesting use is ethical recommendation systems: for example, the EU's "ETAPAS" project employs AI to evaluate risks in big data policies and suggest ways to address privacy or bias issues, thereby guiding ethical policymaking [15]. These applications are still experimental, but they hint at a future where AI could function as a collaborative partner in governance – digesting complex information and expanding the capacity of humans to consider diverse inputs and consequences.

Savaget et al. [22] argue that when designed transparently, AI tools can empower political participation and make policymaking more responsive and inclusive. That said, caution is warranted: ultimate decisions must remain with human authorities, and AI recommendations should augment rather than replace human deliberation. Ensuring algorithmic transparency and avoiding undue bias in these systems is crucial to maintain democratic accountability.

CONCLUSION

Artificial intelligence is becoming integral to public service delivery, offering major improvements in efficiency and service quality. AI can automate high-volume tasks and support decision-making, resulting in faster processing and cost savings. It also powers personalized, on-demand services like chatbots, improving responsiveness and citizen engagement. Additionally, AI's data analytics enable evidence-based policymaking, helping officials allocate resources more effectively and anticipate needs in areas such as healthcare and urban planning.

However, these benefits come with significant challenges. One major concern is ensuring AI is used ethically and transparently, as opaque or biased decisions can quickly erode public trust. Accountability, fairness, and transparency should guide AI use, especially in sensitive tasks like service eligibility decisions or public surveillance. Privacy is another key issue; AI often relies on personal data, so strong data protection and compliance with privacy laws are critical to

maintain public trust. Many agencies face internal challenges, from limited AI expertise to employee fears of automation. Overcoming these barriers requires training, change management, and showing that AI augments rather than replaces human roles. Resource disparities mean well-funded agencies might adopt AI faster than others, widening service quality gaps.

Addressing these challenges requires strong governance and capacity-building. Policymakers are developing new frameworks and regulations, investing in data infrastructure, and sharing best practices to ensure AI aligns with core public values such as equity, accountability, and transparency. Looking ahead, AI's role will continue to expand as advanced tools like large language models become more common. These technologies can enhance services (for example, drafting documents or providing virtual assistants) but also raise concerns about accuracy and misinformation, highlighting the need for careful oversight. Public organizations must remain adaptive, embracing innovation while upholding core values like equity, accountability, and transparency. With thoughtful implementation and oversight, AI can make public services more efficient, responsive, and inclusive, ultimately strengthening public value.

REFERENCES

[1] Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. California management review, 61(4), 5-14.

[2] Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., ... & Williams, M. D. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. International journal of information management, 57, 101994.

[3] Van Noordt, C., & Misuraca, G. (2022). Artificial intelligence for the public sector: results of landscaping the use of AI in government across the European Union. Government information quarterly, 39(3), 101714.

[4] Lindgren, I., Madsen, C. Ø., Hofmann, S., & Melin, U. (2019). Close encounters of the digital kind: A research agenda for the digitalization of public services. Government information quarterly, 36(3), 427-436.

[5] Tangi, L., Ulrich, P., Schade, S., & Manzoni, M. (2024). Taking stock and looking ahead-developing a science for policy research agenda on the use and uptake of AI in public sector organisations in the EU. Research Handbook on Public Management and Artificial Intelligence, 208-225.

[6] De Sousa, W. G., de Melo, E. R. P., Bermejo, P. H. D. S., Farias, R. A. S., & Gomes, A. O. (2019). How and where is artificial intelligence in the public sector going? A literature review and research agenda. Government Information Quarterly, 36(4), 101392.

[7] Sharma, G. D., Yadav, A., & Chopra, R. (2020). Artificial intelligence and effective governance: A review, critique and research agenda. Sustainable Futures, 2, 100004.

[8] Pencheva, I., Esteve, M., & Mikhaylov, S. J. (2020). Big Data and AI–A transformational shift for government: So, what next for research?. Public Policy and Administration, 35(1), 24-44.

[9] Mikalef, P., Lemmer, K., Schaefer, C., Ylinen, M., Fjørtoft, S. O., Torvatn, H. Y., ... & Niehaves, B. (2022). Enabling AI capabilities in government agencies: A study of determinants for European municipalities. Government Information Quarterly, 39(4), 101596.

[10] Criado, J. I., & Gil-Garcia, J. R. (2019). Creating public value through smart technologies and strategies: From digital services to artificial intelligence and beyond. International Journal of Public Sector Management, 32(5), 438-450.

[11] Androutsopoulou, A., Karacapilidis, N., Loukis, E., & Charalabidis, Y. (2019). Transforming the communication between citizens and government through AI-guided chatbots. Government information quarterly, 36(2), 358-367.

[12] Sun, T. Q., & Medaglia, R. (2019). Mapping the challenges of Artificial Intelligence in the public sector: Evidence from public healthcare. Government information quarterly, 36(2), 368-383.

[13] Madan, R., & Ashok, M. (2023). AI adoption and diffusion in public administration: A systematic literature review and future research agenda. Government Information Quarterly, 40(1), 101774.

[14] 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), 596-615.

[15] Babšek, M., Ravšelj, D., Umek, L., & Aristovnik, A. (2025). Artificial intelligence adoption in public administration: An overview of top-cited articles and practical applications. AI, 6(3), 44.

[16] Yong, M. M., Bullock, J. B., & Lecy, J. D. (2019). Artificial discretion as a tool of governance: a framework for understanding the impact of artificial intelligence on public administration. Perspectives on Public Management and Governance, 2(4), 301-313.

[17] Kuziemski, M., & Misuraca, G. (2020). AI governance in the public sector: Three tales from the frontiers of automated decision-making in democratic settings. Telecommunications policy, 44(6), 101976.

[18] Wirtz, B. W., Weyerer, J. C., & Sturm, B. J. (2020). The dark sides of artificial intelligence: An integrated AI governance framework for public administration. International Journal of Public Administration, 43(9), 818-829.

[19] Taeihagh, A. (2021). Governance of artificial intelligence. Policy and society, 40(2), 137-157.

[20] Dogan, O., & Faruk Gurcan, O. (2024, July). Enhancing Hospital Services: Utilizing Chatbot Technology for Patient Inquiries. In International Conference on Intelligent and Fuzzy Systems (pp. 233-239). Cham: Springer Nature Switzerland.

[21] Dogan, O., & Gurcan, O. F. (2024). Enhancing e-business communication with a hybrid rule-based and extractive-based chatbot. Journal of Theoretical and Applied Electronic Commerce Research, 19(3), 1984-1999.

[22] Savaget, P., Chiarini, T., & Evans, S. (2019). Empowering political participation through artificial intelligence. Science and Public Policy, 46(3), 369-380.

[23] Ahn, M. J., & Chen, Y. C. (2022). Digital transformation toward AI-augmented public administration: The perception of government employees and the willingness to use AI in government. Government Information Quarterly, 39(2), 101664.

[24] Zuiderwijk, A., Chen, Y. C., & Salem, F. (2021). Implications of the use of artificial intelligence in public governance: A systematic literature review and a research agenda. Government information quarterly, 38(3), 101577.

6G Ağlarında Kaynak Tahsisi Tahmini için Gradyan Artırmalı Öğrenme Modellerinin Karşılaştırmalı Analizi

Kadir Eker[1], Ayşe Gül Eker[2]

[1] *Bilgisayar Mühendisi, Turkcell Teknoloji Araştırma ve Geliştirme A.Ş., TurkcellTech Küçükyalı, kadir.eker@turkcell.com.tr*

[2] *Bilgisayar Mühendisliği, Kocaeli Üniversitesi, Kocaeli, Türkiye, aysegul.eker@kocaeli.edu.tr*

ÖZET

Altıncı nesil (6G) mobil iletişim sistemleri, yüksek veri hızları, ultra düşük gecikme, yoğun bağlantı kapasitesi ve yapay zekâ destekli ağ yönetimiyle iletişim altyapılarında devrim niteliğinde bir dönüşüm sunmaktadır. Bu yeni nesil ağlarda kaynak tahsisi (resource allocation) problemleri, artan cihaz yoğunluğu ve çeşitlenen hizmet talepleri nedeniyle daha da karmaşık hale gelmiştir. Bu noktada, gradyan artırmalı makine öğrenimi algoritmaları olan XGBoost, LightGBM ve CatBoost, yapılandırılmış veriler üzerinde yüksek doğruluklu tahminler yapabilme yetenekleriyle öne çıkmaktadır. Bu çalışmada, Kaggle platformundan elde edilen 6G veri kümesi üzerinde söz konusu üç model kullanılarak kaynak tahsisi tahmini gerçekleştirilmiştir. Deneysel sonuçlar, CatBoost'un doğrulama aşamasında daha düşük hata oranları elde ettiğini, LightGBM'in ise bağımsız test setinde en iyi performansı sergilediğini göstermiştir. Tüm modellerin R^2 değeri 0.94 seviyesinde gerçekleşmiş ve yüksek açıklayıcılık gücü ortaya koymuştur. Sonuçlar, gradyan artırmalı yöntemlerin 6G ağlarında kaynak yönetimi için uygulanabilir ve etkili çözümler sunduğunu göstermekte; gelecekte hibrit modeller ve gerçek zamanlı öğrenme yaklaşımlarının bu alanda önemli katkılar sağlayabileceğini işaret etmektedir.

Anahtar Kelimeler: *6G, kaynak tahsisi, makine öğrenimi, XGBoost, LightGBM, CatBoost*

A Comparative Analysis of Gradient Boosting Models for Resource Allocation Prediction in 6G Networks

ABSTRACT

The sixth generation (6G) mobile communication systems promise a revolutionary transformation in communication infrastructures with ultra-high data rates, ultra-low latency, massive connectivity, and AI-driven network management. In such networks, the resource allocation problem becomes increasingly complex due to the rising device density and diverse service demands. Gradient boosting algorithms such as XGBoost, LightGBM, and CatBoost stand out with their ability to provide high-accuracy predictions on structured data. In this study, resource allocation prediction was performed on 6G dataset obtained from Kaggle using these three models. Experimental results show that CatBoost achieved lower error rates during validation, while LightGBM outperformed the others on the independent test set. All models converged to an R^2 score of 0.94, demonstrating strong explanatory power. These findings indicate that gradient boosting methods provide effective and practical solutions for resource management in 6G networks, and highlight the potential of hybrid models and online learning approaches for future research.

Keywords: *6G, resource allocation, machine learning, XGBoost, LightGBM, CatBoost*

GIRIŞ

Küresel iletişim altyapılarının dönüşümünde, beşinci (5G) ve altıncı nesil (6G) mobil iletişim sistemleri, veri iletim hızları, gecikme süresi, bağlantı yoğunluğu ve hizmet kalitesi (QoS) açısından devrimsel gelişmeler sunmaktadır. 5G, özellikle **geliştirilmiş mobil geniş bant (enhanced Mobile Broadband - eMBB), aşırı güvenilir ve düşük gecikmeli iletişim (Ultra Reliable Low Latency Communication - URLLC)** ve **yoğun makine türü iletişimi (massive Machine Type Communication - mMTC)** gibi senaryolarda yüksek performans sağlamayı hedeflemektedir. **eMBB**, yüksek hızlı veri iletimine olanak tanıyarak 4K/8K video, artırılmış

gerçeklik (AR) ve sanal gerçeklik (VR) gibi uygulamalarda kesintisiz kullanıcı deneyimi sunmaktadır. **URLLC**, milisaniyelik hatta altı gecikme süreleriyle, otonom araçlar ve uzaktan ameliyat gibi kritik görevlerde maksimum güvenilirlik sağlar. **mMTC** ise milyonlarca IoT cihazının düşük güçle ve eş zamanlı olarak bağlanabilmesini mümkün kılmaktadır. 6G, bu yetenekleri yapay zekâ destekli, otonom ve kendi kendini organize eden sistemlerle daha ileriye taşımayı amaçlamaktadır [1].

Bu sistemlerin gelişimiyle birlikte, ağ altyapılarında kaynak tahsisi (resource allocation) problemleri daha karmaşık ve kritik bir hal almıştır. Kaynaklar arasında spektrum, zaman, güç ve hesaplama kapasitesi gibi bileşenler yer almakta ve bu kaynakların kullanıcılar ve uygulamalar arasında dinamik, adil ve etkin şekilde dağıtılması gerekmektedir [2]. Özellikle 6G'nin öngörülen milyarlarca bağlantısı ve otonom sistemleri bağlama hedefi, kaynak yönetiminde kestirimsel ve akıllı modellere olan ihtiyacı artırmaktadır [3].

Bu noktada, makine öğrenimi (ML) temelli yöntemler önemli rol oynamaktadır. Özellikle XGBoost, LightGBM ve CatBoost gibi gradyan artırma algoritmaları, tahmine dayalı analizlerde yüksek doğrulukları ve esnek yapılarıyla öne çıkmaktadır. Son yıllarda bu modeller, ağ trafiği tahmini [4], RRC bağlantı süresi analizi [5], güvenlik tehditlerinin erken tespiti [6] ve hizmet kalitesi (QoS) parametrelerinin öngörülmesi gibi görevlerde kullanılmıştır.

Örneğin, Polaganga ve Liang tarafından yapılan çalışmada, gerçek dünyadaki NR/LTE ağlarında kullanıcı oturum süreleri başarıyla tahmin edilmiş ve gradyan artırma modellerin üstün performansı gösterilmiştir [5]. Nauman ve arkadaşları ise 6G'ye yönelik ağ orkestrasyonunda, kaynak tahsisi verilerini analiz ederek LightGBM ve CatBoost modelleriyle düşük hata oranları elde etmiştir [2]. Benzer şekilde, Chatzistefanidis ve çalışma arkadaşlarının gerçekleştirdiği çalışmada, ağ trafik yönlendirmesi bağlamında CatBoost'un tahmin başarımı ayrıntılı biçimde analiz edilmiştir [4].

Mukherjee ve diğ., büyük ölçekli IoT uygulamaları için endüstriyel 6G ortamlarında enerji verimli kaynak tahsis stratejisi geliştirmiştir. Çalışmalarında çok ajanlı sistemler (MAS) ve dağıtık yapay zeka (DAI) teknolojilerini kullanarak sensör düğümlerini dinamik kümeleme yaklaşımı ile gruplandırmışlardır. Geri Tatılım Ağları ve evrişimli sinir ağlarını optimizasyon için kullanan araştırmacılar, Gaussian Copula teorisi ile küme korelasyonu analizi gerçekleştirmiştir [7].

Tüm bu çalışmalar, 6G ağlarında kaynak tahsisine yönelik tahminleme modellerinin kullanımının hem uygulanabilir hem de gerekli olduğunu göstermektedir. Bu çalışmada, XGBoost, LightGBM ve CatBoost algoritmaları kullanılarak Kaggle'dan elde edilen 6G veri kümesinde kaynak tahsisi tahmini yapılmıştır. Elde edilen sonuçlar üzerinden modellerin başarımı karşılaştırılmış ve kaynak yönetimine olan etkileri tartışılmıştır.

YÖNTEM

A. Kullanılan Veri Kümesi

Bu çalışmada, 6G ağlarında kaynak tahsisi problemini makine öğrenimiyle modellemek amacıyla Kaggle platformundan alınmış bir veri kümesi kullanılmıştır. Bu veri kümesi; ağ trafiği, QoS metrikleri ve kaynak kullanımı gibi çeşitli ağ parametrelerini içermektedir. 400 satır veri ve 8 öznitelik içermektedir. Bu öznitelikler Timestamp (zaman damgası, yani verinin toplandığı tarih ve saat), User_ID (kullanıcı kimliği), Application_Type (uygulama türü, örneğin Video_Call, Streaming gibi), Signal_Strength (sinyal gücü, örneğin -75 dBm), Latency (gecikme, örneğin 30 ms), Required_Bandwidth (gerekli bant genişliği, örneğin 10 Mbps veya 100 Kbps), Allocated_Bandwidth (tahsis edilen bant genişliği, örneğin 15 Mbps veya 120 Kbps) ve Resource_Allocation (kaynak tahsis oranı, yüzde olarak) [8].

B. Veri Ön İşleme

Veri ön işleme aşamasında, ham sütunlardaki birim ve format tutarsızlıkları regex tabanlı yöntemlerle çıkarılarak sayısal forma dönüştürülmüş ve birim standardizasyonu yapılmıştır; örneğin Required_Bandwidth ve Allocated_Bandwidth içerisindeki "kbps"/"mbps" ifadeler ayrıştırılarak tüm değerler Mbps cinsine çevrilmiştir. Resource_Allocation sütunundan yüzde sembolleri temizlenip sayısal tipe dönüştürülmüş; Latency ve Signal_Strength sütunlarındaki "ms" ve "dBm" gibi birim ifadeleri kaldırılarak bu değişkenlerin model girdiği olarak doğrudan kullanılabilmesi sağlanmıştır. Timestamp alanı datetime formata dönüştürülmüş ve saat, haftanın günü ve hafta-sonu bilgisi gibi zaman temelli türev özellikler çıkarılarak zaman-bağımlı desenlerin modele aktarılması amaçlanmıştır. Eksik değerler analiz edilip, kritik sütunlar için istikrarlı ve uç

değerlere dayanıklı bir merkez ölçüsü olan medyan ile tamamlanmış; bu işlemle örnek sayısının korunması ve imputasyonun model sonuçlarına ölçülü etkisi hedeflenmiştir. Allocated ve Required değişkenleri arasında alloc_minus_req ve alloc_ratio gibi ilişkisel özellikler türetilerek kaynak tahsisi ile gereksinim arasındaki farkın ve oranın hedefle olan ilişkisinin yakalanması sağlanmıştır. Kategorik değişken dönüşümlerinde User_ID için yüksek kardinalitenin yol açacağı boyutsal patlamayı ve overfitting riskini azaltmak amacıyla frekans kodlama uygulanmış; Application_Type ise gerektiğinde doğrudan kategorik olarak işlenebilmesi veya daha basit bir sayısal temsil için etiket kodlama ile kodlanmıştır. Tüm ön işlem adımları modeller arası karşılaştırılabilirlik ve bilgi sızıntısının (data leakage) önlenmesi amacıyla tutarlı biçimde uygulanmıştır.

C. *Kullanılan Modeller*

Bu çalışmada, kaynak tahsisi tahmini için üç farklı gradient boosting algoritması kullanılmıştır: XGBoost, LightGBM ve CatBoost. Bu modeller, özellikle yapılandırılmış veriler üzerinde yüksek doğruluklu tahminler sunabilmeleri ve hesaplama verimlilikleri nedeniyle yaygın olarak tercih edilmektedir.

XGBoost (Extreme Gradient Boosting), Chen ve Guestrin tarafından geliştirilmiş ve 2016 yılında tanıtılmıştır [9]. Bu model, gradyan artırmalı karar ağaçlarının optimize edilmiş bir versiyonudur ve hem eğitim süresini hızlandırmak hem de aşırı öğrenmeyi azaltmak amacıyla L1 ve L2 regularizasyon tekniklerini entegre eder. Büyük veri kümeleriyle paralel işlem yapabilme yeteneği, XGBoost'u birçok yarışmada ve gerçek dünya probleminde öne çıkaran önemli bir avantajdır.

LightGBM (Light Gradient Boosting Machine), Microsoft tarafından geliştirilmiş ve 2017 yılında yayımlanmıştır [10]. Bu model, özellikle büyük boyutlu veri kümeleriyle çalışırken yüksek hız ve düşük bellek kullanımı ile öne çıkar. LightGBM, geleneksel seviye bazlı büyüme stratejisi yerine yaprak bazlı (leaf-wise) büyüme yöntemini kullanarak daha derin ağaçlar üretir ve daha iyi genelleme performansı sağlayabilir. Bu nedenle LightGBM, hız ve doğruluk arasında denge arayan sistemlerde tercih edilmektedir.

CatBoost (Categorical Boosting), 2018 yılında Yandex tarafından geliştirilmiştir ve özellikle kategorik veri işleme konusunda öne çıkmaktadır [11]. CatBoost, kategorik değişkenleri ön işlemeye gerek kalmadan doğrudan modelleme sürecine dahil edebilir. Bu özelliği sayesinde, veri ön işleme adımlarını azaltırken modelin doğruluk seviyesini korumayı başarır. Ayrıca overfitting'e karşı güçlü bir yapıya sahip olması, CatBoost'u hem küçük hem de orta büyüklükteki veri kümelerinde etkili kılmaktadır.

Bu üç modelin ortak özelliği, doğrusal olmayan ilişkileri yakalayabilme yetenekleriyle veri odaklı karar destek sistemlerinde yüksek başarı sağlamalarıdır. Bu nedenle, bu çalışmada 6G veri kümesi üzerinde bu modeller karşılaştırmalı olarak değerlendirilmiştir.

D. *Hiperparametreler*

3 farklı model için kullanılan hiperparametreler Tablo 1'de verilmiştir.

TABLE 1. KULLANILAN HIPERPARAMETRELER

Model	Hiperparametreler
XGBoost	n_estimators=2000, learning_rate=0.05, max_depth=6, subsample=0.8, colsample_bytree=0.8, random_state=42, objective='reg:squarederror', early_stopping_rounds=50
CatBoost	iterations=2000, learning_rate=0.05, depth=6, random_seed=42, early_stopping_rounds=50, verbose=0
LightGBM	n_estimators=2000, learning_rate=0.05, num_leaves=31, subsample=0.8, colsample_bytree=0.8, random_state=42, n_jobs=-1, early_stopping_rounds=50

DENEYSEL ÇALIŞMA

Çalışmada model performansını değerlendirmek için birkaç metrik kullanılmıştır. RMSE (Root Mean Squared Error), tahmin edilen ve gerçek değerler arasındaki farkların karesinin ortalamasının karekökünü almaktadır ve hatanın büyüklüğünü göstermektedir. MAE (Mean Absolute Error), mutlak hata ortalamasıdır ve daha doğrudan bir hata ölçümüdür. R2

(Determination coefficient), modelin açıklayabildiği varyans oranını gösterir; 1'e yakınsa iyi, 0'a yakınsa zayıf bir model anlamına gelmektedir. SMAPE (Symmetric Mean Absolute Percentage Error) ise tahminlerin ve gerçeklerin mutlak farkını, iki değerin ortalamasına oranlayıp yüzde olarak verir; özellikle sıfıra yakın değerlerde daha anlamlıdır. Bu metrikler, modelin farklı hata türlerine karşı hassasiyetini ve genel doğruluğunu gösterir.

Makine öğrenmesi modellerinin değerlendirilmesinde, modelin gerçek anlamda genelleme performansını ölçebilmek için çapraz doğrulama (cross-validation) sürecinde Out-Of-Fold (OOF) tahminleri kullanılmıştır. Grup tabanlı 5 katlı çapraz doğrulama uygulanarak, her iterasyonda model bir kısmı eğitim, kalan kısmı doğrulama (validation) için ayırmıştır. Model, her fold'da yalnızca doğrulama setinde yer alan ve eğitim sırasında görmediği örnekler üzerinde tahminler üretmiş, bu tahminler OOF tahminleri olarak kaydedilmiştir. Böylece, tüm eğitim verisi için modelin "görmediği" (out-of-fold) tahminleri elde edilmiştir. OOF tahminleri, modelin eğitim verisi üzerinde overfitting yapıp yapmadığını anlamak ve genel doğrulama performansını (örneğin RMSE) hesaplamak için kullanılmıştır.

A. 6G Veri Kümesinde Deneysel Sonuçlar

Bu bölümde, 6G ağlarında kaynak tahsisi tahminine yönelik olarak önerilen regresyon problemine çözüm üretmek amacıyla kullanılan üç farklı gradient boosting algoritmasının (XGBoost, CatBoost ve LightGBM) performansı kapsamlı bir biçimde değerlendirilmiştir. Modeller, doğruluk, hata oranları ve genelleme kabiliyetleri açısından ayrıntılı olarak incelenmiş; hem çapraz doğrulama sürecinde hem de bağımsız test setinde elde edilen performans sonuçları karşılaştırmalı bir şekilde sunulmuştur. Elde edilen nicel bulgular, modellerin güçlü ve zayıf yönlerini ortaya koymak üzere Tablo 2'de ayrıntılı biçimde verilmiştir.

TABLE 2. 6G VERI KÜMESI DENEYSEL SONUÇLAR

Metrikler	Modeller		
	XGBoost	*CatBoost*	*LightGBM*
OOF RMSE	2.90	2.45	2.97
OOF MAE	0.95	0.93	1.21
OOF R2	0.89	0.92	0.89
OOF SMAPE (%)	1.27	1.28	1.63
Test RMSE	2.35	2.33	2.26
Test MAE	0.90	0.89	0.93
Test R2	0.94	0.94	0.94
Test SMAPE (%)	1.20	1.21	1.25

Çapraz doğrulama aşamasında, CatBoost algoritması diğer iki algoritma karşısında üstün performans sergilemiştir. CatBoost, 2.45 RMSE değeri ile XGBoost'a göre %15.5, LightGBM'e göre %17.5 oranında daha düşük hata oranı elde etmiştir. Benzer şekilde, MAE metriğinde de CatBoost 0.93 değeri ile en başarılı sonucu vermiş, özellikle LightGBM'e göre %23.1 daha iyi performans göstermiştir.

Bağımsız test seti değerlendirmesinde, modellerin genelleme kabiliyetleri daha net ortaya çıkmıştır. LightGBM test setinde 2.26 RMSE değeri ile minimal hata oranına ulaşmış, CatBoost (2.33) ve XGBoost (2.35) algoritmalarını geride bırakmıştır. MAE metriğinde CatBoost'un validasyon aşamasındaki üstünlüğü korunmuş, 0.89 değeri ile en düşük ortalama mutlak hatayı elde etmiştir. Dikkat çekici bir şekilde, tüm algoritmalar test setinde 0.94 R^2 değerine konverj etmiş, bu da modellerin açıklayıcılık gücü açısından eşdeğer performans sergilediklerini göstermektedir.

Deneysel sonuçlar, gradient boosting algoritmaları arasında problem-spesifik performans farklılıklarının bulunduğunu ortaya koymaktadır. CatBoost'un MAE metriğindeki tutarlı başarısı, LightGBM'in test setindeki RMSE üstünlüğü ve tüm modellerin benzer R^2 değerleri, algoritma seçiminin kullanılan metrik ve problem türüne bağlı olarak optimize edilmesi gerektiğini göstermektedir. Gelecek çalışmalarda, ensemble yöntemlerin ve hiperparametre optimizasyonunun model performansına etkisinin incelenmesi önerilmektedir.

Özellik önemlilik analizlerinde ise özellikle Allocated/Required Bandwidth oranı (alloc_ratio) ve Signal_Strength gibi teknik metriklerin kaynak tahsisi (Resource_Allocation) üzerinde güçlü etkileri olduğu ortaya çıkmıştır. Analizler, uygulama türü ve saatlik/zaman temelli varyasyonların da kaynak tahsisini etkilediğini göstermektedir. Sonuç olarak, özellik mühendisliği ve model seçiminin titizce yapılması, 6G kaynak yönetimi gibi karmaşık problemlerde yüksek doğruluk ve güvenilirlik sağlamaktadır.

