ISSN: 3023 - 8544

ADBA

COMPUTER SCIENCE

VOLUME 1, ISSUE 1, JULY 2024

AN INTERDISCIPLINARY JOURNAL OF COMPUTER SCIENCE

https://journals.**adbascientific**.com/acs

ISSN: 3023 - 8544

ADBA Computer Science
Cutting-Edge Scientific Solutions

Volume: 1 – Issue No: 1 (July 2024)
https://journals.adbascientific.com/acs/issue/view/1

Editorial Board Members

ISSN: 3023 - 8544

Volume: 1 – Issue No: 1 (July 2024)
https://journals.adbascientific.com/acs/issue/view/1

Contents:

ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.1 / No.1 / 2024 / pp.1-7
https://doi.org/10.69882/adba.cs.2024071

Deep Learning in Agriculture: Detection and Analysis of Sugar Beets with YOLOv8

Cem Özkurt [*,α,1] **and Firdevs Süngü** [β,2]

*Department of Computer Engineering, Sakarya University of Applied Sciences, 54050, Sakarya, Turkiye, βDepartment of Electrical and Electronics Engineering, Sakarya University of Applied Sciences, Sakarya, Turkiye, αArtificial Intelligence and Data Science Application and Research Center, Sakarya University of Applied Sciences, 54050, Sakarya, Turkiye.

ABSTRACT In this study, the performance of the YOLOv8 model in detecting sugar beets was evaluated using images obtained from a drone over a sugar beet field. High-resolution drone images were divided into small segments, labeled, and the model was trained using data augmentation techniques. The results obtained during the training and testing phases demonstrated that the model successfully detected sugar beets with high accuracy, precision, recall, and F1 score values. The analysis of label correlograms and result graphs confirmed the model's labeling accuracy and detection capability. These findings indicate that the YOLOv8 model can be an effective tool in agricultural production monitoring and plant health assessment applications. In the future, the model's performance will be more comprehensively evaluated using datasets obtained from different geographical regions and various agricultural products.

KEYWORDS
Sugar beet detection
Drone images
Deep learning
YOLOv8
Agricultural monitoring

INTRODUCTION

Sugar beet is a significant crop worldwide and plays a crucial role in global food security and economy (Yalçınkaya *et al.* 2006). With its high sugar content, sugar beet is a vital component in sugar production, which is a fundamental food item in many households worldwide (Semerci 2016). The crop is cultivated in various parts of the world, including major producers like the United States, France, and Germany. Sugar beet cultivation is a complex process that requires careful planning, precise irrigation, and timely harvesting to ensure optimal yields (Yalçınkaya *et al.* 2006).

Sugar beet is also a significant crop in Turkey, especially in the eastern regions where the climate is more favorable for agriculture (Tursun 2016). The country has a long history of sugar beet production dating back to the early 20th century. Today, Turkey is one of the largest sugar beet producers globally, with a significant portion of its production coming from eastern provinces (Semerci 2016). The country's sugar beet industry is supported by a network of sugar factories, processing plants, and research institutions working together to increase yields, reduce costs, and improve overall efficiency.

Drone technologies offer revolutionary innovations in the agricultural sector and are used in areas such as monitoring plant health, managing irrigation, and increasing crop Drone technologies offer revolutionary innovations in the agricultural sector and are used in areas such as monitoring plant health, managing irrigation, and increasing crop productivity. For instance, aerial photography and multispectral imaging with drones enable farmers to analyze field conditions more quickly and accurately (Zhang and Kovacs 2012). These technologies offer time and cost savings in critical agricultural processes such as disease and pest detection, and unmanned aerial vehicles provide a much more effective solution for surveying large agricultural areas in a short time compared to traditional methods (Tsouros *et al.* 2019). The development of these technologies paves the way for more sustainable and efficient practices in agricultural activities. With the increasing use of unmanned aerial vehicles in agricultural production, studies on the integration and optimization of these technologies are gaining momentum (Bendig *et al.* 2013).

Artificial intelligence and image processing methods are fundamentally transforming data analysis and decision-making processes in the agricultural sector. Specifically, object detection and classification algorithms provide high accuracy in the analysis of agricultural images (Kamilaris and Prenafeta-Boldú 2018). Deep

Manuscript received: 13 June 2024,
Revised: 27 June 2024,
Accepted: 28 June 2024.

[1]cemozkurt@subu.edu.tr (**Corresponding author**)
[2]firdevs.sungu01@gmail.com

learning models such as YOLO (You Only Look Once) offer effective solutions for the automatic identification and monitoring of agricultural products (Redmon *et al.* 2016). These algorithms can be trained on large datasets and provide valuable insights into plant health and productivity (Chlingaryan *et al.* 2018). Additionally, AI-supported image processing techniques are used in the autonomous management of agricultural machinery and precision farming applications. The adoption of AI and image processing technologies in agriculture enhances the optimization and sustainability of production processes (Chlingaryan *et al.* 2018).

The journey from YOLO-v1 to YOLO-v8 showcases the continuous improvement and adaptability of these models. (Hussain 2023) discusses the progression and complementary nature of YOLO models, emphasizing their integration into digital manufacturing and defect detection. (Talaat and ZainEldin 2023) propose an enhanced fire detection approach for smart cities utilizing YOLO-v8, highlighting its efficacy in real-time scenarios. (Terven *et al.* 2023) provide a comprehensive review of YOLO architectures, noting the advancements up to YOLO-v8 and the introduction of YOLO-NAS, which further enhance performance and accuracy in computer vision tasks. Additionally, (Kim *et al.* 2023) demonstrate the application of YOLO-v8 in high-speed drone detection, underlining its capability in rapid and precise object identification. These studies collectively illustrate the versatility and robust performance of YOLO-v8 across diverse applications, marking it as a pivotal development in the field of computer vision.

Modern approaches in sugar beet production involve the integration of new technologies to increase efficiency and ensure environmental sustainability. Advanced agricultural machinery and sensor systems enable more efficient management of sugar beet fields (Hoffmann and Kenter 2018). These systems continuously monitor soil moisture, plant health, and growth rates, providing farmers with real-time data. This allows for the optimization of precision farming practices, fertilization, and irrigation processes, thereby minimizing environmental impacts (Weiss *et al.* 2020).Modern biotechnology methods also play a significant role in the development of disease-resistant and high-yielding sugar beet varieties. These innovative approaches contribute to increased sustainability and economic gains in sugar beet production (Kumar *et al.* 2016).

The use of drone and artificial intelligence technologies in agriculture has the potential to further improve agricultural production processes in the future (Kaya and Goraj 2020). AI algorithms, integrated with big data analytics, can provide decision support systems at every stage of agricultural production processes (Wolfert *et al.* 2017). These technologies will enhance the agricultural sector's ability to adapt to global challenges such as climate change and population growth (Rose *et al.* 2016). Additionally, data-sharing platforms and smart farming networks will facilitate farmers' access to information, contributing to the creation of a collective knowledge base. The widespread adoption of drone and AI technologies in agriculture will enable the development of more sustainable, efficient, and resilient agricultural systems in the future (Eastwood *et al.* 2019).

MATERIALS AND METHODS

Dataset and Resources

The dataset used in this study consists of high-resolution drone images of sugar beet fields obtained from the internet. The images contain sugar beet plants in the green leafy growth stage. To ensure the accuracy and diversity of the images, the dataset, comprising 271 images, was divided into small segments and augmented using various data augmentation techniques. Each image segment was cropped to include sugar beet plants prominently. Subsequently, the images were manually labeled. The labeling process was carried out carefully and meticulously to provide accurate data and enhance the training performance of the model.

The artificial intelligence model was developed following the "Machine Learning Lifecycle" depicted below and in Figure 1.

Figure 1 Machine Learning Lifecycle

Artifical Intelligence

With the advancement of technology today, artificial intelligence, a subject that continues to evolve, made its debut during a meeting in 1956, introduced by John McCarthy (Yılmaz *et al.* 2020). Artificial learning entails the ability of a computer or a machine under computer control to make decisions using mechanisms resembling those of living beings that can learn (Özel, M. A. and Baysal, S. S. and Şahin, M. 2021). In short, Artificial learning (AI) aims to replace human intelligence with machine intelligence (Munakata 1998). Artificial learning systems are those that interpret complex data through various methods to make it more understandable and improve themselves based on the experiences they gain (Aksoy *et al.* 2021).

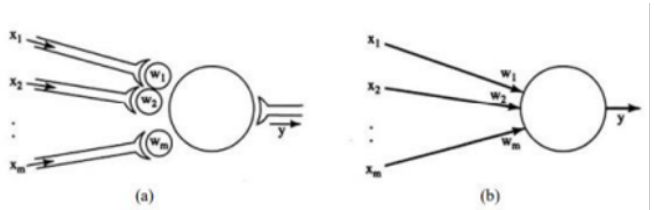

Figure 2 (a) A neuron model preserving the natural neuron image. (b) Another representation of the model (Munakata 1998)

A biological neuron is the fundamental building block of the nervous system. Its main function is to facilitate the transmission of information. It receives, transmits, and responds to stimuli. The artificial neuron shares similarities with it, consisting of structures such as axon, synapse, dendrite, myelin sheath, and nucleus. After defining the neural network architecture, the network enters the training phase. In this stage, the network learns by iteratively adjusting the weights of its connections based on provided examples (Munakata 1998).

Artifical Neural Networks

Artificial neural networks gained recognition through a study conducted by Warren McCulloch and Walter Pitts in 1943. They belong to the subset of artificial intelligence. The mathematical modeling of the neural structure of the human brain, for learning from experiences and remembering methods, is referred to as artificial neural networks. The aim is to model the neuron network of the brain to transfer the learning and decision-making process of the human brain to the computer environment. A neural network (NN) is an abstract computer example of the human brain (Munakata 1998).

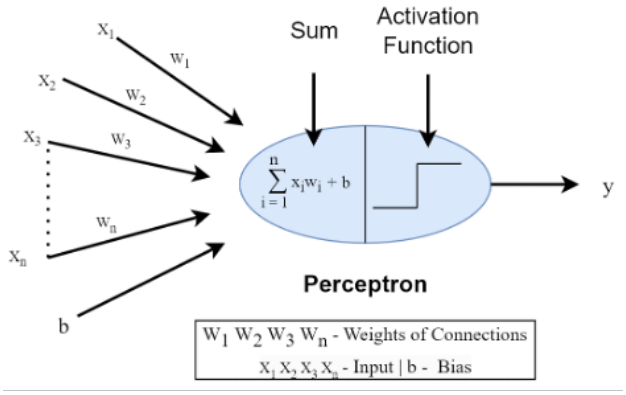

Figure 3 Mathematical model of a neuron (Tan *et al.* 2021)

Artificial neural networks are composed of artificial neurons. They have five basic components: inputs, weights, summation function, activation function, and outputs. Single-layer neural networks consist of input and output layers. These layers are generally used to solve linear problems. There can be one or more neurons in the layers (Yılmaz *et al.* 2020).

Machine Learning

In 1950, Alan Turing anticipated the development of the concept of machine learning and its future impact. Machine learning, a method used in artificial intelligence studies, is considered a subset of artificial intelligence. Deep learning is also a subset of machine learning. The relationship between artificial intelligence, machine learning, and deep learning is shown in Figure 4 (Tan *et al.* 2021).

The manual processing and analysis of very large datasets are not feasible. To address these problems, Machine Learning (ML) methods have been developed. Machine learning is the general term for computer algorithms that model a problem based on the data specific to that problem. The model created with the available dataset and the algorithm used are designed to perform optimally (Atalay and Çelik 2017).

Deep Learning

Following AlexNet's victory in the ImageNet competition in 2012, deep learning models began to be used in subsequent competitions. Deep learning is a subclass of machine learning with one or more hidden layers to gradually extract high-level features from raw data (Kazanç *et al.* 2021).

Deep learning can successfully analyze large datasets and can be applied to any field where data is available (Tan *et al.* 2021). The widespread success of deep learning is attributed to its method of computing outputs. A significant advantage of deep learning compared to traditional techniques is that it does not require an explicit feature extraction stage (Bozkurt 2021).

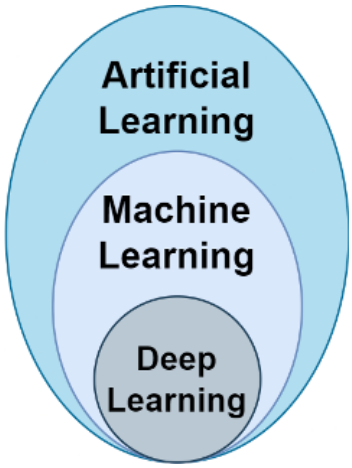

Figure 4 Artificial Intelligence Architecture (Arslan 2021)

Due to advancements in hardware, there has been an increased focus on deep learning studies, which has in turn improved object detection success rates. R-CNN, Faster R-CNN, Single Shot Detector (SSD), and YOLO are some of the deep learning-based object detection methods.

Among these methods, the YOLO algorithm and the DarkNet model offer high processing speed and accuracy. Experiments were conducted for four different versions of the algorithm, and the results were compared. The best results in terms of detection accuracy and speed were achieved with Version-4 algorithm. The success of deep learning methods has been proven in ImageNet classification competitions (Seçkin 2021).

Training and Optimization of YOLOv8 Model

YOLOv8 is a model developed for real-time object detection and offers significant improvements over its previous versions. One of the biggest advantages of YOLOv8 is its ability to provide high accuracy at high speed (Redmon *et al.* 2016). The model has been optimized for sugar beet detection and trained on the dataset prepared for this study. YOLOv8 has the ability to detect objects in a single network without incorporating complex components like region proposal networks (Bochkovskiy *et al.* 2020).

Various data augmentation techniques were used during the model training process. Images were processed with techniques such as rotation, scaling, brightness, and contrast adjustments. These techniques were used to improve the model's generalization ability. The hyperparameters of YOLOv8 were optimized during the training process; these hyperparameters include factors such as learning rate, batch size, and number of epochs. During training, the performance of the model on training and validation sets was monitored, and necessary adjustments were made. The loss function was carefully selected to improve the model's accuracy. The loss function of YOLOv8 focuses on minimizing classification and localization errors.

Additionally, the architecture of the model has been optimized for both speed and accuracy. YOLOv8 can provide fast results even on large datasets with efficient memory usage and computational requirements (Sokolova and Lapalme 2009). The output layers of the model provide class predictions and bounding box coordinates for each object. In this study, the performance of YOLOv8 was evaluated using metrics such as accuracy, error rate, precision, recall, and F1 score. The results showed that the model achieved high accuracy and efficiency in sugar beet detection.

Evaluation Metrics

The performance of the model was evaluated using metrics such as accuracy, loss, precision, recall, F1 score, and mean Average Precision (mAP).

Accuracy Accuracy represents the ratio of correct predictions made by the model to the total predictions. In a classification problem, accuracy is calculated as the ratio of correctly classified examples to the total examples.

TP (True Positives): Correctly predicted positive instances. TN (True Negatives): Correctly predicted negative instances. FP (False Positives): Incorrectly predicted positive instances. FN (False Negatives): Incorrectly predicted negative instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision Precision is a metric that represents the ratio of correct detections made by the model to the total detections. This metric is particularly important to assess the impact of false positives. A high precision value indicates that the majority of detections made by the model are correct. It measures the accuracy of the positive predictions made by the model and is calculated using the following formula:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall Recall, also known as sensitivity or true positive rate, measures the ratio of correct detections made by the model to the total number of actual objects. This metric is particularly important to assess the impact of missed positives (false negatives). A high recall value indicates that the model successfully detects all available sugar beets. Recall measures how well the model detects all sugar beets and is calculated using the following formula:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1 Score F1 score represents the harmonic mean of precision and recall, summarizing the overall performance of the model. This metric balances precision and recall, providing a single value to evaluate the model's performance. It is calculated using the following formula:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

The F1 score is an important metric, especially in imbalanced datasets, because it considers both correct detections and missed detections.

Mean Average Precision (mAP) mAP measures the average accuracy performance of the model across all classes. This metric is obtained by averaging the Average Precision (AP) values calculated for each class. mAP represents the overall detection performance of the model and is calculated using the following formula:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{5}$$

Here, N represents the total number of classes, and AP_i represents the Average Precision value calculated for each class. mAP is an important metric when evaluating the overall performance of the model because it considers the performance across all classes.

RESULTS

This study evaluated the performance of the YOLOv8 model for sugar beet detection, and the results were promising. Various data augmentation techniques were employed during the model training to increase the diversity of the dataset and enhance the model's generalization ability. After applying these augmentation techniques, the dataset expanded to 1355 images. The F1 score graph obtained after the training process demonstrates that the model exhibits high performance in terms of accuracy and precision. In the F1 score graph, it is evident that the accuracy improves and errors decrease as the training progresses.

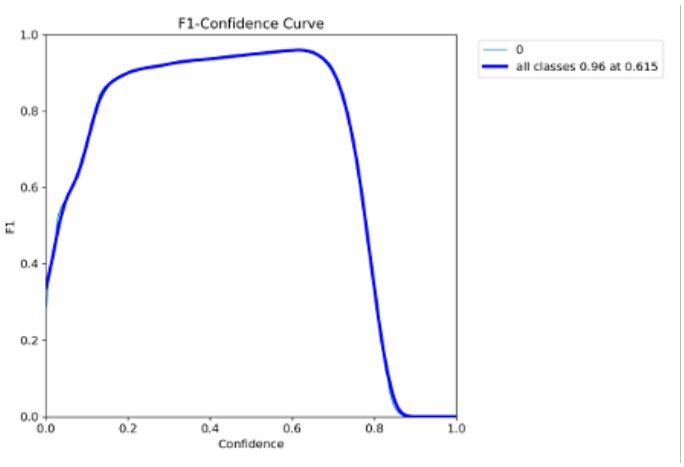

Figure 5 F1 Score Graph

IN-TEXT CITATIONS

A labels correlogram image was used to analyze the correlation between labels and the model's labeling accuracy. This analysis confirms that the model consistently produces accurate labeling. The correlogram shows the relationship between each label and other labels, as well as how accurately the model detects each label. This demonstrates how well the model distinguishes between similar-looking objects and how precise it is.

