Sinopfbd, 2024; 9(2): 369-387 **https://doi.org/10.33484/sinopfbd.1458580**

Sinop Üniversitesi Fen Bilimleri Dergisi
Sinop Uni J Nat Sci

E-ISSN: 2564-7873

https://dergipark.org.tr/tr/pub/sinopfbd

Classification of Heart Diseases with Ensemble Learning Algorithms

Kenan ERDEM[1] Elham Tahsin YASİN[2] Müslüme Beyza YILDIZ[3] and Murat KÖKLÜ[3]

How to cite: Erdem, K., Yasin, E. T., Yıldız, M. B., & Köklü, M. (2024). Classification of heart diseases with ensemble learning algorithms. *Sinop Üniversitesi Fen Bilimleri Dergisi, 9*(2), 369-387. https://doi.org/10.33484/sinopfbd.145880

Research Article

Corresponding Author
Murat KÖKLÜ
mkoklu@selcuk.edu.tr

ORCID of the Authors
K.E: 0000-0001-6002-5873
E.T.Y: 0000-0003-3246-6000
M.B.Y: 0009-0002-0231-687X
M.K: 0000-0002-2737-2360

Received: 25.03.2024
Accepted: 05.08.2024

Abstract

The heart is one of the vital organs of the human body. Preserving heart health is a crucial factor that affects our overall well-being. Heart diseases are considered a prominent health issue of our time and are recognized as one of the leading causes of death worldwide. This underscores the importance of the heart once again. Understanding this critical health issue better, developing early diagnosis techniques, and creating effective treatment plans require continuous research and effort. In this study, performance measurements of three different machine learning algorithms were obtained using a dataset with 18 features from 319795 records of individuals with and without heart disease. The research results indicate that ensemble methods (AdaBoost, Stacking, and Gradient Boosting) can be successfully applied in the diagnosis of heart disease. The classification accuracies of these algorithms are as follows: 88.80% for AdaBoost, 91.50% for Stacking, and 91.60% for Gradient Boosting. Results from this study indicate that successful methods can be used to diagnose heart disease.

Keywords: Heart disease, Artificial Intelligence Techniques, Diagnosis and Classification, Ensemble, Gradient Boosting

Kalp Hastalıklarının Topluluk Öğrenme Algoritmaları ile Sınıflandırılması

[1]Selcuk University, Faculty of Medicine, Department of Cardiology, Konya, Türkiye

[2]Selcuk University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Konya, Türkiye

[3]Selcuk University, Faculty of Technology, Department of Computer Engineering, Konya, Türkiye

Öz

Kalp, insan vücudunun hayati organlarından biridir. Kalp sağlığının korunması genel refahımızı etkileyen çok önemli bir faktördür. Kalp hastalıkları çağımızın en önemli sağlık sorunlarından biri olarak kabul edilmekte ve dünya çapında önde gelen ölüm nedenlerinden biri olarak kabul edilmektedir. Bu da kalbin önemini bir kez daha vurgulamaktadır. Bu kritik sağlık sorununu daha iyi anlamak, erken teşhis teknikleri geliştirmek ve etkili tedavi planları oluşturmak sürekli araştırma ve çaba gerektirmektedir. Bu çalışmada, kalp hastalığı olan ve olmayan bireylere ait 319795 kayıttan elde edilen 18 özellikli bir veri kümesi kullanılarak üç farklı makine öğrenimi algoritmasının performans ölçümleri elde edilmiştir. Araştırma sonuçları, topluluk yöntemlerinin (AdaBoost, Stacking ve Gradient Boosting) kalp hastalığı teşhisinde başarıyla uygulanabileceğini göstermektedir. Bu algoritmaların sınıflandırma doğrulukları aşağıdaki gibidir: AdaBoost için %88.80, Stacking için %91.50 ve Gradient Boosting için %91.60. Bu sonuçlar, kalp hastalığının teşhisinde kullanılabilecek başarılı yöntemlerin varlığını vurgulamaktadır.