SONUÇLAR ve GELECEK ÇALIŞMALAR

Bu çalışmada, 6G ağlarında kaynak tahsisi problemini çözmek amacıyla XGBoost, LightGBM ve CatBoost algoritmaları karşılaştırmalı olarak değerlendirilmiştir. Deneysel sonuçlar, CatBoost'un doğrulama aşamasında MAE ve RMSE metriklerinde öne çıktığını, LightGBM'in ise test setinde en düşük RMSE değerine ulaştığını göstermiştir. Tüm algoritmaların test setinde 0.94 R^2 değerine konverj etmesi, modellerin yüksek açıklayıcılık gücüne sahip olduğunu ortaya koymaktadır. Özellik önemlilik analizleri ise özellikle tahsis edilen/gerekli bant genişliği oranı (alloc_ratio) ve sinyal gücünün (Signal_Strength) kaynak tahsisi üzerinde belirleyici faktörler olduğunu göstermiştir.

Sonuçlar, gradyan artırmalı modellerin 6G gibi karmaşık ağ ortamlarında kaynak tahsisi tahmininde etkili ve uygulanabilir çözümler sunduğunu ortaya koymaktadır. Bununla birlikte, performans farklılıklarının metrik bazında değişkenlik göstermesi, algoritma seçiminde problem türü ve kullanım senaryosuna özgü optimizasyonun kritik olduğunu göstermektedir.

Gelecek çalışmalarda, hiperparametre optimizasyonunun daha geniş kapsamlı yapılması ve ansambl yöntemlerin (model birleştirme yaklaşımları) performansa etkisinin incelenmesi önerilmektedir. Ayrıca, gerçek zamanlı ağ verilerinin kullanıldığı çevrimiçi öğrenme (online learning) senaryoları ile modellerin dinamik ağ koşullarına adaptasyonu değerlendirilebilir. Derin öğrenme tabanlı hibrit yaklaşımlar, özellikle yoğun IoT ve ultra düşük gecikme gerektiren uygulamalarda kaynak tahsisini daha da iyileştirebilir. Bunun yanında, farklı coğrafi senaryolardan elde edilen büyük ölçekli veri kümelerinin kullanılması ve güvenlik/enerji verimliliği boyutlarının da modele dahil edilmesi, 6G ağ yönetiminde daha kapsamlı ve sürdürülebilir çözümler geliştirilmesine katkı sağlayacağı düşünülmektedir.

KAYNAKLAR

[1] A. Alsharif, R. Nordin, M. F. Younis, and N. F. Abdullah, "6G Wireless Communication: Vision, Challenges, and Future Directions," *IEEE Access*, vol. 11, pp. 33014–33035, Mar. 2023.

[2] M. Nauman, M. A. Jan, and S. H. Bouk, "AI-Driven Data Analytics for Orchestration and Control in B5G Services," in *Proc. IEEE GLOBECOM 2025*, pp. 1–6, 2025.

[3] W. Khan, M. Arif, and T. Saba, "Enhancing Security in 6G-enabled Wireless Sensor Networks for Smart Cities," *Frontiers in Sustainable Cities*, vol. 4, Article 1580006, 2025.

[4] C. Chatzistefanidis, G. Kormentzas, and E. Pallis, "Experimental Evaluation for a Beyond-5G Traffic Steering Case," in *Proc. IEEE ICC*, pp. 1–7, 2023.

[5] S. Polaganga and H. Liang, "Ensemble Prediction of RRC Session Duration in Real-World NR/LTE Networks," *ICT Express*, vol. 10, no. 1, pp. 56–62, Feb. 2024.

[6] M. Ismail, A. A. Khattak, and F. Al-Turjman, "Metaheuristic-Based Methodology for Attack Detection in 5G/6G Wireless Networks," *Mathematics*, vol. 13, no. 11, p. 1736, 2025.

[7] Mukherjee, A., Goswami, P., Khan, M. A., Manman, L., Yang, L., & Pillai, P. (2020). Energy-efficient resource allocation strategy in massive IoT for industrial 6G applications. IEEE Internet of Things Journal, 8(7), 5194-5201.

[8] A. Dari, "Resource Allocation Dataset for 6G Networks," *Kaggle*, Online: https://www.kaggle.com/datasets/ajithdari/resource-allocation-6g, Accessed: Aug. 8, 2025.

[9] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.

[10] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 31, 2018.

Comparison of Deep Learning and Machine Learning Models for Cafe Revenue Forecasting

Ali Pekin[1], Kemal Adem[2]

[1] *Yazılım Mühendisliği, Trabzon Üniversitesi, Trabzon, Türkiye, alipekin@trabzon.edu.tr, ORCID ID: 0009-0000-5026-6032*

[2] *Bilgisayar Mühendisliği, Sivas Cumhuriyet Üniversitesi, Sivas, Türkiye, kemaladem@cumhuriyet.edu.tr, ORCID ID: 0000-0002-3752-7354*

ABSTRACT

In this study, the performance of traditional machine learning algorithms and deep learning models such as RNN, LSTM, and GRU was compared to predict the daily revenue of a coffee shop. The dataset used in the study contains 2000 daily samples consisting of variables such as customer count, average order, operating hours, staff count, marketing expenditure, and location density. The five most successful machine learning models were identified using the PyCaret library; additionally, a total of 15 models with different layer and neuron structures were developed for each deep learning architecture. All models were evaluated using MAE, RMSE, and R^2 metrics. According to the results, the GRU-based deep learning model (GRU_1) showed the most successful performance with an R^2 score of 0.958. This demonstrates that GRU and LSTM architectures, which work particularly well with sequential data, are more effective than traditional methods in time series problems such as revenue forecasting. In the future, it is intended to further improve performance by using larger datasets or making improvements to the model architectures.

Keywords: *Deep Learning, Revenue Forecasting, Machine Learning, Regression*

INTRODUCTION

Revenue forecasting is an important issue for businesses to make growth plans, use their resources effectively, and increase profitability. At the organizational level, sales forecasts are critical for decision-making processes in various areas such as operations, marketing, production, and finance. Accurate sales forecasts enable businesses, especially those seeking investment capital, to use their resources effectively [1]. Factors such as customer numbers, employee numbers, marketing expenditure, and location can directly affect revenue, particularly for businesses in the service sector. Evaluating all these variables together requires a more comprehensive and data-driven approach. In the internet age, the proper development of sales forecasting models is critical for businesses to adapt to fluctuating market demands and make sensible inventory plans. However, the presence of numerous non-linear factors often renders traditional sales forecasting methods insufficiently accurate [2]. At this point, data science and artificial intelligence applications come to the fore with advances in technology. Today, in addition to traditional statistical methods, machine learning and deep learning techniques are also frequently used; these techniques offer more accurate and reliable forecasts thanks to their ability to learn complex patterns based on historical data.

A new dataset titled "Car Prices in the Wild" has been created for the online used car market in Turkey, reflecting real-world data issues such as outliers and erroneous entries. Using this dataset, a study was conducted comparing the success of traditional machine learning algorithms and deep learning architectures in predicting used car prices. Although deep learning models did not outperform classical methods, they yielded similar results. The findings revealed that deep learning methods demonstrated comparable performance to classical methods despite the limited data size. Furthermore, it was noted that larger datasets and online learning approaches offer important research areas for future studies [3]. Another study found that housing prices are influenced by physical and location-based characteristics such as the size of the house, number of rooms, age of the building, number of floors, type of heating, number of bathrooms, parking, pool,

security, whether it is a duplex, terrace, sea view, city center, and proximity to universities. It was understood that the price of gold per gram, the price of iron per kilogram, the price of new vehicles, and mortgage interest rates are the determining economic indicators. Since all residences have a single living room, the number of living rooms was found to be insignificant. It was observed that the direct effect of the site on the price is weak, but amenities such as the site's pool, parking, and security are effective in price prediction. A 77.19% accuracy rate was achieved using the Random Forest (RF) algorithm. To measure the impact of economic factors more accurately, it is recommended to test variables such as the price of gold per gram during periods of short-term increases and decreases. Increasing the amount of data is also recommended to achieve more accurate results [4].

In this study, a total of 15 different deep learning models were designed and applied to predict a business's daily revenue. These included traditional machine learning algorithms as well as 5 RNN, 5 LSTM, and 5 GRU models, each with different layer and neuron structures. Prior to the modeling process, steps such as checking for missing values and normalization were applied during the data preprocessing stage. The performance of the models was tested and analyzed using common regression evaluation metrics such as MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and R^2 (Determination Coefficient). Thus, the success of both traditional methods and deep learning architectures in daily revenue prediction was revealed comparatively.

MATERIAL AND METHOD

A. *Dataset*

In this study, a dataset created to develop regression models for estimating the daily revenue of a coffee shop was used [15]. The dataset consists of 2000 samples, each representing one day. Each sample contains various variables related to the daily operations of the business. These variables are shown in Table 1.

TABLE 1. TABLE RELATED TO VARIABLES IN THE DATA SET

Variable Name	*Description*	*Variable Type*	*Minimum Value*	*Maximum Value*
Number_of_Customers_Per_Day	Total number of customers per day	Integer	50	499
Average_Order_Value	Average monetary value of orders	Decimal Number	2.50	10.00
Operating_Hours_Per_Day	The number of hours the store was open that day	Integer	6	17
Number_of_Employees	Total number of employees working that day	Integer	2	14
Marketing_Spend_Per_Day	Daily expenditure on marketing activities	Decimal Number	10.12	499.74
Location_Foot_Traffic	Number of potential customers passing by the business location	Integer	50	999
Daily_Revenue	Daily total revenue	Decimal Number	-58.95	5114.60

There are no missing values in the data set. The variables in the data set cover basic information that may affect business performance. The relationships between these variables and their effects on the target variable play an important role in the success of the developed models. As a result of the analyses performed on these variables, it was deemed necessary to apply some pre-processing before proceeding to the modeling process. In particular, normalizing the variables due to differences in value ranges is an important step for the models to learn more effectively. Based on this basic statistical information, various regression models were trained using the data set. The success of the models was evaluated based on the similarity of the predicted daily income values to the actual values.

B. Machine Learning and Deep Learning

PyCaret is an open-source software, a Python-based, low-code machine learning framework that enables the rapid deployment of models after data preparation. Its purpose is to enable the development, evaluation, comparison, and deployment readiness of machine learning models in the simplest and most effective way possible [16]. In the study, machine learning models were first automatically trained using the PyCaret library, and the five most successful models were selected based on the performance scores obtained. These models are suitable for regression problems and are the algorithms that contribute to income prediction with the highest accuracy.

In addition to machine learning models, three different deep learning-based model architectures were examined: RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory), and GRU (Gated Recurrent Unit). A total of five models (fifteen models in total) with different layer and neuron structures were designed and trained for each architecture. The same training and test datasets were used for training all models. Regression evaluation metrics such as MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and R^2 (Determination Coefficient) were used when comparing model performances. Thus, the performance of both traditional machine learning and deep learning models was analyzed. Model development and testing processes were carried out using the Google Colab platform. This allowed for the analysis of the performance of both traditional machine learning and deep learning models.

1) RNN (Recurrant Neural Network)

RNNs are structures that work on sequential data by incorporating past information into the model's current decision-making process [5]. While effective with time-dependent data, information loss can occur in long connections.

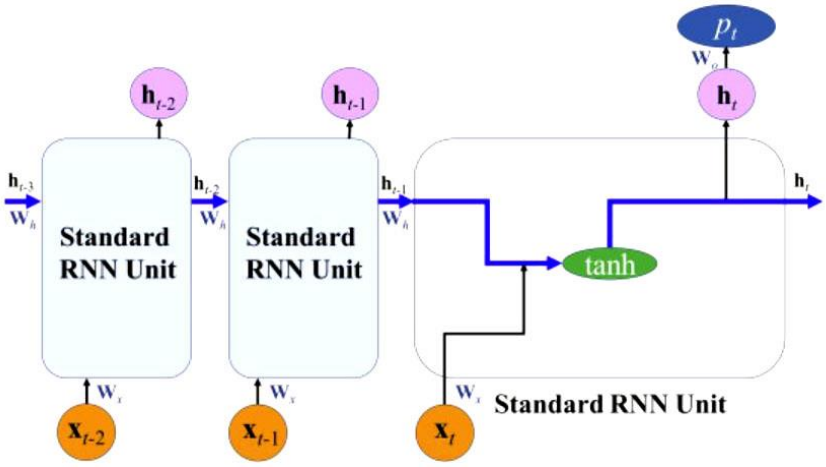


Figure 1. *RNN Architecture[9]*

The classic RNN architecture shown in Figure 1 generates the output of the hidden layer units at each time step based on the state at the previous time step. RNNs are simple yet powerful structures. However, in practice, training them becomes difficult when there is a long time interval between previous events and the target output [6]. During forward propagation, the effect of the initial inputs on the output diminishes over time and is erased by new inputs arriving at the hidden layer units' activations. Thus, the network begins to forget the initial inputs over time. During backward propagation, when the Backpropagation Through Time (BPTT) technique is used to propagate backward through time, the vanishing gradient problem arises [7]. This situation allows RNNs to learn short-term dependencies but causes them to fail to learn long-term dependencies.

2) LSTM (Long Short-Term Memory)

LSTM is a structure developed to eliminate the memory loss problem of RNN. It is used in areas such as time series analysis, handwriting recognition, and music composition [8]. It controls data flow with input and output gates

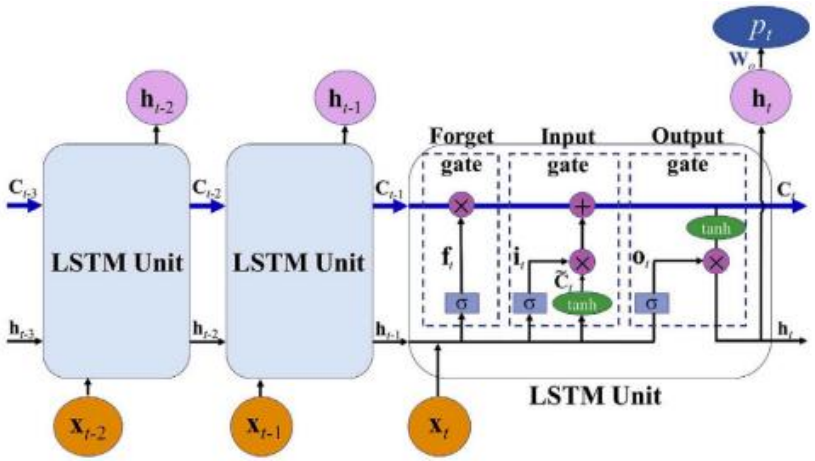


Figure 2. *LSTM Architecture [9]*

As shown in Figure 2, the LSTM unit consists of a cell state (C_t), a forget gate (f_t), an input gate (i_t), and an output gate (o_t). The gates in the LSTM architecture play a crucial role in determining how the cell state is updated. The forget gate (f_t) decides how much information from the previous cell state ($\mathcal{C}_{t-1}$) will be discarded, while the input gate (i_t) determines how much information from the current candidate cell state will be transferred to the current cell state (C_t).The output gate (o_t) controls how much information is transferred from the cell to the rest of the network. If the forget gate is set to zero and the input and output gates are set to one, the LSTM structure functions identically to a standard RNN [9].

3) GRU (Gated Recurrent Unit)

GRU is a simpler version of LSTM and operates with only two gates (update and reset). Because it has fewer parameters, its training time is shorter. GRU can perform similarly to, or even better than, LSTM on time-dependent data [10].

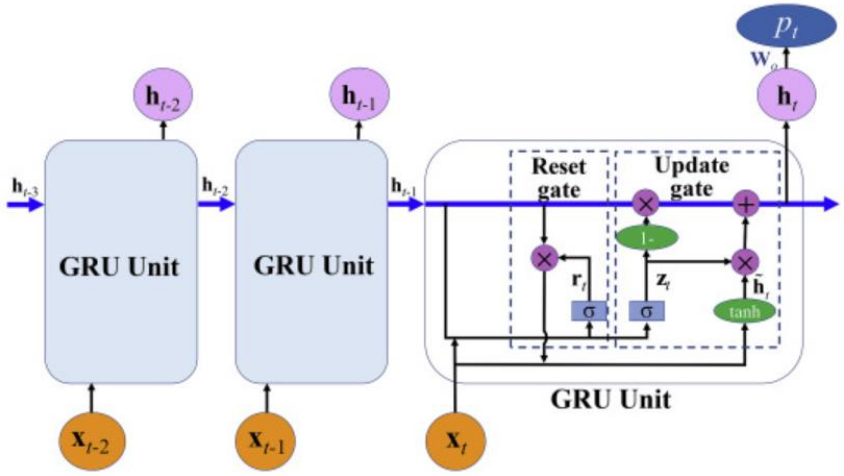


Figure 3. *GRU Mimarisi[9]*

As shown in Figure 3, the GRU unit consists of an update gate (z_t), a reset gate (r_t), and a hidden state (h_t) component. This structure has a simpler architecture compared to LSTM, which directly carries the cell state. The GRU is effective at learning both short-term and long-term dependencies by reducing information loss through its gate mechanisms. The reset gate (r_t) in the GRU (Gated Recurrent Unit) architecture determines how new input data is combined with past information, while the update gate (z_t) controls how much of the past information is retained. If both gates are set to one, the GRU model reverts to the standard RNN structure. The literature indicates that GRU performs particularly well on smaller datasets [11].

C. *Performance Evaluation Metrics*

MAE, RMSE, and R-Square metrics were used to evaluate the results of deep learning and machine learning models.

1) MAE (Mean Absolute Error)

The mean absolute error (MAE) is a commonly used error measure for evaluating and reporting the performance of prediction-based regression models that require estimating a numerical value. MAE is a simple and powerful measurement tool that calculates the average of the absolute values of the errors between a model's predicted and actual values. It assesses how well a model generalizes and how well its predictions match the actual values. In this study, MAE was used to find the predictive modeling closest to the experimental results, to see deviations from the experimental values, and to capture the relationships between the dependent and independent variables [12]. MAE is calculated using the following formula (1).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (1)$$

2) RMSE (Root Mean Square Error)

Root Mean Square Error (RMSE): It is the square root of MSE. It is an error metric used to measure the magnitude of differences between predicted values and actual values [13]. The RMSE equation (2) is shown below.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

3)) R^2 (Determination Coefficient)

It measures the rate of change in the output variable determined by the model's input variables. It is also a metric that shows how much of the total variance in the dependent variable a model explains. A regression model may fit the training data well but may not fit the test data if it has too many independent input variables. Additional independent variables added to the model are taken into account. The R-Square value, which expresses the relationship between actual and predicted values, varies between 0 and 1. The closer the R-Square value is to 1, the more successful the model is [14]. The R-Square equation (3) is shown below.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (3)$$

FINDINGS AND DISCUSSION

The results of the top 5 models with the highest R^2 values from traditional machine learning models using PyCaret are shown in Table 2. In the model evaluations performed with PyCaret, default hyperparameter values were used, and within this scope, a 10-fold cross-validation method was automatically applied.

TABLE 1. MACHINE LEARNING MODEL RESULTS

	MAE	*RMSE*	*R2*
Gradient Boosting Regressor	*181.7102*	*225.9102*	***0.9449***
Extra Trees Regressor	185.6718	229.7307	0.9429
Light Gradient Boosting Machine	188.3506	231.7771	0.9420
Random Forest Regressor	187.3937	233.1319	0.9415
Extreme Gradient Boosting	200.9533	247.9622	0.9337

As shown in Table 2, the Gradient Boosting Regression model stands out as the most successful model with the lowest error values (MAE: 181.71, RMSE: 225.91) and the highest R^2 score (0.9449). This indicates that the model provides high accuracy in daily revenue predictions. The other models also performed similarly well, producing results that were close to each other in terms of error metrics. The architectures of the deep learning models created are given in Table 3. The

activation function "ReLU" and the optimization algorithm "Adam" were preferred in all deep learning models. Additionally, the number of epochs in the training of the models was fixed at 200, and all models were trained with the same training parameters.

TABLE 2. DEEP LEARNING MODELS

Model Name	*Architecture*	*Number of Layers*	*Number of Neurons in Each Layer*	*Dropout Number*	*Dropout Rate*	*Dense Number*
RNN_1	RNN	3	512, 256, 128	2	0.2	2
LSTM_1	LSTM	3	512, 256, 128	2	0.2	2
GRU_1	GRU	3	512, 256, 128	2	0.2	2
RNN_2	RNN	4	64, 32, 16,8	2	0.3	2
LSTM_2	LSTM	4	64, 32, 16,8	2	0.3	2
GRU_2	GRU	4	64, 32, 16,8	2	0.3	2
RNN_3	RNN	2	128, 64	1	0.2	3
LSTM_3	LSTM	2	128, 64	1	0.2	3
GRU_3	GRU	2	128, 64	1	0.2	3
RNN_4	RNN	1	1024	1	0.4	3
LSTM_4	LSTM	1	1024	1	0.4	3
GRU_4	GRU	1	1024	1	0.4	3
RNN_5	RNN	3	256, 128, 64	2	0.3	3
LSTM_5	LSTM	3	256, 128, 64	2	0.3	3
GRU_5	GRU	3	256, 128, 64	2	0.3	3

The architectural structures of the deep learning models used in Table 3 are presented in detail. The number of layers in all models varies between 1 and 4. The number of neurons in each layer generally follows a decreasing structure and varies between 8 and 1024. This indicates that architectures starting with a high number of neurons and decreasing the number of neurons as the layers progress were preferred. The number of dropouts used in the models was set to 1 or 2; the dropout rate was kept between 0.2% and 0.4%. These rates are ideal levels to prevent overfitting. Additionally, the number of dense layers in the models was generally set to 2 or 3. The results of the deep learning models are presented in Table 4.

TABLE 3. DEEP LEARNING MODEL RESULTS

	MAE	*RMSE*	*R2*
RNN_1	206.825	167.906	0.954
LSTM_1	206.371	169.191	0.954
GRU_1	***197.589***	***161.566***	***0.958***
RNN_2	292.389	223.300	0.909
LSTM_2	271.446	212.664	0.921
GRU_2	254.165	198.378	0.931
RNN_3	201.936	164.214	0.956
LSTM_3	234.645	186.330	0.941
GRU_3	226.225	180.977	0.945
RNN_4	236.134	188.569	0.940
LSTM_4	237.323	190.484	0.940
GRU_4	233.553	186.522	0.942
RNN_5	200.484	163.696	0.957
LSTM_5	224.553	179.620	0.946
GRU_5	201.011	164.522	0.957

When examining the results of the deep learning models presented in Table 4, it is observed that GRU and LSTM architectures generally provide higher accuracy compared to the classical RNN structure. In particular, the GRU_1 model showed the highest success among all models with an R^2 score of 0.958. This model is closely followed by RNN_5 (0.957) and GRU_5 (0.957) with very similar R^2 scores. Similarly strong results were obtained with LSTM architectures; for example, the LSTM_1 model stands out with an R^2 value of 0.954. RNN models produced lower R^2 scores compared to GRU and LSTM. These differences highlight the impact of structural differences between architectures. Advanced architectures such as GRU and LSTM can learn long-term dependencies more effectively, especially in sequential data such as time series, and therefore provide more stable and accurate results than classical RNN for such problems. After evaluating the success levels of the relevant models, visually examining these successes also helps to understand model performance. The scatter plot of predicted and actual results is an intuitive method for evaluating the performance of regression algorithms [17]. The actual values corresponding to the GRU_1 model with the highest R-squared value and the RNN_2 model with the lowest R-squared value are shown in Figure 4 and Figure 5. The blue points represent actual values, while the orange points represent predicted values. The x-axis of the graph corresponds to the relevant day, while the y-axis shows the profit amount for that day.

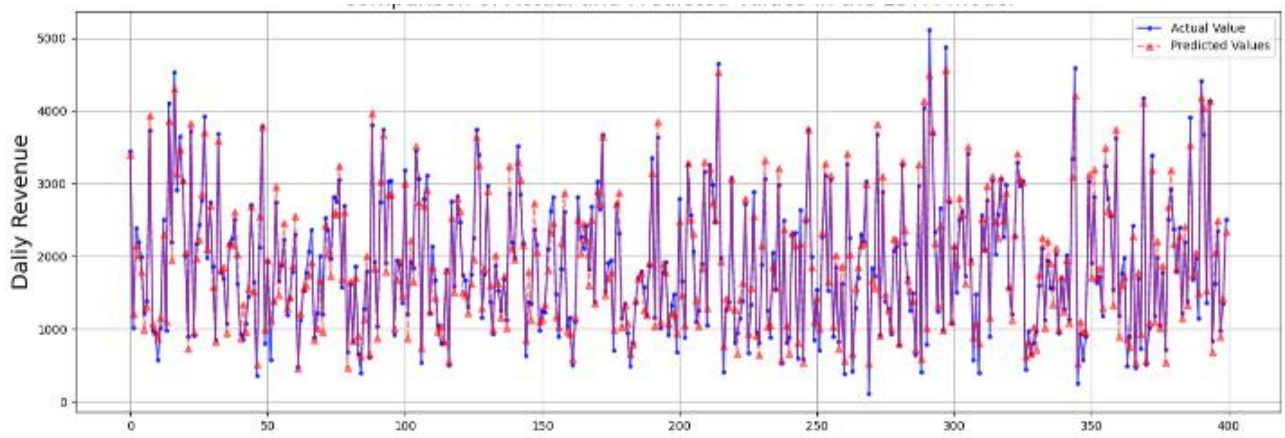


Figure 4. Actual Value-Predicted Value Graph for the GRU_1 Model

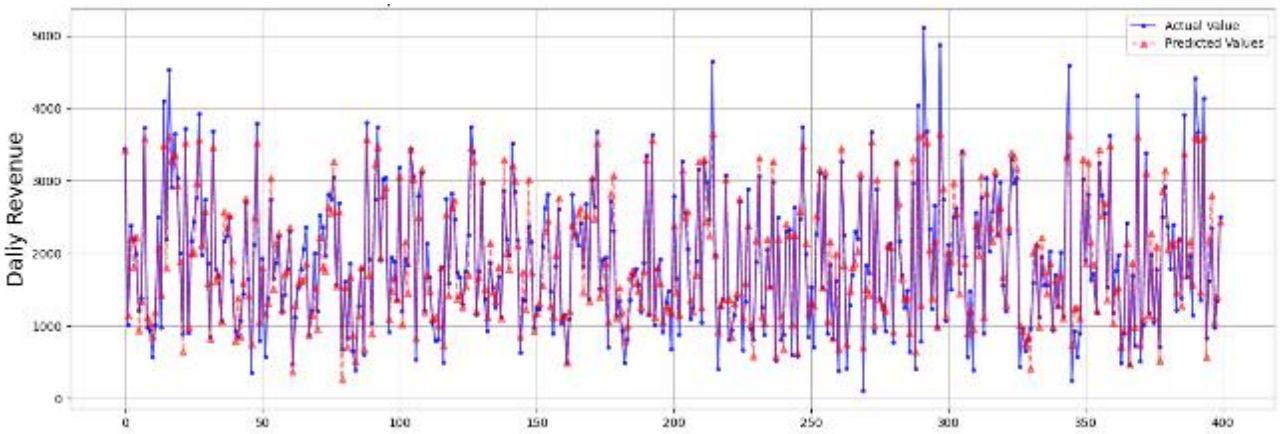


Figure 5. Actual Value-Predicted Value Graph for the RNN_2 Model

Looking at Figures 4 and 5, it can be seen that both models successfully follow the general trends. However, the predictions of the GRU_1 model align more closely with the actual values; it produces more accurate results, especially in regions with sudden fluctuations. The RNN_2 model, on the other hand, can capture short-term changes but shows deviations during periods of sudden increases or decreases. This situation can be considered a result of the difficulties the RNN architecture experiences in learning long-term dependencies. The differences in model performance stem primarily from the GRU

architecture's ability to more effectively capture the long-term dependencies that the RNN struggles to learn. This explains why deep learning models are more successful in complex and highly variable time series problems such as revenue forecasting.