Additionally, as is seen from in the result graph the sugar beet plants detected by the model are accurately identified, and the bounding boxes are correctly placed.

The images used in the testing process were selected to evaluate how the model would perform in real-world applications. The analysis of the test images shows that the model can successfully detect sugar beet plants. The number and locations of sugar beet plants detected by the model were verified by comparing them with ground truth values. These test images demonstrate the practical application potential of the model in the field.

As a result, it has been observed that the YOLOv8 model provides 96.3% accuracy and efficiency in sugar beet detection. The model has yielded successful results in both the training and testing phases. The findings of this study may contribute to productivity and plant health monitoring efforts in sugar beet fields. In the future, it is planned to test the model on larger and more diverse datasets and adapt it to different agricultural products.

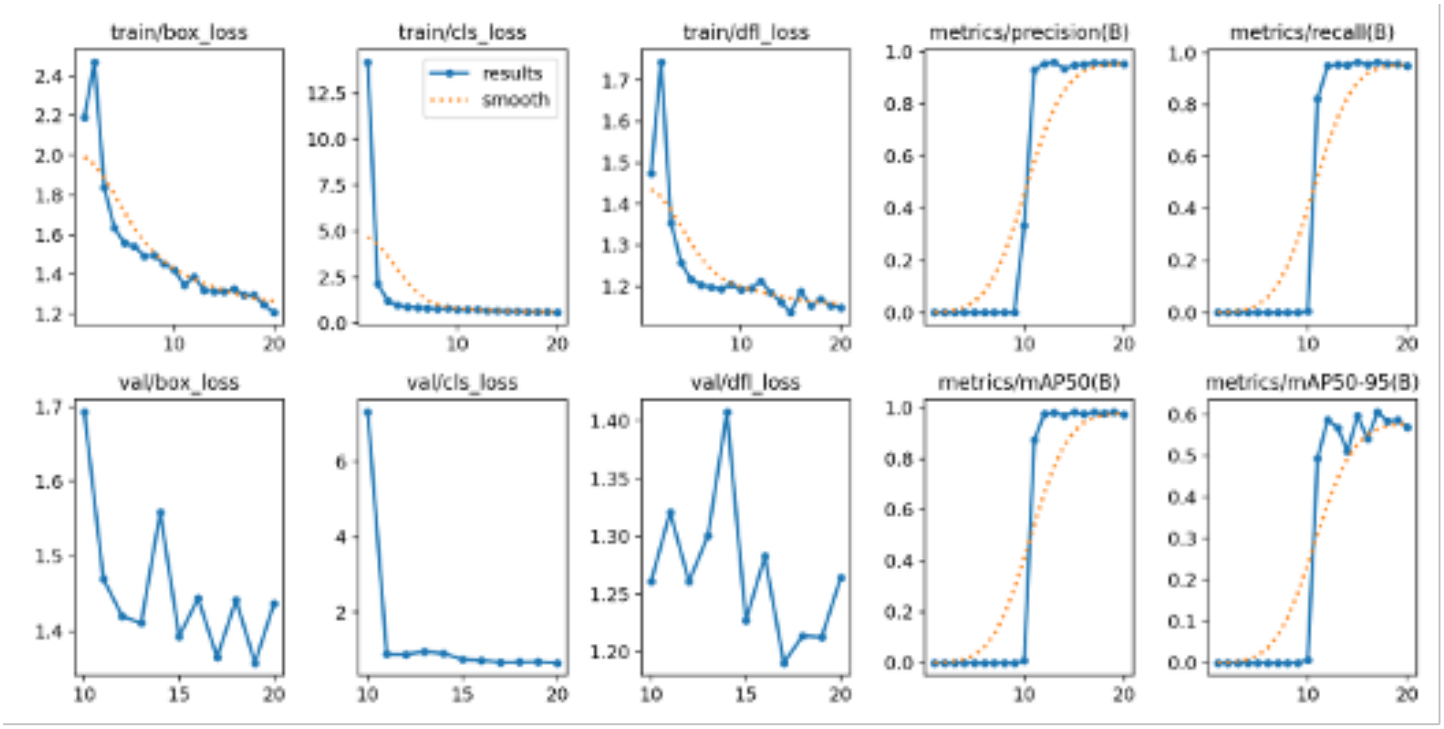

Figure 6 Training and Validation Result Graphs

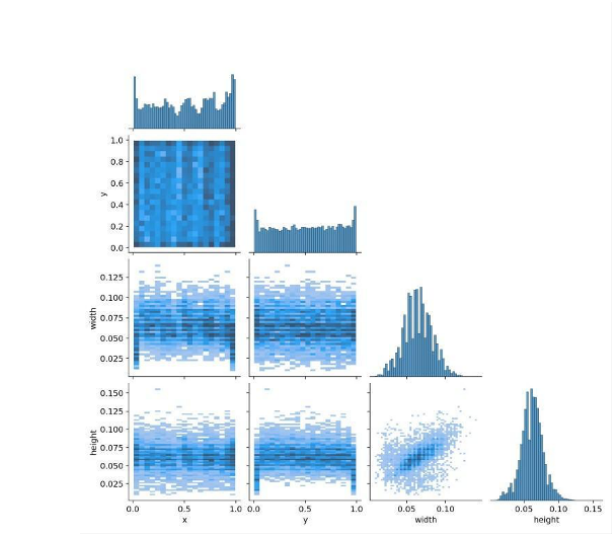

Figure 7 Result of Labels Correlogram

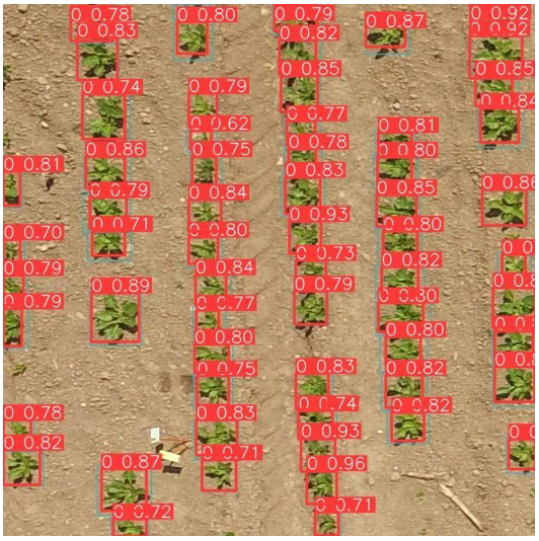

Figure 8 Test Image

DISCUSSION

This study aimed to evaluate the effectiveness of the YOLOv8 model in detecting sugar beets using drone imagery. The results obtained demonstrate that the model can accurately and precisely detect sugar beets. The data augmentation techniques employed during the model's training process have increased the diversity of the dataset and enhanced the model's generalization capability. This has enabled the model to perform successfully not only in specific environments but also in different environmental conditions.

PERFORMANCE METRICS AND EVALUATION OF RESULTS

The evaluation metrics of the model include various criteria such as accuracy, precision, recall, F1 score, and mAP. The result of the performance metrics can be seen in Table 1. Precision measures the ratio of correct detections made by the model, while recall evaluates how well the model can detect all true sugar beet plants [30]. The obtained high precision and recall values indicate that the model minimizes both false positives and false negatives. The F1 score summarizes the overall performance of the model by providing a balanced combination of these two metrics.

Table 1 Performance Metrics

Accuracy	F1	Recall	mAP
96.3%	96%	94.9%	97.4%

The obtained F1 score graph illustrates how the accuracy and error rates of the model improved over time during the training process. It's observed that the model's accuracy increased and errors decreased as the training progressed. This indicates the model's learning capacity and its ability to adapt to the dataset. Additionally, the labels correlogram image allows us to analyze the labeling accuracy and correlation between labels. This analysis confirms that the model produces consistent and accurate labeling.

REAL-WORLD APPLICATIONS OF THE MODEL

The images used during the testing phase were selected to simulate real-world conditions. The analysis of these test images demonstrates that the model can successfully detect sugar beets. This finding indicates that the model can be practically used in agricultural applications. Particularly, such a model is believed to have significant potential for monitoring field productivity and assessing plant health.

LIMITATIONS AND FUTURE WORK

This study has several limitations. Firstly, the dataset used consists of images obtained from a single field. Evaluating the model's performance with datasets obtained from different geographical regions and varying climate conditions is essential for generalizability. Additionally, exploring the applicability of the model to other agricultural products could be an important research topic for future studies.

In the future, the model is planned to be tested on larger and more diverse datasets. Additionally, the aim is to further enhance the model's performance by exploring different deep learning models and data augmentation techniques. Such studies could provide more effective and efficient solutions for monitoring and managing agricultural production.

CONCLUSION

This study has demonstrated that the YOLOv8 model provides high accuracy and efficiency in sugar beet detection. The model has shown successful results in both the training and testing phases. The findings obtained can contribute significantly to productivity and plant health monitoring in agricultural production. Such deep learning models offer significant potential for digital transformation and smart farming applications in the agricultural sector.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Aksoy, B., K. Korucu, Çalışkan, Osmanbey, and H. D. Halis, 2021 İnsansız hava aracı ile görüntü İşleme ve yapay zekâ teknikleri kullanılarak yangın tespiti: Örnek bir uygulama. Düzce Üniversitesi Bilim ve Teknoloji Dergisi **9**: 112–122.

Arslan, E., 2021 *Evrişimli Sinir Ağı Özelliklerine Dayanan Korelasyon Filtreleme ve Veri İlişkilendirme ile Çoklu Nesne Takibi*. Master's thesis, Bursa Uludag University (Turkey).

Atalay, M. and E. Çelik, 2017 Büyük veri analizinde yapay zekâ ve makine Öğrenmesi uygulamaları - artificial intelligence and machine learning applications in big data analysis. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi pp. 155–172.

Bendig, J., A. Bolten, and G. Bareth, 2013 Uav-based imaging for multi-temporal, very high resolution crop surface models to monitor crop growth variability. Photogrammetrie, Fernerkundung, Geoinformation **2013**: 551–562.

Bochkovskiy, A., C.-Y. Wang, and H.-Y. M. Liao, 2020 Yolov4: Optimal speed and accuracy of object detection Available: http://arxiv.org/abs/2004.10934.

Bozkurt, F., 2021 Derin Öğrenme tekniklerini kullanarak akciğer x-ray görüntülerinden covid-19 tespiti. European Journal of Science and Technology pp. 149–156.

Chlingaryan, A., S. Sukkarieh, and B. Whelan, 2018 Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. Computers and Electronics in Agriculture **151**: 61–69.

Eastwood, C., L. Klerkx, M. Ayre, and B. Dela Rue, 2019 Managing socio-ethical challenges in the development of smart farming: From a fragmented to a comprehensive approach for responsible research and innovation. Journal of Agricultural and Environmental Ethics **32**: 741–768.

Hoffmann, C. M. and C. Kenter, 2018 Yield potential of sugar beet – have we hit the ceiling? Frontiers in Plant Science **9**: 1–6.

Hussain, M., 2023 Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. Machines **11**: 677.

Kamilaris, A. and F. X. Prenafeta-Boldú, 2018 Deep learning in agriculture: A survey. Computers and Electronics in Agriculture **147**: 70–90.

Kaya, S. and Z. Goraj, 2020 The use of drones in agricultural production. International Journal of Innovative Approaches in Agricultural Research **4**: 166–176.

Kazanç, M., T. Ensari, and M. Dağtekin, 2021 Videoların derin Öğrenme ile sınıflandırılarak filtrelenmesi. European Journal of Science and Technology pp. 338–342.

Kim, J. H., N. Kim, and C. S. Won, 2023 High-speed drone detection based on yolo-v8. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–2, IEEE.

Kumar, V., M. Baweja, P. K. Singh, and P. Shukla, 2016 Recent developments in systems biology and metabolic engineering of plant–microbe interactions. Frontiers in Plant Science **7**: 1–12.

Munakata, T., 1998 *Fundamentals of the New Artificial Intelligence*, volume 2. Springer, New York.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi, 2016 You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788.

Rose, D. C. *et al.*, 2016 Decision support tools for agriculture: Towards effective design and delivery. Agricultural Systems **149**: 165–174.

Semerci, A., 2016 Tarımsal verimlilik göstergeleriyle avrupa birliği-türkiye tarımı. Journal of Agricultural Faculty of Gaziosmanpasa University **33**: 203–203.

Seçkin, M. E., 2021 *Derin Öğrenme Kullanılarak Trafik Koşullarına Uygun Otonom Araç Uygulaması*. Master's thesis, Bursa Uludag University (Turkey).

Sokolova, M. and G. Lapalme, 2009 A systematic analysis of performance measures for classification tasks. Information Processing and Management **45**: 427–437.

Talaat, F. M. and H. ZainEldin, 2023 An improved fire detection approach based on yolo-v8 for smart cities. Neural Computing and Applications **35**: 20939–20954.

Tan, F. G., A. S. Yüksel, E. Aydemir, and M. Ersoy, 2021 Derin Öğrenme teknikleri ile nesne tespiti ve takibi Üzerine bir İnceleme. European Journal of Science and Technology pp. 159–171.

Terven, J., D. M. Córdova-Esparza, and J. A. Romero-González, 2023 A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine Learning and Knowledge Extraction **5**: 1680–1716.

Tsouros, D. C., S. Bibi, and P. G. Sarigiannidis, 2019 A review on uav-based applications for precision agriculture. Information **10**.

Tursun, N., 2016 Kahramanmaraş İli ve İlçelerinde Şekerpancarı ekim alanlarında sorun olan yabancı otların belirlenmesi June.

Weiss, M., F. Jacob, and G. Duveiller, 2020 Remote sensing for agricultural applications: A meta-review. Remote Sensing of Environment **236**: 0–39.

Wolfert, S., L. Ge, C. Verdouw, and M. J. Bogaardt, 2017 Big data in smart farming – a review. Agricultural Systems **153**: 69–80.

Yalçınkaya, N., M. H. Yalçınkaya, and C. Çılbant, 2006 Avrupa birliği'ne yönelik düzenlemeler Çerçevesinde türk tarım politikaları ve sektörün geleceği Üzerine etkisi. Yönetim ve Ekonomi **13**: 98–118, Ticaret Borsası Genel Sekreteri, MANSA Dr. Coşkun ÇILBANT.

Yılmaz, O., H. Aydın, and A. Çetinkaya, 2020 Faster r-cnn Üzerinde geliştirilen model ile object detection api Üzerinde doğruluk tahmini ve analizi. European Journal of Science and Technology pp. 783–795.

Zhang, C. and J. M. Kovacs, 2012 The application of small unmanned aerial systems for precision agriculture: A review. Precision Agriculture **13**: 693–712.

Özel, M. A. and Baysal, S. S. and Şahin, M., 2021 Derin Öğrenme algoritması (yolo) ile dinamik test süresince süspansiyon parçalarında Çatlak tespiti. European Journal of Science and Technology pp. 1–5.

How to cite this article: Ozkurt, C., and Sungu, F. Deep Learning in Agriculture: Detection and Analysis of Sugar Beets with YOLOv8. *ADBA Computer Science*, 1(1), 1-7, 2024.

ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.1 / No.1 / 2024 / pp.8-13
https://doi.org/10.69882/adba.cs.2024072

Evaluating the Effectiveness of Machine Learning Models in Predicting Student Academic Achievement

Emre Deniz *,[1]
*Department of Computer Engineering, Hitit University, 19030, Corum, Turkiye.

ABSTRACT This study evaluates the effectiveness of various machine learning models in predicting student academic achievement using a dataset of 1000 students. The data includes demographic, psychological, social, and institutional factors. Models such as Linear Regression, Decision Tree Regressor, Random Forest Regressor, K-Nearest Neighbors Regressor, Support Vector Regressor (SVR), Gradient Boosting Regressor (GBR), XGBoost Regressor, and Neural Network (MLP) were employed. Results show that test preparation courses significantly enhance student performance, with SVR and Linear Regression models demonstrating the best predictive performance. The study highlights the importance of optimized educational strategies to enhance academic outcomes.

KEYWORDS
Academic achievement
Linear regression
Machine learning
Student performance

INTRODUCTION

Factors influencing student performance are multifaceted and encompass a wide array of determinants. Demographic factors such as gender, ethnicity, and parental education level have been recognized as crucial influencers of academic success (Korantwi-Barimah *et al.* 2017; Cantekin 2020; Jones *et al.* 2012). Gender has been highlighted as a factor impacting student academic performance, with studies indicating variations in success levels between male and female students (Degé *et al.* 2014). Similarly, ethnicity plays a critical role in student academic success, with research emphasizing the influence of ethnic identity and parents' goals on students' academic achievements (Abbasi *et al.* 2019). Additionally, parental education level is linked to student performance, where higher parental education levels are generally associated with better academic outcomes (Tang 2011).

Psychological factors like achievement motivation, locus of control, and academic self-concept are also key determinants of academic success (Wenglinsky 1996; Bizuneh 2021). Motivation levels and beliefs about one's abilities significantly affect academic performance, with high achievement motivation correlating with greater academic success (Hosova and Duchovicova 2019). Locus of control, representing individuals' beliefs about control over their lives, is associated with academic achievement, where internal locus of control is linked to better performance (Iyengar *et al.* 2022). Academic self-concept, reflecting students' perceptions of their academic abilities, plays a pivotal role in shaping their academic outcomes (Corbière *et al.* 2006).

Social factors including parental involvement, peer influence, and socioeconomic status have been shown to influence student performance (Jaiswal and Choudhuri 2017; Marsh and Yeung 1997; Erkman *et al.* 2010). Parental engagement in education is consistently linked to increased academic success, with supportive family environments contributing positively to students' achievements. Peer influence can also impact student performance, with social networks and friendships influencing academic outcomes . Additionally, socioeconomic status is a significant predictor of academic success, with students from higher socioeconomic backgrounds generally achieving better educational outcomes .