Anahtar Kelimeler: Kalp Hastalığı, Yapay Zeka Teknikleri, Teşhis ve Sınıflandırma, Ensemble, Gradient Boosting

Introduction

In modern societies, heart disease has become a significant health concern. Recognized as one of the leading causes of death worldwide, heart disease plays a central role in medical research. Continuous efforts have been made to gain a deeper understanding of the causes and effects of these diseases and to develop effective diagnostic and treatment methods [1]. Following a better understanding of this crucial health problem, new technologies such as machine learning are gaining importance for their potential contributions in this field [2]. Moreover, the ability of machine learning algorithms to quickly and reliably analyze large amounts of data facilitates the diagnostic process for individuals with and without heart disease [3, 4].

In their study, Mohan et al. [5] achieved an accuracy level of 88.7% in the detection of heart disease using the Hybrid Random Forest with a Linear Model (HRFLM) model. They did not impose restrictions on feature selection and used all features. The heart disease data were collected from the UCI Machine Learning Repository, comprising 297 patient records in the dataset with 13 features. In cases where an individual does not have heart disease, the value is set to 0. For patients with heart disease, values range from 1 to 4, representing the severity of the disease, with scaling indicating seriousness (4 being the highest) [5].

Repaka et al. [6] employed the Naive Bayes Bayesian algorithm in the design and implementation of Smart Heart Disease Prediction (SHDP). They utilized the UCI dataset, allocating 80% of the dataset for training and the remaining 20% for testing. With the Naive Bayes Bayesian algorithm, they achieved an accuracy of 89.77% [6].

Anitha & Sridevi [7] utilized the UCI dataset. When comparing the KNN, Naive Bayes, and SVM algorithms, they found that the Naive Bayes algorithm detected heart disease with an accuracy of 86.6% [7].

Shah et al. [8] aimed to conceptualize the probability of developing heart disease in patients. They used the dataset available from the Cleveland database in the UCI repository for patients heart disease. The dataset contained 303 samples and 76 features, however they considered only 14 features for testing. They utilized the WEKA tool for preprocessing the dataset in ARFF format (attribute-relation file format). The K-Nearest Neighbors, Naive Bayes, and Random Forest algorithms showed the best results in this model, achieving accuracies of 78.94%, 88.15%, and 84.21%, respectively [8].

Motarwar et al. [9] utilized the Cleveland dataset in their research. They trained the model using 80% of the data (242 samples) and predicted the remaining 20% (61 samples). To predict the probability of developing heart disease, they employed machine learning algorithms, such as Random Forest, Naive

Bayes, Support Vector Machine, Hoeffding Decision Tree, and Logistic Model Tree. Random Forest achieved the highest accuracy with an initial accuracy of 88.52% [9].

Junaid and Kumar [10] employed a hybrid algorithm in their study, combining Naive Bayes, Support Vector Machine (SVM), and Artificial Neural Network (ANN) algorithms. The accuracy, precision, and recall values they obtained were 88.54%, 82.11%, and 91.47%, respectively [10].

Sharma and Parmar [11] utilized the UCI dataset in their study for the detection of heart disease. They evaluated algorithms such as KNN, SVM, Naive Bayes, and Random Forest. Deep Neural Networks (DNN) using Talos optimization outperformed other optimizations, providing a higher accuracy of 90.76% [11].

Anbuselvan [12] utilized the UCI machine learning dataset for their project. In supervised learning models, Logistic Regression, and the ensemble technique XGBoost, Random Forest achieved better results with an accuracy of 86.89% compared to other methods such as Naive Bayes, Support Vector Machine, K-Nearest Neighbors, and Decision Tree algorithms [12].

Kavitha et al. [3] proposed a new machine learning approach to predict heart disease in their project. They used the Cleveland heart disease dataset, which contained 303 samples and approximately 14 features. Seventy percent of the dataset was used for training, and the remaining 30% was used for testing. The hybrid model, consisting of a combination of Random Forest and Decision Tree, demonstrated an accuracy level of 88.7% [3].

Rani et al. [13] in their research on predicting heart disease, employed Support Vector Machine, Naive Bayes, Logistic Regression, Random Forest, and Adaboost classifiers. The Random Forest classifier yielded the most accurate results with an accuracy rate of 86.6%. They used the Cleveland heart disease dataset from the UCI (University of California, Irvine) machine learning repository [13].