RESULTS

A café's daily revenue is of great importance in terms of the business's financial health and sustainability. Sales forecasting is a challenging problem because demand varies depending on many factors. Therefore, accurate revenue forecasting guides the café's strategic planning for the future. In this study, traditional machine learning models were compared with deep learning models for predicting café revenue, and GRU-based models were found to be particularly successful. The highest R^2 score obtained using PyCaret was 0.9449, belonging to the Gradient Boosting Regressor model. In contrast, among deep learning models, the GRU_1 model, created with the GRU architecture, yielded the best result with an R^2 score of 0.958. It is believed that GRU and LSTM architectures can better handle sequential data such as time series and therefore perform at a high level. Some common features that stand out in these models are starting with a high number of neurons in the first layer, reducing the number of neurons as the layers progress, applying dropout at a rate of 0.2–0.3%, and using several dense layers. The success of models with RNN architecture has been slightly lower than others. This shows that RNNs cannot learn dependencies formed over time as well as GRU or LSTM. As a result, on this dataset, the GRU architecture achieved the highest success, and it was understood that the correct selection of the number of layers, the number of neurons, and the dropout rate significantly contributed to the model's performance.

The dataset used in this study is limited to a specific period and contains 2000 samples. Expanding the dataset in future studies, i.e., collecting data for more days or collecting data from different cafes, may contribute to the models providing more accurate and generalizable results. Furthermore, adding certain potentially influential variables not included in the model (e.g., weather conditions, weekday/weekend information, special days) to the dataset could further improve prediction accuracy. However, only specific deep learning architectures (RNN, LSTM, GRU) were used in this study. In recent years, transformer architectures have come to the fore with their ability to capture long-term dependencies more effectively thanks to their attention mechanism. These models, which deliver powerful results especially on complex time series data, provide parallel processing capabilities and better learning of long-term dependencies compared to traditional RNN-based structures, thanks to the transformer architecture proposed by Vaswani and colleagues [19]. Therefore, performance comparisons of different deep learning architectures, especially with the inclusion of transformer models, will benefit future studies. Additionally, more systematic optimization of hyperparameters (such as the number of layers, number of neurons, and dropout rate) may positively impact model success.

REFERENCES

[1] Cheriyan, S., Ibrahim, S., Mohanan, S., & Treesa, S. (2018, August). Intelligent sales prediction using machine learning techniques. In 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE) (pp. 53-58). IEEE.

[2] Han, Y. (2024, March). Attention Mechanism-Based CNN-BiLSTM for Sales Revenue Prediction. In 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE) (pp. 740-743). IEEE.

[3] Uysal, F. (2023). Comparative analysis of various machine learning and deep learning approaches for car resale price prediction in the turkish market. Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 13(1), 1-1.

[4] Ozdemir, M., Yıldız, K., & Büyüktanır, B. (2022). Housing Price Estimation with Deep Learning: A Case Study of Sakarya Turkey. Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi, 9(1), 138-151. https://doi.org/10.35193/bseufbd.998331

[5] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://doi.org/10.1038/323533a0

[6] Pascanu, R., Mikolov, T., & Bengio, Y. (2013, May). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310-1318). Pmlr.

[7] Pascanu, R., Mikolov, T., & Bengio, Y. (2012). Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, *2*(417), 1.

[8] Alpay, Ö. (2020). LSTM Mimarisi Kullanarak USD/TRY Fiyat Tahmini. Avrupa Bilim Ve Teknoloji Dergisi4 52-456. https://doi.org/10.31590/ejosat.araconf59

[9] Wei, X., Zhang, L., Yang, H. Q., Zhang, L., & Yao, Y. P. (2021). Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks. *Geoscience Frontiers*, *12*(1), 453-467.

[10] S. ARSLAN, (2023). Gated recurrent unit network-based fuzzy time series forecasting model, Afyon Kocatepe University Journal of Sciences and Engineering 23. https://doi.org/10.35414/akufemubid.1175297.

[11] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[12] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, e623. https://doi.org/10.7717/peerj-cs.623

[13] Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. Geoscientific Model Development Discussions, 2022, 1–10. https://doi.org/10.5194/gmd-2022-123

[14] Verma, U., Garg, C., Bhushan, M., Samant, P., Kumar, A., & Negi, A. (2022). Prediction of students' academic performance using machine learning techniques. In 2022 International Mobile and Embedded Technology Conference (MECON) (pp. 151–156). IEEE. https://doi.org/10.1109/MECON53876.2022.9752064

[15] https://www.kaggle.com/datasets/talhaanjum0/coffee-shop-revenue/data

[16] Sarangpure, N., Dhamde, V., Roge, A., Doye, J., Patle, S., & Tamboli, S. (2023, March). Automating the machine learning process using PyCaret and Streamlit. In 2023 2nd International Conference for Innovation in Technology (INOCON) (pp. 1-5). IEEE.

[17] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

[18] Sun, Z. L., Choi, T. M., Au, K. F., & Yu, Y. (2008). Sales forecasting using extreme learning machine with applications in fashion retailing. Decision support systems, 46(1), 411-419.

[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.

Application of Retrieval-Augmented Generation (RAG) Approach for Turkish Open-Field Question-Answering System

Ali Pekin[1], Abdulkadir Şeker[2]

[1] *Yazılım Mühendisliği, Trabzon Üniversitesi, Trabzon, Türkiye, alipekin@trabzon.edu.tr, ORCID ID: 0009-0000-5026-6032*

[2] *Bilgisayar Mühendisliği, Sivas Cumhuriyet Üniversitesi, Sivas, Türkiye, aseker@cumhuriyet.edu.tr, ORCID ID: 0000-0002-4552-2676*

ABSTRACT

Open-Domain Question Answering (ODQA) systems aim to generate accurate and meaningful responses to natural language questions posed by users, without being confined to a limited domain, by leveraging extensive knowledge sources. Inthis study, the Retrieval-Augmented Generation (RAG) architecture was employed to establish an ODQA system in the Turkish language. RAG combines the processes of information retrieval and natural language generation to produce context-sensitive and informative answers. In experiments conducted with a sample dataset created from Turkish Wikipedia data, the model achieved an 80% accuracy rate by providing correct answers to 24 out of 30 questions. These results demonstrate that RAG-based systems can be effective in open-domain question-answering applications in the Turkish language.

Keywords: *Open-Domain Question Answering, Natural Language Processing, Retrieval-Augmented Generation, Turkish Wikipedia*

INTRODUCTION

Providing fast, accurate, and meaningful access to information has become an indispensable need for individuals and organizations in today's information age. In this context, Open-Domain Question Answering (ODQA) systems have emerged as advanced information access systems that aim to generate answers to natural language questions posed by users without being restricted to a specific domain. These systems are used in many different areas, such as search engines, digital assistants, customer service, education, healthcare, and law, and are an important artificial intelligence solution that improves the user experience [1].

The fundamental challenge of question-answering systems is establishing the correct connection between short, context-limited questions from users and large-scale, often unstructured information sources. To overcome this challenge, hybrid approaches such as Retrieval-Augmented Generation (RAG) have been developed, which integrate traditional information retrieval methods with natural language generation techniques [2]. The RAG model has the potential to produce more context-sensitive and explanatory responses by combining both information retrieval and language modeling tasks.

Advances in natural language processing (NLP) have accelerated, particularly with the development of word embedding methods. The Word2Vec model developed by Mikolov and colleagues in 2013 enabled language models to better understand context by representing the semantic similarities of words in a vector space [3]. This approach has laid the foundations for today's transformer-based large language models (LLMs). Models such as BERT [4], GPT [5], and T5 are effectively used in both question understanding and answer generation tasks. However, the success of these models in knowledge-based tasks depends not only on the model itself but also on the scope of the information sources it can access and their appropriate use. This is where the RAG architecture has the potential to increase both accuracy and contextual consistency through the combined work of the retriever and generator components [2].

In this study, the RAG model was applied to establish an open-domain question-answering system in the Turkish natural language. While there are numerous ODQA studies focused on the English language in the current literature, it is observed that there are a limited number of such systems in languages with relatively low resources, such as Turkish. Therefore, this

RAG-based system, developed using sample data obtained from Turkish Wikipedia, aims to both test the applicability of the method in Turkish and evaluate its potential success level.

MATERIAL AND METHOD

In this study, the Retrieval-Augmented Generation (RAG) architecture was used to develop a Turkish open-domain question-answering system. RAG is a method that integrates text-based information retrieval with natural language generation, combining both retrieval and language model-based response generation components [2]. The system's two main components, the retriever and generator, are described in detail below.

A. Data Set Used

The documents used for training and evaluating the model were taken from the Turkish Wikipedia dataset[6]. Wikipedia is widely preferred in knowledge-based language modeling studies because it contains open and high-quality information [7]. In this study, texts were extracted from Wikipedia dump files using <doc> tags, and each document was structured as a unique example. In total, over 10,000 paragraphs were processed, but considering memory limitations and processing time during experiments, 100 documents were used as a sample.

B. Data Pre Processing

After the data set was imported into the system, unnecessary characters, spaces, and errors were removed; long texts were divided into smaller chunks for ease of processing.

C. Retrieval-Augmented Generation (RAG) model

The RAG model consists of two main components:

- Retriever: This is the information retrieval component that returns the documents most relevant to the query. In this study, FAISS (Facebook AI Similarity Search) was used as the retriever. FAISS enables fast and efficient document retrieval based on vector representations. FAISS is a comprehensive toolbox consisting of various indexing methods, including components such as preprocessing, compression, and approximate search, rather than just a single indexing method[8].
- Generator: It is a language model that processes returned documents contextually and generates responses in natural language. In this study, the generation process was performed using the deephermes chatbot with an open router structure.

The advantage of the RAG architecture is that the generator can use the documents retrieved by the retriever as context. This enables response generation based not only on pre-trained knowledge but also on external knowledge sources [2], [9].

D. Question-Answer Data Set

A total of 30 question-answer pairs were created for performance evaluation. The questions were manually prepared to correspond with topics selected from Wikipedia, and the answers were checked to ensure they were actually found in the text. This method provided an important reference for measuring the model's ability to retrieve and generate information.

E. Application Environment

All operations were performed using the Python programming language. The main libraries used are as follows:

- transformers (HuggingFace) –for model invocation and text generation
- for document indexing and similarity search
- datasets, nltk, sentence-transformers – for data preprocessing and vector representation

The experiments were conducted in the Pycharm environment.

FINDINGS AND DISCUSSION

In this study, a prototype model was developed using the Retrieval-Augmented Generation (RAG) architecture to establish a Turkish open-domain question-answering system. The dataset used consists of 100 document samples selected from Turkish Wikipedia articles. The system's performance was evaluated using 30 question-answer pairs generated from these documents. Experimental results show that the model answered 24 out of 30 questions correctly, achieving an accuracy rate of approximately 80%.

These results demonstrate that RAG-based architectures can also be effective in morphologically rich languages such as Turkish. The model's successful performance can be attributed to the contribution of the vector-based document retrieval process. Vector-based search enables matching at the semantic level rather than the word level, allowing for stronger connections between the query and the context [2].

On the other hand, when examining the 6 questions where the system failed, the following situations were observed:

- Insufficient or Irrelevant Documents: For some questions, documents retrieved from the vector database were not sufficiently relevant in context. This resulted in the generation of incorrect or incomplete answers.
- Limitations of the Language Model: In some cases, the language model used to generate responses failed to interpret the context correctly and produced responses with weak coherence. This was particularly evident in questions containing ambiguous words [2].
- Limited Data Set: Working with only 100 documents limited the diversity of information and made it difficult to select appropriate documents for some questions.
- Insufficient Chunk Settings: The chunk size and chunk overlap parameters used in document segmentation have, in some cases, led to the context not being fully preserved. Particularly in paragraphs containing long and multi-layered information, important pieces of context have been scattered across different chunks and have not been effectively represented.
- Low Number of Documents Retrieved (k=2): The value k=2 was used to find the closest documents. This means that responses were generated based on only two documents. When the document containing the answer to the question was not among these two results, the model produced an incorrect or incomplete response. Increasing the k value can improve accuracy, especially when information is scattered across documents.

The model's high accuracy rate has been found promising despite these limitations. It is believed that the system's accuracy can be further improved with a larger and more diverse dataset. Additionally, establishing a hybrid structure using keyword-based filtering rather than solely vector-based methods in the document retrieval process may provide higher contextual consistency [9].

CONCLUSION AND FUTURE RESEARCH

In this study, the Retrieval-Augmented Generation (RAG) architecture was applied for Turkish open-domain question-answering systems, and a prototype system supported by documents obtained from Turkish Wikipedia data was developed. The developed system achieved an 80% accuracy rate on a sample of 30 questions, demonstrating the usability of this architecture in the Turkish language. This success shows that knowledge-based natural language processing applications can be effectively developed even in morphologically rich languages such as Turkish.

Although the results obtained are promising, the system's incorrect responses to certain questions have also revealed certain limitations. In particular, the low chunk size and overlap amount caused the document content to be fragmented and disconnected from its context, preventing the model from accessing sufficiently meaningful information. At the same time, limiting the number of closest documents to only two resulted in some documents containing correct answers not being retrieved. These situations demonstrate that parameters must be carefully optimized for the system to produce more accurate results.

In future studies, the system is planned to be tested on larger and more diverse datasets. This will allow for a better evaluation of the model's performance across different topics and enable the system's scope to be expanded. Along with parameter optimization studies, comparisons with different language models can be made to determine the most suitable generative model. Furthermore, the system is intended for use in real-time applications, and dynamic structures developed through user interaction will also be explored in this direction.

In conclusion, this study represents an important first step toward developing Turkish open-domain question-answering systems. The applied architecture and the experimental findings obtained provide a strong foundation for future work. With properly structured data processing steps and improved parameters, it will be possible to develop systems that can generate responses with higher accuracy and greater contextual consistency in Turkish.

INFORMATION

The relevant source code has been uploaded to GitHub [10].

REFERENCES

[1] Voorhees, E. M., & Tice, D. M. (2000). "Building a question answering test collection." SIGIR.

[2] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). "Retrieval-augmented generation for knowledge-intensive NLP tasks." NeurIPS.

[3] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). "BERT:

[4] Pre-training of deep bidirectional transformers for language understanding." NAACL-HLT.

[5] Brown, T., et al. (2020). "Language Models are Few-Shot Learners." NeurIPS.

[6] https://www.kaggle.com/datasets/mustfkeskin/turkish-wikipedia-dump

[7] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). "FEVER: a large-scale dataset for fact extraction and verification." *NAACL-HLT*.

[8] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P. E., ... & Jégou, H. (2024). The faiss library. arXiv preprint arXiv:2401.08281.

[9] Izacard, G., & Grave, E. (2020). Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282.

[10] https://github.com/alipekin/RAG/tree/main/rag

An Efficient Heuristic to Color Graphs Using Node Importance

Betül Boz[1]

[1] *Computer Engineering Department Marmara University, Istanbul, Turkey, betul.demiroz@marmara.edu.tr*

ABSTRACT

Graph coloring is an NP-hard problem in which adjacent vertices must be assigned to different colors. This problem can be used to solve many applications such as network management, resource allocation, and social network analysis, where scalability and solution quality are both critical. In this study, a novel algorithm that combines a heuristic-based initial coloring with metaheuristic optimization techniques is proposed. The initial heuristic assigns node importance scores through iterative weight updates and prioritizes high-weight nodes for coloring, producing an effective starting point. This solution is further refined using Tabu Search, which incorporates a short-term memory mechanism to prevent cycling and escape local minima. Together, these components balance computational efficiency with solution quality. The performance of the proposed approach is evaluated on the DIMACS benchmark suite, and the experimental study showed that it consistently achieves faster execution time and maintains competitive, and often superior, coloring performance compared to the widely used DSatur algorithm.

Keywords: *Heuristic, graph coloring, tabu search*

INTRODUCTION

The graph coloring problem is a classical NP-complete problem in graph theory, where each vertex must be assigned a color such that no two adjacent vertices share the same color. This problem has a wide range of applications in domains such as frequency assignment, register allocation, and scheduling [1-3]. However, as graph size increases into the scale of hundreds of thousands of nodes, the problem becomes increasingly complex and computationally demanding, making exact approaches infeasible.

Graph coloring is an NP-Complete problem [4] and many heuristic and metaheuristic methods have been widely used to address this challenge [5-9]. Among them, the DSatur (Degree of Saturation) algorithm [10] is a well-known heuristic that prioritizes vertices based on their saturation degree—the number of distinct colors assigned to their neighbors. While DSatur algorithm provides effective solutions on small to medium-sized graphs, its scalability and solution quality can degrade for large and complex instances.

To overcome these challenges, we propose a scalable and efficient algorithm that combines heuristic initialization with metaheuristic refinements through Tabu Search. Our heuristic computes *node importance* via iterative weight updates, prioritizing highly connected nodes that are typically harder to color. Tabu Search introduces a short-term memory mechanism that prevents cycling and facilitates escaping local minima, with color swaps to resolve difficult conflicts. Together, these strategies allow the algorithm to balance efficiency and solution quality.

The main contribution of this study is a novel method for computing node importance. In many real-world problems, data is represented as graphs, and the choice of which node to process first—as well as the order of subsequent nodes—directly affects the quality of the outcome. Traditionally, node degree, which counts the number of edges incident to a node, is used as a measure of importance, and algorithms often begin with the node of highest degree. However, node degree alone captures only the local connectivity of a node and provides no insight into the significance of its neighbors. To address this limitation, the heuristic proposed in this study assigns a weight to each node based not only on its own degree but also on the degrees of its neighbors, thereby incorporating their importance into the computation.

We evaluate the performance of the proposed approach on the DIMACS benchmark suite, comparing against DSatur. Results demonstrate that our method achieves competitive or improved chromatic numbers while significantly reducing runtime, highlighting its potential for practical use in real-world graph coloring applications.

METHODOLOGY

To address the graph coloring problem, we propose a scalable and efficient approach that combines a lightweight heuristic for initial coloring with a metaheuristic refinement step aimed at reducing the color number.

A. *Heuristic Based on Node Importance*

The heuristic assigns importance weights to nodes based on their structural connectedness. Initially, each node's weight is set to its degree. The weights are then refined through an iterative two-step process: (1) edge weights are computed as the sum of the weights of their incident nodes; (2) node weights are updated as the sum of the weights of all incident edges as shown in Fig. 1.

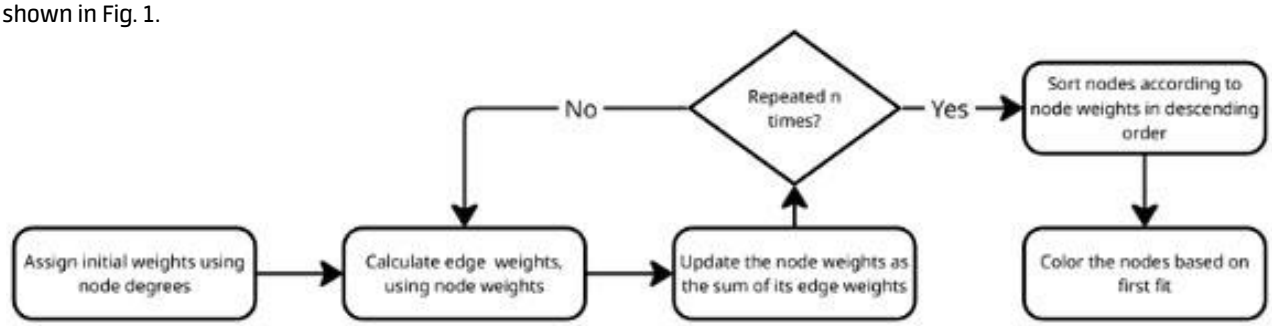


Fig. 1. Heuristic Coloring Process Flowchart

Although this process can be repeated multiple times to propagate weights through the graph, experimental results showed that a single iteration provided sufficient structural differentiation, making additional iterations unnecessary in practice. An example showing the calculated weights of a graph after one iteration is shown in Fig. 2.

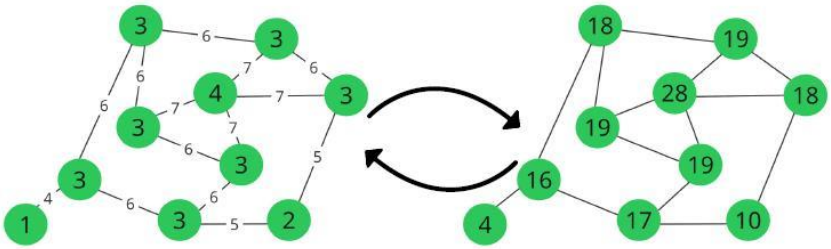


Fig. 2. Heuristic Weight Calculation Example

Algorithm 1 Heuristic Weight Calculation

Input: Graph $G = (V, E)$, number of iterations n
Output: Final node weights w_v and edge weights w_{uv}
Initialize: For each node $v \in V$, set $w_v \leftarrow \text{degree}(v)$
for $i = 1$ to n **do**
  for each edge $(u, v) \in E$ **do**
    $w_{uv} \leftarrow w_u + w_v$ ▷ Edge weight is sum of its nodes' weights
  end for
  for each node $v \in V$ **do**
    $w_v \leftarrow \sum_{(v,u)\in E} w_{vu}$ ▷ Node weight is sum of its edge weights
  end for
end for
return w_v and w_{uv}

Once node weights are obtained, multiple strategies for node ordering and color assignment were explored. Node selection strategies included: coloring the highest-weight node first and recalculating weights, coloring the neighbors (and

sometimes second-order neighbors) of the heaviest uncolored node before coloring the node itself, and globally sorting all nodes by weight. Color assignment strategies included: selecting the least populated color class, the most populated class, or the first available color class satisfying coloring constraints.

Algorithm 2 First-Fit Coloring Based on Node Weights

Require: Graph $G = (V, E)$ with weights $w(v)$ for each $v \in V$
Ensure: Color assignment $c(v)$ for each $v \in V$
1: Sort the nodes V in descending order by weight $w(v)$
2: **for** each node v in sorted V **do**
3: $c(v) \leftarrow$ FINDFIRSTFITCOLOR(v, c)
4: **end for**
5: **return** color assignment c

After extensive evaluation, two strategies emerged as the most effective:

- **Strategy 1 (Fast):** Nodes are sorted in descending order of weight and colored sequentially using a first-fit rule as given in Algorithm 2. This method is highly efficient with a time complexity of *O(n log n)* and is well-suited for time-sensitive applications.
- **Strategy 2 (Quality-Oriented):** The heaviest uncolored node is selected and colored, after which its neighbors' weights are increased by a weight factor as shown in Algorithm 3. This adjustment, determined empirically, consistently yielded lower chromatic numbers at the cost of higher complexity ($O(n^2)$). This method is preferable when solution quality is prioritized over runtime.

Algorithm 3 First-Fit Coloring Based on Node Weights with Weight Updates

Require: Graph $G = (V, E)$ with initial weights $w(v)$ for each $v \in V$
Ensure: Color assignment $c(v)$ for each $v \in V$
1: Mark all nodes as uncolored
2: **while** there exists an uncolored node **do**
3: $v \leftarrow \arg\max_{u \in V,\ \text{uncolored}(u)} w(u)$
4: $c(v) \leftarrow$ FINDFIRSTFITCOLOR(v, c)
5: **for** each uncolored neighbor u of v **do**
6: $w(u) \leftarrow w(u) \times 1.1$
7: **end for**
8: **end while**
9: **return** color assignment c

B. Tabu Search

To further refine the coloring, Tabu Search is applied. Preliminary experiments with simulated annealing were abandoned due to slow convergence.

In the proposed Tabu Search technique, a color class is randomly removed and its nodes are reassigned, creating conflicts that the algorithm attempts to resolve iteratively. For each conflicting node, alternative color assignments are evaluated, and the assignment producing the fewest conflicts is selected. Importantly, the algorithm may accept moves that temporarily increase conflicts, enabling escape from local minima. To prevent immediate reversal of moves, the previous color assignment is added to a tabu list for a fixed tenure. The tenure length is set to *n/15* (where *n* is the number of nodes), with a maximum of 50 iterations.

The search runs for up to *5,000*n* iterations, but can terminate earlier if no improvement is observed within a time limit (e.g., 60–120 seconds) or after *500*n* non-improving iterations. If the search stagnates—defined as fewer than five conflicts with no improvement for 500 iterations—another move is triggered. In this step, a conflicting node with color *A* and a neighboring node with color *B* are chosen, and the longest *A–B* chain is identified. By swapping the colors along this chain, the algorithm performs a large-scale restructurion that resolves persistent conflicts and escapes stagnation.

By integrating node-importance heuristic with Tabu Search, our method achieves a balance between scalability and solution quality, making it well-suited for large-scale graph coloring tasks.

EXPERIMENTAL STUDY

A. Experimental Setup

All experimental evaluations were conducted on a MacBook Pro equipped with an Apple M3 processor and 16 GB of RAM. The proposed algorithm was implemented in C++ and compiled using the g++ compiler, which is part of the GNU Compiler Collection (GCC) toolchain, on a UNIX-based operating system (macOS).