In conclusion, student performance is influenced by a complex interplay of factors, encompassing individual characteristics like motivation and self-concept, social influences such as parental involvement and peer relationships, and broader institutional practices and educational environments. Understanding these multifaceted determinants is crucial for developing effective strategies to support student success and enhance academic achievement.

The main research question of this study is: "What are the key factors that influence student performance and how to determine the relative effects of these factors on student academic achievement?"

This research question aims to analyze the effects of demographic, psychological, social and institutional factors on student performance and determine the importance of these factors. In the realm of machine learning, various regression models have been

Manuscript received: 13 June 2024,
Revised: 28 June 2024,
Accepted: 28 June 2024.

[1]emredeniz@hitit.edu.tr (**Corresponding author**)

extensively studied and applied across different domains to predict outcomes and make informed decisions. Among the popular regression models are Linear Regression, Decision Tree Regressor, Random Forest Regressor, K-Nearest Neighbors Regressor, Support Vector Regressor (SVR), Gradient Boosting Regressor (GBR), XGBoost Regressor (XGB), and Neural Network (Multi-Layer Perceptron, MLP). These models have been employed in diverse fields such as healthcare, environmental science, physics, and more to address a wide range of prediction tasks (Hassanzadeh *et al.* 2022).

Linear Regression, a fundamental and widely used regression model, forms the basis for many predictive analytics tasks. It establishes a linear relationship between the input variables and the target variable, making it a simple yet effective tool for prediction. Decision Tree Regressor operates by recursively partitioning the data into subsets based on certain features, creating a tree-like structure to make predictions. Random Forest Regressor, an ensemble method built on decision trees, combines multiple trees to improve prediction accuracy and reduce overfitting (Azar *et al.* 2022). K-Nearest Neighbors Regressor predicts the target variable by considering the 'k' nearest data points in the feature space.

Support Vector Regressor (SVR) utilizes support vectors to find the optimal hyperplane that best separates the data points in a high-dimensional space. Gradient Boosting Regressor sequentially builds multiple weak learners to create a strong predictive model, minimizing errors at each step. XGBoost Regressor, known for its efficiency and performance, implements gradient-boosted decision trees and is considered a state-of-the-art model for structured data, often outperforming deep learning models in regression tasks (Ferreira *et al.* 2024).

In the context of specific applications, these regression models have been leveraged for various predictive tasks. For instance, in the medical field, machine learning models like Random Forest, Support Vector Machine (SVM), and XGBoost have been utilized for survival prediction in diseases such as ovarian cancer. These models play a crucial role in analyzing patient data and making informed decisions regarding treatment and prognosis (Fei *et al.* 2019). Moreover, in environmental science, regression models like Gradient Boosting Regressor, Linear Regression, K-Nearest Neighbors Regressor, Random Forest Regressor, and XGBoost have been employed to predict outcomes related to solar energy harvesting and air pollution forecasting. These models aid in optimizing processes, enhancing efficiency, and making data-driven decisions in environmental research (Gonçalves *et al.* 2023). Furthermore, in physics and material science, regression models such as XGBoost, Random Forest, and Support Vector Regression have been utilized for tasks like predicting reduction potentials for complexes and conducting single-molecule conductance measurements.

In the domain of public health, machine learning models have been instrumental in predicting outcomes related to pandemics like COVID-19. Regression models such as Linear Regression, Support Vector Machine Regressor, Random Forest Regressor, and XGBoost Regressor have been employed to forecast disease outbreaks, analyze mortality rates, and guide public health interventions. These models provide valuable insights for policymakers and healthcare professionals to make informed decisions and mitigate the impact of health crises (Belho and Rawat 2023). Overall, the diverse applications of regression models in various fields underscore their significance in predictive analytics, decision-making, and knowledge discovery. By leveraging the strengths of different regression algorithms, researchers and practitioners can extract valuable insights from data, optimize processes, and drive innovation across a wide range of domains.

MATERIALS AND METHODS

In this study, a data set containing performance data of 1000 students was used (SPSScientist 2018). The variables included in the data set are:

Gender: Female and male

Ethnicity: Group A, Group B, Group C, Group D, Group E

Parental Level of Education: Some high school, high school, some college, associate's degree, bachelor's degree, master's degree

Lunch Type: Free/reduced and standard

Test Preparation Course: None and completed

Course Scores: Mathematics, reading and writing scores (between 0-100)

In the data preprocessing stage, categorical variables were converted to numerical values and numerical variables were normalized. This study was carried out using exploratory data analysis, correlation analysis and various machine learning models (linear regression, decision trees, random forest, K-Nearest Neighbors, Support Vector Regressor, Gradient Boosting Regressor, XGBoost Regressor, Multi-Layer Perceptron).

Data Preprocessing

First of all, missing and incorrect data in the data set were checked. Categorical variables were converted to numerical values and numerical variables were normalized. These operations are important to make the data set suitable for machine learning models.

Figure 1 shows the detailed exploratory anaylsis of the data. The distributions of the variables in the data set were examined using histograms. Additionally, a correlation matrix was created to understand the linear relationships between variables.

Gender: The number of male and female students in the data set is almost equally distributed.

Ethnicity: Although there is no significant difference between ethnicity groups, Group C and Group D seem to have the most students.

Parental Level of Education: The majority of parents have received education up to undergraduate level.

Lunch Type: The majority of students receive standard lunch.

Test Preparation Course: The majority of students have not completed the test preparation course.

Mathematics, Reading and Writing Scores: These scores show a wide distribution and the density is concentrated around the average score.

The correlation matrix which showed in Figure 2 evaluates linear relationships between variables. The findings obtained in the correlation analysis are as follows:

Test Prep Course: Mathematics correlates positively with reading and writing scores. This shows that students who completed the test preparation course received higher scores.Math, Reading, and Writing Scores: There are strong positive correlations between these three scores. Students who perform well in one subject often perform well in other courses.

Machine Learning Models

In this study, various machine learning models were used to predict student performance. Training and performance evaluation of the models were performed by hyperparameter optimization using GridSearchCV. The models and hyperparameter settings used are:

Linear Regression Model: LinearRegression Hyperparameters: No hyperparameter tuning is done.

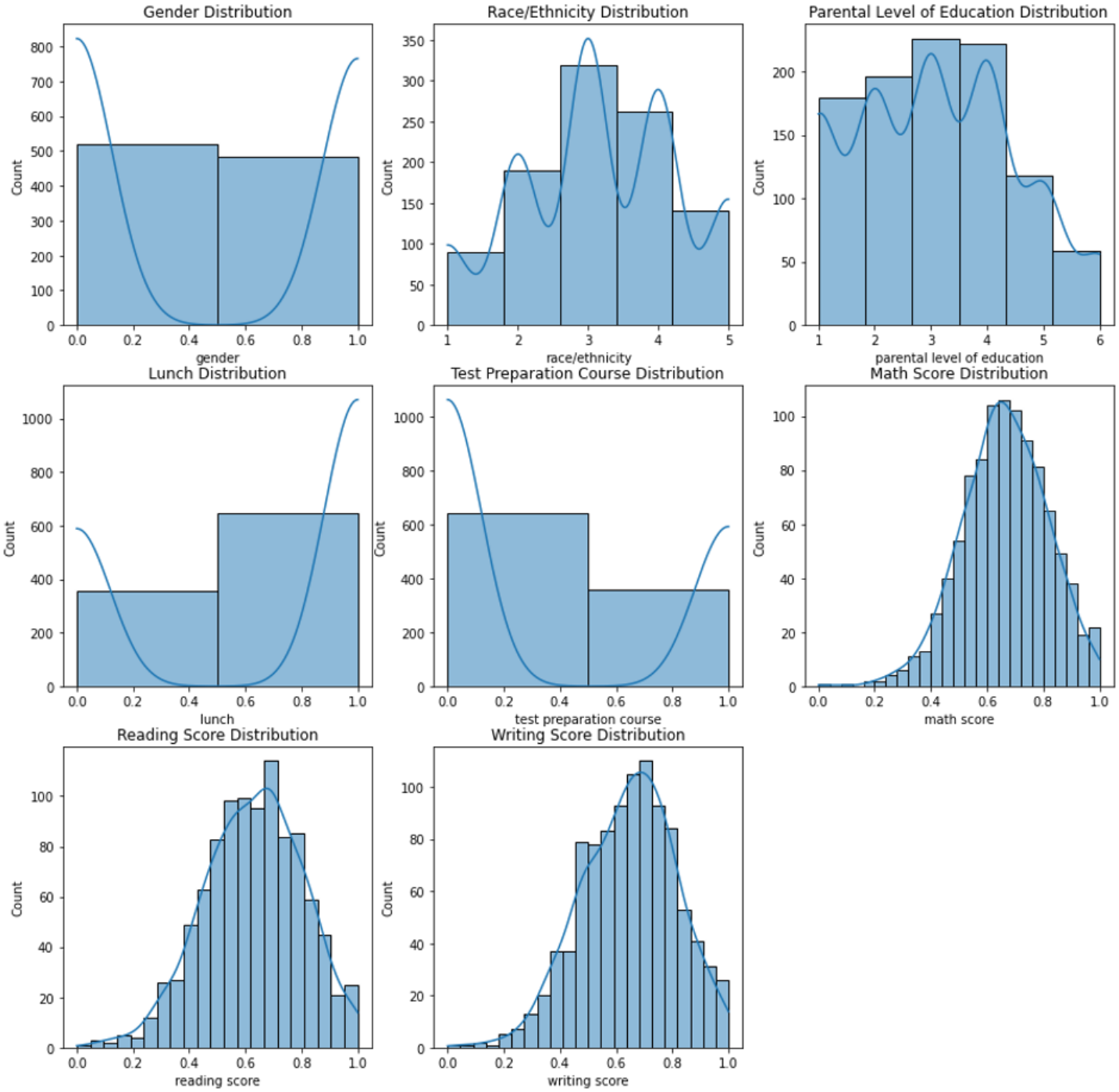

Figure 1 Exploratory Data Analysis

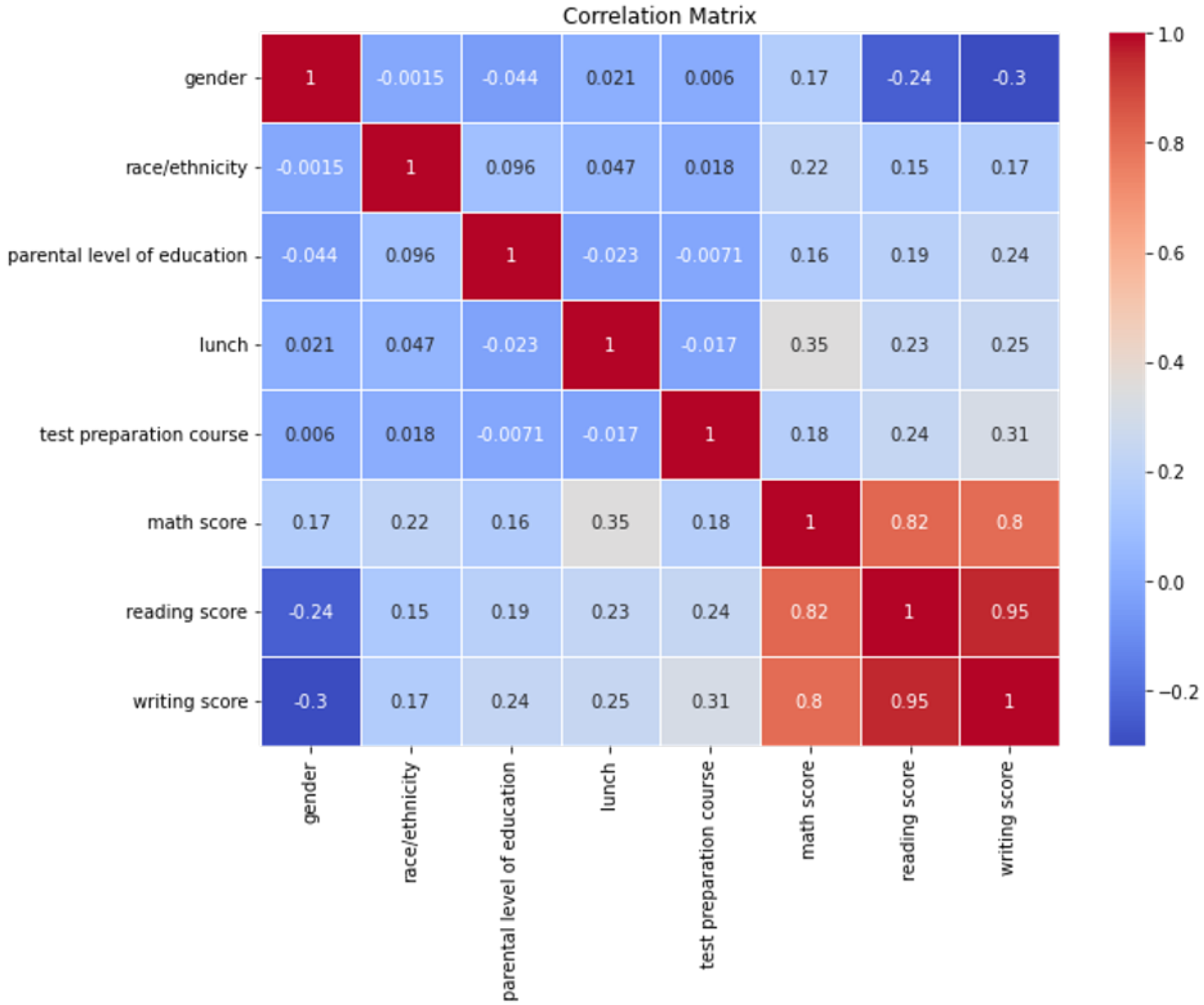

Figure 2 Correlation Matrix

Decision Tree Model: DecisionTreeRegressor Hyperparameters: 'max-depth': 10, 'min-samples-leaf': 4, 'min-samples-split': 10

Random Forest Model: RandomForestRegressor Hyperparameters: 'max-depth': 10, 'min-samples-leaf': 2, 'min-samples-split': 10, 'n-estimators': 200

K-Nearest Neighbors (KNN) Model: KNeighborsRegressor Hyperparameters: 'algorithm': 'brute', 'n-neighbors': 9, 'weights': 'distance'

Support Vector Regressor (SVR) Model: SVR Hyperparameters: 'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'

Gradient Boosting Regressor (GBR) Model: GradientBoostingRegressor Hyperparameters: 'learning-rate': 0.1, 'max-depth': 3, 'n-estimators': 100, 'subsample': 1.0

XGBoost Regressor (XGB) Model: XGBRegressor Hyperparameters: 'learning-rate': 0.1, 'max-depth': 3, 'n-estimators': 100

Neural Network (MLP) Model: MLPRegressor Hyperparameters: 'activation': 'tanh', 'alpha': 0.05, 'hidden-layer-sizes': (50, 100, 50), 'learning-rate': 'constant', 'solver': 'adam'

RESULTS

The performance of various machine learning models was evaluated using Mean Squared Error (MSE) and R-squared (R^2) metrics, as shown in Table 1. The Support Vector Regressor (SVR) and Linear Regression models demonstrated the best performance, with MSE values of 0.0028 and 0.0028 and R^2 values of 0.8866 and 0.8854 respectively. These models showed superior ability in predicting student performance accurately.

The strong performance of these models suggests that linear relationships among variables play a significant role in predicting academic achievement. Additionally, models like Random Forest and Gradient Boosting also showed high accuracy, indicating their robustness in handling complex data interactions. The results highlight the importance of optimizing test preparation strategies and suggest that focusing on interrelated academic subjects can lead to enhanced student performance.

Table 1 Results of Machine Learning Models

Model	Mean Squared Error (MSE)	R-squared (R^2)
Linear Regression	0.0028	0.8854
Decision Tree	0.0046	0.8116
Random Forest	0.0036	0.8533
K-Nearest Neighbors	0.0054	0.7798
Support Vector Regressor	0.0028	0.8866
Gradient Boosting Regressor	0.0030	0.8755
XGBoost Regressor	0.0032	0.8685
Neural Network (MLP)	0.0036	0.8512

CONCLUSION

This study has demonstrated the effectiveness of various machine learning models in predicting student academic achievement and highlighted the significant impact of test preparation courses on student performance. The findings indicate that demographic factors such as gender and ethnicity are not direct determinants of academic success, suggesting that educational policies should focus on enhancing educational experiences and preparation.

Educational institutions are recommended to prioritize test preparation courses and integrate data-driven approaches to identify and support students at risk of underperforming. Policies should be designed to foster interconnected learning across subjects to maximize student achievement. Future research should aim to validate these findings using larger and more diverse datasets and explore the long-term effects of different educational strategies.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Abbasi, H., V. Mehdinezhad, and M. Shirazi, 2019 Impact of jigsaw technique on improving university students' self-concept. Educational Research in Medical Sciences **8**.

Azar, A. S., S. B. Rikan, A. Naemi, J. B. Mohasefi, H. Pirnejad, *et al.*, 2022 Application of machine learning techniques for predicting survival in ovarian cancer. BMC Medical Informatics and Decision Making **22**.

Belho, K. and M. S. Rawat, 2023 Response of hydro-meteorological hazards to environmental degradation in kohima district of nagaland, north east india. International Journal of Scientific Research in Science, Engineering and Technology pp. 339–349.

Bizuneh, S. M., 2021 Belief in counselling effectiveness, academic self-concept as correlates of academic help-seeking behavior among college students. Journal of Education and Practice .

Cantekin, Ö. F., 2020 The effects of academic self-concept and organizational factors on academic achievement. Bartın University Journal of Faculty of Education **9**: 26–35.

Corbière, M., F. Fraccaroli, V. Mbékou, and J. Perron, 2006 Academic self-concept and academic interest measurement: A multisample european study. European Journal of Psychology of Education **21**: 3–15.

Degé, F., S. Wehrum, R. Stark, and G. Schwarzer, 2014 Music lessons and academic self-concept in 12- to 14-year-old children. Musicae Scientiae **18**: 203–215.