Jindal et al. [14] developed a cardiovascular disease detection model in their study using three machine learning classification models (Logistic Regression, Random Forest, and KNN). They utilized the UCI repository for their dataset, which includes 304 patients from different age groups and 13 medical features. The model which was applied using the KNN and Logistic Regression, achieved an average accuracy of 85%. Among these algorithms, KNN was the most effective, reaching an accuracy of 88.52% [14].

Goel [15] collected a dataset consisting of 13 features and 383 individual values. Among the algorithms, Logistic Regression, KNN, Naïve Bayes, Decision Tree, and Random Forest, SVM achieved the highest accuracy rate of 86% [15].

Boukhatem et al. [16] utilized four classification methods, namely Multilayer Perceptron (MLP), Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB), for cardiovascular disease detection in their study. The SVM model exhibited the best performance with an accuracy of 91.67%. They used the Heart Disease UCI dataset from Kaggle for cardiovascular disease detection [16].

Sugendran and Sujatha [17] used an Enhanced Genetic Algorithm (EGA) based Fuzzy Weight update Support Vector Machine (FWSVM) algorithm in their research to predict the early stages of heart disease. They employed the Cleveland heart disease dataset from the open-source UCI repository to validate their proposed model. The dataset contains 303 samples and 76 features. The EGA-FWSVM classifier, utilizing fuzzy weighted evaluation, achieved an accuracy of 91.68% [17].

Erdem et al. [18] emphasized the significance of early diagnosis and identification of risk factors in combating heart disease, a leading cause of global mortality. Recognizing the challenges in traditional diagnosis methods, the study explores the efficacy of seven machine learning algorithms on a dataset with 4238 records and 16 patient characteristics. Naive Bayes, Decision Trees, Random Forests, Support Vector Machines (SVM), Artificial Neural Networks (ANNs), K Nearest Neighbors, and Logistic Regressions achieve accuracies of 78.9%, 79.9%, 83.9%, 70.9%, 83.7%, 83.4%, and 85.5%, respectively [18].

The literature in this section delves deeply into the topic of heart disease by various aspects, risk factors, and potential treatments. Additionally, it focuses on the application of artificial intelligence techniques, such as machine learning for the classification and detection of heart diseases. Throughout this section, numerous studies have been conducted to illuminate advancements and findings in the field of cardiology. The data presented in Table 1 encompasses previous research related to heart diseases.

Different machine learning algorithms have been suggested for the classification of heart diseases. The results obtained in the research using ensemble methods in the effective diagnosis of heart disease, paving the way for more accurate and efficient diagnostic tools in healthcare.

***Table 1**. Summary of previously published studies on heart diseases*

Dataset	Methods	Accuracy	References
297	Hybrid Random Forest with Linear Model (HRFLM)	88.70%	[5]
-	Naive Bayes	89.70%	[6]
76 features and 14 attributes	Naive Bayes	86.6%	[7]
303 samples and 76 features	Naive Bayes	88.15%	[8]
303	Random Forest	88.52%	[9]
76 features and 14 attributes	Hybrid Naïve Bayes, Support Vector Machine, and Artificial Neural Network	88.54%	[10]
303 samples and 14 features	Using Talos optimization for Deep Neural Network	90.76%	[11]
303 samples and 14 features	Random Forest	86.89%	[12]
303 samples and 14 features	Hybrid Random Forest and Decision Tree	88.70%	[3]
303 samples and 76 features	Random Forest	86.60%	[13]
13 medical features and 304 patients	K-Nearest Neighbor	88.52%	[14]
13 features and 383 individual values	Support Vector Machine	86%	[15]

Table 1 continued…

Dataset	Methods	Accuracy	References
303 samples and 13 features	Support Vector Machine	91.67%	[16]
303 samples and 76 features	EGA-FWSVM	91.68%	[17]
4238 records and 16 patient characteristics	NB, DT, RF, SVM, ANNs, KNN, LR.	78.90%, 79.90%, 83.90%, 70.90%, 83.70%, 83.40%, 85.50%	[18]

Materials and Methods

The scope of the article involves the use of a single dataset for the detection of heart disease. For this purpose, AdaBoost, Stacking, and Gradient Boosting algorithms were employed. The steps followed to complete the research are illustrated in Figure 1, and the study was conducted successfully.