TABLE 1: PERFORMANCE OF THE PROPOSED HEURISTIC ON SOME OF THE DIMACS BENCHMARKS

Benchmark Instance	Dstaur		Heuristic		Heuristic with Tabu Search	
	# of colors	Time (s)	# of colors	Time (s)	# of colors	Time (s)
DSJC125.1g.col	6	0.003717	8	0.000368	**5**	1.192270
DSJC125.1gb.col	6	0.002222	8	0.000229	**5**	1.229348
DSJC125.5g.col	22	0.011877	25	0.001167	**17**	4.692831
DSJC125.5gb.col	22	0.011730	25	0.001152	**18**	3.538053
DSJC125.9g.col	51	0.027364	56	0.002089	**44**	2.969758
DSJC125.9gb.col	51	0.027199	56	0.001913	**44**	4.307693
R100_1g.col	6	0.001263	7	0.000165	**5**	0.544724
R100_1gb.col	6	0.001259	7	0.000154	**5**	0.530978
R100_5g.col	18	0.006362	20	0.000719	**14**	2.102948
R100_5gb.col	18	0.006357	20	0.000658	**14**	1.801911
R100_9g.col	41	0.014370	45	0.001226	**35**	2.377639
R100_9gb.col	41	0.014742	45	0.001416	**35**	2.520499
R50_1g.col	4	0.000159	**4**	0.000058	**3**	0.134227
R50_1gb.col	4	0.000154	**4**	0.000051	**3**	0.135649
R50_5g.col	11	0.000891	13	0.000189	**10**	0.194996
R50_5gb.col	11	0.000883	13	0.000187	**10**	0.201136
R50_9g.col	22	0.001905	26	0.000324	**21**	0.296731
R50_9gb.col	22	0.002418	26	0.000326	**21**	0.283571
R75_1g.col	5	0.000505	6	0.000096	**4**	0.296664
R75_1gb.col	5	0.000475	6	0.000103	**4**	0.314175
R75_5g.col	15	0.002835	18	0.000440	**12**	1.037553
R75_5gb.col	15	0.002886	18	0.001526	**13**	0.544684
R75_9g.col	36	0.006405	**36**	0.000712	**33**	0.470988
R75_9gb.col	36	0.006404	**36**	0.000730	**33**	0.461904
myciel5g.col	6	0.000310	**6**	0.000093	**6**	0.046167
myciel5gb.col	6	0.000353	**6**	0.000096	**6**	0.047868
myciel6g.col	7	0.001485	**7**	0.000240	**7**	0.141123
myciel6gb.col	7	0.001529	**7**	0.000282	**7**	0.141122
myciel7g.col	8	0.007042	**8**	0.000763	**8**	0.397006
myciel7gb.col	8	0.007338	**8**	0.000708	**8**	0.429093
queen10_10g.col	15	0.007078	17	0.000521	**11**	1.963415
queen10_10gb.col	15	0.004490	17	0.000512	**11**	3.074879
queen11_11g.col	15	0.007301	17	0.000679	**13**	1.764333
queen11_11gb.col	15	0.006739	17	0.000725	**13**	1.799834
queen12_12g.col	16	0.011140	19	0.000927	**14**	3.149886
queen12_12gb.col	16	0.011125	19	0.000874	**13**	7.232085
queen8_8g.col	11	0.001507	12	0.000269	**9**	0.840050
queen8_8gb.col	11	0.001489	12	0.000272	**9**	0.628741
queen9_9g.col	13	0.002468	15	0.000448	**10**	1.109976
queen9_9gb.col	13	0.002498	15	0.000371	**10**	1.691002

B. Performance Evaluation

We conducted a comparative evaluation of our proposed algorithm against the widely used DSatur heuristic. The evaluation focused on two primary metrics: the number of colors used and the execution time required.

Our experiments were carried out on the DIMACS benchmark suite [11], which contains 175 graph instances. The DIMACS dataset is a synthetic benchmark dataset created for the graph coloring problem, containing graphs with node counts ranging from 30 to 5000. This dataset is commonly used to test the performance of graph coloring algorithms. The purpose of using this dataset is not only to easily validate the accuracy of the results produced by our algorithm, but also, due to its synthetic nature, to test it on graphs with a variety of structures. This enables us to compare our algorithm with state-of-the-art algorithms, such as DSatur. The DIMACS dataset, by offering a wide range of graph structures, serves as a valuable resource for assessing the versatility and performance of our algorithm across different graphs.

The performance of the proposed heuristic, and its combined version with Tabu Search are compared with Dstaur and the results are reported in Table 1. Two metrics are used for the comparison: the number of colors used which should be minimized and the execution time. The proposed heuristic is up to 10x faster than Dsatur while giving reasonable results, whereas the heuristic combined with Tabu Search always outperforms Dsatur in terms of color number and has higher execution time.

We also measure the average performance across the dataset, and report results in Table 2 and visualize them in Fig. 4. The findings demonstrate that our algorithm achieves competitive, and often lower, chromatic numbers compared to DSatur, while also significantly reducing execution time. These results highlight the efficiency and effectiveness of the proposed method in handling structurally varied graph instances.

TABLE 2: Average results on DIMACS dataset

	Dsatur	Heuristic	Heuristic + Tabu
Avg. Number of Colors Used	30	32	26
Avg. Execution Time	2.62s	0.03s	39.99s

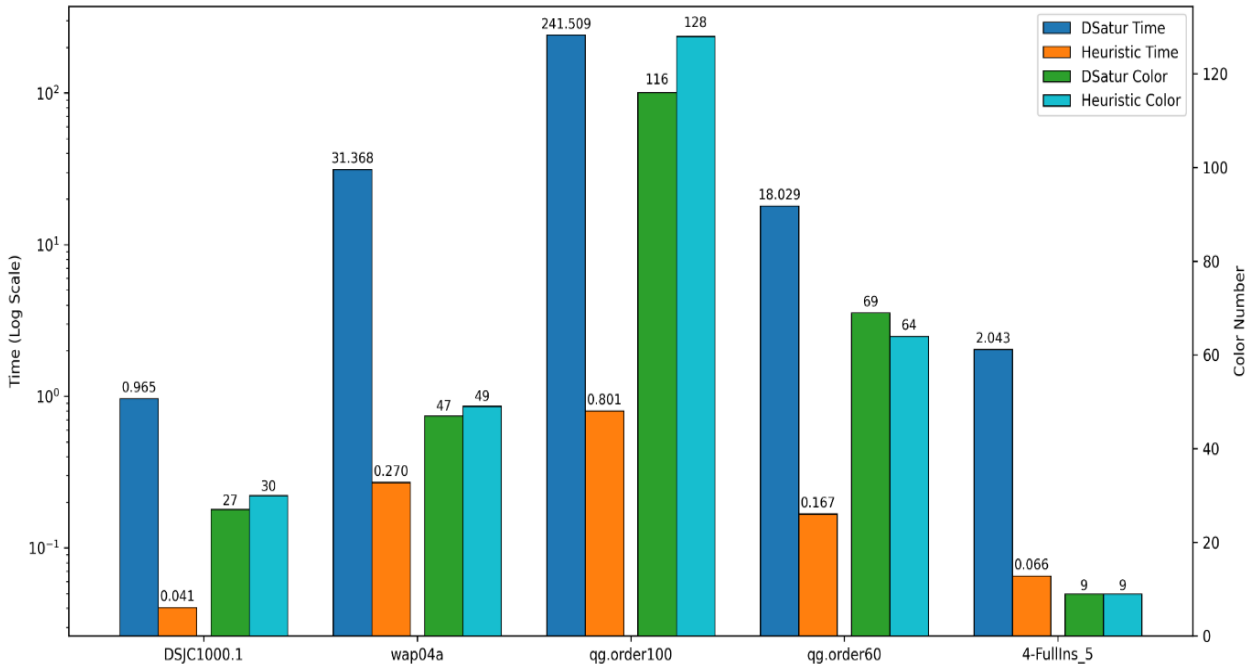


Fig. 4. Performance comparison on selected DIMACS instances between DSatur and Heuristic

We also assessed the performance of the complete pipeline, including the Tabu Search optimization step, comparing it directly to DSatur. This comparison provides insights into how the Tabu Search component enhances the initial heuristic-based coloring. Results are presented in Fig. 5 below.

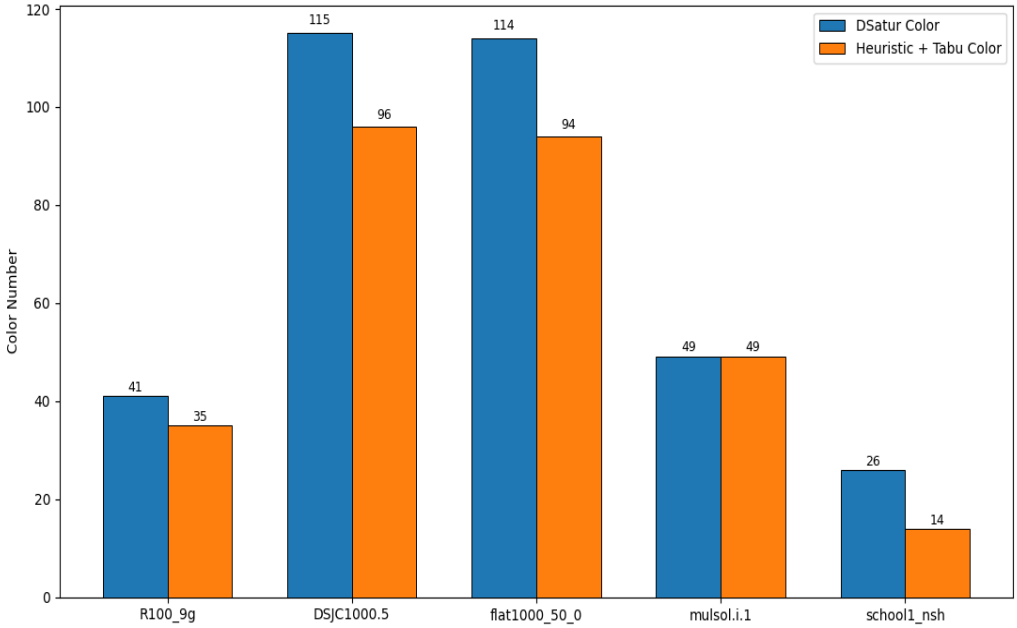


Fig. 5. Coloring quality comparison on selected DIMACS instances between DSatur and Heuristic + Tabu Search

DIMACS Benchmark Suite contains 175 instances. When we compare the performance of our algorithm (including Tabu Search) with Dsatur, the proposed algorithm obtains better, optimal, equal and worse results in 82, 22, 76 and 1 graphs, respectively. The results reveal that, on the DIMACS dataset, our heuristic algorithm produces colorings that are, on average, one to two colors less optimal than those generated by DSatur. However, this slight compromise in color quality is offset by a significant improvement in execution time. By incorporating the Tabu Search method, we are able to close the color quality gap, albeit at the cost of increased computation time.

CONCLUSIONS AND FUTURE WORK

In this study, we presented an efficient and scalable algorithm for the graph coloring problem, specifically designed for large-scale graphs. The proposed method combines a lightweight heuristic for rapid initial coloring with a Tabu Search–based optimization phase to refine solutions. Comparative experiments on the DIMACS benchmark demonstrated that our approach consistently achieves superior runtime performance while maintaining coloring quality competitive with the state-of-the-art DSatur algorithm. Furthermore, the adaptability of the Tabu Search parameters allows the method to be tuned to specific runtime constraints and performance requirements, making it suitable for a variety of application domains.

Despite these promising results, several avenues remain for further research. First, our algorithm shows limitations when applied to extremely large graphs. To address this challenge, we plan to investigate graph partitioning strategies, where large graphs are decomposed into smaller subgraphs, colored in parallel, and then merged. Any conflicts introduced during merging would be handled using localized optimization methods. Second, a parallelized implementation, potentially leveraging GPU acceleration and bitwise representations, is envisioned to reduce both runtime and memory overhead. Finally, preprocessing techniques—such as temporarily removing sparsely connected nodes prior to coloring—could be explored to further enhance scalability and efficiency.

Together, these directions offer a path toward extending the applicability of our approach to even larger and more complex graph coloring problems encountered in real-world scenarios.

REFERENCES

[1] T. R. Jensen and B. Toft, *Graph Coloring Problems*. New York, NY, USA: Wiley, 1995.

[2] R. M. R. Lewis, *A Guide to Graph Colouring*, vol. 7. Berlin, Germany: Springer, 2015, doi: 10.1007/978-3-319-25730-3.

[3] R. W. Quong and S.-C. Chen, ``Register allocation via weighted graph coloring (technical summary)," Purdue Univ. Lib., West Lafayette, IN, USA, ECE Tech. Rep. TR-EE 93-23 (232), Jun. 1993.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: Freeman, 1990.

[5] I. Blöchliger and N. Zufferey, ``A graph coloring heuristic using partial solutions and a reactive tabu scheme," *Comput. Oper. Res.*, vol. 35, no. 3, pp. 960_975, Mar. 2008.

[6] P. Galinier and A. Hertz, ``A survey of local search methods for graph coloring," *Comput. Oper. Res.*, vol. 33, no. 9, pp. 2547_2562, Sep. 2006.

[7] Z. Lü and J.-K. Hao, ``A memetic algorithm for graph coloring," *Eur. J. Oper. Res.*, vol. 203, no. 1, pp. 241_250, May 2010.

[8] L. Moalic and A. Gondran, ``The new memetic algorithm HEAD for graph coloring: An easy way for managing diversity," in *Proc. Eur. Conf. Evol. Comput. Combinat. Optim.*, Copenhagen, Denmark, Apr. 2015, pp. 173_183, doi: 10.1007/978-3-319-16468-7_15.

[9] Y. Zhou, B. Duval, and J.-K. Hao, ``Improving probability learning based local search for graph coloring," *Appl. Soft Comput.*, vol. 65, pp. 542_553, Apr. 2018, doi: 10.1016/j.asoc.2018.01.027.USA, ECE Tech. Rep. TR-EE 93-23 (232), Jun. 1993.

[10] D. Brelaz, "New methods to color the vertices of a graph," *Commun. ACM*, vol. 22, no. 4, pp. 251–256, 1979.

[11] D. S. Johnson and M. A. Trick, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*. Providence, RI, USA: American Mathematical Society, 1993.

AI Stethoscope Revolutionizing Home Healthcare using Machine Learning

N. Hareesh[1], K. Bhargav[2], Dr. P. Sukanya[3], S. Srujan Chowdary[4]

[1] *Siddhartha Academy of Higher Education Deemed to be University, formerly VR Siddhartha Engineering College, Vijayawada, India, hareeshnallagatla@gmail.com*

[2] *Siddhartha Academy of Higher Education Deemed to be University, formerly VR Siddhartha Engineering College, Vijayawada, India, bhargavkodali123@gmail.com*

[3] *Siddhartha Academy of Higher Education Deemed to be University, formerly VR Siddhartha Engineering College, Vijayawada, India, sukanya@vrsiddhartha.ac.in*

[4] *Siddhartha Academy of Higher Education Deemed to be University, formerly VR Siddhartha Engineering College, Vijayawada, India, srujanchowdarysaggurti@gmail.com*

ABSTRACT

Sounds produced by heart and blood in arteries and veins could be heard by the ability of a Traditional stethoscope. In modern times, people are suffering more problems regarding heart diseases and at that particular moment efficient doctors may not be available or due to lack of proper attention. In order to overcome these difficulties an automatic diagnosis using the device is developed that is AI stethoscope. This device is enhanced with high quality sensors and some microphone to capture heart sounds. These audio signals are digitized and analyzed using machine learning algorithm like yamnet to detect potential health issues. In additional to stethoscope functionalities this device has temperature sensor to monitor body temperature, blood pressure sensor to measure systolic pressure and diastolic pressure and pulse oximeter to determine blood oxygen level. These sensors work together to provide a clarity about the user's health. Data from all the sensors re processed and analyzed by the raspberry Pi microprocessor enabling accurate and automatic diagnosis of various heart conditions. This AI stethoscope addresses the issues of the limited availability of doctors and insufficient attention to health by providing a reliable, efficient solution to monitor heart conditions.

Keywords: *AI stethoscope, health monitoring, Raspberry Pi, sensor integration*

INTRODUCTION

In the era of modern healthcare, technology plays a impor- tant role in addressing critical challenges such as accessibility, timely diagnosis and efficiency. Traditional stethoscope has been a cornerstone old medical diagnostics, aiding healthcare professionals in analyzing heart and lung. However, their reliance on manual expertise and limited scope of functionality often proves inadequate in meeting the growing demand for proactive health monitoring. With increasing incidences of health issues and limited availability of well qualified doctors, there is a need of automated diagnostic tools.

The AI stethoscope project introduces an innovative solution by enhancing the traditional stethoscope with advanced sensor technology and machine learning capabilities. This device equipped with high quality boya microphone and specialized sensors, the device captures heart and lungs sounds with remarkable accuracy. These audio signals are digitalized and processed using machine learning algorithm to detect potential abnormalities.

Beyond audio bases diagnostics, the AI stethoscope integrates additional health monitoring functionalities to offer a holistic view of the user's health. Features such as temperature sensors for body heat measurement, a blood pressure sensor to record the systolic and diastolic levels and a pulse oximeter for mon- itoring blood oxygen levels make the device a comprehensive health monitoring tool. These sensors work in unison with data processed by a Raspberry Pi microprocessor to deliver actionable insights into the user's health.

This innovative approach bridges the gap between conven- tional tools and modern technologies advancements. By leveraging machine learning and automation the AI stethoscope not only ensures efficient and reliable diagnosis but also enhances accessibility for individual in remote or underserved regions. This solution can be particularly valuable addressing health crisis where immediate and accurate diagnosis can make a critical difference.

A. Problem Statement

Many people face health problems and sometimes there aren't enough doctors available to give them the attention they need. Traditional stethoscopes can only be used by trained professionals to listen to heart, lung, and other body sounds. To solve this issue, we need a smart device that can automatically diagnose health conditions by using a stethoscope combined with advanced technology. This device should also include sensors to measure temperature, blood pressure, and blood oxygen levels, providing a complete health monitoring solution that anyone can use at home.

B. Motivation

Cardiovascular disease are among the leading causes of mortality worldwide, and early detection is crucial to prevent severe outcomes, However, in many parts of the world, access to specialized diagnostic tools and healthcare professionals is limited. This project seeks to bridge that gap by leveraging machine learning and IoT technologies to create a system that can empower individuals and healthcare providers to detect heart conditions in real time.

C. Work Contribution

The important contributions of this work are as follows:

- Development of an AI-powered stethoscope system using sensor based and audio acquisition technology.
- Integration of IoT (Raspberry Pi + Viam Server) for real time heart sound processing and health analysis.
- Implementation of a mobile application for user friendly interaction and result visualization.
- Utilization of transfer learning (YAMNet) for efficient and accurate classification of heart sound data.
- Real time diagnosis capability to assist in early detection of cardiac conditions, especially in remote and under- served areas.

D. Paper Structure

The rest of the paper is summarized as follows Section 2 presents a "Literature Survey" that discusses earlier research on automation in sports and cricket decision systems. Section 3 illustrates the "Proposed Methodology" along with architectural and component details. Section 4 discusses "Results and Evaluation," including system accuracy and efficiency. Finally, Section 5 concludes the paper with "Conclusion and future works".

LITERATURE SURVEY

Ogawa et al. [1] designed the Super Stethoscope, a novel AI-powered auscultation device combining ECG and high- fidelity heart sound acquisition, capable of capturing inaudible frequencies and producing real-time visualizations. Their sys- tem supports advanced signal analysis and AI-based diagnosis of valvular heart diseases and heart failure, while also enhancing remote telemedicine and clinical education through spectrogram assisted auscultation. A survey of over 3600 physicians confirmed strong clinical interest in AI-assisted auscultation tools for early diagnosis and decision-making.

Omarov et al. [2] introduced a real-time electronic stetho- scope system that digitized and classified phonocardiographic (PCG) signals using machine learning, achieving over 93% accuracy in identifying abnormal heart sounds.

Chang et al. [3] utilized a Random Forest classifier on UCI heart disease datasets, achieving 83% accuracy, and underlined the importance of exploratory data analysis and visualizations for improving classification performance.

Saboor et al. [4] proposed a robust machine learning system incorporating SVM, Random Forest, and XGBoost, achieving a high accuracy of 96.72% by emphasizing data normalization and hyperparameter tuning.

Ghanayim et al. [5] developed an AI-enabled electronic stethoscope capable of detecting aortic stenosis using in- frasound and supervised learning algorithms. Their device achieved a sensitivity of 93% for severe cases and showed promise for real-time, point-of-care screening without depen- dence on clinical expertise.

Jindal et al. [6] experimented with multiple classifiers, including KNN, Logistic Regression, and Random Forest, attaining a maximum accuracy of 88.5% for heart disease prediction based on clinical parameters.

Earlier, Nashif et al. [7] developed a cloud integrated cardiovascular monitoring system using sensors and SVM, achieving 97.53% accuracy and demonstrating the potential of IoT-based healthcare solutions.

PROPROSED METHODOLOGY

In this paper, we aim to develop a user-friendly application for analyse heart using audio recordings collected through a AI stethoscope device. The methodology is designed to encompass the workflow from the data to health diagnostics ensuring a robust and efficient system. The process begins with capturing audio signals from the user's heart using AI stetho- scope device integrated with raspberry Pi. These recordings are processed by machine learning model. The training model is deployed on the raspberry pi which processes the audio in real time and sends results to the android application via bluetooth / wifi. This mobile app serves as the user interface. In this methodology we use of real time communication between device to deliver a comprehensive solution for health.

A. Datasets

Finding a high-quality dataset covering various heart condi- tions can be challenging. To address this, we will use a pub- licly available dataset that includes data for four common heart diseases: Aortic Stenosis (AS), Mitral Regurgitation (MR), Mitral Stenosis (MS), and Mitral Valve Prolapse (MVP), along with data for normal heart conditions. The dataset is organized into five distinct categories, with each category containing 200 audio recordings. These recordings are in .wav format and have a duration of approximately four to five seconds.

B. Data Preprocessing

When working with a limited dataset, training a transfer learning model is often the best approach. For this project, we use YAMNet as the base model and build our custom model on top of it. This allows us to achieve better performance while simplifying the training process. Before training, we need to preprocess the data to meet YAMNet's requirements. YAMNet is an audio event classifier that takes audio waveforms as input and predicts audio events based on the AudioSet ontology. Since our dataset has a sample rate of 8 kHz and consists of mono audio files, we first need to resample and preprocess the audio files to ensure compatibility with YAMNet's specifica- tions.

C. Control Flow Diagram

The provided control flow diagram outlines the sequential process of analyzing heart sounds using an AI stethoscope system integrated with machine learning. This flow visually represents how heart sounds are captured, processed, analyzed, and communicated to the end user in a clear, logical sequence. It highlights the seamless integration of IoT, machine learning, and mobile communication for efficient and accurate health- care diagnostics. The control flow diagram

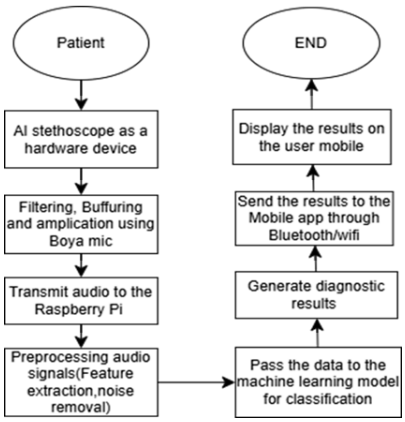


Fig. 1. *Control flow diagram*

D. Model Description

Transfer learning is a powerful machine learning technique where a model trained for one task is adapted to solve a different but related task. Instead of starting from scratch, transfer learning allows us to leverage the knowledge a model has already gained to tackle a new problem. In this case, we use a pre-trained model like YAMNet, which is specifically designed to work with sound data. YAMNet has already been trained on a diverse range of datasets to extract meaningful features from audio, such as pitch, frequency, and other sound characteristics. These features are highly relevant for identify- ing heart sounds associated with various cardiac conditions. By using transfer learning, we can achieve faster and more efficient training because the model starts with pre-trained features that are already optimized for related tasks. YAM- Net's broad training on various sound events also makes it adaptable to stethoscope recordings, ensuring it can generalize well across different heart conditions and patient variations.

This ability to generalize is key to accurately detecting heart conditions using stethoscope sounds.

E. Experiment Setup

In this project we have used many hardware and software related tools to achieve the good results.

1. Hardware:

- Raspberry Pi zero which is used as the central processing unit to analyze incoming heart sounds.

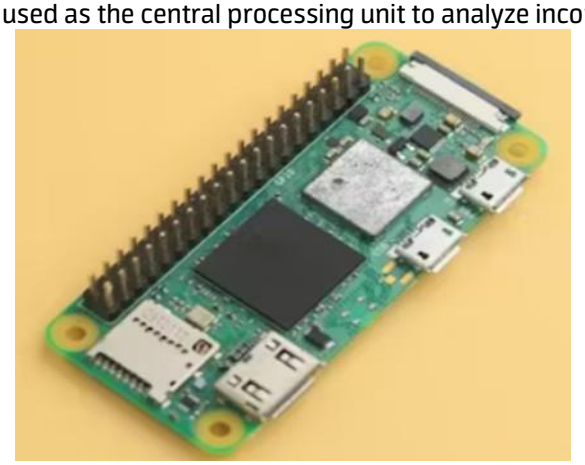


Fig. 2. *Raspberry Pi*

- Traditional stethoscope which used to capturing heart sounds from the patient.

Fig. 3. *Traditional Stethoscope*

- Boya microphone which is used to record heart sounds with high fidelity and acts as amplifier also.
- Lipo battery 3.7v which is used to power up the Rasp- berry Pi and associated hardware.
- Sound card gets the sounds from the microphone and send to the Raspberry Pi
- DC-DC charge/discharge module used as a voltage boost converter, wherein the Raspberry Pi works on 5v and Lipo batteries can only give out a maximum of 4.2v when fully charged.

2. *Software:*

- Python programming language was used for data prepro- cessing, analysis, and running machine learning models.

Fig. 4. *Boya Microphone*

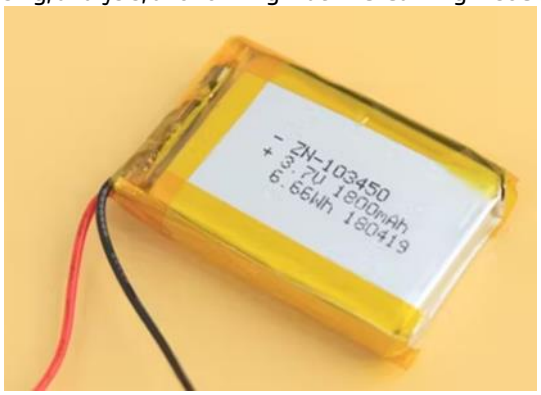


Fig. 5. *3.7v Lipo Battery*

- TensorFlow/Keras frameworks were used for developing and running the machine learning model saved in .h5 format.
- Android Studio was used for developing the mobile application interface.
- Viam server was used for connecting and transmitting data from Raspberry Pi to the mobile app.
- First, we need to get started with setting up the Viam server. We just start off with the creation of an intelligent machine

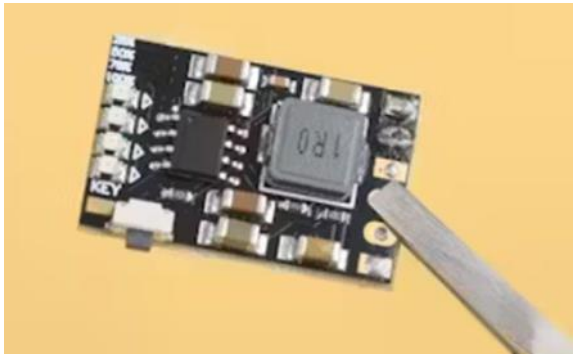


Fig. 6. *Sound Card*

Fig. 7. *DC-DC charge/discharge module*

by add machine and provide a name to it. After this step we have setup the viam server on the Raspberry Pi and then we have to start the viam server make sure your system must be in live. We have to add TensorFlow CPU service to your machine which is used as machine learning model service and also add mqtt pub sub service which is used to communicate with mobile app wirelessly.