Erkman, F., A. Caner, H. Sakız, B. Börkan, and K. Şahan, 2010 Influence of perceived teacher acceptance, self-concept, and school attitude on the academic achievement of school-age children in turkey. Cross-Cultural Research **44**: 295–309.

Fei, X., Q. Zhang, and Q. Ling, 2019 Vehicle exhaust concentration estimation based on an improved stacking model. IEEE Access **7**: 179454–179463.

Ferreira, R. A. S., S. F. H. Correia, L. Fu, P. Georgieva, M. Antunes, *et al.*, 2024 Predicting the efficiency of luminescent solar concentrators for solar energy harvesting using machine learning. Scientific Reports **14**.

Gonçalves, D. M., R. Henriques, and R. S. Costa, 2023 Predicting metabolic fluxes from omics data via machine learning: Moving from knowledge-driven towards data-driven approaches. Computational and Structural Biotechnology Journal **21**: 4960–4973.

Hassanzadeh, H., J. Boyle, S. Khanna, B. Biki, and F. Syed, 2022 Daily surgery caseload prediction: Towards improving operating theatre efficiency. BMC Medical Informatics and Decision Making **22**.

Hosova, D. and J. Duchovicova, 2019 Gender differences in self-concept of gifted pupils. In *CBU International Conference Proceedings*, volume 7.

Iyengar, R. N., G. Gouri, M. Kumar, and Y. Yanjana, 2022 Academic self concept and academic achievement of indian cbse school students. National Journal of Community Medicine **12**: 405–410.

Jaiswal, S. K. and R. Choudhuri, 2017 Academic self concept and academic achievement of secondary school students. American Journal of Educational Research **5**: 1108–1113.

Jones, M. H., S. Audley, and S. M. Kiefer, 2012 Relationships among adolescents' perceptions of friends' behaviors, academic self-concept, and math performance. Journal of Educational Psychology **104**: 19–31.

Korantwi-Barimah, J. S., A. Ofori, E. Nsiah-Gyabaah, and A. M. Sekyere, 2017 Relationship between motivation, academic self-concept and academic achievement amongst students at a ghanaian technical university. International Journal of Human Resource Studies **7**.

Marsh, H. W. and A. S. Yeung, 1997 Causal effects of academic self-concept on academic achievement: Structural equation models of longitudinal data. Journal of Educational Psychology **89**: 41–54.

SPSScientist, 2018 Students performance in exams [data set].

Tang, S., 2011 The relationships of self-concept, academic achievement and future pathway of first year business studies diploma students. International Journal of Psychological Studies **3**.

Wenglinsky, H., 1996 Measuring self-concept and relating it to academic achievement: Statistical analyses of the marsh self-description questionnaire. ETS Research Report Series **1996**.

How to cite this article: Deniz, E. Evaluating the Effectiveness of Machine Learning Models in Predicting Student Academic Achievement. *ADBA Computer Science*, 1(1), 8-13, 2024.

ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.1 / No.1 / 2024 / pp.14-18
https://doi.org/10.69882/adba.cs.2024073

Enhancing Anomaly Detection in Large-Scale Log Data Using Machine Learning: A Comparative Study of SVM and KNN Algorithms with HDFS Dataset

Yusuf Alaca [*,1], **Erdal Başaran** [α,2] **and Yüksel Çelik** [β,3]

[*]Department of Computer Engineering, Hitit University, 19030, Corum, Turkiye, [α]Department of Computer Engineering, Agrı İbrahim Cecen University, Agrı, Turkiye, [β]Department of Computer Engineering, Karabük University, Karabük, Turkiye.

ABSTRACT As information technology rapidly advances, servers, mobile, and desktop applications are easily attacked due to their high value. Therefore, cyber attacks have raised great concerns in many areas. Anomaly detection plays a significant role in the field of cyber attacks, and log records, which record detailed system runtime information, have consequently become an important data analysis object. Traditional log anomaly detection relies on programmers manually inspecting logs through keyword searches and regular expression matching. While programmers can use intrusion detection systems to reduce their workload, log data is massive, attack types are diverse, and the advancement of hacking skills makes traditional detection inefficient. To improve traditional detection technology, many anomaly detection mechanisms, especially machine learning methods, have been proposed in recent years. In this study, an anomaly detection system using two different machine learning algorithms is proposed for large log data. Using Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) algorithms, experiments were conducted with the Hadoop Distributed File System (HDFS) log dataset, and experimental results show that this system provides higher detection accuracy and can detect unknown anomaly data.

KEYWORDS
Anomaly detection
KNN
SVM
Machine learning
HDFS

INTRODUCTION

In information technology infrastructures, many components and assets are interconnected and continuously interacting. Therefore, determining the cause of cyber attacks is challenging (A. Oliner and Xu 2012). Log records are considered a primary data source because they capture the runtime information of software (Sillito and Kutomi 2020). Detecting anomalies in log records is difficult due to several factors. The primary reasons include the rapidly increasing volume of logs (H. Mi and Cai 2013), the simultaneous generation of diverse log records (W. Xu and Jordan 2009), and changes in the nature of log recording due to software updates (Elbasani and Kim 2021).

In the existing literature, anomaly detection has been performed on various types of log records, including failure prediction and management (Tan and Gu 2010), RAS logs (Z. Zheng and Beckman 2010), health logs (Elbasani and Kim 2021), event logs (T. Pitakrat and Hoorn 2014), activity logs (H. Saadatfar and Deldari 2012), transactional and operational log records (T. Jia and Xu 2017), and more. Additionally, parsing log records has been achieved using frequency pattern mining (Vaarandi 2003), clustering (H. Hamooni and Mueen 2016), and natural language processing (NLP) techniques (X. Duan and Yin 2021).

In this study, the machine learning algorithms K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) are used for fast and effective anomaly detection. Analyses were conducted using the Hadoop Distributed File System (HDFS) dataset, which has been employed in numerous studies (M. Du and Srikumar 2017), achieving high success rates.

Manuscript received: 20 May 2024,
Revised: 27 June 2024,
Accepted: 28 June 2024.

[1]yusufalaca@hitit.edu.tr(**Corresponding author**)
[2]erdalbasaran@agri.edu.tr
[3]yukselcelik@ karabuk.edu.tr

DATASET DESCRIPTION AND PREPARATION

In this study, experiments were conducted using the HDFS dataset. This dataset has been labeled as normal and abnormal by Hadoop experts. Table 1 shows the time span, number of log lines, and the amount of labeled abnormal data in this dataset. The HDFS log dataset was collected from over 200 heterogeneous sources of Amazon and consists of 11,175,629 lines of log data. The HDFS log data records operations such as partitioning, replicating, and deleting within a specific block using block_id. This dataset comprises 575,061 log blocks with 16,838 labeled as abnormal by Hadoop experts (M. Du and Srikumar 2017).

The analysis of log data involves using numerical and categorical data as input, which requires the raw log data to be cleaned, sorted, and normalized. Figure 1 shows the log parsing steps. Each raw log entry consists of two parts: a timestamp and a complementary log part. The timestamp records the time of each log entry. Since timestamps in different formats are regular expressions, they can be easily extracted from raw log data during the log parsing stage. The log identifier is a token that identifies multiple processes or message exchanges within the system.

081109 203615 148 INFO dfs.DataNode$PacketResponder: PacketResponder 1 for block blk_38865049064139660 terminating

Date	081109
Time	203615
Level	INFO
Component	dfs.DataNode$PacketResponder
Event Template	PacketResponder 1 for block <*> terminating
Parameters	blk_38865049064139660

Figure 1 Steps of Log Parsing

After the log parsing steps, the data needs to be digitized. The word2vec (Church 2017) algorithm has been used to convert the textual parts of the log data into numerical values. The Mean/Mode method commonly used in the literature has been employed to address missing data, and to mitigate the impact of missing data, all missing values have been replaced with zero (Lin and Tsai 2020). Following digitization, anomaly labels generated by Hadoop experts have been appended to the end of the dataset. In the label column, 0 is used for normal data and 1 for abnormal data.

PROPOSED METHOD

Detecting anomalies in log analysis is quite challenging because log data consists of both numerical and categorical data. To enable the analysis of this data, it first undergoes preprocessing. Through log parsing, features are extracted from the dataset and transformed into a vectorized form. Subsequently, this vectorized dataset is analyzed using machine learning algorithms to detect anomalies.

Figure 2 illustrates the architecture of the proposed method. Particularly, the utilization of the word2vec algorithm for digitization during log parsing has had a significant impact on the high performance of experimental results. By employing this method, multiple machine learning algorithms have been utilized for anomaly detection from log records, resulting in high success rates. The

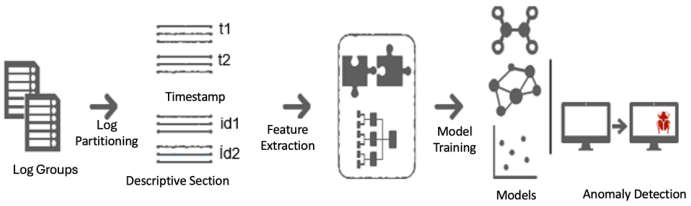

Figure 2 Architecture of the Proposed Anomaly Detection Method

log parsing process is crucial for using data in machine learning algorithms.

SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) are used for classification problems using supervised learning. Typically, they classify by drawing a line on a plane to maximize the distance between points of two classes (M. A. Hearst and Scholkopf 1998). The main objective of classification is to determine which class future data belongs to. In Figure 3, the data is divided into two classes, black and white. A line is drawn to separate these two classes, and the area between them is called the margin. The larger the margin, the better the two classes are separated. W denotes the weight vector, x denotes the input vector, and b denotes the bias. Using these values, the margin region remains between ±1.

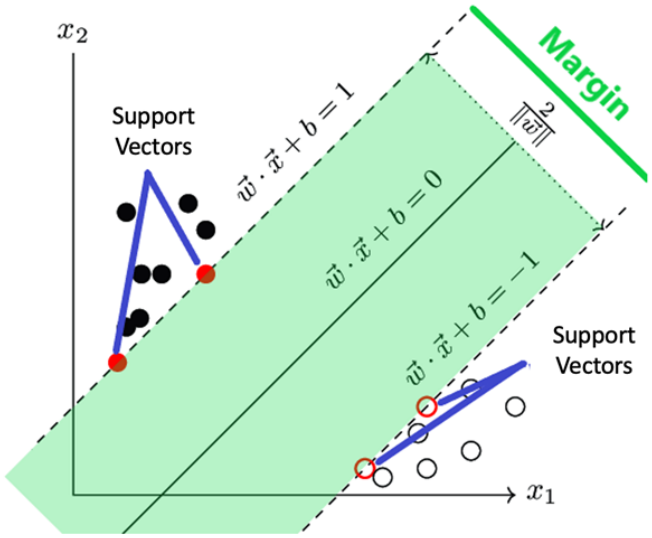

Figure 3 Working principle of Support Vector Machines

To classify low-dimensional data more efficiently, the kernel method is employed. This method expands the available data by multiplying it with kernel functions without increasing the dimensionality of the data, making it more meaningful (Steinwart and Christmann 2008). Two of the kernels used are the Polynomial and the Gaussian RBF cubic kernel. The Polynomial kernel enables processing of data from 2 dimensions to 3 or more dimensions (Moghaddam and Hamidzadeh 2016). It classifies by calculating the similarity of each point to a specific point using a normal distribution. The spread of the distribution is controlled by the gamma hyperparameter. A smaller gamma parameter leads to a wider distribution. To avoid overfitting, the gamma value should be reduced while for underfitting, it should be increased. In this

Table 1 Characteristics of the HDFS Log Dataset

Dataset	Duration	Number of Log Lines	Number of Anomalies (Blocks)
HDFS	38.7 hours	11,175,629	16,838

Table 2 Data with Missing Values Completed by Digitizing Using Word2Vec

Column 1	Column 2	Column 3	...	Column 23	Column 24	Labels
5	5	5	...	0	0	0
5	22	9	...	23	21	0
22	5	5	...	0	0	1
22	26	26	...	4	21	0
5	9	11	...	23	21	1
5	26	3	...	21	0	1

study, classification methods using normal distribution along with polynomial and cubic kernels were employed, resulting in a high success rate.

K NEAREST NEIGHBORS ALGORITHM

kNN is a supervised learning algorithm used for both classification and regression problems. It finds the k nearest neighbors to a new point and makes predictions based on those neighbors (G. Guo and Greer 2003; Ö. Tonkal and Kocaoğlu 2021). Three different distance calculation methods have been used in this study. The Euclidean distance is used to measure proximity in the kNN algorithm. Euclidean distance linearly measures the distance between two points. The calculation of Euclidean distance between points P=(x1, x2, ..., xn) and Q=(y1, y2, ..., yn) is given in Equation 1.

$$D_{PQ} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

The Minkowski distance is expressed with a general formula and is used to define various distance metrics for different values of p. It is a generalization of distance metrics such as the Euclidean distance commonly used in machine learning, clustering, and data mining applications. The Minkowski distance between any two points P and Q, where P=(x1, x2, ..., xn) and Q=(y1, y2, ..., yn), is calculated according to Equation 2.

$$D_{PQ} = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{2}$$

The Mahalanobis Distance is a distance measurement system used in computer science and many other fields. Its main difference from other measurement systems is that it performs distance separation on an elliptical plane. The Mahalanobis distance is calculated as the square root of the product of the difference between the value vector and the mean, the inverse of the covariance matrix, and the transpose of the difference between the value vector and the mean. Equation 3 illustrates the calculation of the Mahalanobis distance.

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{3}$$

PERFORMANCE METRICS FOR EVALUATING THE PROPOSED METHOD

In this study, the success of the proposed method was assessed using the following criteria sequentially. Accuracy and Precision measurements were conducted according to Equations 4 and 7, respectively. These equations utilize parameters such as TN (true negatives), TP (true positives), FN (false negatives), and FP (false positives). The F-Score derived from the cumulative sum of Accuracy and Precision was calculated in Equation 8. Additionally, Precision was computed in Equation 7, and Specificity was determined in Equation 6.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{F-Score} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{8}$$

Table 3 Classification Test Results

Classification Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F-Score (%)
SVM					
Linear	0.9794	0.9944	0.9644	0.9655	0.9797
Polynomial	0.9958	0.9978	0.9939	0.9939	0.9958
Cubic	0.9978	1	0.9956	0.9956	0.9978
kNN					
Euclidean	0.9694	0.9806	0.9583	0.9592	0.9698
Minkowski	0.9725	0.9856	0.9594	0.9605	0.9729
Mahalanobis	0.9761	0.9950	0.9572	0.9588	0.9766

CONCLUSION

Detecting anomalies from large log data is quite challenging. In this study, log parsing was conducted using word2vec on datasets containing both numerical and categorical data such as the HDFS dataset. Experimental test results have demonstrated high success using machine learning algorithms such as SVM and kNN. In the future, testing success results with different machine learning algorithms is planned.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

A. Oliner, A. G. and W. Xu, 2012 Advances and challenges in log analysis. Communications of the ACM **55**: 55–61.

Church, K. W., 2017 Word2Vec. Natural Language Engineering **23**: 155–162.

Elbasani, E. and J. D. Kim, 2021 LLAD: Life-Log Anomaly Detection Based on Recurrent Neural Network LSTM. Journal of Healthcare Engineering **2021**.

G. Guo, D. B. Y. B., H. Wang and K. Greer, 2003 KNN model-based approach in classification. In *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pp. 986–996, Springer.

H. Hamooni, J. X. H. Z.-G. J., B. Debnath and A. Mueen, 2016 Logmine: Fast pattern recognition for log analytics. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1573–1582, ACM.

H. Mi, Y. Z. M. R.-T. L., H. Wang and H. Cai, 2013 Toward fine-grained unsupervised scalable performance diagnosis for production cloud computing systems. IEEE Transactions on Parallel and Distributed Systems **24**: 1245–1255.

H. Saadatfar, H. F. and H. Deldari, 2012 Predicting job failures in AuverGrid based on workload log analysis. New Generation Computing **30**: 73–94.

Lin, W.-C. and C.-F. Tsai, 2020 Missing value imputation: a review and analysis of the literature (2006–2017). Artificial Intelligence Review **53**: 1487–1509.

M. A. Hearst, E. O.-J. P., S. T. Dumais and B. Scholkopf, 1998 Support vector machines. IEEE Intelligent Systems and their Applications **13**: 18–28.

M. Du, G. Z., F. Li and V. Srikumar, 2017 DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 1285–1298.

Moghaddam, V. H. and J. Hamidzadeh, 2016 New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier. Pattern Recognition **60**: 921–935.

Sillito, J. and E. Kutomi, 2020 Failures and Fixes: A Study of Software System Incident Response. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 185–195, IEEE.

Steinwart, I. and A. Christmann, 2008 *Support Vector Machines*. Springer Science & Business Media.

T. Jia, P. C. Y. L.-F. M., L. Yang and J. Xu, 2017 Logsed: Anomaly diagnosis through mining time-weighted control flow graph in logs. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pp. 447–455, IEEE.

T. Pitakrat, O. K. F. K., J. Grunert and A. V. Hoorn, 2014 A framework for system event classification and prediction by means of machine learning. In *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*, pp. 173–180, ACM.

Tan, Y. and X. Gu, 2010 On predictability of system anomalies in real world. In *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 133–140, IEEE.

Vaarandi, R., 2003 A data clustering algorithm for mining patterns from event logs. In *Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003)*, pp. 119–126, IEEE.

W. Xu, A. F. D. P., L. Huang and M. I. Jordan, 2009 Detecting large-scale system problems by mining console logs. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*,

pp. 117–132, ACM.
X. Duan, H. C. W. Y., S. Ying and X. Yin, 2021 OILog: An online incremental log keyword extraction approach based on MDP-LSTM neural network. Information Systems **95**: 101618.
Z. Zheng, R. G. S. C., Z. Lan and P. Beckman, 2010 A practical failure prediction with location and lead time for blue gene/p. In *2010 International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 15–22, IEEE.
Ö. Tonkal, E. B. Z. C., H. Polat and R. Kocaoğlu, 2021 Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking. Electronics **10**.