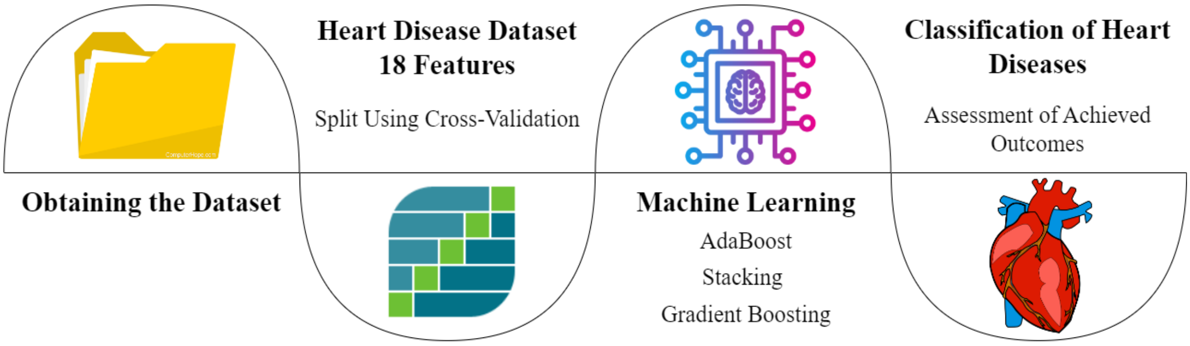


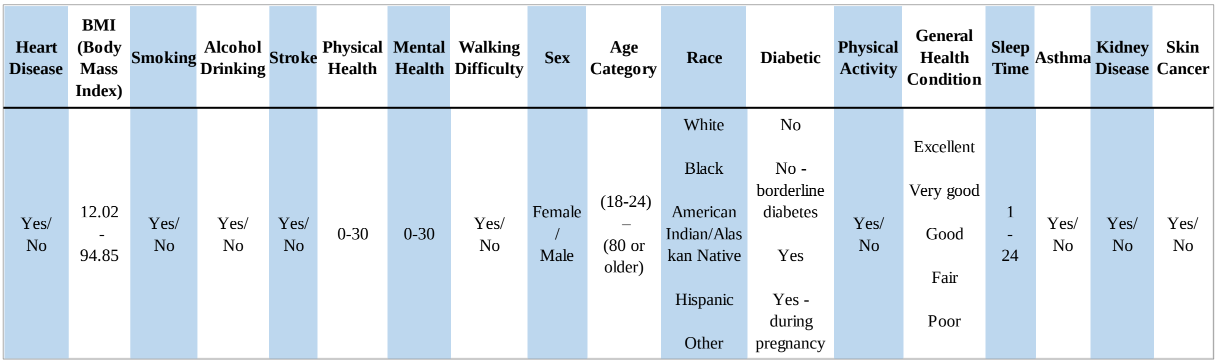
Figure 1. *Flow diagram depicting the evaluation of classification performances for heart disease detection*

Dataset

The dataset used for diagnosing heart disease is the (Heart Disease) dataset, obtained from the Kaggle data sharing site. Originally published by Abu Bakar Siddique Mahi [19], this dataset encompasses 18 different patient features [20-23]. In total, there are 319795 records were included in the dataset. The values and value ranges of the features in this dataset are presented in Figure 2 [24]. The data were split into training and testing sets using the cross-validation technique. Dividing the dataset into 10 and using 1 part as test and the rest as train. Then changing the test in the second fold and leaving the rest as train data. Table 2 provides an overview of the patient characteristics in the heart disease dataset.