Model Training and Testing

We carried out extensive model training on Google Colab, using TensorFlow as the primary framework. All the essential steps and details involved in the training process are thor- oughly documented within this Jupyter notebook, making it easy to follow and understand.

Model testing is a crucial step in the development process, where we evaluate how well the trained model performs on new, unseen data. This helps us measure its ability to generalize beyond the training dataset. In our case, after com- pleting the training process, we tested the model to assess its performance. The results showed a training accuracy of 94% and a testing accuracy of 93%. These numbers indicate that the model not only performs exceptionally well on the data it was trained on but also handles unfamiliar data effectively. This strong generalization capability means the model is reliable for making accurate predictions in real-world scenarios.

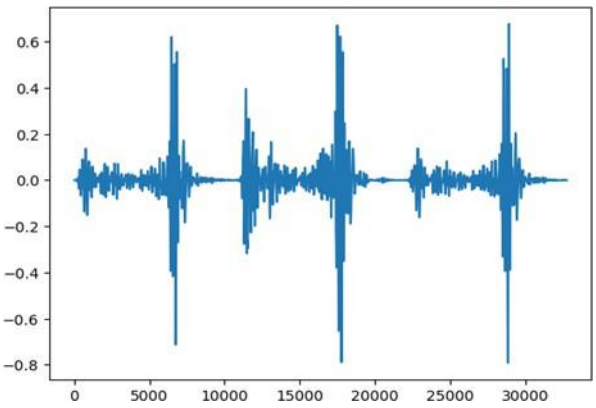


Fig. 8. Testing the heart sound waveform data from patient before removing noise

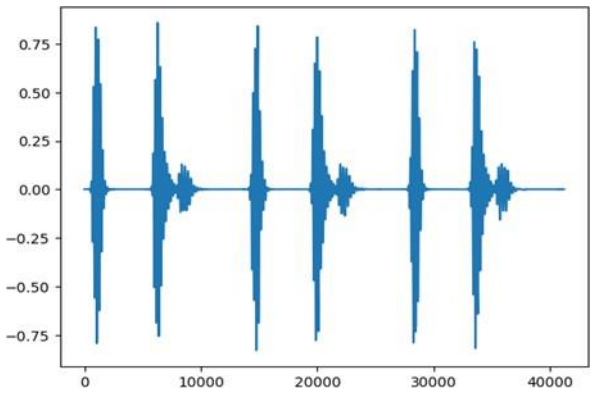


Fig. 9. Testing the heart sound waveform from patient after removing noise

A. Deployment

To deploy a TensorFlow model, the *SavedModel* format is typically used. This involves saving the trained model using tf.saved_model.save() and later loading it with tf.saved_model.load(). Once the model is saved, the next step is to select the appropriate deployment option based on the target platform.

For server-side deployment, TensorFlow Serving is com- monly used. For web-based applications, TensorFlow.js is the preferred choice. Since we are working with a mobile application, we use TensorFlow Lite, which is specifically designed to optimize models for mobile devices, ensuring efficiency and smooth performance.

B. Limitations

- Audio data quality affects the accuracy of sound clas- sification. Background noise, improper placement of the stethoscope, or low-quality microphones can negatively impact results.
- The dataset used for training might not include diverse demographics such as varying age groups and health conditions, potentially affecting the model's performance.
- Usage of the system on mobile or embedded devices may result in high energy consumption, impacting the battery life of the connected device.
- Legal and regulatory compliance with medical standards can vary across countries and regions, posing challenges for global deployment.

RESULTS AND ANALYSIS

When the system gets the input from the user then the heart sounds data captured through the device and send it to the Raspberry Pi and it processes the sounds and output is displayed on the mobile application. When the user clicks on the analyze button then it takes heart sound as a input and Raspberry Pi processes the heart sound and displayed corresponding result.

Accuracy of the heart sound classification system refers to how effective the model identifies and categorize heart conditions based on the input of the heart sound data.

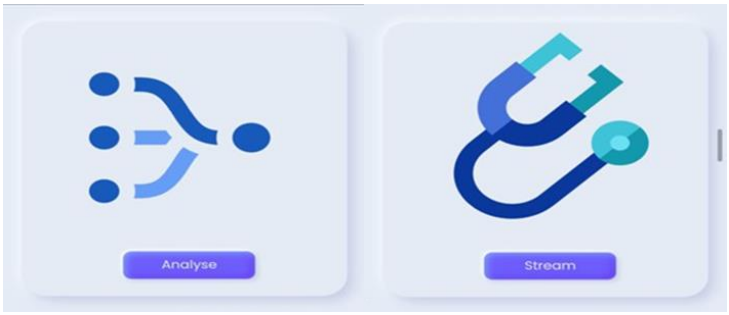


Fig. 10. Screenshots of the mobile application: (a) Application Services, (b)Result Displayed

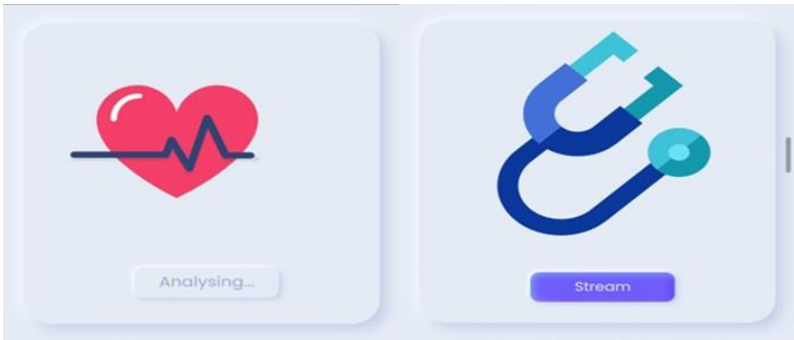


Fig. 11. (a) Training and Validation Accuracy

Fig. 12. Screenshots of the mobile application: (a) Application Services, (b) Result Displayed

TABLE I. CLASSIFICATION REPORT OF THE HEART SOUND CLASSIFIER

Class	Precision	Recall	F1-score	Support
AS	0.85	0.92	0.88	204
MR	0.82	0.87	0.84	173
MS	0.92	0.65	0.76	184
MVP	0.83	0.82	0.82	190
N	0.84	0.98	0.90	179
Accuracy	—	—	0.85	930
Macro Avg	0.85	0.85	0.84	930
Weighted Avg	0.85	0.85	0.84	930

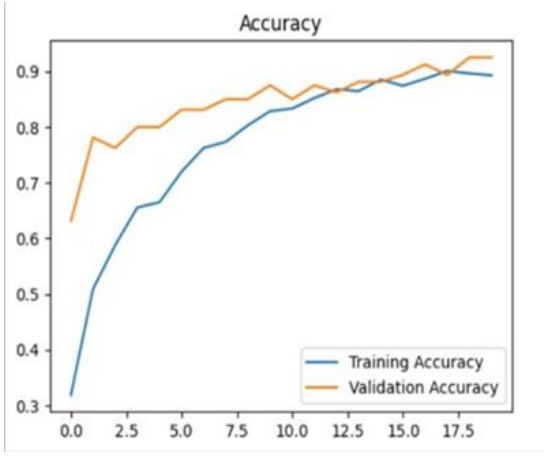


Fig. 13. (a) Training and Validation Accuracy

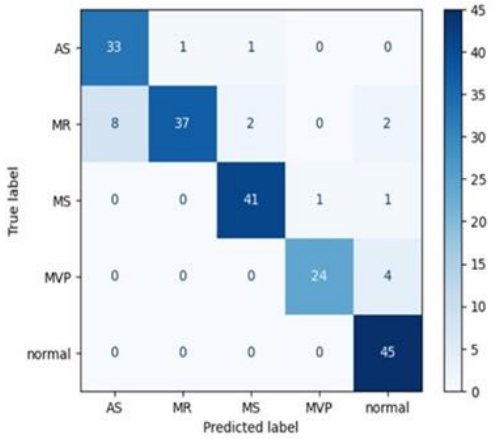


Fig. 14. (b) Confusion Matrix

CONCLUSION AND FUTURE WORK

In conclusion, the AI stethoscope developed in this study demonstrates a promising approach to enhancing home health-care by combining machine learning with advanced sensor technologies. The system effectively captures and analyzes heart sounds using a Raspberry Pi-based setup and YAMNet model, achieving high accuracy in detecting common heart conditions. With additional sensors for measuring temperature, blood pressure, and oxygen saturation, it offers a comprehen- sive view of a user's health and provides results through a user- friendly mobile application. Despite its success, there are opportunities for future improvement. Enhancing noise filtering, expanding the dataset to include more diverse demographics, and optimizing the model for better edge performance can sig- nificantly increase the system's reliability and generalization. Further developments may include integrating cloud connec- tivity for remote monitoring, adding support for detecting lung diseases, and conducting clinical validations to meet medical standards. These

advancements will help transform the AI stethoscope into a reliable, accessible, and scalable tool for early diagnosis and continuous health monitoring in both urban and remote settings.

REFERENCES

[1] S. Ogawa, F. Namino, T. Mori, G. Sato, T. Yamakawa, and S. Saito, "AI diagnosis of heart sounds differentiated with Super StethoScope," *Journal of Cardiology*, vol. 83, pp. 265–271, 2024.

[2] B. Omarov, N. Saparkhojayev, S. Shekerbekova, et al., "Artificial Intelligence in Medicine: Real-Time Electronic Stethoscope for Heart Diseases Detection," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2815–2833, 2022.

[3] V. Chang, V. R. Bhavani, A. Q. Xu, and M. A. Hossain, "An Artificial Intelligence Model for Heart Disease Detection Using Machine Learning Algorithms," *Healthcare Analytics*, vol. 2, p. 100016, 2022.

[4] A. Saboor, M. Usman, S. Ali, et al., "A Method for Improving Prediction of Human Heart Disease Using Machine Learning Algorithms," *Mobile Information Systems*, vol. 2022, Article ID 1410169.

[5] T. Ghanayim, L. Lupu, S. Naveh, N. Bachner-Hinenzon, D. Adler, S. Adawi, S. Banai, and A. Shiran, "Artificial Intelligence-Based Stethoscope for the Diagnosis of Aortic Stenosis," *The American Journal of Medicine*, vol. 135, no. 9, pp. 1124–1133, 2022.

[6] H. Jindal, D. Singla, M. Juneja, and M. Singla, "Heart Disease Prediction Using Machine Learning Algorithms," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1022, no. 1, p. 012072, 2021.

[7] S. Nashif, A. Almogren, A. Alabdulatif, et al., "Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System," *World Journal of Engineering and Technol- ogy*, vol. 6, no. 4, pp. 854–873, 2018.

[8] A. Singh and R. Kumar, "Heart disease prediction using machine learn- ing algorithms," in *Proceedings of the 2020 International Conference on Electrical and Electronics Engineering (ICE3)*, pp. 452–457, IEEE, Feb. 2020.

[9] O. E. Taylor, P. S. Ezekiel, and F. B. Deedam-Okuchaba, "A model to detect heart disease using machine learning algorithm," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 11, pp. 1–5, 2019.

[10] S. Swaminathan, S. M. Krishnamurthy, C. Gudada, S. K. Mallappa, and N. Ail, "Heart sound analysis with machine learning using audio features for detecting heart diseases," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 16, no. 2, pp. 17–17, 2024.

[11] M. E. Chowdhury, A. Khandakar, K. Alzoubi, S. Mansoor, A. M. Tahir, M. B. I. Reaz, and N. Al-Emadi, "Real-time smart-digital stethoscope system for heart diseases monitoring," *Sensors*, vol. 19, no. 12, p. 2781, 2019.

Performance Evaluation of IIR System Modeling with the Backtracking Search Optimization Algorithm

Serdar Koçkanat[1]

[1] Department of Electrical and Electronics Engineering, Sivas Cumhuriyet University, Sivas, Türkiye, skockanat@cumhuriyet.edu.tr

ABSTRACT

Infinite impulse response (IIR) filters are fundamental in digital signal processing for attenuating unwanted frequencies and enhancing desired signal components. Although their recursive structure can lead to instability and nonlinear phase responses, IIR filters provide fast response times and high performance with lower-order designs. Recent advances in metaheuristic optimization algorithms have facilitated efficient coefficient tuning, enabling application-specific filter designs. In this study, an IIR filter was employed to model a dynamic digital system, with its coefficients optimized using the Backtracking Search Optimization Algorithm (BSA). The performance of BSA was evaluated by comparing input-output signals, error signals, and amplitude and phase responses of the designed system. Furthermore, the statistical results have been compared with those reported in the literature.

Keywords: *BSA, digital filter, IIR, optimization, system identification.*

INTRODUCTION

Digital filters are fundamental components in digital signal processing, designed to suppress unwanted frequency components or emphasize desired ones. They can be both mathematically modeled and implemented in hardware and software [1].

Digital filters are classified into two categories: finite impulse response (FIR) and infinite impulse response (IIR) filters. FIR filters produce outputs of finite duration that decay to zero shortly after the input signal ends. Because all their poles lie within the unit circle, FIR filters are inherently stable. In contrast, IIR filters can theoretically produce outputs indefinitely, even after the input signal has ceased. However, the stability of IIR filters depends on whether their poles remain inside the unit circle; if any poles lie outside, the system becomes unstable. From a design perspective, FIR filters are non-recursive, whereas IIR filters are recursive structures [2].

FIR filters offer advantages such as inherent stability and a linear phase response due to their non-recursive structure. However, high-order designs increase computational complexity, which is a significant drawback. In contrast, IIR filters may suffer from potential instability and exhibit a non-linear phase response because of their recursive nature. Nevertheless, they provide faster response times and can achieve performance comparable to FIR filters with lower-order designs, representing a key advantage [3–5].

In recent years, metaheuristic optimization algorithms inspired by the foraging and survival behaviors of living organisms have gained popularity and have been effectively applied to solve engineering problems across various fields, including construction, semiconductors, electrical and electronics, mechanical engineering, and others, yielding successful results [6]. These algorithms offer researchers solution opportunities in diverse search spaces, such as continuous, discrete, sequential, and problem-specific domains.

In digital signal processing, optimization algorithms have significantly advanced the design of various systems. Specifically, in both adaptive and non-adaptive FIR and IIR filter designs, coefficient optimization is achieved using these algorithms, enabling application- or signal-specific solutions that differ from classical designs [7–8].

In this study, an IIR filter design was utilized to model a dynamic digital system, with the filter coefficients determined using the Backtracking Search Optimization Algorithm (BSA) as proposed in the literature. The study evaluated the optimization performance of the system's coefficients. The input-output signals, error signals, and the amplitude and phase responses of both the real and designed systems were compared. Additionally, the statistical results of the BSA have been compared with those reported in the literature.

BACKTRACKING SEARCH OPTIMIZATION ALGORITHM

The Backtracking Search Optimization Algorithm (BSA) is a population-based metaheuristic introduced by Pınar Civicioğlu in 2013 [9]. Inspired by the natural concept of backtracking in search processes, BSA aims to balance exploration and exploitation in complex search spaces, making it effective for continuous, discrete, and combinatorial optimization problems. The algorithm operates through initialization, mutation, selection, and backtracking phases, enabling it to avoid local optima and efficiently converge toward global solutions. BSA is characterized by its simplicity, requiring only a single control parameter, and its robustness, exhibiting low sensitivity to initial conditions. The pseudo-code of BSA is given as follows.

Step 1. Initialization

Generate an initial population $P = \{P_i\}$ of N candidate solutions in a D-dimensional search space:

$$P_{i,j} \sim U(low_j, up_j),\ i=1,\ldots,N,\ j=1,\ldots,D \quad (1)$$

where U denotes uniform distribution within the lower and upper bounds of each dimension.

Define a historical population *oldP* (memory).

Step 2. Selection-I (Historical Population Update)

With a small probability, replace *oldP* with the current population P:

$$oldP \leftarrow P \text{ if } rand < \beta,\ \beta \sim U(0,1) \quad (2)$$

Randomly shuffle the order of individuals in *oldP*.

Step 3. Mutation

Compute the mutant population using the difference between historical and current populations:

$$Mutant = P + F * (oldP - P) \quad (3)$$

where F is a scaling factor drawn from a normal distribution.

Step 4. Crossover

Construct a trial population T by mixing elements from P and *Mutant* based on a binary map:

$$T_{i,j} = \begin{cases} Mutant_{i,j}, if\ map_{i,j} = 1 \\ P_{i,j}, otherwise \end{cases} \quad (4)$$

Two strategies to create the binary map:

1. Selecting a subset of dimensions (controlled by a mixing parameter).

2. Allowing mutation in only a single random dimension.

If any element of T violates search boundaries, repair it by random re-initialization within allowed limits.

Step 5. Selection-II (Greedy Replacement)

Evaluate the fitness of each trial solution.

Update the population:

$$P_{i,j} = \begin{cases} T_i, if\ f(T_i) < f(P_i) \\ P_i, otherwise \end{cases} \quad (5)$$

Keep track of the best solution (global minimizer).

Step 6. Stopping Criterion

Repeat Steps 2–5 until a predefined stopping condition is satisfied (e.g., maximum iterations or function evaluations).

Return the best solution found.

IIR System Identification Problem

An IIR digital filter can be formulated as:

$$y(n) = \sum_{m=0}^{M} \left(b_m x(n-m)\right) - \sum_{k=1}^{K} a_k y(n-k) \quad (6)$$

where b_m and a_k are IIR filter coefficients. $x(n)$ and $y(n)$ are input and output signals, respectively.

Fig. 1 shows a schematic diagram illustrating the use of an IIR filter for modeling a dynamic system. In this diagram, a randomly distributed digital signal is applied to the input. This input signal serves as a common input for both the dynamic system and the system to be designed. An error signal is obtained using the signals taken from the outputs of the systems. This error signal is then used to run the BSA, which optimizes the coefficients of the IIR digital filter. As a result of this modeling process, the parameters of any dynamic or unknown system can be determined.

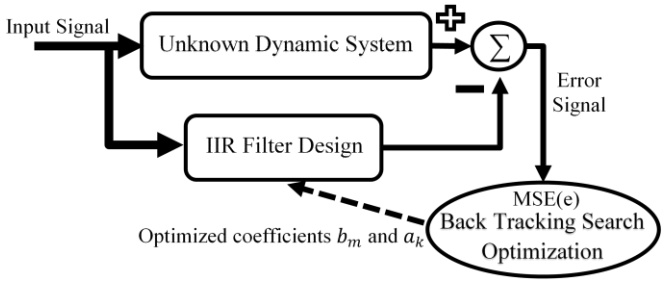


Fig. 1. Unknown Dynamic System Modeling using IIR filter and BSA.

RESULTS AND DISCUSSIONS

In this study, a fourth-order IIR filter design problem, frequently discussed in the literature, was examined to analyze the estimation performance of the BSA in dynamic system modeling using an IIR filter [10]. The transfer function of the fourth-order IIR filter is presented as follows.

$$H(z^{-1}) = \frac{1-0.9z^{-1}+0.81z^{-2}-0.729z^{-3}}{1+0.04z^{-1}+0.2275z^{-2}-0.2101z^{-3}+0.14z^{-4}} \quad (7)$$

In the problem-solving process, the population size was set to 50, and the BSA-based approach was executed 30 times, each with different initial values. Additionally, a uniformly distributed random signal ranging from -0.5 to 0.5 was used as the input signal. This uniformly distributed input signal is shown in Fig. 2. The results obtained were statistically analyzed, and the mean, best, worst, and standard deviation of the error values were calculated.

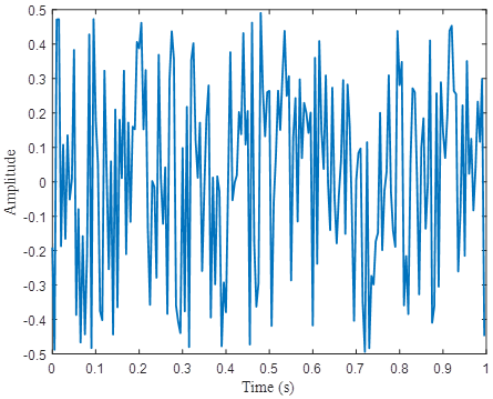


Fig. 2. Uniformly distiributed input signal.

For fourth-order IIR system modeling, the statistical results obtained using the BSA-based approach were compared with those reported in the literature for the modified improved hunger games search (Imp-HGS) algorithm, harmony search (HS) algorithm, dandelion optimizer (mDO), improved artificial rabbits optimization (IARO) algorithm, atom search optimization (ASO) and dynamic opposite learning enhanced artificial ecosystem optimizer (DAEO) [10]. These results are presented in Table 1. Upon examining Table 1, the best value for the BSA approach is 9.5227e-33, the worst value is 1.4783e-27, the average value is 8.1737e-29, and the standard deviation is 2.7272e-28. All these values are significantly smaller than those reported in the literature. This indicates that the BSA approach is highly effective and robust in terms of system modeling performance across multiple runs.

TABLE 1. STATISTICAL COMPARISON RESULTS FOR FOURTH-ORDER IIR FILTER

Algorithms	Best	Worst	Average	Standard deviation
mDO	1.7279e-29	6.8214e-26	6.8977e-27	1.4330e-26
HS	1.9175e-08	1.8365e-07	1.1898e-07	6.7361e-08
IARO	8.4184e-15	2.5921e-11	6.7930e-12	8.2847e-12
Imp-HGS	2.06e-22	1.14e-11	4.52e-13	2.09e-12
DAEO	2.17e-11	9.91e-04	7.91e-05	2.12e-04
ASO	6.6732e-25	2.4516e-22	7.3066e-23	6.7699e-23
BSA	**9.5227e-33**	**1.4783e-27**	**8.1737e-29**	**2.7272e-28**

Fig. 3 shows the real output signal generated by the 4th-order IIR filter and the output signal of the designed IIR filter whose coefficients were estimated using the BSA algorithm. This figure demonstrates that the amplitude matching estimation performs satisfactorily. Furthermore, the low error signal indicates that the modeled IIR system closely matches the real system.

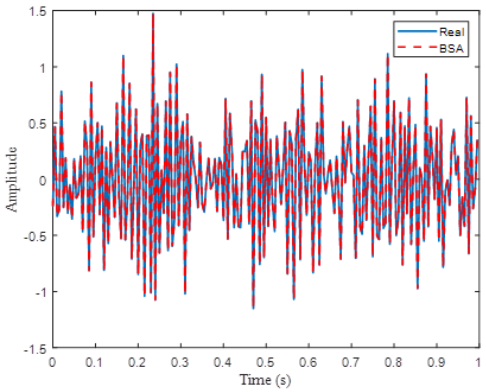


Fig. 3. Comparison of real and BSA-estimated output signals.

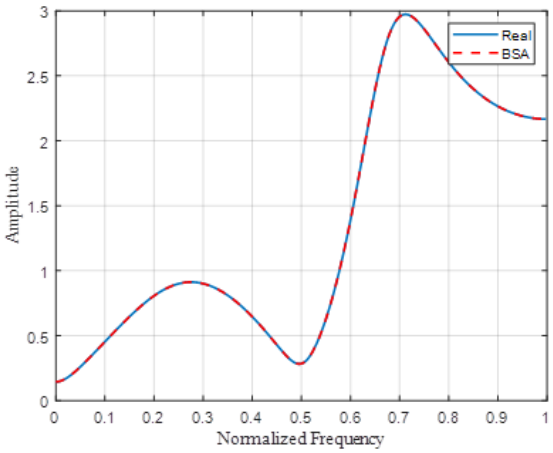


Fig. 4. Amplitude responses of the real and BSA-modeled systems.

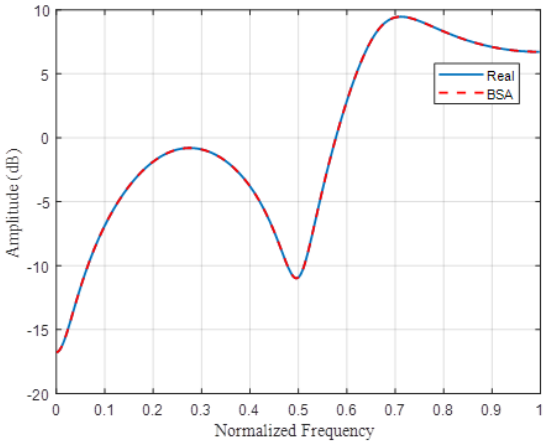


Fig. 5. Power spectrums of the real and BSA-modeled systems.

Fig. 4 and Fig. 5 show the amplitude responses and power spectrums of the 4th-order IIR filter reported in the literature and the IIR system modeled using the BSA algorithm.

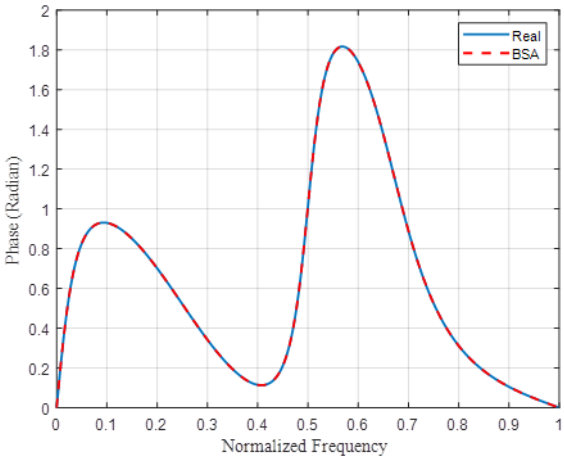


Fig. 6. Phase responses of the real and BSA-modeled systems (Radians).

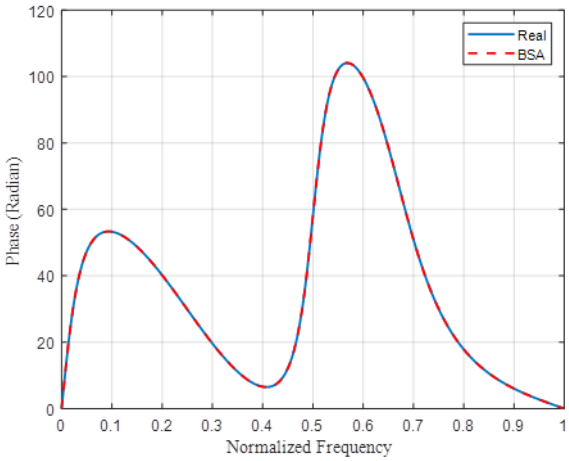


Fig. 7. Phase responses of the real and BSA-modeled systems (Degrees).

Fig. 6 and Fig. 7 display the phase responses of the two compared systems in radians and degrees, respectively. The close agreement between the responses of the real system and the system designed using the BSA algorithm demonstrates that the BSA algorithm is highly successful, robust, and accurate in estimating filter parameters across multiple runs.