How to cite this article: Alaca, Y., Basaran, E., and Celik, Y. Enhancing Anomaly Detection in Large-Scale Log Data Using Machine Learning: A Comparative Study of SVM and KNN Algorithms with HDFS Dataset. *ADBA Computer Science*, 1(1), 14-18, 2024.

ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.1 / No.1 / 2024 / pp.19-25
https://doi.org/10.69882/adba.cs.2024074

Environmental Sustainability through AI: A Case Study on CO_2 Emission Prediction

Cem Özkurt [*,α,1]

*Department of Computer Engineering, Sakarya University of Applied Sciences, 54050, Sakarya, Turkiye, αArtificial Intelligence and Data Science Application and Research Center, Sakarya University of Applied Sciences, 54050, Sakarya, Turkiye.

ABSTRACT In this study, the Biogeography-Based Optimization (BBO) algorithm was effectively utilized to predict carbon dioxide (CO_2) emissions. In the context of combating global warming and climate change, making accurate and reliable CO_2 emission predictions is critically important for developing environmental policies and strategies. Accordingly, the motivation for our study is to contribute to environmental decision-making processes by improving the accuracy of CO_2 emission predictions. BBO is a nature-inspired optimization method used to analyze complex relationships and identify significant features within a dataset. The focus of the study is to accurately predict the "share_global_coal_co2" parameter, and for this purpose, the BBO algorithm was employed to identify the 20 most influential features. The analyses revealed that the Gradient Boosting algorithm provided the lowest Mean Squared Error (MSE) value of 0.347408, indicating that the model can make predictions closer to the actual data. Additionally, the use of interpretable artificial intelligence models such as SHAP and LIME made the model's predictions more understandable and clearly demonstrated the impact of specific features on the predictions. The results obtained provide significant guidance for environmental policymakers and energy experts. The effectiveness of the BBO algorithm in predicting CO_2 emissions can contribute to more informed and data-driven decisions in environmental analysis and policy-making processes. This study emphasizes the importance of artificial intelligence and optimization techniques in achieving sustainability goals and helps develop more effective strategies in environmental management.

KEYWORDS
CO_2 emissions
Biogeography-Based optimization (BBO)
Explainable AI
SHAP
LIME

INTRODUCTION

Addressing global environmental issues and climate change, the reduction of carbon dioxide (CO_2) emissions and CO_2 capture technologies play a crucial role. In this context, the Biogeography-Based Optimization (BBO) algorithm has been employed to estimate the "share_global_coal_co2" parameter. Inspired by natural biogeographic processes, BBO is an optimization algorithm used to analyze complex relationships and identify significant features within a dataset. This study utilizes a dataset comprising various parameters, including share global coal CO_2, share global cumulative gas CO_2, share global cumulative LUC CO_2, share global cumulative flaring CO_2, cumulative oil CO_2, share global gas CO_2, share global cumulative coal CO_2, cumulative CO_2 including LUC, total GHG, cumulative LUC CO_2, share global cumulative cement CO_2, CO_2 growth percentage, cumulative cement CO_2, CO_2 including LUC per GDP, oil CO_2, coal CO_2, temperature change from GHG, consumption CO_2 per GDP, share global LUC CO_2, cement CO_2, cumulative coal CO_2, primary energy consumption, other industry CO_2, share global cumulative oil CO_2, CO_2 per GDP, cumulative gasCO_2, temperature change from CO_2, nitrous oxide, share global cumulative CO_2 including LUC, and share global CO_2.

By applying the BBO algorithm, the study aims to reduce these parameters to the 20 most influential ones for accurately predicting the "share_global_coal_co2" parameter. Artificial intelligence techniques significantly contribute to reducing CO_2 emissions and developing sustainable energy solutions. For instance, Delanoë *et al.* (2023) evaluated the positive and negative impacts of AI mod-

Manuscript received: 16 May 2024,
Revised: 13 June 2024,
Accepted: 25 June 2024.

[1]cemozkurt@subu.edu.tr (**Corresponding author**)

els on CO_2 emissions reduction. In this study, three different AI models were utilized for energy demand management in Brazilian households, photovoltaic power forecasting in Tunisia, and the electric vehicle routing problem in Sweden and Luxembourg. The results indicated that AI models can achieve significant CO_2 reductions depending on the context. Yan *et al.* (2020) developed a hybrid artificial intelligence model to predict the physical and chemical changes in coal seams during CO_2 geological sequestration. This model, integrating back propagation neural network (BPNN), genetic algorithm (GA), and adaptive boosting algorithm (AdaBoost), was optimized to accurately predict coal strength alterations due to CO_2 adsorption. The study demonstrated that the hybrid model could effectively and accurately predict these changes. Qerimi and Sergi (2022) examined the legislative processes related to carbon capture and storage (CCS) technology. This study emphasized the importance of CCS and AI technologies in achieving climate goals and argued for the necessity of new regulations to govern the development, design, and deployment of such technologies.

Another study explored the use of artificial neural network (ANN) tools to enhance the efficiency of CO_2 storage projects by predicting critical performance indicators like methane recovery and CO_2 injection. The findings showed that ANN models could accurately predict performance in CO_2 storage projects. Thanh *et al.* (2022) used hybrid artificial intelligence models to predict the deliverability of underground natural gas storage sites. This study highlighted the importance of developing intelligent systems that can accurately predict natural gas storage deliverability in various geological formations. Stef *et al.* (2023) investigated the impact of high-quality institutional measures on global CO_2 emissions reduction.

The study revealed that effective climate change policies must be associated with improvements in property rights protection, citizen participation in elections and freedom of expression, and corruption control. Heo *et al.* (2022) developed an explainable artificial intelligence (XAI) model to create a net-zero carbon roadmap for the petrochemical industry. This model produced various scenarios of offshore wind power and conducted techno-economic and environmental assessments. The findings underscored the feasibility and effectiveness of AI-driven net-zero carbon solutions.

This literature review highlights the significance and application areas of artificial intelligence and optimization algorithms in reducing CO_2 emissions. By examining the effectiveness of the BBO algorithm in predicting the "share_global_coal_co2" parameter, this study aims to contribute to more informed and data-driven decision-making processes in environmental analysis and policy development.

MATERIALS AND METHODOLOGY

This dataset contains various climate variables, greenhouse gas emissions, and economic indicators. Compiled to examine global carbon emissions and the environmental impact of various human activities, this dataset consists of a total of 32 different parameters. Parameters like "share_global_coal_co2" represent coal-related carbon dioxide emissions, while others like "cumulative_oil_co2" and "cumulative_gas_co2" indicate emissions from oil and gas sources, respectively. Economic indicators such as "gdp" and "consumption_co2_per_gdp" can be used to analyze the relationship between economic growth and greenhouse gas emissions. This dataset can be utilized with various machine learning models to predict the "share_global_coal_co2" parameter and can play a significant role in issues such as energy policy development and climate change strategy determination.

Dataset

This dataset contains various climate variables, greenhouse gas emissions, and economic indicators. Compiled to examine global carbon emissions and the environmental impact of various human activities, this dataset consists of a total of 32 different parameters. Parameters like "share_global_coal_co2" represent coal-related carbon dioxide emissions, while others like "cumulative_oil_co2" and "cumulative_gas_co2" indicate emissions from oil and gas sources, respectively. Economic indicators such as "gdp" and "consumption_co2_per_gdp" can be used to analyze the relationship between economic growth and greenhouse gas emissions. This dataset can be utilized with various machine learning models to predict the "share_global_coal_co2" parameter and can play a significant role in issues such as energy policy development and climate change strategy determination.

Biogeography-based Optimization (BBO)

BBO is a natural optimization algorithm based on biogeography principles, inspired by natural processes modeling the dispersion and migration of biological species among habitats. This algorithm utilizes mathematical models representing the quality of habitats and migration rates between species to optimize the fitness values of the population. The fundamental working principle of BBO involves the movement of the best individuals among habitats to improve the fitness values of a population initially generated randomly. This process enhances and optimizes the fitness values of the population over time. BBO can be effective in complex and multi-dimensional optimization problems, although it may require appropriate parameter settings and modeling tailored to the problem context.

Machine Learning

Machine learning is a branch of artificial intelligence where computer systems have the ability to learn from data. These systems create models using data to perform specific tasks or solve problems, and they can analyze new data or make predictions using these models. Machine learning is data-driven as it relies on learning from experiential data. Fundamentally, it is a combination of disciplines such as data analysis, statistics, mathematics, and computer science. Machine learning algorithms are commonly used in various tasks such as classification, regression, clustering, dimensionality reduction, and pattern recognition. Examples include decision trees, support vector machines, gradient boosting machines, and deep learning networks. Machine learning has a wide range of applications across various industries and fields, including healthcare, finance, automotive, retail, and more. However, training these models requires careful management of factors such as proper hyperparameter tuning and data quality.

Artificial Neural Networks (ANNs) are a machine learning model that mimics the workings of the human brain and has been successfully used in many fields in recent years. This model enables information processing and learning by forming a network of neurons, the basic units of a neural network. Artificial neural networks have structures consisting of multiple layers; each layer receives inputs from the previous layer and processes them. These processes are typically carried out with non-linear activation functions. Artificial neural networks can handle a wide range of data but may require large amounts of data and have lengthy training times. However, using a subfield called deep learning, they can exhibit superior performance in large and complex datasets. One of the fundamental advantages of artificial neural networks is their ability to optimize learning capabilities with various architectures and

hyperparameters. However, it's important to deal with issues such as overfitting and ensure good generalization to data outside the training set.

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

In Equation 1, "z" represents the value of the objective function, while "w_i" and "x_i" represent the components of the weight and input data, respectively. "b" is a constant term. The summation calculates the value of this objective function by taking a combination of weights and inputs over a specific dataset. Thus, the BBO algorithm attempts to find the best solution in a particular problem by optimizing this equation.

XGBoost Recently, XGBoost (eXtreme Gradient Boosting) has become increasingly popular for classification and regression problems, especially for structured data, yielding effective results. This machine learning algorithm constructs a strong predictor by combining many weak predictors, often referred to as decision trees. Using a technique called gradient boosting, at each step of the model, a new predictor is added to minimize the loss (error). XGBoost applies this gradient descent to decision trees and sequentially adds weak predictors. Thus, the model learns increasingly complex relationships and makes more accurate predictions (Chen and Guestrin 2016). One significant advantage of XGBoost is its ability to be optimized for different datasets and problems by adjusting its hyperparameters. Additionally, it can work quickly and handle large datasets well. However, its high performance and flexibility come at a cost. As the complexity of the model increases, both training and prediction times may increase. Therefore, it's essential to consider factors like computational resources and hyperparameter optimization when using XGBoost.

$$F(x) = L(\theta) + \Omega(\theta) \tag{2}$$

In Equation 2, "$F(x)$" represents the predicted value of the target variable, "$L(\theta)$" represents the loss function, and "$\Omega(\theta)$" represents the regularization term. Essentially, the XGBoost algorithm adds weak predictors sequentially by minimizing the residuals from the previous model's predictions. In this equation, "$L(\theta)$" calculates the error between the predicted and actual values of the features, and "$\Omega(\theta)$" is the regularization term that limits the complexity of the model. Thus, the XGBoost algorithm aims to optimize prediction performance by creating the most suitable model through the combination of the loss function and regularization term.

$$L(\theta) = \sum_{i=1}^{n} -(y_i \log\log(\hat{y}_i) + (1 - y_i) \log\log(\hat{y}_i)) \tag{3}$$

In Equation 3, the expression "$L(\theta)$" represents a loss function that measures how far the model's predictions are from the true labels. θ represents the model parameters. In the equation, "y_i" symbolizes the true label values, and "$\hat{y}_i$" represents the predictions made by the model. This loss function, used for binary classification problems, calculates the negative log probability sum over the predicted class probabilities for each data point's true class labels. Thus, the algorithm aims to learn the best model parameters by minimizing this loss function and aims to increase classification accuracy.

$$\Omega(\theta) = \gamma T + \frac{\lambda}{2} \sum_{j=1}^{T} w_j^2 \tag{4}$$

In Equation 4, the expression "$\Omega(\theta)$" represents a regularization term that limits the complexity of the model. γ and λ are regularization parameters and regularization coefficients, respectively. In the equation, "T" represents the number of trees, and "w_j" represents the weights of each tree. This regularization term is used to reduce the tendency of the model to overfit and improve its generalization ability. The first term controls the number of trees and their complexity, while the second term regulates the complexity of the trees by the square of their weights. This regularization term aims to prevent overfitting by helping the model become simpler and more generalizable. Thus, the XGBoost algorithm uses this regularization term to control the complexity of the model while minimizing the loss function.

LightGBM is a machine learning algorithm that has gained popularity recently, particularly for working effectively with large datasets. This algorithm constructs decision trees using the gradient boosting method, but unlike other traditional gradient boosting-based methods, LightGBM builds trees by considering specific features. This allows it to create more efficient trees by taking into account different levels of importance of features in the dataset (Zhang *et al.* 2020). LightGBM can handle large datasets well because it utilizes parallel computing capabilities to reduce training and prediction times. Additionally, it offers advantages such as low memory usage and high scalability. However, it's essential to properly adjust some hyperparameters of LightGBM; otherwise, you might encounter issues like overfitting or other problems that could negatively impact the model's performance.

$$F_m = F_{m-1}(x) + \eta \cdot h_m(x) \tag{5}$$

In Equation 5, "F_m" represents the m-th prediction of the function, while "$F_{m-1}(x)$" represents the sum of predictions made in the (m-1)th stage. η represents the learning rate, and "$h_m(x)$" represents the weak predictor added in the m-th stage. LightGBM constructs a tree-based model using the gradient boosting method. This equation shows adding a new tree to the current predictions at each stage of the model and adding the predictions of the tree with the learning rate to the total predictions. This process allows the model to learn complex relationships in the dataset without increasing complexity and preventing overfitting. Thus, LightGBM is successfully used across a wide range of applications, providing high accuracy and fast training times.

Random Forest (RF) is a widely used machine learning algorithm for classification and regression problems. This algorithm constructs a predictor by aggregating many decision trees. Each decision tree is trained on a subset of data, randomly sampled from the original dataset (bootstrap sampling), and features selected randomly (subspace sampling). Then, each tree makes its prediction, and in the case of classification, the final prediction is made by voting, or in regression, by taking the average. Random Forest can learn more complex decision boundaries than a single tree and reduces the risk of overfitting. Additionally, it is robust to noise and missing data in the input dataset. However, it's crucial to adjust RF's hyperparameters (e.g., number of trees, size of feature subsets, etc.), as otherwise, its performance may degrade or overfitting may ocur (Liaw and Wiener 2002).

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{6}$$

In Equation 6, "$f(x)$" represents the predicted value of the function, "N" is the number of data points, and "$f_i(x)$" represents the

prediction of each decision tree. The Random Forest algorithm is an ensemble learning technique where many decision trees are collectively constructed, and the average prediction of each tree is taken. In this equation, "$f_i(x)$" shows the prediction made by each tree individually, while the term "$\frac{1}{N}$" provides the final prediction by averaging the predictions of all trees. This method balances the variance and errors within each tree while collectively obtaining a stronger and more generalized prediction. As a result, the Random Forest algorithm addresses complexity in the dataset while reducing the risk of overfitting, thus providing stable and accurate predictions.

Support Vector Machine (SVM) is a powerful machine learning algorithm used for classification and regression tasks. It determines a decision boundary in the feature space for classification or predicts a regression function, aiming to achieve the widest possible margin between classes, supported by a subset of training data points called support vectors (Gunn 1998). SVM works effectively on linearly separable problems and can handle nonlinear problems by transforming the feature space using kernel functions. Its advantages include effectiveness with high-dimensional datasets, reducing the risk of overfitting, and its ability to handle various data structures through different kernel functions. However, it's crucial to adjust SVM's hyperparameters (e.g., the C parameter, kernel type, etc.) correctly to avoid performance degradation or overfitting.

$$f(x) = w^T x + b \tag{7}$$

In Equation 7, "$f(x)$" represents the predicted classification for input data "x", while "w" denotes the weight vector and "b" stands for the bias term. SVM classifies data by creating a separation line between two classes. The "w" vector indicates the normal and slope of the separation plane, while the "b" bias term represents the distance of the plane from the origin. SVM utilizes this equation to find an optimal separation plane and is typically effective in classification problems.

$$margin = \frac{2}{\|w\|} \tag{8}$$

Equation 8 defines a concept known as the "margin" in the Support Vector Machine (SVM) algorithm. The "margin" indicates how far the separation plane is from the data points. The equation divides the norm (length) of the weight vector "$\|w\|$" and multiplies the result by two to calculate the margin. SVM aims to maximize the margin to find the best separation plane or hyperplane. Thus, SVM typically provides a wide margin for classifying data points and can better adapt to new data.

$$minimize \frac{1}{2}\|w\|^2 \tag{9}$$

Equation 9 represents the norm (length) of the weight vector "$\|w\|$". The goal of SVM is to classify data points by creating a separation plane between two classes. This equation is a mathematical expression used to determine the decision boundaries of SVM. By using this equation, SVM optimizes the weight vector and bias term (b) to find the equation of the separation plane (or hyperplane). Thus, it creates a separation plane that best classifies the data points. In summary, SVM aims to minimize the result of this equation to classify data points optimally.

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \tag{10}$$

Equation 10 represents the kernel function in the Support Vector Machine (SVM) algorithm. It denotes the measure of the dot product between two input vectors. "$\phi(x)$" symbolizes a feature map that transforms the input data into a higher-dimensional feature space. This function enables SVM to classify linearly inseparable data. By using this kernel function, SVM makes the data linearly separable and then finds a separation plane (or hyperplane). This kernel function plays a critical role in making the data linearly separable.