Table 2. *Patient characteristics in the heart disease dataset*

Patient Characteristics					
1	Heart Disease	7	Mental Health Status	13	Physical Activity
2	BMI (Body Mass Index)	8	Walking Difficulty	14	General Health Status
3	Smoking	9	Gender	15	Sleep Duration
4	Alcohol Consumption	10	Age Category	16	Asthma
5	Stroke	11	Race	17	Kidney Disease
6	Physical Health Status	12	Diabetes Status	18	Skin Cancer



Heart Disease	BMI (Body Mass Index)	Smoking	Alcohol Drinking	Stroke	Physical Health	Mental Health	Walking Difficulty	Sex	Age Category	Race	Diabetic	Physical Activity	General Health Condition	Sleep Time	Asthma	Kidney Disease	Skin Cancer
Yes/ No	12.02 - 94.85	Yes/ No	Yes/ No	Yes/ No	0-30	0-30	Yes/ No	Female / Male	(18-24) – (80 or older)	White Black American Indian/Alaskan Native Hispanic Other	No No - borderline diabetes Yes Yes - during pregnancy	Yes/ No	Excellent Very good Good Fair Poor	1 - 24	Yes/ No	Yes/ No	Yes/ No

Figure 2. *Values and value ranges of the features in the dataset*

Cross Validation

Cross-validation is an important evaluation method that better assesses how a machine learning model will generalize to real-world data and measures the model's performance more reliably. Another objective of the model is to detect issues such as overfitting. The most common cross-validation technique is known as "k-fold cross-validation" [25]. In this method, the dataset is divided into k subsets. Then, the model is trained and tested k times. For each training-test pair, performance metrics of the model are recorded. Ultimately, a performance value is obtained based on the number of iterations. K-fold cross-validation may incur additional costs, particularly in large datasets, as it requires k rounds of model training and testing [26]. Although extra time is recuired to ensure a more accurate performance assessment of the model, this is disadvantage [27]. The diagram of the cross-validation method is presented in Figure 3.

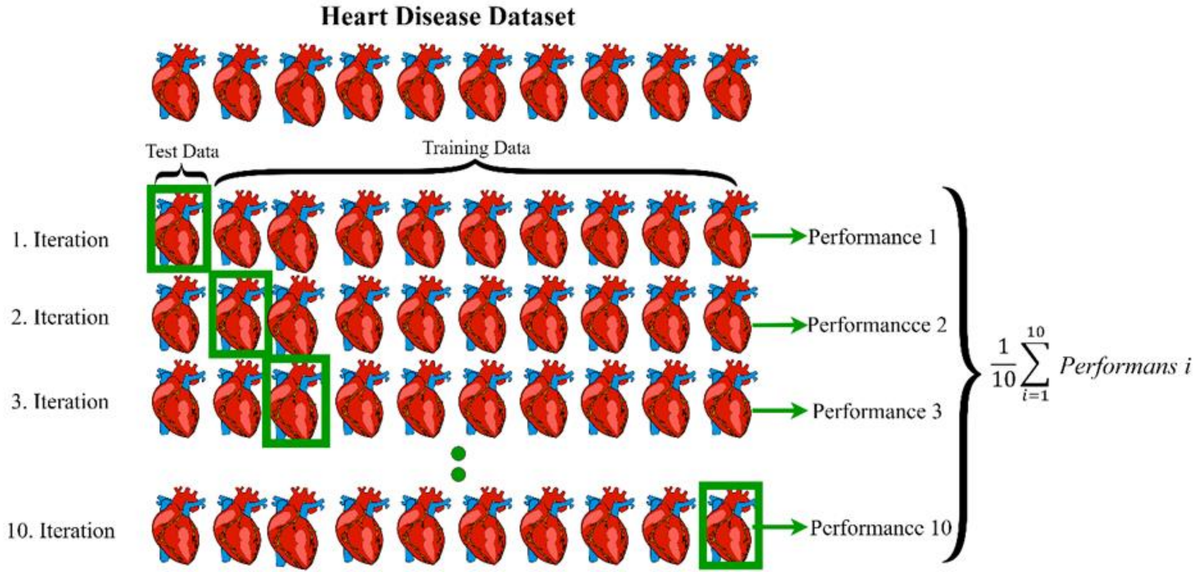


Figure 3. *Diagram of the cross-validation method*

Performance Metric and Confusion Matrix

The confusion matrix is a metric table used to evaluate the performance of classification algorithms in machine learning and statistical modeling. In classification problems, it can be considered of as a summary table that shows instances of data that a model classifies correctly or incorrectly. For a binary classification problem, the confusion matrix attempting to distinguish between two classes includes four different terms: True Positives, True Negatives, False Positives, and False Negatives [28, 29]. These terms, and their explanations, are included in Table 3.