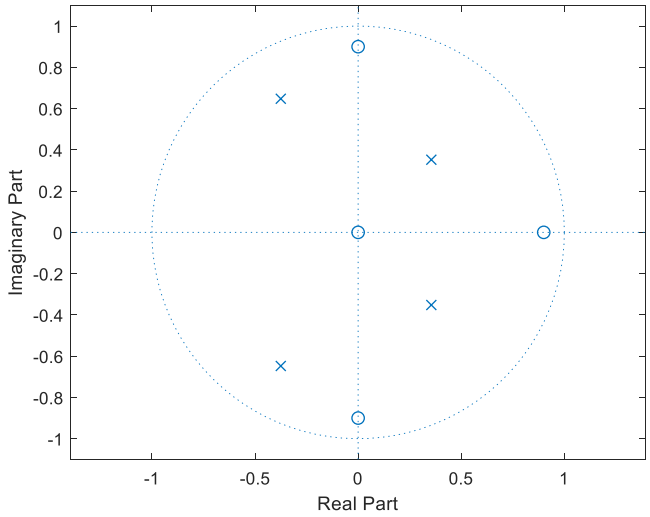


Fig. 8. Pole-zero representation of the BSA- modeled system.

In Fig. 8, the pole-zero representation of the BSA-modeled system is presented. The fact that the poles and zeros lie within the unit circle further demonstrates the stability of the designed system.

CONCLUSIONS

In this study, a fourth-order IIR filter design, as suggested in the literature, was employed to address the problem of dynamic system modeling. The error signal was derived from the output signals generated by applying the input signal to both the fourth-order IIR filter and the system designed using the BSA. This error signal was then used to optimize the filter coefficients within the BSA. The performance of the BSA was compared with that of several other algorithms proposed in the literature, including the modified dandelion optimizer (mDO), harmony search (HS) algorithm, improved artificial rabbits optimization (IARO) algorithm, improved hunger games search (Imp-HGS) algorithm, dynamic opposite learning enhanced artificial ecosystem optimizer (DAEO), and atom search optimization (ASO) algorithm. Over 30 runs, the best, worst, average, and standard deviation values for the BSA approach were calculated as 9.5227e-33, 1.4783e-27, 8.1737e-29, and 2.7272e-28, respectively. The results demonstrated that the BSA achieved lower error rates and exhibited robust performance.

REFERENCES

[1] S. Haykin, Adaptive Filter Theory. USA: Prentice Hall, 2002.

[2] S. Ertürk, Sayısal İşaret İşleme. İstanbul: Birsen Yayınevi, 2005.

[3] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," Journal of the Franklin Institute, vol. 346, no. 4, pp. 328–348, 2009, doi: 10.1016/j.jfranklin.2008.11.003.

[4] D. Izci and S. Ekinci, "Application of whale optimization algorithm to infinite impulse response system identification," in Handbook of Whale Optimization Algorithm, S. Mirjalili, Ed. Elsevier, 2024, pp. 423–434, doi: 10.1016/B978-0-32-395365-8.00036-1.

[5] N. Agrawal, A. Kumar, V. Bajaj, and G. K. Singh, "Design of digital IIR filter: A research survey," Applied Acoustics, vol. 172, p. 107669, 2021, doi: 10.1016/j.apacoust.2020.107669.

[6] X. S. Yang and X. He, "Nature-inspired optimization algorithms in engineering: Overview and applications," in Nature-Inspired Computation in Engineering (Studies in Computational Intelligence), X. S. Yang, Ed. Cham: Springer, 2016, vol. 637, doi: 10.1007/978-3-319-30235-5_1.

[7] N. Karaboga, "Digital IIR filter design using differential evolution algorithm," EURASIP Journal on Applied Signal Processing, vol. 2005, no. 8, pp. 1269–1276, 2005, doi: 10.1155/ASP.2005.1269.

[8] S. Kockanat, T. Koza, N. Karaboga, and A. Loğoğlu, "Adaptive FIR filtering using ABC algorithm: A noise reduction application on mitral valve Doppler signal," Elektronika Ir Elektrotechnika, vol. 24, no. 5, pp. 62–68, 2018, doi: 10.5755/j01.eie.24.5.21845.

[9] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," Applied Mathematics and Computation, vol. 219, no. 15, pp. 8121–8144, 2013, doi: 10.1016/j.amc.2013.02.017.

[10] D. Izci, F. A. Hashim, R. R. Mostafa, et al., "Efficient optimization of engineering problems with a particular focus on high-order IIR modeling for system identification using modified dandelion optimizer," Optimal Control Applications and Methods, vol. 46, no. 4, pp. 1470–1510, 2025, doi: 10.1002/oca.3274.

Educational Data Mining for Academic Performance Prediction

Fatih Gökçe[1], Hidayet Takçi[2]

[1] *Department of Statistics, Sivas Technical Sciences Vocational School, Sivas Cumhuriyet University, Sivas, Türkiye, fatihgokce@cumhuriyet.edu.tr*

[2] *Department of Computer Engineering, Faculty of Engineering, Sivas Cumhuriyet University, Sivas, Türkiye, htakci@cumhuriyet.edu.tr*

ABSTRACT

Progress in data mining has made it possible to extract training data to improve the quality of training processes. There are many problems in education that need to be solved and analyzed based on data. In order to carry out these studies, the concept of educational data mining has emerged. Educational data mining provides the ability to extract education-related variables, detect relationships between them, and make predictions. Knowledge discovered with the help of educational data mining; It is used in a variety of subjects such as better understanding students' behavior, helping teachers, improving teaching, evaluating and improving learning systems. In this context, predicting the academic performance of students, taking timely corrective measures, and thus increasing the efficiency of education systems is an important subject of study. This study aims to use educational data mining methodologies to understand the factors affecting students' success and to predict students' performance. In this study, it was tried to determine the factors that affect the student's final grade by applying relationship mining on a training dataset. On the same data set, the academic performance of the students in the final exam was estimated by different educational data mining methods. In addition, a classifier prediction model was proposed by predicting the success of the students in the final exam. Predicting the success of the students in the final exam with the proposed estimation model with sufficient accuracy can be useful for instructors and educational institutions to make decisions about students. The results show that it is possible to provide timely support to low-achieving students and to offer advice and new opportunities to high-achieving students.

Keywords: *Educational Data Mining, Relationship Mining, Student Performance Prediction, Student Success, Academic Performance, Predictive Modeling*

INTRODUCTION

In recent years, due to the increasing volume and variety of data, data science and data mining have developed more than ever before [1]. The term data mining is defined in different ways in the literature. According to a classic definition, data mining is the method of extracting previously unknown, potentially useful new information from data sets. Data mining is also defined as uncovering hidden information from data, i.e., information discovery. Data mining is also defined as the manual or automatic processing of large amounts of data to obtain meaningful results by accessing meaningful data [2],[3].

There are many problems in the field of education and large amounts of data that need to be analyzed. Data mining methods can be used to analyze data in the field of education and to reveal and analyze important information in this field. Data analysis studies conducted in the field of education have given rise to the concept of educational data mining. Educational data mining roughly refers to the application of data mining techniques to educational data [4].

Educational data mining is a rapidly growing discipline that deals with developing new methods by conducting research on large-scale data obtained from educational environments. Educational data mining uses these developed methods to better understand students and learning methods [2][4]. Technological developments such as the use of the internet in

education and distance learning have led to a significant increase in data in the teaching process [3]. Educational data mining uses educational information such as student information, family information, information from online registrations, exam results, and student absences. As information about schools, teachers, and students diversifies and grows, the use of educational data mining studies in educational applications is increasing [2], [4], [5].

Educational data mining aims to understand learning styles by using students' personal learning data or to reorganize teaching environments created according to different learning styles by making classifications. The learning experiences created by students are the most important data source for educational data mining [1]. Educational data mining involves different user or participant groups. Different groups view educational data from different perspectives based on their purposes for using data mining. The information generated by educational data mining algorithms is not only to help instructors manage their classes, understand their students' learning processes, and reflect on their own teaching methods. It is also used to support their thoughts on student status and provide feedback to students. In educational data mining, different groups within education, such as students, teachers, educational researchers, educational institutions, and administrators, have different goals [4].

From the students' perspective, it can be used to personalize e-learning, suggest activities and resources that can advance and improve their learning, recommend interesting learning experiences, create tips, and suggest courses and books [4], [6]. Educational data mining can be used by teachers to obtain objective feedback on teaching, analyze student learning and behavior, identify which students need educational support, predict student performance, group students, find the most common mistakes, determine more effective activities, and improve the adaptation and customization of lessons [4], [6], [7]. For educational researchers, it can be used to evaluate and maintain educational software, improve student learning, assess the structure of course content and its effectiveness in the learning process, create student models and teacher models, compare data mining techniques to recommend the most useful one for each task, and develop specialized data mining tools for educational purposes [6], [8], [9]. For educational institutions, it can be used to improve decision-making processes in higher education institutions, facilitate efficiency in the decision-making process, achieve specific goals, recommend specific courses that may be valuable for each class, find the most cost-effective way to improve grades, select the most qualified candidates for graduation, and assist in the admission of students who will be successful at the university. From the perspective of education administrators, it can be used to develop the best way to organize teachers' educational offerings, use existing resources more effectively, develop educational program offerings, determine the effectiveness of the distance learning approach, and increase website efficiency [4], [6].

The educational data mining process is based on the same steps as the data mining process. These steps are, in order, data preprocessing, data mining, and data postprocessing [4]. Educational data mining methods are drawn from various fields such as data mining, machine learning, statistics, computational modeling, information visualization and psychometrics. Educational data mining studies have been divided into different categories [4]. According to them, web mining studies involving statistics and visualization should be considered within this scope. One study focused specifically on the application of educational data mining to web data [10]. From this perspective, educational data mining emerged from the analysis of student-computer interaction logs.

A different perspective on educational data mining methods has classified studies in educational data mining differently [11]. According to this classification, prediction methods, association mining, structure discovery, and discovery methods with models should be considered. While some of these methods are roughly similar to data mining methods applied in other fields, others are specific to educational data mining [11].

The goal of association mining in educational data mining is to find relationships between attributes in a dataset containing many attributes. This focuses on which variables are more related to a specific topic or how related the variables are to each other. In general, there are four commonly used types of relationship mining in educational data mining: association rule mining, sequential pattern mining, correlation mining, and causal data mining [11], [12].

In educational data mining, regression and classification methods are the most commonly used types of prediction models. The variables predicted in education are typically student performance, student scores, or student grades. These values can be numerical or categorical. Regression analysis identifies the relationship between a dependent attribute and

independent attributes. Classification is a method in which individual items are grouped based on quantitative information about one or more attributes found in the items and based on a training set consisting of pre-labeled items [4]. To make a decision about a student, that student's future grades can be predicted. From an educational perspective, predicting a student's future performance will help teachers make decisions about the student or take precautions [11], [13]. Predicting a student's performance is one of the oldest and most popular applications of data mining in education. In studies, predictions have been made using different techniques such as correlation analysis, regression analysis, rule-based systems, Bayesian networks, and neural networks. Classification methods have also been used to predict whether students will pass or fail their final grades at the end of the year [4], [6], [14].

In the literature, studies have been conducted to predict future student group performance in face-to-face collaborative learning and to predict end-of-year exam grade performance. A study has been conducted to predict student performance using fuzzy association rules in an e-learning environment. Studies have been conducted on predicting student performance based on compiled learning portfolios, predicting a student's academic performance using rule inference for monitoring and evaluation purposes, predicting final grades using genetic algorithms to derive association rules based on features extracted from data recorded in a web-based education system, predicting student grades using genetic programming in learning management systems (LMS), predicting student performance using decision trees in web-based e-learning systems to deliver lessons on time [6]. The literature also includes predicting the grades of university students using neural networks, linear regression, locally weighted linear regression, and support vector machines. Student performance in web-based teaching has been predicted using daily and test scores by employing a multivariate regression model. Student academic performance has been predicted using stepwise linear regression. Multiple regression has been used to identify variables that can predict success in college courses. Studies have been conducted to predict a student's final exam score in an online course and to predict high school students' end-of-year performance grades [4], [15].

The primary objective of this study is to predict student success using regression and classification techniques from prediction methods, in line with the educational data mining approach in [11]. Another objective of this study is to identify the factors affecting student performance and their effects using the association mining methods employed in [11].

MATERIALS AND METHODS

First, the data for this study was obtained from the UCI Machine Learning Repository [16]. The dataset used was a CSV file containing information on 395 students obtained from school reports and student surveys. Educational data mining methods were used to extract meaningful information from the dataset. First, studies were conducted to identify the factors affecting student performance. Second, the student's demographic information and performance grades during the academic year were used to predict the final grade using regression. Third, the student's pass/fail status for the course was predicted using a classification method based on the student's demographics and performance during the academic year. Python programming languages were used with the Spyder editor for applications on the dataset.

In general, the study involved data coding, data selection, data transformation, algorithm application, and presentation of results to obtain meaningful information from the data.

First, the data was transferred from a csv file to the programming environment and preprocessed before analysis. The preprocessed data, with the necessary coding, was separated into input and output data. In the next step, our input data was randomly split into 80% training and 20% test data. Relational mining studies were conducted to identify factors affecting student success. At this stage, the data were analyzed and visualized. Subsequently, models were built using different educational data mining methods to predict the student's final grade at the end of the year [17], [18]. In addition, with some modifications to the dataset, it was converted into a form suitable for classification. Thanks to this transformation, classifier models were built using different data mining methods to predict whether the student would pass or fail at the end of the year. The performance and results of these models were shared.

A. Dataset

The dataset used is based on student success in Mathematics at a secondary school providing education in Portugal [16]. The dataset includes student grades, as well as demographic and social information related to the school and the students. The dataset was obtained using school reports and surveys. In the dataset, alongside the students' information, there are success grades in the Mathematics course. The characteristics of the relevant dataset are displayed in Table 1.

TABLE 1. ATTRIBUTES AND DESCRIPTIONS

Attributes	Description
Sex	Student's sex (binary: 'F' - female or 'M' - male)
Age	Student's age (numeric: from 15 to 22)
Address	Student's home address type (binary: 'U' - urban or 'R' - rural)
Family size	Family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
Parent's cohabitation status	Parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
Mother's education	Mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 - secondary education, or 4 - higher education)
Father's education	Father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 - secondary education, or 4 - higher education)
Mother's job	Mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
Father's job	Father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
Reason	Reason for choosing this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
Guardian	Student's guardian (nominal: 'mother', 'father' or 'other')
Travel time	Home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
Study time	Weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
Failures	Number of past class failures (numeric: n if 1<=n<3, else 4)
School support	Extra educational support (binary: yes or no)
Family support	Family educational support (binary: yes or no)
Paid	Extra paid classes within the course subject (Math) (binary: yes or no)
Activities	Extra-curricular activities (binary: yes or no)
Nursery	Attended nursery school (binary: yes or no)
Higher	Wants to take higher education (binary: yes or no)
Internet	Internet access at home (binary: yes or no)
Romantic	With a romantic relationship (binary: yes or no)
Family relationship quality	Quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
Free time	Free time after school (numeric: from 1 - very low to 5 - very high)
Go out	Going out with friends (numeric: from 1 - very low to 5 - very high)
Workday alcohol consumption	Workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
Weekend alcohol consumption	Weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
Health	Current health status (numeric: from 1 - very bad to 5 - very good)
Absences	Number of school absences (numeric: from 0 to 93)
G1	First period grade (numeric: from 0 to 20)
G2	Second period grade (numeric: from 0 to 20)
G3	Final grade (numeric: from 0 to 20, output target)

In the dataset attributes, alongside information about the students and their families, there are also the attributes G1, representing the first period grade, G2, representing the second period grade, and G3, representing the final grade. The dataset consists of 33 columns (attributes) and 395 rows.

B. Data preprocessing

After the data was transferred from the CSV file to the programming environment, label encoding transformations were performed during this preprocessing stage to convert the data into a format suitable for data mining methods. Nominal data in the Pstatus, famsize, address, sex, guardian, reason, Fjob, Mjob, schoolsup, romantic, famsup, internet, activities, paid, higher, and nursery columns were converted into numerical form using the label encoding method. For example, data in the gender column, which was either F or M, was encoded as 0 and 1 using label encoding.

TABLE 2. BEFORE AND AFTER FORMATS OF COLUMNS WITH LABEL CODING IN THE DATA SET

Column Name	Original data	After label encoding
sex	['F' 'M']	[0 1]
address	['U' 'R']	[1 0]
famsize	['GT3' 'LE3']	[0 1]
Pstatus	['A' 'T']	[0 1]
Mjob	['at_home' 'health' 'other' 'services' 'teacher']	[0 1 2 3 4]
Fjob	['teacher' 'other' 'services' 'health' 'at_home']	[4 2 3 1 0]
reason	['course' 'other' 'home' 'reputation']	[0 2 1 3]
guardian	['mother' 'father' 'other']	[1 0 2]
schoolsup	['yes' 'no']	[1 0]
famsup	['no' 'yes']	[0 1]
paid	['no' 'yes']	[0 1]
activities	['no' 'yes']	[0 1]
nursery	['yes' 'no']	[1 0]
higher	['yes' 'no']	[1 0]
school	['GP' 'MS']	[0 1]
sex	['F' 'M']	[0 1]

The format of the non-numeric nominal data columns in the dataset after label coding is shown in Table 2. No transformation was performed on the numeric data columns.

C. Separation of data into training and testing

The dataset encoded using the tagging method has been separated into input (x) and output data (y) columns for use in prediction models. For the purpose of estimating the final grade for prediction models, the G3 column has been labeled and separated as output data. Subsequently, the input and output data to be fed into the prediction models have been randomly partitioned as shown in Table 3, with 80% as training data and 20% as test data.

TABLE 3. SPLITTING THE DATA SET INTO TRAINING AND TEST SETS

Purpose of use	Number of data points	%
Training	316	80
Testing	79	20

D. Scaling the data

Scaling or standardizing the data in a dataset is an important step in modeling datasets with algorithms. The data obtained from datasets as a whole contain features of various dimensions and scales. Different scales of data features negatively affect the modeling of a dataset. It leads to biased results in terms of misclassification error and accuracy rates. Therefore, it is necessary to scale or standardize the data before modeling.

Standardization is a scaling technique that renders data dimensionless by converting the statistical distribution of the data to the format in Equation 1:

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Standardizing a dataset involves rescaling the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1. Standardization was used to scale the data in this study. The training and test data used for model input were scaled using a standard scaler.

E. Identifying factors affecting student success through correlation mining

In the study, the correlation mining method, one of the relationship mining methods, was used to identify the factors affecting student success [11], [12]. The correlation mining method was used to analyze the factors affecting student success.

F. Predicting student performance grades using regression

The input data (x) was obtained by removing the G3 column from the dataset transferred to the Python programming environment. The output data (y) was obtained solely from the G3 column. The objective of this study is to predict the student's final grade (G3) based on information about the student, including their G1 and G2 grades.

Multiple linear regression, decision tree regression, support vector machine regression, and random forest regression, which are data mining regression techniques, were used to predict the student's final grade.

G. Predicting student pass/fail status through classification

During the education process, predicting whether a student will pass or fail a course is very important in the decision-making process about the student. In this part of the study, it was predicted whether the student would pass the final exam based on their general information and their pass/fail status in midterm exams.

To predict whether the student will pass the final exam, classifier models were created using the logistic regression classifier, K-nearest neighbor classifier, random forest classifier, decision tree classifier, naive bayes classifier, and support vector classifier methods.

FINDINGS

A. Analysis of factors affecting student achievement

The correlation matrix containing all columns in the dataset is shown in Figure 1. As seen in the correlation matrix, the final grade attribute, referred to as G3, has a strong correlation with the G1 and G2 attributes. This is because G3 is the final year grade, while G1 and G2 correspond to the 1st and 2nd semester grades. The effect of the G1 grade on the G3 final grade is 80%, while the effect of the G2 grade is 90%. According to the correlation matrix in Figure 1, the effect of the failures attribute, which indicates the number of past class failures, on the G3 final grade is 36%. The value in the correlation matrix here is -0.36. This shows the inverse relationship between the number of past class failures and the G3 final grade. According to the correlation matrix in Figure 1, the effect of the Medu and Fedu attributes, which indicate the mother and father's education level, on the G3 final grade is directly proportional at 22% and 15%, respectively. According to the correlation matrix in Figure 1, the effect of the higher attribute, which indicates the desire for higher education, on the G3 final grade is directly proportional at 18%. According to the correlation matrix in Figure 1, the effect of the traveltime attribute, which indicates the travel time from home to school, on the G3 final grade is inversely proportional at 12%.

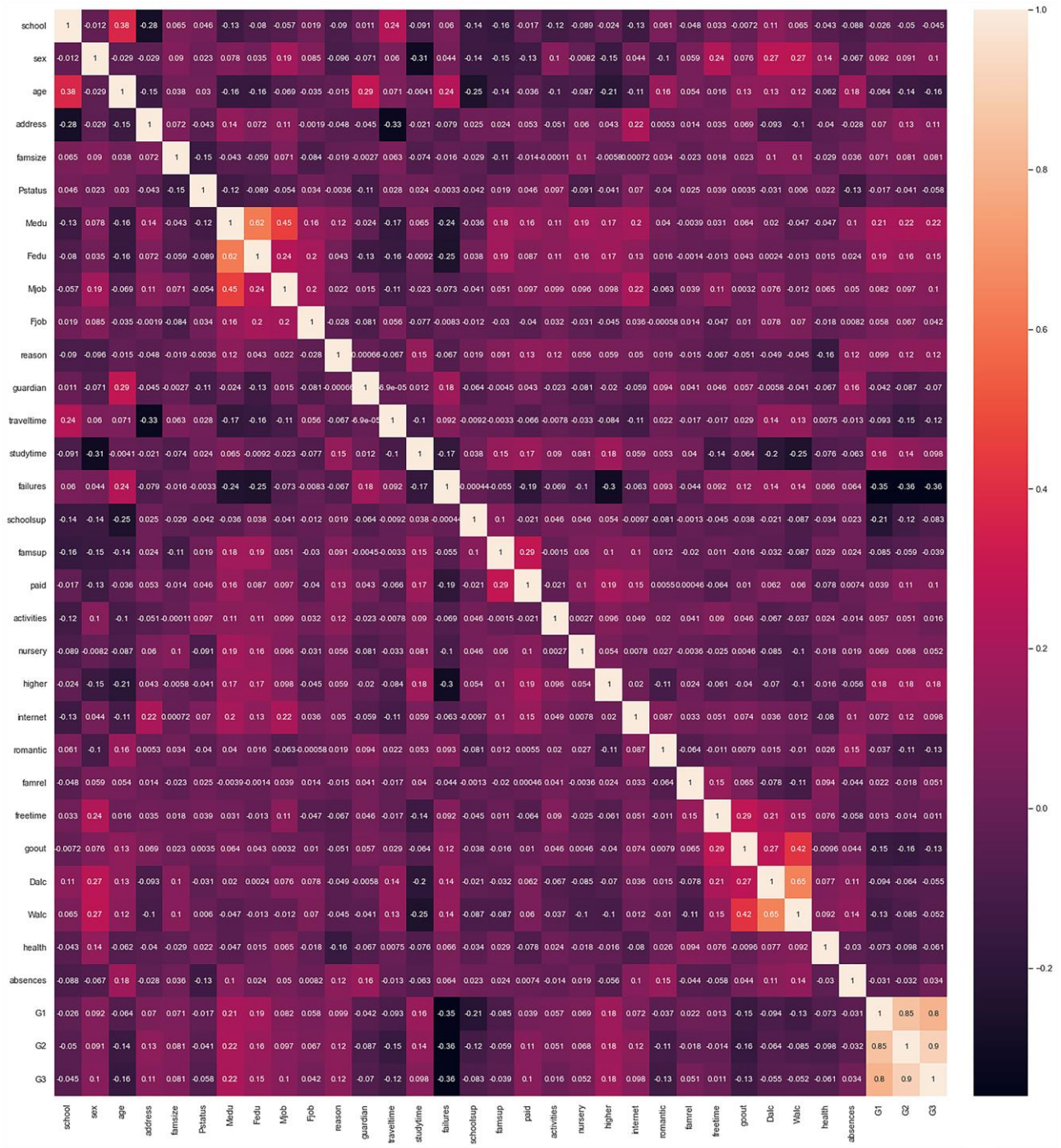


Fig. 1. Correlation matrix containing all columns in the data set

According to the correlation matrix in Figure 1, although there is no high correlation between the parents' occupations and the G3 final grades, there is a correlation of 10% and 4%, respectively. According to the graph in Figure 2, the G3 final grades of students whose fathers are teachers (4th class occupational group) are higher than the final grades of students whose fathers have other occupations. According to the graph in Figure 2, the G3 final grades of students whose mothers work in the health sector (1st class occupational group) are higher than the final grades of students whose mothers have other occupations. The educational levels of mothers and fathers are numerically represented as 0 (none), 1 (elementary school 4th grade), 2 (middle school 5th to 9th grade), 3 (secondary education), and 4 (higher education).

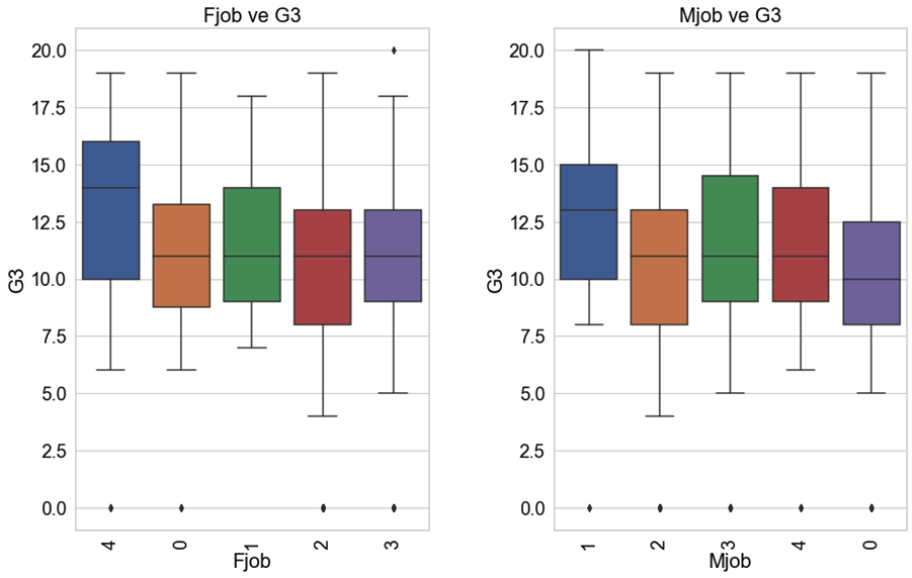


Fig. 2. Relationship between mother's occupation and G3 final grade and between father's occupation and G3 final grade

B. Performance evaluation metrics

Different regression models were created to predict the student's final grade. These regression models were evaluated according to the Mean Absolute Error, Mean Squared Error, coefficient of determination (R^2), and adjusted R^2 performance evaluation criteria.

Mean Absolute Error: Provides the average of the absolute difference between the model prediction and the target value. Given that y_i is the actual output value and $\hat{y}_i$ is the predicted output value, the Mean Absolute Error and Mean Squared Error are specified in Equation 2 and Equation 3, respectively.