K-Nearest Neighbors (KNN) is a widely used machine learning algorithm for classification and regression problems. This algorithm relies on the nearest neighbors of a data point to determine its class or value. The distance between each data point and all other points in the feature space is calculated, and then the K nearest neighbors of the input data point are selected. In classification, a prediction is made based on the classes of these neighbors, while in regression, the average of the neighbors' values is used. KNN is a non-parametric algorithm, meaning it makes no assumptions about the underlying data distribution. It's also a versatile algorithm that can effectively handle both numerical and categorical data. However, it may suffer from computational inefficiency with large datasets, and selecting the correct value for K is crucial as it can affect the performance of the model (Kramer and Kramer 2013).

$$d(x_i, x_j) = \sqrt{\sum_{p=1}^{n} (x_{i,p} - x_{j,p})^2} \tag{11}$$

In Equation 11, "$d(x_i, x_j)$" represents the Euclidean distance between two data points, "x_i" and "x_j". "n" denotes the dimensionality of the data points, while "$x_{i,p}$" and "$x_{j,p}$" respectively represent the "p"-dimensional features of data points "i" and "j". The KNN algorithm uses the labels of neighboring data points to classify a new data point. This distance measurement determines the similarity or distance between one data point and another. The KNN algorithm classifies a new data point by considering the closest neighbors up to a specified "k" number. This distance measurement plays a fundamental role in the classification process of KNN, evaluating the relationship between data points to provide the most appropriate classification.

$$y = \text{mode}(y_1, y_2, \ldots, y_k) \tag{12}$$

In Equation 12, "y" represents the predicted value of the target variable, while "$y_1, y_2, \ldots, y_k$" denote the class labels of neighboring data points. The "mode" function determines the most frequently occurring class label among the neighboring data points, i.e., it takes the mode value. The KNN algorithm considers the labels of the nearest neighbors to classify a data point. In this method, the predicted class for a new data point is often the mode value obtained from the class labels of its neighbors. Thus, the KNN algorithm classifies a data point based on its neighboring points, and this equation explains this classification process.

$$y = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{13}$$

In Equation 13, it represents the classification process of the K-Nearest Neighbors (KNN) algorithm. While "y" denotes the predicted class of a new data point, "y_i" indicates the class labels of neighboring data points. "k" specifies the number of neighbors used. This equation is a fundamental step in the classification process of the KNN algorithm. The predicted class for a new data point is determined by taking the average of the class labels of its k nearest neighbors. Thus, the data point adopts the predicted class based on the class labels of its neighbors. This equation expresses a simple yet effective classification method of the KNN algorithm.

Gradient Boosting is a powerful machine learning technique for both classification and regression problems. This method constructs a strong predictor by combining many weak predictors. Each weak predictor focuses on correcting the errors of previous predictions. Using an optimization algorithm called gradient descent, Gradient Boosting attempts to minimize these errors. This process continues with the addition of a new weak predictor at each step, gradually reducing the model's errors and making more accurate predictions. One advantage of Gradient Boosting is its ability to combine predictors of different types, usually decision trees, which enhances its capability to learn different data structures and relationships. However, it's crucial to properly adjust the hyperparameters of Gradient Boosting, such as learning rate, tree depth, and number of trees, or else issues like overfitting or longer training times may arise (Bentéjac *et al.* 2021).

$$F_0(x) = 0 \tag{14}$$

In Equation 14, it represents the initial prediction in the Gradient Boosting algorithm. "$F_0(x)$" represents the initial prediction of a new data point, and this initial value is zero. The Gradient Boosting algorithm constructs a prediction model by sequentially adding weak predictors. Initially, the prediction model starts at zero. This equation specifies the beginning of the process for building the prediction model in the Gradient Boosting algorithm.

$$F_m(x) = F_{m-1}(x) + \rho * h_m(x) \tag{15}$$

In Equation 15, "$F_m(x)$" represents the prediction of the new model, while "$F_{m-1}(x)$" denotes the prediction of the previous model, and "$h_m(x)$" represents the m-th weak predictor. "ρ" indicates the learning rate. The Gradient Boosting algorithm uses this equation when adding the next weak predictor to the current prediction model. In other words, in each iteration, the predictions of the current model are updated by adding the predictions of the new predictor multiplied by the learning rate. This way, the algorithm controls the effect of the predictor added in the next step. This equation explains the process of iteratively improving the prediction model in the Gradient Boosting algorithm.

Explainable Artificial Intelligence (XAI) is a branch developed to understand and explain the decisions and predictions of machine learning and artificial intelligence models. XAI emerges from the effort to interpret the inner workings of complex models in a way that is more suitable for human understanding. These techniques contribute to addressing significant issues such as increasing the model's reliability by explaining why and how decisions are made, identifying errors, and addressing fairness and ethical concerns (Ali *et al.* 2023). XAI encompasses various techniques that can help understand the features, variables, and relationships underlying a model's predictions.

These include assessing feature importance, visualizing prediction boundaries, providing instance-based explanations, and analyzing interactions between features. However, XAI methods can themselves be complex, often depending on the complexity of the model, and it's crucial to strike a balance between explainability and model performance.

SHAP (Shapley Additive Explanations), an explainable artificial intelligence (XAI) technique that uses Shapley values to explain the contribution of each feature to model predictions. Shapley values originate from cooperative game theory and estimate the contribution of a feature to a model prediction by considering its value when combined with other features. SHAP is commonly used to make complex machine learning models (such as deep learning or gradient boosting) interpretable. This technique measures the impact of each feature on predictions while also showing how this impact varies for a specific example or observation. Thus, it provides a detailed understanding of why and how a particular prediction was made. SHAP can be used for tasks such as evaluating feature importance, examining interactions between features, and explaining how each feature contributes to model predictions. However, SHAP values can be challenging to interpret, and they can be computationally expensive when working with large datasets or complex models (Das and Rad 2020).

$$\phi_i(f) = \frac{1}{N!} \sum_{\pi} \Big[f(x_{\pi(i)}) - f(x_{\pi}) \Big] \tag{16}$$

Equation 16 represents an explanation method used in the SHAP (SHapley Additive exPlanations) algorithm to measure the contributions of features to the model prediction. "$\phi_i(f)$" represents the contributions of different features, while "f" denotes the model prediction, "N" is the number of data points, "π" is a permutation of data points, "$(x_{\pi(i)})$" represents the i-th data point in a specific permutation, and "x_{π}" denotes the permutation itself. This equation considers all permutations of data points to calculate the contribution of each feature to the model prediction. The SHAP algorithm is used to understand complex model predictions and explain the impact of each feature on the prediction.

LIME is a technique used in the field of explainable artificial intelligence (XAI) to explain how model predictions are made for a specific example or observation. Regardless of the complexity of the model, LIME makes any machine learning model interpretable. This technique creates a surrogate model to understand which features or variables are influential in making a prediction at a particular point. The surrogate model selects features to mimic the predictions of the original model locally. LIME generates data samples by making random changes around the data point to create the local model, and uses the predictions of each sample in the original model. Finally, by examining the behavior of the local model on these generated samples, LIME explains how the original model made a specific prediction. LIME is particularly useful when working with complex models and in situations where understanding why certain predictions are made is difficult. However, interpreting LIME results can be challenging, and careful parameter tuning may be required to obtain accurate results (Das and Rad 2020).

$$e(x) = \arg\min_{g \in \mathcal{G}} (f, g, \pi_x) + \Omega(g) \tag{17}$$

In Equation 17, an explanation method in the LIME (Local Interpretable Model-agnostic Explanations) algorithm is described. "$e(x)$" represents the explanation of a particular example, while the "arg min" operator denotes the one with the smallest value within a given set. "g" represents the model prediction, "f" is the true function, and "π_x" indicates the weights of other examples around the

sample "x". "$\Omega(g)$" is a term limiting the complexity of the model. This equation is formulated as an optimization problem to find the explanation model that best explains the prediction of a particular example. LIME provides local explanations to understand the decisions of complex machine learning models.

RESULTS

In this study, the Biogeography-Based Optimization (BBO) algorithm was used to predict the "share_global_coal_co2" parameter. BBO is an optimization algorithm inspired by natural biogeographic processes, used to determine complex relationships in the dataset and identify the most important features. In this study, the BBO algorithm was used to predict the "share_global_coal_co2" parameter, and the 20 most effective parameters among other parameters in the dataset were identified. BBO assists in predicting the "share_global_coal_co2" parameter by selecting from various parameters in the dataset, thereby helping to predict it more accurately. Therefore, the BBO algorithm was used to predict the "share_global_coal_co2" parameter and understand the impact of specific features. The 20 most important features identified by the BBO algorithm were fed into machine learning models. They were evaluated using metrics such as Mean Squared Error (MSE) and R-squared Score.

Mean Squared Error (MSE) is a metric used to evaluate predictions made by machine learning algorithms. This measurement calculates the average of the squared differences between the actual values and the predicted values. Lower MSE values indicate that the predictions are closer to the actual values, while higher MSE values indicate that the predictions are farther from the actual values. Therefore, MSE is an important measure used to assess a model's predictive ability.

The R-squared (R^2) score is a metric used to evaluate predictions made by machine learning algorithms. This score determines how well a model fits and explains the data. R-squared measures the proportion of the variance in the dependent variable that is predictable from the independent variables. The values typically range from 0 to 1; the closer it is to 1, the better the model explains the data. However, it can also take negative values, indicating that the model performs worse than a horizontal line. In summary, the R-squared score is a measure of how well a model fits the data.

The evaluation results based on the top 20 features identified by the BBO algorithm are presented in Table 1. The Gradient Boosting algorithm, which gave the lowest Mean Squared Error (MSE) value, has been explained using interpretable artificial intelligence models SHAP and LIME. The explanations are presented in Figures 1 and 2.

In Figure 1, the average impact of various greenhouse gas (GHG) emissions and carbon dioxide (CO_2) sources on the model output is assessed using SHAP values. In the figure, total greenhouse gas emissions (total_ghg) and the share of global cumulative cement CO_2 (share_global_cumulative_cement_co2) stand out as the factors with the highest average impact on the model output. Nitrous oxide (nitrous_oxide) and the share of global CO_2 (share_global_co2) also have significant impacts, while factors such as other industry CO_2 (other_industry_co2) and the share of global cumulative coal CO_2 (share_global_cumulative_coal_co2) have less pronounced effects.

Lower impacts are observed among factors such as cumulative land use change CO_2 (cumulative_luc_co2) and temperature change from GHG (temperature_change_from_ghg). Overall, the figure demonstrates that greenhouse gases and various CO_2 emission sources contribute to model outcomes to varying degrees.

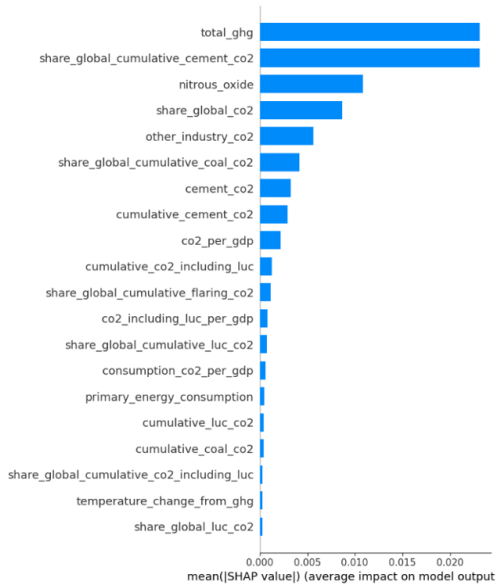

Figure 1 Output of SHAP, an explainable artificial intelligence model

Figure 2 The output of LIME, an explainable artificial intelligence model

In Figure 2, the "GradientBoostingRegressor" algorithm and the interpretable artificial intelligence model "LIME" were used to create predictions for the share of global coal emissions. The predicted value is determined as "-0.20". According to the analysis, the features that the model gives the most importance to are "coal_co2" and "share_global_luc_co", and it is observed that changes in these features have significant effects on the prediction. Additionally, as indicated in the visual, there is a negative correlation with the "share_global" feature's prediction, meaning that as this value increases, the predicted share of global coal emissions decreases

CONCLUSION

Our results demonstrate that the Biogeography-Based Optimization (BBO) algorithm is an effective method for predicting the "share_global_coal_co2" parameter. The BBO algorithm has achieved successful outcomes by utilizing various parameters in the dataset to identify complex relationships and determine significant features. In this study, the 20 most important features identified by the BBO algorithm were integrated into machine learning models to evaluate prediction performance.

Evaluations conducted using metrics such as Mean Squared Error (MSE) and R-squared score have indicated that the Gradient Boosting algorithm provides the lowest MSE value , suggesting that the predictions are closer to the actual values. These results support the use of interpretable artificial intelligence models such as SHAP and LIME to enhance the accuracy of the model.

Table 1 Results of evaluation metrics of machine learning algorithms.

Machine learning	Mean Square Error (MSE)	R-squared Score
ANN	4.017079	-9.101274
XGBoost	0.572459	0.987030
LightGBM	0.372869	0.991552
RF	0.410437	0.990700
SVM	17.084162	0.612933
KNN	26.406132	0.401730
Gradient Boosting	0.347408	0.972128

SHAP and LIME analyses have rendered the model's predictions more understandable. Specifically, the GradientBoostingRegressor algorithm and LIME model utilized to predict the share of global coal emissions have emphasized the effects of specific features such as "coal_co2" and "share_global_luc_co" on the prediction. These findings offer valuable insights that can be utilized in making critical decisions in areas such as energy policies and environmental management strategies.

In conclusion, this study demonstrates that the Biogeography-Based Optimization algorithm is an effective method for predicting the "share_global_coal_co2" parameter. Furthermore, it underscores the importance of utilizing interpretable artificial intelligence models to elucidate and render predictions more comprehensible. These findings serve as a valuable guide for environmental policymakers and energy experts.

Availability of data and material

Not applicable.

Conflicts of interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The author has no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Ali, S., T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, *et al.*, 2023 Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence. Information fusion **99**: 101805.

Bentéjac, C., A. Csörgő, and G. Martínez-Muñoz, 2021 A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review **54**: 1937–1967.

Chen, T. and C. Guestrin, 2016 Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Das, A. and P. Rad, 2020 Opportunities and challenges in explainable artificial intelligence (xai): A survey. arXiv preprint arXiv:2006.11371 .

Delanoë, P., D. Tchuente, and G. Colin, 2023 Method and evaluations of the effective gain of artificial intelligence models for reducing co2 emissions. Journal of Environmental Management **331**: 117261.

Gunn, S. R., 1998 Support vector machines for classification and regression. ISIS technical report **14**: 5–16.

Heo, S., J. Ko, S. Kim, C. Jeong, S. Hwangbo, *et al.*, 2022 Explainable ai-driven net-zero carbon roadmap for petrochemical industry considering stochastic scenarios of remotely sensed offshore wind energy. Journal of Cleaner Production **379**: 134793.

Kramer, O. and O. Kramer, 2013 K-nearest neighbors. In *Dimensionality reduction with unsupervised nearest neighbors*, pp. 13–23, Springer.

Liaw, A. and M. Wiener, 2002 Classification and regression by randomforest. R news **2**: 18–22.

Qerimi, Q. and B. S. Sergi, 2022 The case for global regulation of carbon capture and storage and artificial intelligence for climate change. International Journal of Greenhouse Gas Control **120**: 103757.

Stef, N., H. Başağaoğlu, D. Chakraborty, and S. B. Jabeur, 2023 Does institutional quality affect co2 emissions? evidence from explainable artificial intelligence models. Energy Economics **124**: 106822.

Thanh, H. V., A. Zamanyad, M. Safaei-Farouji, U. Ashraf, and Z. Hemeng, 2022 Application of hybrid artificial intelligent models to predict deliverability of underground natural gas storage sites. Renewable Energy **200**: 169–184.

Yan, H., J. Zhang, N. Zhou, and M. Li, 2020 Application of hybrid artificial intelligence model to predict coal strength alteration during co2 geological sequestration in coal seams. Science of the total environment **711**: 135029.

Zhang, Y., C. Zhu, and Q. Wang, 2020 Lightgbm-based model for metro passenger volume forecasting. IET Intelligent Transport Systems **14**: 1815–1823.

How to cite this article: Ozkurt, C. Environmental Sustainability through AI: A Case Study on CO_2 Emission Prediction. *ADBA Computer Science*, 1(1), 19-25, 2024.

ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.1 / No.1 / 2024 / pp.26-35
https://doi.org/10.69882/adba.cs.2024075

Optimizing Diabetes Prediction: Addressing Data Imbalance with Machine Learning Algorithms

Khalid Hani Abushahla *,1 **and Muhammed Ali Pala** α,2

*Biomedical Engineering Department, Graduate Education Institute, Sakarya University of Applied Sciences, 54050, Sakarya, Turkiye, αBiomedical Technologies Application and Research Center & Electrical and Electronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, 54050, Sakarya, Turkiye.

ABSTRACT

Imbalanced datasets pose significant challenges in various fields including the classification of medical conditions such as diabetes. This study investigates six methodologies for handling imbalanced diabetes datasets aiming to enhance classification performance through diverse preprocessing techniques. The methodologies are evaluated using multiple models: Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, KNN, Naive Bayes, XGBoost, LightGBM, and CatBoost. The preprocessing techniques include simple implementation, data standardization, normalization, standardization with K-Fold cross-validation, and two variations incorporating the SMOTE oversampling technique. The effectiveness of each methodology is assessed based on accuracy, precision, recall, and F1 scores across different classifiers. Results indicate that standardization combined with K-Fold cross-validation consistently enhances model performance. Additionally, the integration of the SMOTE technique significantly improves results, especially for Gradient Boosting and SVM classifiers. Among the tested models, CatBoost demonstrated exceptional performance in handling imbalanced datasets, achieving an accuracy of 95.18%, precision of 91.10%, recall of 95.52%, and an F1 score of 93.26%. This study underscores the importance of tailored preprocessing techniques in improving the classification of imbalanced medical datasets, highlighting their potential to enhance predictive accuracy in critical applications.