Table 3. *Confusion matrix with explanations*

		Actual Class	
		Positive	***Negative***
Predicted Class	***Positive***	**TP** True Positives (TP) represents the number of data points that the model correctly predicted as positive.	**FP** False Positives (FP) represents the number of data points that the model incorrectly predicted as positive when they actually should be negative.
	Negative	**FN** False Negatives (FN) represents the number of data points that the model incorrectly predicted as negative when they actually should be positive.	**TN** True Negatives (TN) represents the number of data points that the model correctly predicted as negative.

A performance metric is a criterion or measure used to assess and evaluate the success level of a model or system. These metrics help us understand the strengths and weaknesses of a model by determining various features such as accuracy, precision, recall, F1 score, among others [30]. Confusion matrix is used to calculate various measurements when evaluating the performance of a model [28, 31].

Accuracy is a good choice when there is a balanced distribution among classes in the dataset, and the sizes of the classes are similar. It shows the ratio of correctly predicted data points to all data points [32].

$$(TN + TP)/(TN + FP + TP + FN) \tag{5}$$

Precision is a good choice when the class distribution is imbalanced or when the cost of false positives is high. It shows the ratio of correctly predicted positive data points to the total predicted positive data points.

$$TP/(TP + FP) \tag{6}$$

Recall, also known as Sensitivity, is important when the cost of false negatives (FN) is high or when the class of primary interest is rare and crucial. It shows the ratio of correctly predicted positive data points to the total actual positive data points.

$$TP/(TP + FN) \tag{7}$$

F1 Score is important when there is imbalance among classes or when the cost of false positives and false negatives is comparable. It is a metric that combines precision and recall.

$$2 * (Precision * Recall)/(Precision + Recall) \tag{8}$$

Ensemble Learning Techniques in Machine Learning

Throughout this research endeavor, we leveraged the capabilities of AdaBoosting, Stacking, and Gradient Boosting techniques to refine our analytical framework. This section provides a nuanced exposition of each method, elucidating their distinct applications and contributions to the overarching methodology implemented in our study.

Ensemble methods provide notable benefits compared with individual machine learning techniques, principally because they can merge many models to attain superior performance and generalization. These techniques frequently produce greater accuracy than individual models by mitigating the risk of overfitting and enhancing resilience. Ensemble approaches can decrease the variability of predictions by taking the averaging of numerous models. This is especially advantageous for minimizing the influence of outliers and noise in the data. Ensemble approaches have a tendency to exhibit superior generalization capabilities when applied to unknown data, leading to more dependable and consistent predictions. Moreover, these techniques are adaptable and varied, able to integrate several models such as decision trees, neural networks, and logistic regression, so utilizing the advantages of each to enhance overall performance. Boosting and bagging are techniques that are especially developed to address the problem of overfitting. Boosting aims to repair the errors made by weak learners, while bagging decreases overfitting by averaging the predictions of many models trained on distinct subsets of the data. When comparing the three main ensemble approaches - boosting, stacking, and bagging - it becomes apparent that each method has its own distinctive attributes and advantages. Boosting is a technique that applies weak learners to the data in a sequential manner, where each learner corrects the errors made by the previous one. This method provides exceptional precision and the capability to manage intricate data patterns, demonstrating excellent performance even on datasets with uneven distributions. Nevertheless, the process of boosting might be susceptible to overfitting if not adequately regularized and requires significant computer resources. Some examples of algorithms are AdaBoost, Gradient Boosting, and XGBoost. Stacking is a technique that entails training several base learners and then using a meta-learner to merge their predictions. This approach provides significant adaptability by integrating various models, frequently resulting in enhanced performance by capitalizing on the advantages of distinct models. Nevertheless, the implementation and fine-tuning of stacking are more intricate, resulting in a greater computational burden since many models need to be trained. Bagging is a technique that entails training numerous models separately on various subsets of the data, which are generated using bootstrapping. The predictions of these models are then averaged. This method effectively decreases variance and overfitting, is straightforward to execute, and may be parallelized. Nevertheless, its performance is diminished when applied to datasets with significant bias and it can be computationally burdensome when dealing with really big datasets. Some examples of algorithms are Random Forest and Bagged Decision Trees.