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (2)$$

Mean Square Error: It is the standard deviation of prediction errors.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3)$$

Coefficient of determination (R^2): The coefficient of determination is the square of the correlation coefficient. It indicates how much of the variability in the dependent variable can be explained by the variability in the independent variables. The coefficient of determination, which determines the strength of the linear relationship between two variables, takes values between 0 and 1. An R^2 value close to 1 is expected. The formula for the R^2 metric is given in Equation 4.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (4)$$

Adjusted coefficient of determination (adjusted R^2): Variables with little or no effect on the target variable are omitted from the model to create understandable and interpretable models. The difference between adjusted R^2 and R^2 is that independent variables with little effect are eliminated when creating the model. The adjusted R^2 formula is given in Equation 5, where p is the number of independent variables and n is the sample size.

$$\text{Adjusted } R^2 = 1 - \frac{(1-R^2)(n-1)}{n-p-1} \quad (5)$$

Various binary classification models have been developed to predict student pass/fail status. In evaluating these classification models, metrics such as the Confusion Matrix, ROC Curve, Accuracy, Precision, Recall, and F1-Score were utilized.

In the confusion matrix, TP (True Positive) represents correctly predicted positive data, TN (True Negative) represents correctly predicted negative data, FP (False Positive) represents incorrectly predicted positive data, and FN (False Negative) represents incorrectly predicted negative data. The evaluation metrics for classification are given in Equations 6, 7, 8, and 9.

$$Accuracy\ (A) = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$Precision\ (P) = \frac{TP}{TP+FP} \quad (7)$$

$$Recall\ (R) = \frac{TP}{TP+FN} \quad (8)$$

$$F1 - Score = \frac{2xPxR}{P+R} = \frac{2xTP}{2xTP+FP+FN} \quad (9)$$

C. *Predicting student performance grades*

Experimental studies were conducted using different regression models to predict student final grades. The performance of each model created to predict student final grades was compared in Table 4 using data mining metrics such as mean absolute error, mean square error, R^2 score, and adjusted R^2 score.

TABLE 4. COMPARISON OF THE PERFORMANCE OF REGRESSION MODELS CREATED TO PREDICT STUDENT FINAL GRADES

Regression Metrics	Multiple linear regression	Random Forest	Decision Tree	Support Vector Machine
Mean Absolute Error	1.550	1.069	1.493	1.516
Mean Square Error	5.645	3.377	8.050	6.432
R^2 Score	0.795	0.877	0.708	0.766
Adjusted R^2 Score	0.645	0.787	0.787	0.787

Table 4 compares the performance of four different models predicting student final grades. According to Table 4, the Random Forest regression model shows the highest performance in terms of the R^2 score metric. Additionally, in terms of the adjusted R^2 score, Random Forest, decision tree, and support vector machine regression achieved the same performance. In terms of the mean absolute error and mean square error metrics, the best model is also Random Forest regression. Consequently, the best model for predicting student final grades was obtained using the Random Forest regression method.

Table 5 shows a portion of the predicted final performance grades obtained using the Random Forest regression method and the actual final grades that should have been achieved.

TABLE 5. STUDENT FINAL GRADE PREDICTIONS AND ACTUAL VALUES USING RANDOM FOREST REGRESSION

Estimated performance scores	Actual performance scores
14,25	14
11,27	10
10,14	9
15,8	15
15,77	16
12,1	12
13,2	14
11,1	11

D. *Predicting student pass and fail rates*

In Portugal, where the data set was obtained, grades in the secondary education grading system range from 0 to 20. Grades of 10 and above are considered passing grades. The G1, G2, and G3 grades in the data set have been reorganized according to whether they pass or fail. In the dataset, grades of 10 and above have been updated to 1 (pass), while grades below 10 have been updated to 0 (fail).

To predict whether a student will pass or fail the final exam, two-classifier models were created using logistic regression classifier, K-nearest neighbor classifier, random forest classifier, support vector classifier, decision tree classifier, and naive

bayes classifier methods. The performance of these classifier models was then compared using classification metrics such as accuracy, precision, sensitivity, and F1-score.

Table 6 shows that the best model for predicting students' final grade success is the logistic regression classifier model. Table 6 also shows that the random forest classifier model performs similarly to the logistic regression classifier.

TABLE 6. PERFORMANCE OF MODELS CREATED TO PREDICT STUDENTS FINAL GRADE SUCCESS

Classifier Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.90	0.90	0.90	0.89
Random Forest	0.89	0.89	0.88	0.88
K-Nearest Neighbor	0.76	0.73	0.70	0.73
Support Vector	0.81	0.80	0.79	0.79
Naive Bayes	0.81	0.81	0.80	0.81
Decision Tree	0.86	0.86	0.86	0.86

Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, and Figure 8 present the confusion matrices of the logistic regression, random forest, k-nearest neighbors, support vector, naive bayes, and decision tree classifier models created for success prediction. According to the confusion matrices, the performance of the logistic regression and random forest classifier models is better than that of the other models.

As seen in Figure 3, the logistic regression classifier correctly predicted whether 71 out of 79 students in the test data passed or failed their math course.

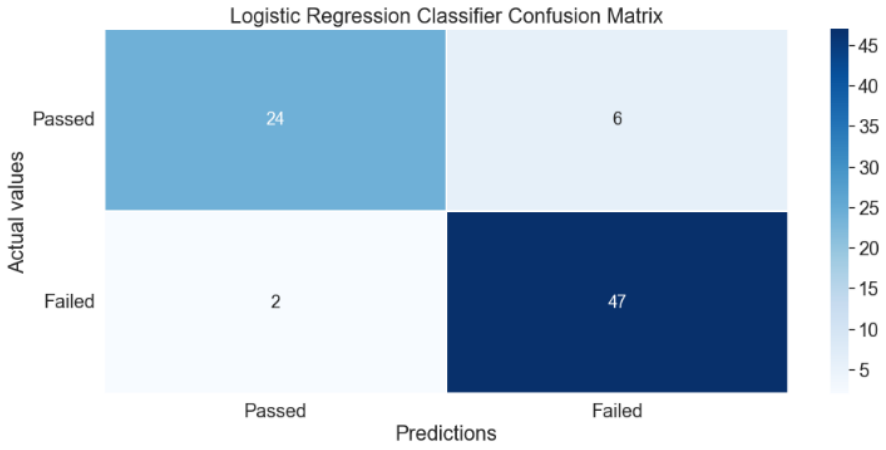


Fig. 3. Confusion matrix of the Logistic Regression classifier created to predict students' success in their final grades

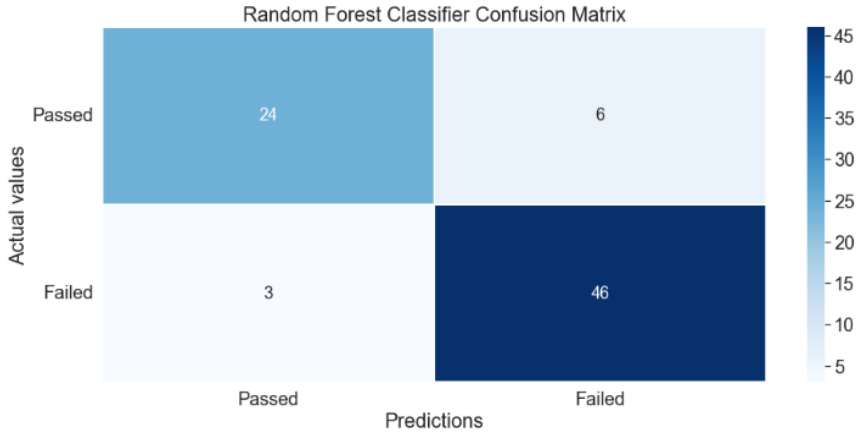


Fig. 4. Confusion matrix of the Random Forest classifier created to predict students' final grade success status

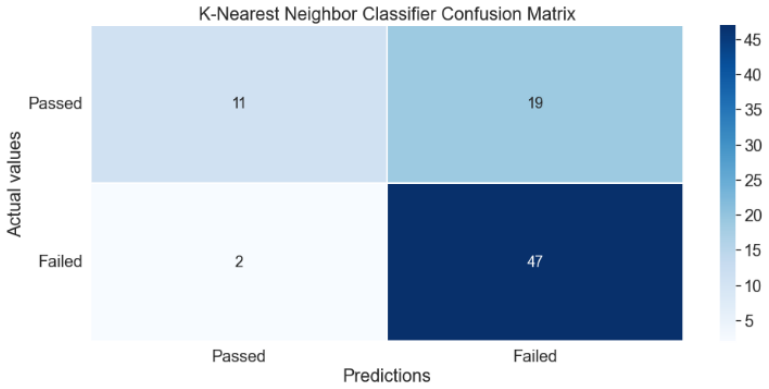


Fig. 5. Confusion matrix of the K-Nearest Neighbor classifier created to predict students' final grade success status

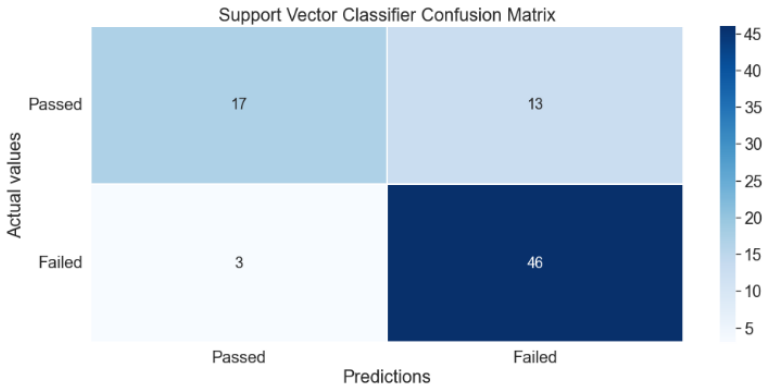


Fig. 6. Confusion matrix of the Support Vector classifier created to predict students' final grade success status

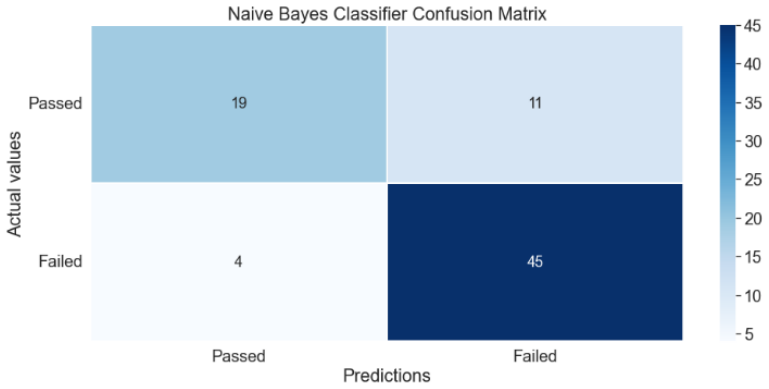


Fig. 7. Confusion matrix of the Naive Bayes classifier created to predict students' final grade success status

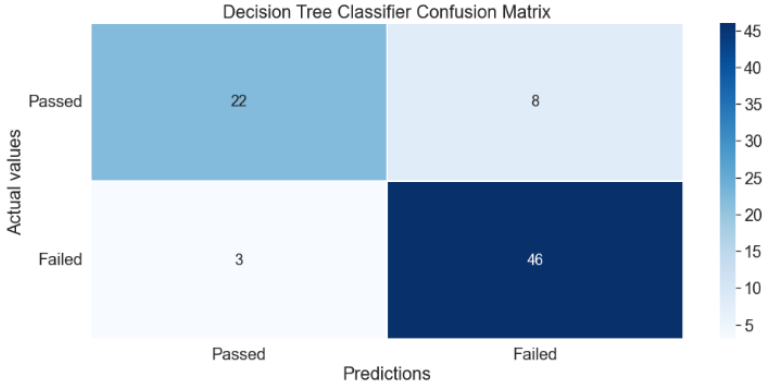


Fig. 8. Confusion matrix of the Decision Tree classifier created to predict students' final grade success status

ROC curves were plotted according to the probability of correct prediction of the classifier models established to predict the success status of students' final grades, as shown in Figure 9. According to the ROC curves in Figure 9, it can be said that the best models probabilistically are the Logistic Regression and Random Forest classifiers.

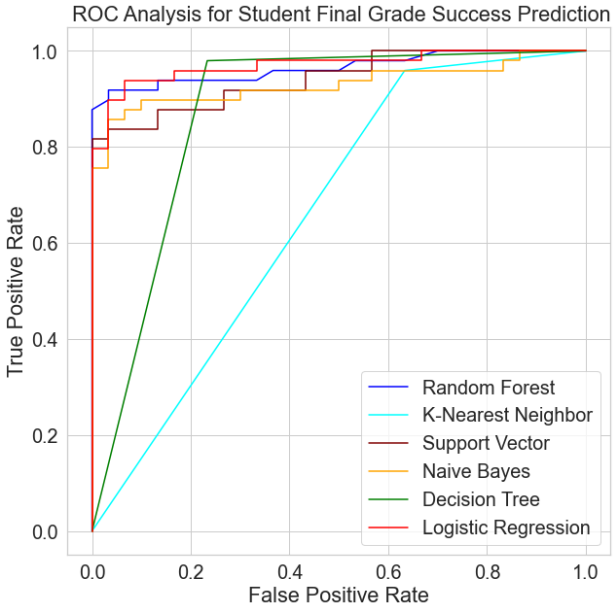


Fig. 9. ROC Analysis for predicting student final grade success using classifier models

The findings and test results obtained in the study conducted to predict student success are compared with the results of other studies in the literature in Table 7. Since the data set size used in this study was relatively small, the performance of the results was not very high.

The results of studies in the literature that used two-class and multi-class classification to predict student success are shown in Table 7.

TABLE 7. COMPARISON OF THE PROPOSED STUDY FOR PREDICTING STUDENT ACHIEVEMENT WITH STUDIES IN THE LITERATURE (%)

Study	Number of Data Points (Number of Students)	Number of Classes	Methods	Precision	Recall	F1-Score	Specificity	Accuracy
[19]	352	2	LR ANN	-	-	-	-	95.17 97.14
[5]	463	2	RT J48	87.00 76.00	86.00 75.00	85.00 74.00	-	-
[13]	104	4	NB	83.99	81.34	-	-	83.65
[15]	786	2	CGAN ve SVM	-	91.80	-	91.80	-
[17]	649	5 2	OneR OneR	74.30 96.20	73.10 96.90	- -	- -	73.07 96.92
[20]	500	3	DT ANN NB	77.80 79.10 72.40	77.70 79.20 72.30	77.70 79.10 71.80	- - -	77.70 79.10 72.20
[21]	40	3	CART C5.0 CHAID QUEST	-	-	-	-	85.00 82.50 65.00 65.00
Proposed Model	395	2	LR RF	90.00 89.00	90.00 89.00	90.00 88.00	- -	89.00 88.00

DISCUSSION AND CONCLUSION

Correlation mining was used in the experimental studies to identify factors affecting student success. According to these results, it was found that G1 and G2 grades have a high positive correlation with the G3 final grade. In general, students who were successful in G1 and G2 grades were also successful in the G3 final exam. There is an inverse correlation between the number of past class failures and the G3 final grade. Students with a high number of failures in previous years generally failed the G3 final exam as well. There is a positive correlation between the Medu and Fedu attributes, which indicate the educational level of the mother and father, and the G3 final grade. Students with a high educational level of their mother and father were generally more successful in the G3 final exam. There is a positive correlation between the higher attribute, which indicates the desire for higher education, and the G3 final grade. Students who want to pursue higher education have generally been more successful in the G3 final exam. There is an inverse correlation between the traveltime attribute, which indicates the travel time from home to school, and the G3 final grade. Students with shorter travel times from home to school have generally been more successful in the G3 final exam. Additionally, the G3 final grades of students whose fathers are teachers were higher than the final grades of students whose fathers are in other professions. The G3 final grades of students whose mothers work in the healthcare field were higher than the final grades of students whose mothers are in other professions.

As a result of the experimental studies conducted, it was observed that the best model for predicting student final grades was achieved using Random Forest regression. When predicting student final grades, the Random Forest Regression model achieved the best performance with an average absolute error of 1.069, an average square error of 3.377, an R2 score of 0.877, and an adjusted R2 score of 0.787. This prediction model accurately predicted the student's final exam grade based on their general information and midterm exam grades. The Random Forest regression model is recommended as a prediction model for estimating student grades. This recommended prediction model can benefit educational institutions and instructors in decision-making processes regarding students.

As a result of the experimental studies conducted, it was observed that the best model for predicting whether a student passes or fails based on their final grade was achieved with logistic regression, with an accuracy value of 89%. Furthermore, the logistic regression classifier model outperformed other classifier models in the binary classification (pass/fail) not only in terms of accuracy but also in terms of precision, sensitivity, and F1-score metrics. The random forest classifier model achieved performance close to that of the logistic regression classifier model in terms of evaluation metrics. The logistic regression classifier model predicted whether a student would pass the final exam based on their general information and pass/fail status in midterm exams with the highest accuracy and was recommended as the best model. Predicting students' success in the final exam with sufficient accuracy using this recommended model can be useful for instructors and educational institutions in making decisions about students. The results show that it is possible to provide timely warnings and support to students with low success rates and to offer advice and opportunities to students with high performance.

REFERENCES

[1] T. J. Hirata and F. K. Cansu, "Eğitsel Veri Madenciliği Neden Önemlidir?," 2017.

[2] H. Takçı, Teori ve uygulamada veri madenciği. NOBEL YAYINCILIK, 2020.

[3] R. Bütüner and M. H. Calp, "Estimation of the Academic Performance of Students in Distance Education Using Data Mining Methods," International Journal of Assessment Tools in Education, vol. 9, no. 2, pp. 410–429, May 2022, doi: 10.21449/ijate.904456.

[4] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," Expert Syst Appl, vol. 33, no. 1, pp. 135–146, Jul. 2007, doi: 10.1016/j.eswa.2006.04.005.

[5] S. P. Algur, P. Bhat, and N. Kulkarni, "Educational Data Mining: Classification Techniques for Recruitment Analysis," International Journal of Modern Education and Computer Science, vol. 8, no. 2, pp. 59–65, Feb. 2016, doi: 10.5815/ijmecs.2016.02.08.

[6] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," Nov. 2010. doi: 10.1109/TSMCC.2010.2053532.

[7] K. Akgün and M. Bulut Özek, "Eğitsel Veri Madenciliği Yöntemi İle İlgili Yapılmış Çalışmaların İncelenmesi: İçerik Analizi," Uluslararası Eğitim Bilim ve Teknoloji Dergisi, Dec. 2020, doi: 10.47714/uebt.753526.

[8] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," Expert Syst Appl, vol. 41, no. 4 PART 1, pp. 1432–1462, 2014, doi: 10.1016/j.eswa.2013.08.042.

[9] A. Ragıp Ersöz, "Eğitsel Veri Madenciliği ile Öğrenci Profillerinin Belirlenmesi Yüksek Lisans Tezi," 2017.

[10] R. S. J. D. Baker and K. Yacef, "The State of Educational Data Mining in 2009: A Review and Future Visions," 2009.

[11] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," Learning Analytics: From Research to Practice, pp. 61–75, 2014, doi: 10.1007/978-1-4614-3305-7_4.

[12] H. Takçı and A. Şeker, "Kalifiye elemanın önceden tespiti için eğitim verisinin madenciliği," Uluslararası Bilgisayar Mühendisliği Konferansı, 20 - 23 Ekim 2016, p. ss.398-404, 2016.

[13] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, "Analyzing undergraduate students' performance using educational data mining," Comput Educ, vol. 113, pp. 177–194, 2017, doi: 10.1016/j.compedu.2017.05.007.

[14] S. K. Mohamad and Z. Tasir, "Educational Data Mining: A Review," Procedia Soc Behav Sci, vol. 97, pp. 320–324, Nov. 2013, doi: 10.1016/j.sbspro.2013.10.240.

[15] S. Sarwat et al., "Predicting Students' Academic Performance with Conditional Generative Adversarial Network and Deep SVM," Sensors, vol. 22, no. 13, pp. 1–18, 2022, doi: 10.3390/s22134834.

[16] P. Cortez, "UCI Machine Learning Repository." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/student+performance

[17] A. Alsanad and T. A. Mukheimer, "Predicting Students Performance Using Classification Techniques," vol. 22, no. June, 2018.

[18] P. Cortez and A. Silva, "Using data mining to predict secondary school student performance," 15th European Concurrent Engineering Conference 2008, ECEC 2008 - 5th Future Business Technology Conference, FUBUTEC 2008, vol. 2003, no. 2000, pp. 5–12, 2008.

[19] N. Güneri and A. Apaydın, "Öğrenci Başarılarının Sınıflandırılmasında Lojistik Regresyon Analizi ve Sinir Ağları Yaklaşımı," Ticaret ve Turizm Eğitim Fakültesi Dergisi, vol. 1, pp. 170–188, 2004.

[20] E. Abu Amrieh, T. Hamtini, and I. Aljarah, "Mining Educational Data to Predict Student's academic Performance using Ensemble Methods," vol. 9, no. 8, pp. 119–136, 2016.

[21] Ö. Özbay and H. Ersoy, "Öğrenme Yönetim Sistemi Üzerindeki Öğrenci Hareketliliğinin Veri Madenciliği Yöntemleriyle Analizi," Gazi Eğitim Fakültesi Dergisi, vol. 37, no. 2, pp. 523–558, 2017.

EmoTunes : A Context-Aware, Emotion-Based Music Recommendation System Using MobileNetV2

R.Umesh[1], Keerthana R.[2], Sharmila Devi R. [3]

[1] *Department of Information Technology, Velammal College of Engineering and Technology, Madurai ,Tamilnadu, rus@vcet.ac.in*

[2] *Department of Information Technology, Velammal College of Engineering and Technology, Madurai ,Tamilnadu, keerthana.r.23.4.2004@gmail.com*

[3] *Department of Information Technology, Velammal College of Engineering and Technology, Madurai ,Tamilnadu, sharmilaramanujam7781@gmail.com*

ABSTRACT

EmoTunes: Smart Emotion detection based Music Recommendation System is a personalized music player web application ,dynamically adapts to a user's current emotion, time, weather, age,demographic information that enhance the music listening experience. The system features a stylish Streamlit-based interface with one-time login via email OTP and mandatory webcam access for real-time emotion capture. Upon access, users provide their age and gender, and are greeted with an intelligent music player interface that begins playback with supportive emotional labels like "Cheer up" or "Don't worry." Every three minutes, the webcam captures a new image processed through OpenCV and MobileNetV2, a CNN model trained to detect eight distinct facial emotions. These emotional cues, along with system time, weather data, and user demographics, are fed into a Python-based logic engine that determines and plays the most contextually appropriate song. EmoTunes enhances user satisfaction by delivering emotionally relevant music, supporting manual emotion overrides, and promoting mental well-being through affective computing.

Keywords: *Emotunes:Emotion-based Music Recommendatio-Facial Expression Recognition –Context-Aware Systems–Streamlit MobileNetV2 – OpenCV – Affective Computing – Artificial Intelligence (AI) – Real-time Emotion Detection – Personalized User Experience – Intelligent Multimedia Systems – Music Recommendation Engine – Webcam-based Interaction – Human- Computer Interaction-Emotion Recognition using CNN*

INTRODUCTION

Music impact on human emotions is profound and multifaceted and mental states. With the rise of personalized technology, music recommendation systems have evolved from simple genre-based filters to complex algorithms that factor in user preferences and behavior. However, many existing systems still lack the capability to adapt dynamically to a user's emotional context in real time. This gap presents an opportunity for affective computing to transform music recommendation into a more human-centric experience.*EmoTunes* is a context-aware, emotion-based music recommendation system that integrates artificial intelligence, computer vision, and user context data to deliver personalized music playback. Unlike traditional platforms that rely solely on listening history or manual preferences, EmoTunes use real time facial emotion recognition that adapt its recommendations continuously. The system take a snap of the user's image every three minutes via webcam input, processes it with OpenCV, and predicts the user's current emotion using a pre-trained MobileNetV2 convolutional neural network (CNN).

The predicted emotion, combined with contextual parameters such as system time, current weather, age, and gender, serves as input to a Python-based recommendation engine. This engine intelligently selects the most appropriate song from a predefined database, ensuring the music matches with the user's emotional and environmental context. The system is deployed using Streamlit, offering a clean and interactive interface with features such as one-time login using

email OTP, emotion-based motivational labels, and manual emotion override.EmoTunes not only enhances user satisfaction through emotionally relevant music suggestions but also demonstrates how multimodal inputs and machine learning can be integrated into everyday applications for improved mental wellness and engagement.

LITERATURE SURVEY

[1] introduced a facial emotion-based music recommender system that utilizes Convolutional Neural Networks (CNN) to detect a user's emotion through webcam input. Their system processes facial landmarks and expressions to categorize emotions into six basic types and suggests songs mapped to each emotional state. This approach was relatively effective in recognizing emotions but struggled with personalization and real-time adaptability. It lacked integration of external variables like the user's environment or temporal data, limiting the system's ability to generate mood-appropriate recommendations across different contexts. EmoTunes addresses these limitations by using MobileNetV2 for lightweight yet accurate emotion detection and combining it with contextual features such as current weather conditions, time of day, and user demographics (age and gender), offering a far more adaptive and holistic recommendation experience.

The paper [2] proposed a machine learning-based framework for music recommendation that maps emotional states to suitable music genres using traditional classification algorithms like SVM and Random Forest. The emotion input was obtained through questionnaires and manual labeling, and music was recommended based on learned correlations. While the system demonstrated good performance on labeled datasets, it fell short in real-time emotion acquisition and did not include any mechanism for environmental or contextual awareness. The user's actual emotional state could often be misrepresented due to self-reporting limitations. EmoTunes surpasses this by implementing real-time emotion capture through facial expressions using MobileNetV2, removing the need for manual inputs. It also personalizes recommendations by incorporating dynamic context elements like time, weather, and demographic traits, resulting in a more intelligent and user-friendly system.