KEYWORDS

Machine learning
Imbalanced dataset
Diabetes classification
Ensemble learning

INTRODUCTION

Diabetes is a long-term metabolic disease characterized by hyperglycemia (elevated blood glucose levels) due to deficiencies in the function of insulin secretion or both that damages the heart, blood vessels, eyes, kidneys, nerves, and heart over time (Association 2009). The most common kind of diabetes, Type 2, usually appears in adulthood as a result of either insufficient or resistant insulin production. Over the past 30 years, its prevalence has skyrocketed globally across all income categories. The hallmark of type 1 diabetes, also known as juvenile or insulin-dependent diabetes, is insufficient insulin production by the pancreas. For those with diabetes, having affordable access to treatment—in particular insulin—is essential. By 2025, the global goal is to stop the rise in diabetes and obesity. Approximately 422 million people worldwide suffer from diabetes, most of whom live in low- and middle-income countries. The disease is directly responsible for 1.5 million fatalities per year. Over the past few decades, there has been a steady increase in both the number of cases and the incidence of diabetes. Therefore, a tool that can help physicians identify this fatal disease earlier and halt its course is desperately needed (Abdulhadi and Al-Mousa 2021).

Machine learning techniques offer immense potential to enhance medical research and clinical care, particularly as providers increasingly utilize electronic health records. Two areas ready to benefit from the application of ML in the medical field are diagnosis and outcome prediction (Shivahare *et al.* 2024). This encompasses the potential identification of high-risk scenarios for

Manuscript received: 30 May 2024,
Revised: 27 June 2024,
Accepted: 30 June 2024.

[1]22500305002@subu.edu.tr
[2]pala@subu.edu.tr

medical emergencies, such as relapse or transitioning into another disease state (Sidey-Gibbons and Sidey-Gibbons 2019). Recent successes include predicting the progression from pre-diabetes to type 2 diabetes using routinely-collected electronic health record data (Anderson *et al.* 2016).

Though in machine learning and AI, class imbalance in datasets is a common issue in real-world dataset analysis, particularly in industries like healthcare, finance, and telecommunications. It can lead to negative effects if incorrectly classified minority cases are identified. Two strategies have been developed in research: external techniques to rebalance distributions before training and internal algorithms to manage imbalance directly. The research aims to provide solid solutions for handling class imbalance in real-world data analysis situations (Ramyachitra and Manikandan 2014).

Several studies have explored predicting diabetes using various datasets and criteria resulting in varying accuracies and performance levels. Chang et al. (2022) evaluated interpretable machine learning models within the Internet of Medical Things (IoMT) using the Pima Indians diabetes dataset with random forest outperforming Naïve Bayes and J48 decision tree across multiple metrics (Chang *et al.* 2023). Naz & Ahuja (2020) focused on predicting diabetes onset achieving high accuracy rates with Deep Learning showing the highest accuracy (Naz and Ahuja 2020). Rajni & Amandeep (2019) introduced the RB-Bayes framework combining methods to improve prediction accuracy emphasizing early detection (Rajni and Amandeep 2019). Bhoi et al. (2021) employed multiple machine learning algorithms with Logistic Regression emerging as the top performer (Bhoi *et al.* 2021). Patra & Khuntia (2021) introduced the sdknn classifier showing significant improvement over conventional techniques (Patra and Khuntia 2021). Miao (2021) developed prediction models highlighting glucose, insulin, and BMI's correlations with diabetes and the Support Vector Classifier's potential (Miao 2021). Mousa et al. (2023) examined machine-learning models for diabetes diagnosis with LSTM performing best in capturing temporal dependencies (Mousa *et al.* 2023).

The prediction of diabetes using various machine learning models has been a topic of extensive research yielding diverse levels of accuracy and performance. Numerous algorithms have been used in studies ranging from interpretable models like Naive Bayes and random forest to deep learning strategies and ensemble frameworks that combine several techniques. Even though research shows notable improvements, the problem of class imbalance in diabetes datasets continues to be a major barrier to predictive accuracy. This study aims to address the problem of imbalanced diabetes datasets by investigating six methodologies to enhance classification performance through diverse preprocessing techniques. We evaluate the effectiveness of these methodologies using multiple models including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, KNN, Naive Bayes, XGBoost, LightGBM, and CatBoost. The preprocessing techniques explored include simple implementation, data standardization, normalization, standardization with K-Fold cross-validation, and two variations incorporating the SMOTE oversampling technique. The models' effectiveness is assessed based on accuracy, precision, recall, and F1 scores. This study underscores the importance of tailored preprocessing techniques in improving the classification of imbalanced medical datasets, highlighting their potential to enhance predictive accuracy in critical applications.

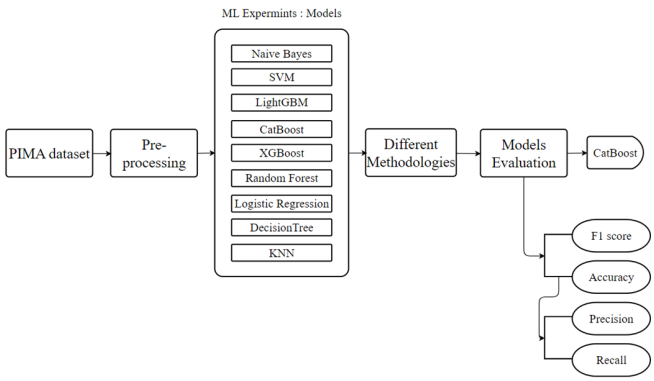

Figure 1 Graphical abstract of the proposed method

METHODOLOGY AND MATERIAL

In our study, we employ a comprehensive methodology to address the challenge of imbalanced datasets. Through detailed exploration, we delve into various preprocessing techniques, validation strategies, and data splitting methodologies to confront this issue head-on. Our approach involves precise experimentation to analyze the methods and differences between these approaches, revealing their strengths and weaknesses. By examining the results, we pointed out the most effective approach, one that not only mitigates data imbalance effects but also maximizes predictive performance. Our dedication to methodical exploration ensures optimal results and a deeper understanding of the underlying dynamics within our datasets.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial first step in the knowledge discovery process, where data scientists use a series of analysis operations (such as filtering, aggregating, and visualizing data) to interactively explore unknown datasets (Milo and Somech 2020). Before formal modeling, graphical representations, and visualizations, EDA seeks to conduct preliminary data investigations to find patterns, evaluate presumptions, and test hypotheses. Explanatory and comparative charts are a common feature of data visualizations, which help to clearly convey both concrete and abstract concepts. When issues are identified and resolved, the accuracy of diabetes diagnosis can be increased. Key aspects and hidden trends in the data can be summarized to help detect difficulties (**?**). Through Table 1, Figure 2, and Figure 3, some substantial insights of the PIMA dataset were illustrated with understandable visuals.

Understanding and Visualizing Data

This dataset includes data from studies on diabetes among women who identify as Pima Indian and who live in Phoenix, Arizona, USA. The women in the dataset are 21 years of age and older. It has eight different numeric variables and 768 entries (**?**). The selected target variable classes are labeled as follows: 1 denotes a positive diabetes test, while 0 denotes a negative test. The name of the dataset, descriptions of the data types, and corresponding roles are shown in Table 1. EDA clarified the dataset and showed that it included 268 individuals with diabetes and 500 values of sample patients who were not diabetic. This makes the imbalance in the dataset its primary challenge. For reference, refer to Figure 3.

Table 1 Dataset description

Main Criteria	Data Type	Input/Target	Notes
Outcome	Categorical	Target	0: No diabetes / 1: Diabetes
Pregnancies	Numerical	Input	-
Glucose	Numerical	Input	-
Blood Pressure	Numerical	Input	-
Skin Thickness	Numerical	Input	-
Insulin	Numerical	Input	-
BMI	Numerical	Input	-
Diabetes Pedigree Function	Numerical	Input	-
Age	Numerical	Input	-

Table 2 Statistical summary of the dataset

	Count	Mean	Std	Min	25%	50%	75%	Max
Pregnancies	768	3.845	3.37	0	1	3	6	17
Glucose	768	120.895	31.973	0	99	117	140.25	199
Blood Pressure	768	69.105	19.356	0	62	72	80	122
Skin Thickness	768	20.536	15.952	0	0	23	32	99
Insulin	768	79.799	115.244	0	0	30.5	127.25	846
BMI	768	31.993	7.884	0	27.3	32	36.6	67.1
Diabetes Pedigree Function	768	0.472	0.331	0.078	0.244	0.372	0.626	2.42
Age	768	33.241	11.76	21	24	29	41	81
Outcome	768	0.349	0.477	0	0	0	1	1

Pre-processing of the Data

Pre-processing is the primary prerequisite for working with datasets. Firstly, outliers in the dataset are addressed using Z-score calculation, where data points exceeding a specified threshold are replaced with NaN values. Subsequently, missing values and zero values in specific columns are handled. The only columns which are excluded from the zero values checking are the "Pregnancies" because it's a true value that many women haven't been pregnant before, and the "Outcome" column because 0 there demonstrates no diabetes diagnosis. Initially, missing values are identified and replaced with the mean of their respective columns. Then zero values in selected columns are replaced with the mean value as well. Then the data was split into 2 splits: training with 80% of the data and testing with 20%. Finally, a confirmation of the replacements is provided. Cumulatively, these processes guarantee that the dataset is free of outliers, NaNs, missing data, and zero values, making it appropriate for activities involving machine learning.

K-Fold Cross-Validation

To assess the generalization performance of the models, K-Fold cross-validation is employed with a predefined number of folds (k=5) (Murugan *et al.* 2023). This technique divides the dataset into k subsets, with each subset serving as a testing set and the remaining data as the training set. The procedure is repeated k times, and the average performance metrics across all folds are determined, assuring the models' durability and dependability beyond the original train-test split (Sohil *et al.* 2013).

Imbalanced Dataset and Solution

Classifiers are designed to categorize objects based on their attributes. However, in practical scenarios, datasets often exhibit class imbalance, where certain classes have significantly fewer instances compared to others. This class imbalance poses a challenge for traditional classification algorithms as they tend to be less accurate in predicting minority classes. This phenomenon

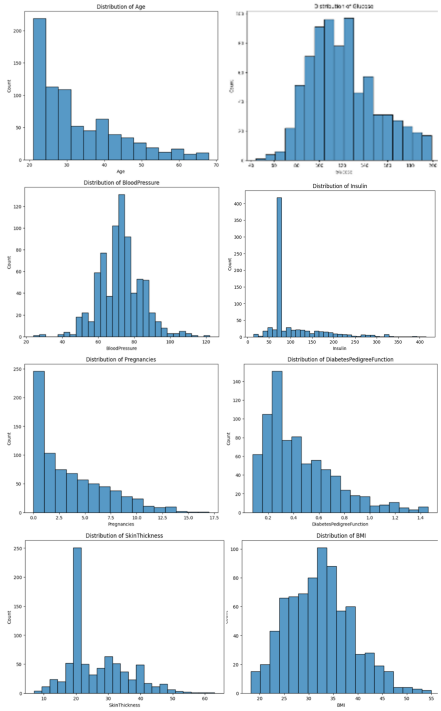

Figure 2 Distribution of the 6 input columns values [Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age]

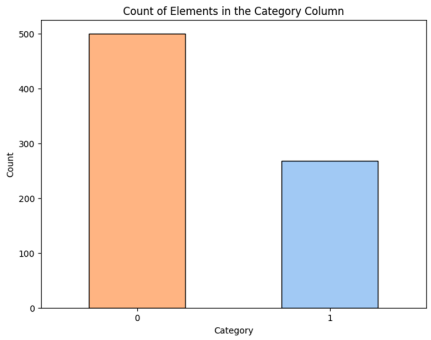

Figure 3 Outcome column values distribution [classification goal]

is known as the class imbalance problem. To address this issue, various techniques have been proposed, including over and under-sampling methods (Pradipta *et al.* 2021). Examples of that are SMOTE (Synthetic minority oversampling approach), ADASYN (Adaptive Synthetic Sampling Method), Borderline-SMOTE, and Safe-Level SMOTE (Gosain and Sardana 2017). These techniques aim to rebalance the dataset by adjusting the class distribution, thereby enhancing the performance of classification algorithms on imbalanced data (Pradipta *et al.* 2021). To address the issue of imbalanced dataset problem, the SMOTE technique was utilized.

SMOTE (Synthetic Minority Over-sampling Technique) introduced by Chawla et al. in 2002 addresses class imbalance by generating synthetic samples in the minority class rather than simply replicating existing samples, thus avoiding overfitting. To further enhance accuracy and mitigate overfitting, the SMOTE algorithm was refined. This method creates artificial minority instances along the line segments connecting minority samples and their 'k' nearest neighbors within the minority class. The 'k' nearest neighbors are randomly selected based on the desired oversampling rate. However, a limitation of SMOTE is its tendency to oversimplify the minority class space without considering the majority class, potentially leading to increased overlap between classes (Gosain and Sardana 2017).

Feature Scaling of the Dataset

MinMaxScaler and StandardScaler are both preprocessing techniques commonly used in machine learning (D.K. *et al.* 2019). MinMaxScaler rescales features to a predetermined range, typically between 0 and 1 (Powers 2020), whereas StandardScaler transforms features to have a mean of 0 and a standard deviation of 1. The primary distinction between them lies in their treatment of outliers and the resulting shape of the distribution.MinMaxScaler may distort data in the presence of outliers, whereas StandardScaler exhibits less sensitivity to them (de Amorim *et al.* 2023). In this study, both feature scalling tools were employed across various experiments to assess their impact on model training and the performance achieved when using normalized data.

Proposed Machine Learning Models

In this study, ten different machine learning models were employed to analyze the dataset and their outcomes were compared. While ensemble models have shown promising performance in prior research, this study explored and compared various ensemble models beside conventional ones in the domain of machine learning experimentation.

Logistic regression (LR) is a model that predicts the probability of a binary outcome by assessing the odds of the event occurring versus not occurring using predictor variables (y = 0 or 1). It employs the natural logarithm of these odds as a regression function. Odds ratios quantify the impact of predictors on the outcome, with

the exponential of the regression coefficients providing these ratios. While logistic regression doesn't have a straightforward formula for estimation like linear regression, it involves iterative processes to converge on the best estimates (LaValley 2008).

A Decision Tree is generated from a collection of labeled training examples, each described by a set of attribute values paired with a class label. Given the expansive search options, decision-tree learning generally follows a greedy, top-down, and recursive approach commencing with the full training dataset and an unfilled tree. It selects an attribute that optimally divides the training data as the root split, subsequently segregating the data into distinct subsets based on the attribute's values. This process repeats recursively for each subset until all instances within a subset share the same class label (Su 2024).

Random Forest is created by combining different tree predictors so that every tree in the forest is dependent on the values of a random vector that is randomly sampled and has the same distribution for every tree. As the number of trees in a forest increases, the generalization error converges a.s. to a limit. The strength of each individual tree in the forest and the correlation between them determine the generalization error of a forest of tree classifiers. Each node can be split using a random feature selection process, which produces error rates that are more resilient to noise but still compare favorably to Adaboost (Rigatti 2017). Internal estimates track correlation, inaccuracy, and strength and are used to illustrate how the number of features employed in the splitting changes. The importance of each variable is also determined by internal estimations. These concepts also apply to regression (Breiman 2001).

Support Vector Machines (SVM) is a powerful algorithm based on Vapnik-Chervonenkis theory designed for supervised learning classification problems. It aims to find the optimal separating surface or hyperplane between two classes using kernel functions and slack variables for noisy data. SVM maximizes margins, the separation between the decision boundary and support vectors to maximize confidence in predictions and generalization ability, ensuring robustness and good generalization to new data (Bhavsar and Panchal 2024).

K-Nearest Neighbor (KNN) is an algorithm that is a straightforward yet effective machine learning method utilized for both classification and regression tasks. It operates by grouping data into coherent clusters or subsets and classifying new input based on its similarity to previously trained data. Essentially, the input is assigned to the class with the most nearest neighbors. While KNN is widely used due to its simplicity and effectiveness, it also possesses several weaknesses. To address these shortcomings, modified versions of the KNN algorithm have been developed through prior research efforts. These variants aim to enhance efficiency by mitigating the limitations of the original KNN approach (Taunk *et al.* 2019).

The basis of the Naive Bayes classifier is a probabilistic approach. Under the presumption that the existence of one feature in a class is unrelated to the existence of another feature in the same class, it applies Bayes' theorem. To estimate the probabilities of a particular category, one uses the joint probabilities of terms and categories. This independence assumption makes it possible to study each term's parameters separately, which speeds up calculation. A set of conditional probabilities and a structural model make up the Bayesian network (Kumari *et al.* 2021).

Gradient Boosting is a fundamental ensemble learning technique developed by Jerome H. Friedman in the late 1990s. It involves iteratively improving predictive models by training weak learners like decision trees to rectify errors from previous models. By focusing on the residuals or gradients of the loss function from the previous model, it reduces prediction errors and assembles an ensemble of models each refining the previous model's predictive accuracy.

XGBoost, an evolution of Gradient Boosting, was developed by Tianqi Chen in 2014 and quickly gained prominence for its efficiency and scalability. Building upon the principles of Gradient Boosting, XGBoost introduces advanced regularization techniques, parallel and distributed computing capabilities, and a comprehensive set of hyperparameters. By optimizing the model's architecture and training process, XGBoost significantly enhances performance while mitigating overfitting. Its versatility and robustness have made it a staple in data science competitions and real-world applications alike (Chen and Guestrin 2016).

LightGBM, a cutting-edge gradient boosting framework, emerged from the labs of Microsoft in 2016, engineered by Guolin Ke et al. Unlike traditional approaches, LightGBM employs novel tree-growing algorithms like Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to enhance speed and efficiency. By prioritizing the most informative data points during tree construction and leveraging histogram-based algorithms, LightGBM achieves unparalleled performance on large-scale datasets. With native support for categorical features and a rich set of hyperparameters, LightGBM empowers users to build high-quality models with minimal computational resources (Ke *et al.* 2016).