AdaBoost

In the initial stage, a weak learner model is created when data samples have equal weights. It is evaluated on the data samples, and as misclassified examples' weights increase, the weights of correctly classified examples decrease. Then, subsequent models are created by focusing on the previous errors. The predictions of all models are combined predominantly with weights [33]. The combination of these steps results in a strong model. AdaBoost can achieve higher accuracy in classification problems by combining low-performing models [34]. The diagram of a two-class AdaBoost classifier designed to distinguish individuals with and without heart disease is shown in Figure 4.

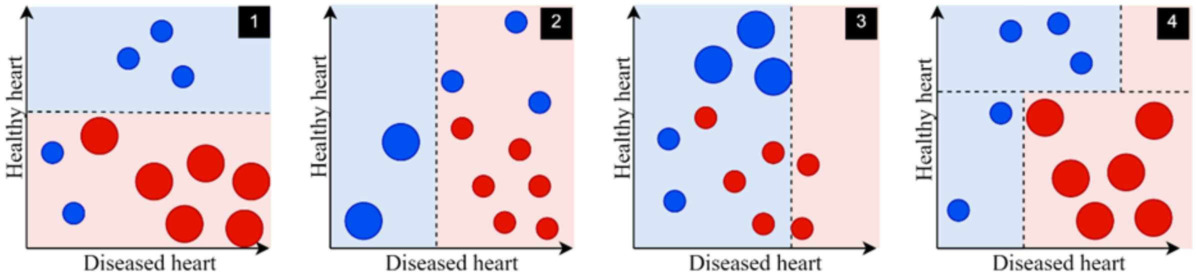


Figure 4*. Two-Class AdaBoost diagram*

Stacking

Stacking is a machine learning algorithm that is used as an ensemble learning technique. Ensemble learning aims to improve predictions or classifications by combining the results of multiple learning algorithms. Stacking is a new framework in ensemble learning that uses meta-learners to combine the results generated by each base learner [35, 36]. Base learners are referred to as first-level learners, and combiners are called meta-learners or second-level learners. Stacking first trains the first-level learner using the initial training dataset. Then, the output of the first-level learner is used as the input feature for the meta-learner. Finally, a new dataset is created by using the relevant original labels as new labels to train the meta-learner. If the learners at the first level use the same type of learning algorithm, theyit are called homogeneous ensembles; otherwise, they are called heterogeneous ensembles [37-40]. The diagram of an example Stacking classifier is shown in Figure 5. Stacking is a method used in machine learning that creates a meta-model by combining numerous basic models. The Stacking widget incorporates an Aggregate input that is used to merge the input models. The models used for the heart disease dataset were AdaBoost and Gradient Boosting.