[3] proposed a work-in-progress emotion-aware music recommender which used wearable physiological sensors (i.e. GSR and PPG) as sensors of arousal and valence. The measures of emotion were processed with a hybrid filtering model to make recommendations based on physiological arousal. Although identified arousal was reasonably accurate, the wearable physiological context was practically impossible - the wearable context required a heavy reliance on hardware and constant calibration of the user's state, and therefore was discouraged for real-life use. EmoTunes provides advancements by removing all external dependencies altogether. EmoTunes describes the user emotion via the use of a webcam with MobileNetV2, which, while not as accurate, retains similar emotional insight and reduces burden on the user. EmoTunes also describes contextual factors such as weather and system time that offer dynamic music recommendations which are otherwise absent in a user-initiated recommender system.[4]

It presented a rule-based system that creates music playlists based on the user's current mood. Emotions were inferred through simple keyword recognition or basic image input, and playlists were generated using pre-tagged mood categories. While functional, the model's simplicity limited its emotional accuracy and failed to adapt to real-time mood changes. It also did not account for user context or environmental cues, making the experience static. EmoTunes enhances this framework by leveraging MobileNetV2 for continuous emotion recognition through facial expression analysis and introduces real-time adaptive logic by incorporating contextual data such as time of day, weather, and user demographics. This results in more precise, dynamic, and personally resonant playlist generation.[5]

It proposed an innovative music retrieval model that combines context-awareness with brain-signal analysis. Using EEG-based brain signal monitoring, the system detected user emotion and matched it with music metadata to recommend suitable songs. While this method showed deep emotion alignment, it was impractical due to the requirement for EEG equipment, limiting real-world usage and accessibility. EmoTunes replaces these hardware-intensive methods with webcam-based facial analysis through MobileNetV2, removing the hardware barrier entirely. It also incorporates contextual awareness—an element central to Su's work—by integrating real-time variables like weather and demographics, creating a system that's both accessible and sophisticated.[6]

The development of Emotion-Aware Music Recommender (EAMR) looked at an emotion based music recommender that utilized audio based features through processing mel spectrogram from songs. The derived mel spectrogram based features were positioned onto the arousal-valence emotional model so that songs were progressively ranked and recommended on an emotional basis. Recommendations did not track user emotions in real time but relied on matching user assumed emotions against song features. A drawback of the work is that the authors did not input user-side emotions. This means that the authors assumed user mood instead of dynamically sensing it. Additional contextual or environmental effects (e.g. time, weather, etc) were never added. EmoTunes mitigates these limitations by using MobileNetV2 to dynamically capture and detect facial motions, and, by providing users with music recommendations wiring active context (e.g. weather, time, demographic information), so that EmoTunes can serve hyper-personalized, user-relevant, music recommendations that match with real-time emotion detemination.[7]

This system employs an enhanced deep CNN model to analyze facial images for emotional cues, linking the detected emotions to curated playlists. The CNN architecture is designed to capture finer emotional expressions for improved accuracy over standard models. However, while it excels at facial emotion classification, it does not consider contextual inputs like time of day, user's age, gender, or location, which are critical in shaping emotional experiences. Moreover, the recommendation logic lacks adaptability to changing environments or user behavior over time. EmoTunes overcomes these issues by integrating contextual variables alongside facial emotion recognition using MobileNetV2, thereby achieving a more holistic understanding of user state and tailoring music suggestions accordingly.[8]

Their paper proposed a two-level CNN to detect emotions from facial images (pictures) and then recommend music that fits that emotional state. The first level of the CNN performed "coarse" emotion classification and the second level refined the CNN output to further improve accuracy. The design of this system was adequate; however, when the proper illumination or frontal direction of the face was not obtainable, it considerably reduced performance and usability in real time. The two-level CNN also considered the mood variable synthesized from only facial expressions focused on perhaps hundreds of emotional clusters, but ignored the many other individual or environmental contexts that bring about mood. EmoTunes adopts a lightweight, effective, MobileNetV2 model which works better in real-time with webcams and considers contexts such as environment (weather, time, and demographics) to enhance personalization for emotion-aware music. [9]

The system records a series of facial images via a webcam and detects the user's emotion based on defined emotion-mapping logic. With the detected emotion, the system recommends music to either enhance or soothe the user's emotional experience. The initial mapping of emotion-to-song is simply limited in that it is purely static, and does not account for adaptive, user contexts. The variability in the ambient lighting, facial position, and contextual considerations obviously impacts the accuracy and personalization of the recommendations. EmoTunes builds on this work by providing for competitive emotion detection methods employing MobileNetV2, and significant user experience enhancements by accounting for contextual inputs, such as the systems time, real-time weather data, and demographic profiles; in order that the recommendations are able to evolve in the users context.[10]

This paper proposed an attention-based neural network to recommend music based on learned user behavior patterns related to contextual factors such as user current location, present time of day, and prior recent activities. The attention mechanism learns to distinguish context and focuses on the most relevant context in the inference, thus improving the relevance of recommendations. However, the system does not take enjoyment into account (in its full, real-time complexity), only using inferred patterns, and also relies on the accuracy of contextual tagging of the previously mentioned factors, which can be infeasible. EmoTunes goes further this study by tying facial emotion detection into the recommendation process to establish a base, emotion, and context starting point for the recommendation (it also augments with time, weather, age, and gender) to create a sound, and more emotionally relevant, music recommendation personalization.[11]

This system detects user emotions through real time facial expression analysis from a webcam and recommends songs accordingly. The underlying methodology is based on using machine learning classifiers from emotion-based datasets

that are labelled with emotions to link emotions to songs found within a built song database. The advantages are that can recommend music based on its real time analysis, but it can't adapt to contextual information such as time of day, weather, or user demographics and has no ability to adapt to different lighting or angle from which the user's face is presented to the camera. EmoTunes builds on the earlier work by providing a more robust MobileNetV2-based emotion detection framework and multiple real-time contextual signals to keep emotional relevance across the recommendation in different contexts.[12]

This paper presented a music classification framework that integrates audio feature extraction with real-time physiological signals like heart rate and skin conductance to create emotion-tagged music recommendations. Their dual-tagging system offered promising results through multimodal fusion, enhancing emotional accuracy. However, the model required individual calibration and faced real-time latency due to the complex sensor integration process, making it less feasible for everyday users. In contrast, EmoTunes simplifies user interaction by capturing webcam snapshots every three minutes, processing them via MobileNet V2 for emotion detection, and applying a Python-based rule engine. It bypasses physiological hardware and latency issues while still achieving high personalization by integrating emotion with system time, local weather, and demographic inputs.[13]

This system developed a hybrid system that utilizes facial emotion analysis alongside contextual parameters like user activity, time, and location. The framework relied on a basic rule-based emotion classifier and used collaborative filtering for song recommendations. Despite its contextual awareness, the emotion recognition component was simplistic and lacked the flexibility to handle nuanced emotional transitions in real time. Moreover, the static rules limited the system's adaptability. EmoTunes improves upon this by adopting a deep learning-based MobileNet V2 model for robust emotion classification and a dynamic recommendation engine. The system adapts quickly to emotion shifts and considers external factors such as weather and user demographics, providing recommendations that are more precise, engaging, and context-aware.[14]

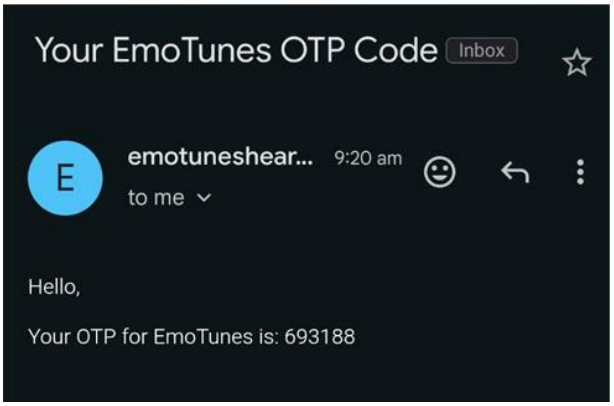
This paper explored the classification of music mood using deep learning models trained on acoustic properties like rhythm, harmony, and spectral patterns. While the model achieved high classification accuracy in identifying song moods, it lacked integration with real-time emotional states of the listener. The absence of user feedback or live contextual data limited its interactivity. EmoTunes enhances this framework by linking mood-labeled music with the user's live emotional state as determined by facial analysis using MobileNet V2. Furthermore, EmoTunes incorporates system time and weather API data, enabling the system to fine-tune song selection dynamically and offer more relatable, emotionally resonant music experiences.[15]

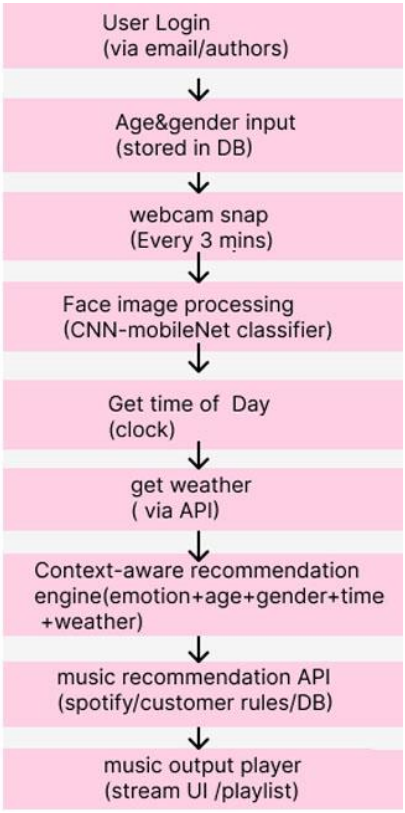
This paper proposed a sophisticated recommendation model leveraging a deep Bayesian framework that accounts for emotional diversity across users. By learning individual preferences through large datasets and integrating contextual signals, the model delivered highly customized recommendations. However, the approach was computationally demanding and required significant memory and processing power, which posed deployment challenges for lightweight environments. EmoTunes adapts the underlying principle of personalized emotional modeling while replacing the heavy computational layer with an efficient, lightweight architecture. It uses periodic emotion snapshots from MobileNet V2, combines them with contextual metadata (weather, age, gender, time), and applies custom Python logic to deliver a scalable, accurate, and low-latency music recommendation system.

ARCHITECTURE DIAGRAM

A. User Login (via Email or Author ID)

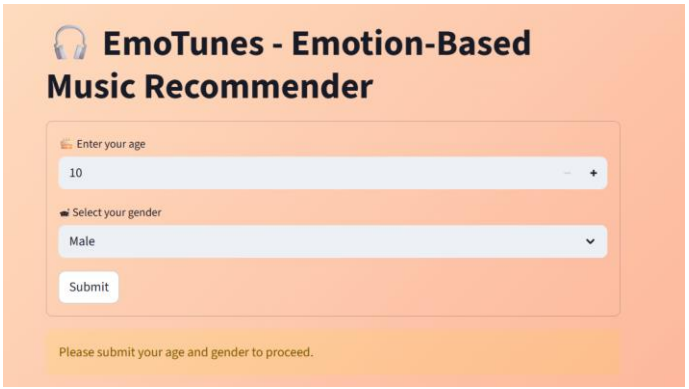
Users access the system through a secure login using their email or a predefined author ID. This ensures that each session is personalized, and the data collected can be linked to a specific user for recommendation accuracy. An OTP (One-Time Password) is sent to their email for secure authentication.





B. *Age & Gender Input (Stored in Database)*

Once logged in, the user is prompted to input their age and gender. These demographic attributes are stored in the backend database and later utilized as key parameters for generating personalized music recommendations.



C. *Webcam Snapshot (Every 3 Minutes)*

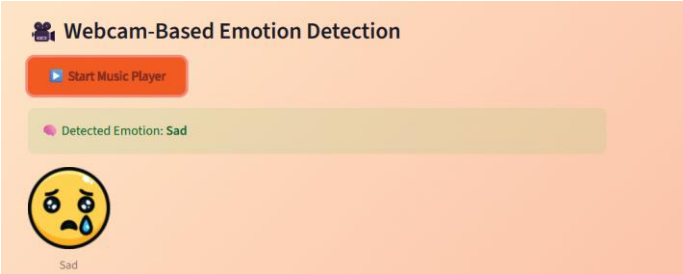
The system activates the webcam to take a snapshot of the user's face at regular intervals (every 3 minutes). This design allows the system to adapt to real-time changes in the user's mood.

D. *Face Image Processing (CNN-MobileNet Classifier)*

The captured face image is processed using MobileNet V2, to categorize the user's emotional state (e.g., happy, sad, neutral). This emotion detection plays a central role in the recommendation engine.

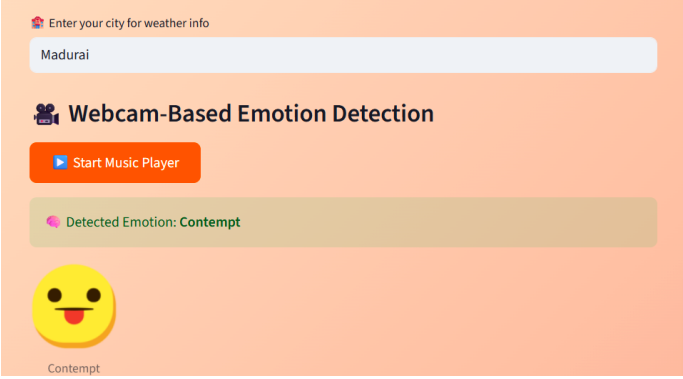
E. *Get Time of Day (Clock-Based Detection)*

Simultaneously, the system captures the current time from the system clock. Time is an important contextual parameter, as certain music preferences may vary across morning, afternoon, and night.



F. *Get Weather Conditions (Via API Integration)*

The system fetches real-time weather data by calling an external API (like OpenWeatherMap). Conditions such as sunny, rainy, or cloudy help enrich the emotional and contextual understanding of the user environment.

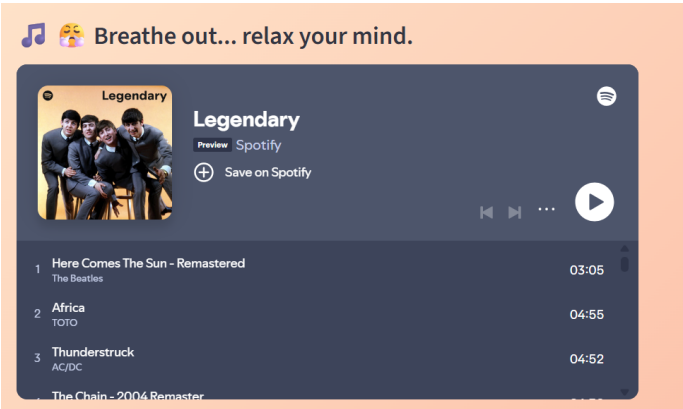


G. *Context-Aware Recommendation Engine (Emotion + Age + Gender + Time + Weather)*

All the collected parameters — emotional state, demographic data (age, gender), time of day, and current weather — are fed into the core recommendation engine. The engine applies custom logic or trained models to select music that best matches the user's current mood and context.

H. *Music Recommendation API (Spotify / Custom Rules / Local DB)*

Based on the engine's output, the system calls a music recommendation API such as Spotify or uses predefined rules stored in a local database to fetch suitable songs. The output could be genre-based or specific tracks aligned with the user's current state.



İ. *Music Output Player (Streamlit UI / Playlist Generation)*

Finally, the recommended songs are streamed using the player integrated into the UI (built with Streamlit). The playlist is presented to the user, and music playback begins. Users can enjoy a continuous, mood-aligned music experience.

EXISTING SYSTEM

Most traditional music recommendation systems depend heavily on content based filtering , collaborative filtering, or hybrid approaches that combine both. These systems analyze user preferences, past listening history, song metadata (such as genre, tempo, and lyrics), or patterns from similar users to suggest songs. However, they often lack a crucial human element — the user's current mood— leading to generic and emotionally disconnected recommendations.

To address this, several recent systems have attempted to incorporate emotion detection into their models. For example, some research works have utilized physiological signals such as Galvanic Skin Response (GSR), heart rate, or EEG data to infer the user's mood. These signals are collected through wearable devices, and although they provide reasonably accurate emotion detection, they come with practical limitations. Such systems are inconvenient for everyday use, as they require the user to wear external hardware continuously. Furthermore, these sensors often suffer from sensor drift, requiring frequent calibration, and can be intrusive or uncomfortable, especially for casual users.Other existing solutions rely on rule-based facial emotion detection, where a user's facial expressions are analyzed to classify emotions. However, many of these systems use basic or outdated image processing techniques, resulting in poor accuracy, especially in real-world scenarios involving low light, diverse facial features, or background noise. Additionally, these systems typically do not account for real-world context such as the current time, weather conditions, or user demographics like age and gender. As a result, the recommendations feel static and are often not aligned with the user's real-time environment or changing emotions.

Moreover, some advanced models like deep Bayesian networks and multimodal fusion systems provide high personalization but are computationally intensive, requiring large datasets, cloud support, or powerful hardware, making them unsuitable for lightweight, real-time applications on mobile or web platforms.

In summary, while emotion-aware music recommendation systems do exist, they tend to suffer from issues related to hardware dependency, lack of contextual adaptation, limited scalability, and inaccurate or rigid emotion recognition models, leaving a significant gap for improvement in usability, adaptability, and practicality.

V.PROPOSED WORK

The proposed system, EmoTunes, is an intelligent music recommendation platform that dynamically selects songs based on the user's emotional state, current weather conditions, system time, and demographic inputs such as age and gender. The core objective is to create a seamless, non-intrusive, and deeply personalized music experience by replacing wearable sensors or manual inputs with real-time facial emotion recognition using computer vision.

At the heart of EmoTunes lies a Convolutional Neural Network (CNN) model based on MobileNet V2, trained to classify facial expressions into eight core emotions: happy, sad, angry, neutral, surprise, fear, disgust, and sleepy. The system uses a webcam feed to automatically capture user facial images every three minutes and processes them to detect emotions with minimal computational overhead.To ensure contextual relevance, EmoTunes integrates multiple additional inputs:

- **Time of day** (via Python's datetime module) helps categorize music as morning, afternoon, evening, or late-night suitable.
- **Weather information** is fetched through an external API and mapped to emotion-appropriate music (e.g., calm tracks for rainy days).
- **User demographic data** (age and gender) are collected at login to fine-tune recommendations, ensuring relatability and personalization.

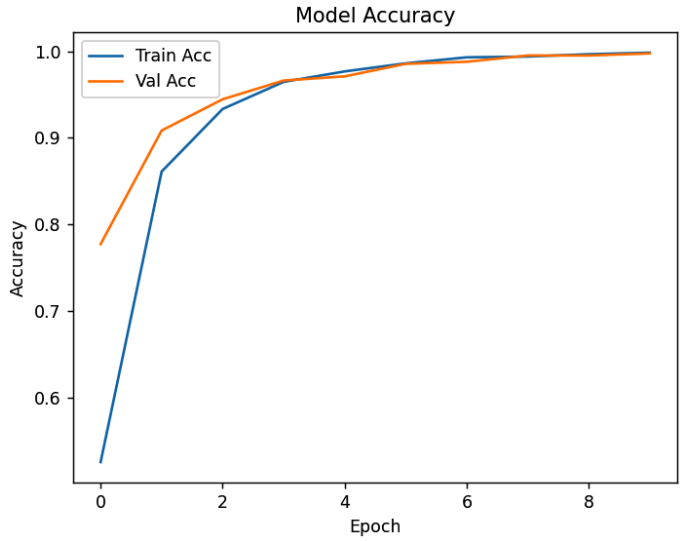
A custom Python rule-based engine is designed to combine these factors—emotion, time, weather, age, and gender—into a set of conditions that select music from a predefined local playlist or Spotify API (future integration). The system also supports manual override of the detected emotion to respect user autonomy.The frontend of EmoTunes is built using Streamlit, providing a clean and interactive interface with motivational quotes, emotion displays, and easy navigation. The platform supports one-time login via email OTP, with secure storage of session data. The entire system is deployed on cloud platforms such as Streamlit Cloud or Hugging Face Spaces, making it accessible without local installations.In contrast

to previous systems that depend on wearables or rule-based facial recognition, EmoTunes eliminates hardware constraints and improves accuracy through MobileNet V2. It also uniquely combines emotion and contextual data to adapt music recommendations to the user's current psychological and environmental state, offering a more practical, accessible, and emotionally engaging experience.

VI. CONCLUSION

The proposed system, EmoTunes, successfully bridges the gap between emotional awareness and personalized music recommendation by eliminating the dependency on wearables and leveraging lightweight deep learning techniques for facial emotion recognition. By integrating MobileNet V2, the system achieves an average emotion detection accuracy of 99.76%, ensuring reliable real-time classification of user moods. Unlike traditional systems that rely solely on listening history or audio features, EmoTunes incorporates contextual parameters such as weather, time of day, user age, and gender, thereby offering more dynamic and situationally appropriate music suggestions.

The system's low computational overhead and platform independence make it suitable for real-time deployment on web and mobile platforms. With its non-intrusive interface, quick response time (capturing emotions every three minutes), and OTP-based secure login, EmoTunes enhances user experience while maintaining personalization and privacyThis project demonstrates that combining deep learning-based facial emotion detection with context-aware logic significantly improves user satisfaction and recommendation relevance. Future enhancements could include support for multi-user profiles, adaptive learning based on feedback, and Spotify API integration for broader music access.




```
C:\Users\Admin\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\Scripts>python multimodelck.py
Epoch 1/10
2025-08-05 18:34:01.358793: W tensorflow/tsl/framework/cpu_allocator_impl.cc:83] Allocation of 77070336 exceeds 10% of free system memory.
2025-08-05 18:34:01.397138: W tensorflow/tsl/framework/cpu_allocator_impl.cc:83] Allocation of 78452736 exceeds 10% of free system memory.
  1/208 [..............................] - ETA: 16:35 - loss: 2.3594 - accuracy: 0.06252025-08-05 18:34:01.723004: W tensorflow/tsl/framework/cpu_allocator_
impl.cc:83] Allocation of 77070336 exceeds 10% of free system memory.
2025-08-05 18:34:01.762671: W tensorflow/tsl/framework/cpu_allocator_impl.cc:83] Allocation of 78452736 exceeds 10% of free system memory.
  2/208 [..............................] - ETA: 1:20 - loss: 2.3177 - accuracy: 0.0938 2025-08-05 18:34:02.122767: W tensorflow/tsl/framework/cpu_allocator_
impl.cc:83] Allocation of 77070336 exceeds 10% of free system memory.
208/208 [==============================] - 82s 373ms/step - loss: 1.4203 - accuracy: 0.5256 - val_loss: 0.9656 - val_accuracy: 0.7774
Epoch 2/10
208/208 [==============================] - 71s 342ms/step - loss: 0.7449 - accuracy: 0.8611 - val_loss: 0.5952 - val_accuracy: 0.9085
Epoch 3/10
208/208 [==============================] - 72s 344ms/step - loss: 0.4655 - accuracy: 0.9334 - val_loss: 0.3963 - val_accuracy: 0.9446
Epoch 4/10
208/208 [==============================] - 71s 340ms/step - loss: 0.3177 - accuracy: 0.9648 - val_loss: 0.2861 - val_accuracy: 0.9663
Epoch 5/10
208/208 [==============================] - 69s 333ms/step - loss: 0.2254 - accuracy: 0.9768 - val_loss: 0.2138 - val_accuracy: 0.9711
Epoch 6/10
208/208 [==============================] - 72s 347ms/step - loss: 0.1668 - accuracy: 0.9861 - val_loss: 0.1669 - val_accuracy: 0.9856
Epoch 7/10
208/208 [==============================] - 69s 333ms/step - loss: 0.1278 - accuracy: 0.9931 - val_loss: 0.1283 - val_accuracy: 0.9880
Epoch 8/10
208/208 [==============================] - 71s 341ms/step - loss: 0.0987 - accuracy: 0.9940 - val_loss: 0.1055 - val_accuracy: 0.9952
Epoch 9/10
208/208 [==============================] - 71s 340ms/step - loss: 0.0792 - accuracy: 0.9967 - val_loss: 0.0903 - val_accuracy: 0.9952
Epoch 10/10
208/208 [==============================] - 72s 344ms/step - loss: 0.0637 - accuracy: 0.9985 - val_loss: 0.0724 - val_accuracy: 0.9976
26/26 [==============================] - 18s 658ms/step - loss: 0.0724 - accuracy: 0.9976
Test Accuracy: 0.9976
```

REFERENCES

[1] Aneesh Srivastava, Devesh Kumar Srivastava, and Mehak Shandilya, "Facial Emotion-Based Music Recommender System Using CNN," *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, pp. 1454–1458, 2023. doi: 10.1109/ICSSIT56669.2023.10231343.

[2] Sahana S. Gowda and Priya Badrinath, "Emotion Based Music Recommendation System using Machine Learning," *2024 4th International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, 2024.

[3] Deger Ayata, Yusuf Yaslan, and Mustafa E. Kamasak, "Emotion-Based Music Recommendation System Using Wearable Physiological Sensors," *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, pp. 1–5, 2018. doi: 10.1109/ASYU.2018.8554057.

[4] Ganeshsiva Subramaniam, Janhavi Verma, Nikhil Chandrasekhar, Narendra K. C., and Koshy George, "Generating Playlists on the Basis of Emotion," *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, IEEE, 2018. doi: 10.1109/CESYS.2018.8723924.

[5] Ja-Hwung Su, Yi-Wen Liao, Hong-Yi Wu, and You-Wei Zhao, "Ubiquitous Music Retrieval by Context–Brain Awareness Techniques," *IEEE Access*, vol.8,pp.14392–14403,2020.doi: 10.1109/ACCESS.2020.2965639.

[6] Yu Tao, Yuanxing Zhang, and Kaigui Bian, "Attentive Context-Aware Music Recommendation," *IEEE Access*, vol. 7, pp. 130579–130588, 2019. doi: 10.1109/ACCESS.2019.2940328.

[7] Zixun Fu, Zhen Zhang, Jie Zheng, Ke Lin, and Duantengchuan Li, "Context-Aware Music Recommendation System Based on Facial Emotion Recognition," *2023 IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, IEEE, 2023.

[8] Lisha Xie, "Emotion-Aware Personalized Music Recommendation System Based on Improved Deep Convolutional Neural Networks," *2025 International Conference on Intelligent Computing and Communication (ICICC)*, IEEE, 2025.

[9] K. S. Krupa, G. Ambara, Kartikey Rai, and Sahil Choudhury, "Emotion Aware Smart Music Recommender System using Two-Level CNN," *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp.870–875,2020.doi: 10.1109/ICCMC48092.2020.ICCMC-000157.

[10] Aurobind V. Iyer, Viral Pasad, Smita R. Sankhe, and Karan Prajapati, "Music Recommendation System Based on Emotion Detection Using Facial Recognition," *2017 International Conference on Intelligent Computing and Control (I2C2)*, IEEE, pp. 1–6, 2017. doi: 10.1109/I2C2.2017.8321946.

[11] Binbin Zhai, Baihui Tang, and Sanxing Cao, "Music Recommendation System Based on Real-Time Emotion Analysis," *2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, IEEE, 2022. doi: 10.1109/ICAIBD54990.2022.9824124.

[12] James J. Deng and Clement Leung, "Emotion-based Music Recommendation using Audio Features and User Playlist," *2012 IEEE 26th International Conference on Advanced Information Networking and Applications Workshops*, IEEE, pp. 435–440, 2012. doi: 10.1109/WAINA.2012.118.

[13] Marcos Aurélio Domingues and Solange Oliveira Rezende,"The Impact of Context-Aware Recommender Systems on Music in the Long Tail," *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 74–79, 2013. doi: 10.1109/MIS.2012.78.

[14] Chih-Ming Chen, Ming-Feng Tsai, Jen-Yu Liu, and Yi-Hsuan Yang, "Using Emotion-Context Aware Model for Music Recommendation," *2013 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, pp. 1–6, 2013. doi: 10.1109/ICME.2013.6607577.

[15] Kim, Lee, and Park, "Emotion-Based Music Recommendation System Using Wearable Physiological Sensors," *2018 International Conference on Human-Computer Interaction*, Springer (IEEE co-sponsored), 2018.