CatBoost developed by Yandex researchers in 2017, CatBoost revolutionizes gradient boosting with its intrinsic handling of categorical features. Created by Daniil Osokin and others, CatBoost automates categorical data encoding, sparing users from tedious preprocessing tasks. Employing advanced regularization techniques like ordered boosting and dynamic tree level regularization, CatBoost effectively combats overfitting while preserving predictive accuracy. Furthermore, its GPU-accelerated training and built-in visualization tools make it a formidable choice for practitioners seeking both performance and interpretability in their models (Prokhorenkova *et al.* 2019).

Evaluation Metrics To assess the models, we employed various metrics commonly used in machine learning evaluations such as the confusion matrix and its derived metrics: Accuracy, Precision, Recall, and F1 Score (Arias-Duart *et al.* 2023). Additionally, the ROC graph was displayed alongside the confusion matrix (Salih and Abdulazeez 2021).

RESULTS AND DISCUSSION

In the study, the methodology tackles the challenge of dealing with imbalanced datasets head-on. Various preprocessing techniques, validation strategies, and data splitting methodologies have been implemented to address this issue comprehensively. Through rigorous experimentation, we've meticulously examined the nuances and disparities between these methods, meticulously dissecting both their strengths and weaknesses.

By scrutinizing the results meticulously, we've identified the most effective approach, one that not only mitigates the effects of data imbalance but also maximizes predictive performance. Our dedication to methodical exploration ensures that we not only achieve optimal results but also gain a deeper understanding of

Table 3 Evaluation Metrics Definition, Formulas, and Ideal Values

Metric	Formula	Definition	Ideal Situation
Confusion Matrix	Table of [TP, TN, FP, FN]	An error matrix is another name for a confusion matrix. It facilitates our analysis of each categorization model's performance. It provides a clear picture of the efficiency of your classification method (Duvva 2024).	The ideal confusion matrix has values only along the diagonal (Duvva 2024).
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Proportion of correct predictions out of total predictions (Luque *et al.* 2019).	1.0
Precision	$\frac{TP}{TP+FP}$	Proportion of true positive predictions out of all positive predictions made by the model (Luque *et al.* 2019).	1.0
Recall	$\frac{TP}{TP+FN}$	Proportion of true positive predictions out of all actual (Luque *et al.* 2019).	1.0
F1 Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of precision and recall balances both metrics (Luque *et al.* 2019).	1.0
ROC	Graph of True Positive and False Positive rates (Raschka 2014).	Helpful resources for choosing categorization models according to how well they perform in terms of True Positive and False Positive rates. Random guessing is represented by the diagonal of a ROC graph and classification models that lie below the diagonal are thought to be less accurate than random guessing (Raschka 2014).	A perfect classifier would have a True Positive Rate of 1 and a False Positive Rate of 0 placing it in the upper left corner of the graph.

the underlying dynamics within our datasets.

The findings of the study present six distinct methodologies for handling our preserved dataset. In this section, the performance of various machine learning classifiers under different preprocessing techniques is presented and analyzed: Simple Implementation, Data Standardization, Data Normalization, Standardization With K-Fold, and Standardization With K-Fold and SMOTE. The study presents six distinct methodologies for handling a preserved dataset focusing on the performance of various machine learning classifiers under different preprocessing techniques. Simple implementation without standardization showed varied performance across classifiers with Support Vector Machine (SVM) exhibiting the highest accuracy.

Applying standardization to the data resulted in consistent improvements across classifiers, particularly benefiting Logistic Regression, Random Forest, and LightGBM. Normalization yielded mixed results with Logistic Regression showing the highest accuracy. Standardization with K-Fold validation provided robust estimates of model performance. Logistic Regression consistently emerged as the top performer across all preprocessing methods. Further exploration included Standardization with K-Fold and SMOTE, which notably improved model performance, particularly for Random Forest and SVM.

An alternative approach involving standardization, SMOTE oversampling, and K-Fold cross-validation showcased significant improvements in various classifiers, with CatBoost exhibiting impressive results. Comparing all methodologies revealed varying degrees of success, with the combined use of standardization, K-Fold cross-validation, and SMOTE proving effective. Random Forest and SVM consistently performed well across different methodologies. The findings offer valuable insights in addressing data preprocessing and class imbalance challenges in machine learning tasks, emphasizing the importance of careful experimentation and customization based on dataset characteristics and problem requirements.

Table 4 Evaluation metric results for each trained model across the six applied methodologies

Normalization of Data				
Classifier	Acc	Prec	Rec	F1
Logistic Regression	77.27%	70.83%	57.62%	66.02%
Decision Tree	72.43%	61.21%	67.19%	64.05%
Random Forest	75.99%	66.57%	65.43%	66.00%
Gradient Boosting	75.32%	63.93%	70.91%	67.24%
SVM	72.76%	61.20%	67.45%	64.18%
KNN	71.68%	59.96%	71.24%	64.95%
Naive Bayes	74.68%	62.50%	72.73%	67.23%
XGBoost	71.43%	58.79%	77.27%	66.21%
LightGBM	72.08%	59.09%	70.91%	64.46%
CatBoost	74.03%	63.16%	65.45%	64.29%
Standardization With K Fold				
Classifier	Acc	Prec	Rec	F1
Logistic Regression	72.44%	72.77%	57.62%	64.13%
Decision Tree	76.89%	69.35%	60.74%	64.80%
Random Forest	76.09%	68.91%	60.74%	64.13%
Gradient Boosting	76.30%	69.05%	61.16%	64.86%
SVM	72.21%	61.96%	68.60%	65.05%
KNN	72.16%	63.64%	71.24%	67.22%
Naive Bayes	75.13%	64.95%	63.64%	63.81%
XGBoost	73.18%	61.14%	68.75%	63.95%
LightGBM	74.38%	65.34%	65.91%	63.59%
CatBoost	76.43%	69.08%	59.87%	63.80%
Standardization, K Fold and SMOTE Implementation				
Classifier	Acc	Prec	Rec	F1
Logistic Regression	75.66%	61.57%	70.91%	67.03%
Decision Tree	62.82%	54.29%	61.22%	57.03%
Random Forest	77.60%	66.44%	66.45%	65.69%
Gradient Boosting	74.87%	62.28%	72.16%	66.76%
SVM	76.69%	68.94%	71.43%	69.77%
KNN	71.95%	63.76%	71.24%	67.27%
Naive Bayes	74.38%	61.60%	75.00%	67.39%
XGBoost	74.50%	61.89%	68.45%	64.69%
LightGBM	75.49%	61.43%	68.30%	65.70%
CatBoost	76.43%	64.53%	73.30%	68.45%
Standardization, K FOLD and SMOTE Implementation (Variation)				
Classifier	Acc	Prec	Rec	F1
Logistic Regression	77.34%	63.46%	74.63%	69.69%
Decision Tree	78.39%	63.95%	74.30%	74.30%
Random Forest	80.10%	100.00%	100.00%	100.00%
Gradient Boosting	89.19%	70.73%	90.67%	85.41%
SVM	82.53%	70.94%	74.70%	77.21%
KNN	100.00%	100.00%	100.00%	100.00%
Naive Bayes	74.74%	69.09%	70.91%	66.00%
XGBoost	100.00%	100.00%	100.00%	100.00%
LightGBM	100.00%	100.00%	100.00%	100.00%
CatBoost	95.18%	91.10%	95.52%	93.26%

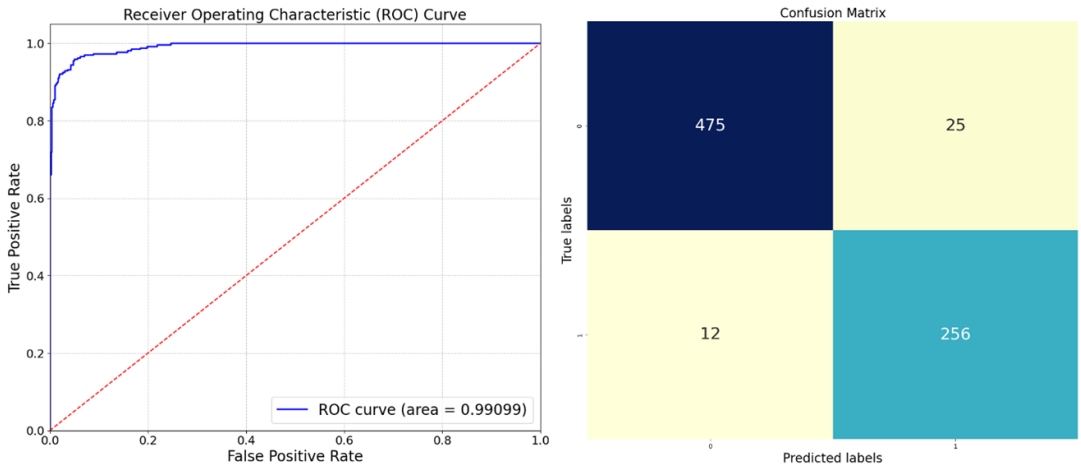

Figure 4 The confusion matrix and ROC curve illustrate the performance of the top-performing model (CatBoost) following the implementation of Standardization with K-Fold and SMOTE techniques.

Table 5 Classification performance of other classifiers in the literature

Reference	Models	Best Performance
(Chang et al., 2022) (Chang *et al.* 2023)	NB, RF, and J48 DT	F1-score: 85.17%
(Naz & Ahuja, 2020) (Naz and Ahuja 2020)	ANN, NB, DT, and DL	Accuracy: 98.07%
(Rajni & Amandeep, 2019) (Rajni and Amandeep 2019)	SVM, NB, KNN, and RB-Bayes framework	Accuracy: 72.9%
Bhoi et al. (2021) (Bhoi *et al.* 2021)	CT, SVM, k-NN, NB, RF, NN, AdaBoost (AB), LR	F1-score: 76%
Patra & Khuntia (2021) (Patra and Khuntia 2021)	Standard Deviation K Nearest Neighbor (SDKNN) classifier	Accuracy: 83.2%
(Miao, 2021) (Miao 2021)	SVM	Accuracy: 87.01%
(Mousa et al., 2023) (Mousa *et al.* 2023)	LSTM, RF, CNN	Accuracy: 85%
This study	LR, DT, RF, GB, SVM, KNN, NB, XGBoost, LightGBM, CatBoost	Accuracy: 94.27%, Precision: 89.16%, Recall: 95.15%, and F1 score: 92.06%

CONCLUSION

In conclusion, this article addressed the common challenge of imbalanced datasets, particularly in the context of classifying medical conditions such as diabetes. It investigated six distinct methodologies aimed at addressing the challenges posed by imbalanced datasets with a specific focus on classifying imbalanced diabetes datasets. The primary objective is to mitigate these challenges through customized preprocessing techniques. Through comprehensive evaluation using various classifiers and performance metrics such as accuracy, precision, recall, and F1 scores, it is evident that standardization, particularly when integrated with K-Fold cross-validation, consistently enhances model performance across classifiers. Moreover, the integration of the SMOTE oversampling technique significantly boosts model performance, particularly noted in Gradient Boosting and SVM classifiers.

Notably, CatBoost emerges as a proficient tool in handling imbalanced datasets, demonstrating impressive accuracy, precision, recall, and F1 scores adapted to the applied preprocessing techniques. These findings underscore the importance of customized preprocessing techniques in effectively addressing the challenges posed by imbalanced datasets, particularly in the context of diabetes classification, delving into the complexities of handling such datasets and highlighting the significance of employing appropriate preprocessing strategies to improve the classification of imbalanced medical datasets, thereby augmenting predictive accuracy in critical healthcare applications. Finally, among the models tested, CatBoost demonstrated exceptional performance in handling imbalanced datasets, achieving an accuracy of 95.18%, precision of 91.10%, recall of 95.52%, and an F1 score of 93.26%.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Abdulhadi, N. and A. Al-Mousa, 2021 Diabetes Detection Using Machine Learning Classification Methods. In *2021 International Conference on Information Technology (ICIT)*, pp. 350–354, IEEE.

Anderson, J. P., J. R. Parikh, D. K. Shenfeld, V. Ivanov, C. Marks, *et al.*, 2016 Reverse Engineering and Evaluation of Prediction Models for Progression to Type 2 Diabetes: An Application of Machine Learning Using Electronic Health Records. Journal of Diabetes Science and Technology **10**: 6–18.

Arias-Duart, A., E. Mariotti, D. Garcia-Gasulla, and J. M. Alonso-Moral, 2023 A Confusion Matrix for Evaluating Feature Attribution Methods. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3709–3714, IEEE.

Association, A. D., 2009 Diagnosis and Classification of Diabetes Mellitus. Diabetes Care **32**: S62–S67.

Bhavsar, H. and M. H. Panchal, 2024 A Review on Support Vector Machine for Data Classification Unpublished.

Bhoi, S. K., S. K. Panda, K. K. Jena, P. A. Abhisekh, S. Sahoo, *et al.*, 2021 Prediction of Diabetes in Females of Pima Indian Heritage: A Complete Supervised Learning Approach.

Breiman, L., 2001 Random Forests. Machine Learning **45**: 5–32.

Chang, V., J. Bailey, Q. A. Xu, and Z. Sun, 2023 Pima Indians Diabetes Mellitus Classification Based on Machine Learning (ML) Algorithms. Neural Computing & Applications **35**: 16157–16173.

Chen, T. and C. Guestrin, 2016 XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM.

de Amorim, L. B. V., G. D. C. Cavalcanti, and R. M. O. Cruz, 2023 The choice of scaling technique matters for classification performance. Applied Soft Computing **133**: 109924.

D.K., T., P. B.G, and F. Xiong, 2019 Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques. Pattern Recognition Letters **128**: 544–550.

Duvva, P., 2024 Did the Confusion Matrix Ever Confuse You? https://medium.com/wicds/did-the-confusion-matrix-ever-confuse-you-5fe869c10739, Accessed: March 9, 2024.

Gosain, A. and S. Sardana, 2017 Handling Class Imbalance Problem Using Oversampling Techniques: A Review. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 79–85, IEEE.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, *et al.*, 2016 LightGBM: A Highly Efficient Gradient Boosting Decision Tree Unpublished.

Kumari, S., D. Kumar, and M. Mittal, 2021 An Ensemble Approach for Classification and Prediction of Diabetes Mellitus Using Soft Voting Classifier. International Journal of Cognitive Computing in Engineering **2**: 40–46.

LaValley, M. P., 2008 Logistic Regression. Circulation **117**: 2395–2399.

Luque, A., A. Carrasco, A. Martín, and A. de Las Heras, 2019 The Impact of Class Imbalance in Classification Performance Metrics Based on the Binary Confusion Matrix. Pattern Recognition **91**: 216–231.

Miao, Y., 2021 Using Machine Learning Algorithms to Predict Diabetes Mellitus Based on PIMA Indians Diabetes Dataset. In *2021 the 5th International Conference on Virtual and Augmented Reality Simulations*, pp. 47–53, ACM.

Milo, T. and A. Somech, 2020 Automating Exploratory Data Analysis via Machine Learning: An Overview. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 2617–2622, ACM.

Mousa, A., W. Mustafa, R. B. Marqas, and S. H. M. Mohammed, 2023 A Comparative Study of Diabetes Detection Using The Pima Indian Diabetes Database. University of Duhok Journal **26**: 277–288.

Murugan, S., P. K. Sivakumar, C. Kavitha, A. Harichandran, and W.-C. Lai, 2023 An Electro-Oculogram (EOG) Sensor's Ability to Detect Driver Hypovigilance Using Machine Learning. Sensors **23**.

Naz, H. and S. Ahuja, 2020 Deep Learning Approach for Diabetes Prediction Using PIMA Indian Dataset. Journal of Diabetes and Metabolic Disorders **19**: 391–403.

Patra, R. and B. Khuntia, 2021 Analysis and Prediction of Pima Indian Diabetes Dataset Using SDKNN Classifier Technique. IOP Conference Series: Materials Science and Engineering **1070**: 012059.

Powers, D. M. W., 2020 Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation .

Pradipta, G. A., R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, 2021 SMOTE for Handling Imbalanced Data Problem: A Review. In *2021 Sixth International Conference on Informatics and Computing (ICIC)*, pp. 1–8, IEEE.

Prokhorenkova, L., G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, 2019 CatBoost: Unbiased Boosting with Categorical Features Unpublished.

Rajni and A. Amandeep, 2019 RB-Bayes Algorithm for the Prediction of Diabetic in Pima Indian Dataset. International Journal of Electrical and Computer Engineering **9**: 4866–4872.

Ramyachitra, D. D. and P. Manikandan, 2014 Imbalanced Dataset Classification and Solutions: A Review. International Journal of Computing and Business Research **5**.

Raschka, S., 2014 An Overview of General Performance Metrics of Binary Classifier Systems Unpublished.

Rigatti, S. J., 2017 Random Forest. Journal of Insurance Medicine **47**: 31–39.

Salih, A. A. and A. M. Abdulazeez, 2021 Evaluation of Classification Algorithms for Intrusion Detection System: A Review. Journal of Soft Computing and Data Mining **2**.

Shivahare, B. D., J. Singh, V. Ravi, R. R. Chandan, T. J. Alahmadi, *et al.*, 2024 Delving into Machine Learning's Influence on Disease Diagnosis and Prediction. The Open Public Health Journal **17**: e18749445297804.

Sidey-Gibbons, J. A. M. and C. J. Sidey-Gibbons, 2019 Machine Learning in Medicine: A Practical Introduction. BMC Medical Research Methodology **19**: 64.

Sohil, F., M. U. Sohali, and J. Shabbir, 2013 *An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)*, volume 6. Springer, 7th edition.

Su, J., 2024 A Fast Decision Tree Learning Algorithm Unpublished.

Taunk, K., S. De, S. Verma, and A. Swetapadma, 2019 A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 1255–1260, IEEE.

How to cite this article: Abushahla, K. H., and Pala, M. A. Optimizing Diabetes Prediction: Addressing Data Imbalance with Machine Learning Algorithms. *ADBA Computer Science*, 1(1), 26-35, 2024.