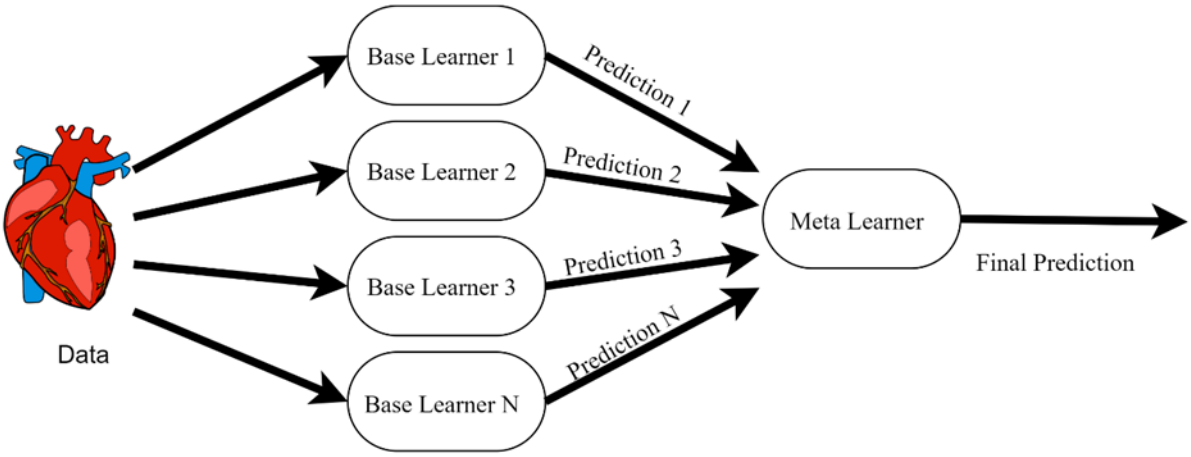


Figure 5. *Stacking diagram*

Gradient Boosting

Gradient Boosting is a widely used ensemble learning method in the field of machine learning. This technique builds weak learner models sequentially, allowing each subsequent model to focus on the errors of the previous ones. As a result, a new model is created, leading to the development of a strong predictive model. Gradient Boosting is particularly effective in regression and classification problems, providing high performance [41]. The diagram of the gradient boosting algorithm is shown in Figure 6.

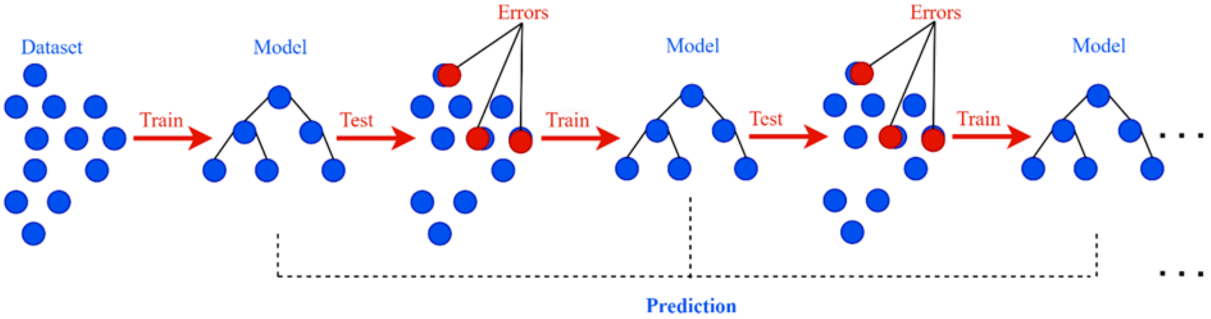


Figure 6. Diagram of the Gradient Boosting algorithm

The AdaBoost method was used in our investigation with predefined parameter settings. The decision tree was chosen as the base estimator, and a total of 50 estimators were used in the boosting procedure. The learning rate for AdaBoost was adjusted to 1.00000, which increased the importance of the contribution of each weak learner. In addition, the classification boosting approach utilized the SAMME.R algorithm, whereas for regression tasks, the loss function used was linear. A specific set of parameters was employed to efficiently train the models using gradient boosting. In this instance, the ensemble employed a total of 100 trees, while maintaining a moderate learning rate of 0.100. In order to manage the growth of trees in the group, a particular parameter was used to restrict the number of levels in each tree to a maximum of 3. The parameter configurations were meticulously selected to strike a balance between the complexity of the model and its predictive accuracy across different tasks in our investigation.

Experimental Results

The classification results using AdaBoost, Stacking, and Gradient Boosting methods are presented in this section. In the dataset used in the study, there are a total of 319795 records. The hardware specifications used to run these algorithms are shown in Table 4.

***Table 4.** Specifications of the hardware used in the study*

HARDWARE UNIT	FEATURES
CPU	Intel® Core i7™ 12700 K 3.61 GHz
RAM	64 GB
Graphics Card	NVIDIA GeForce RTX 3080 Ti
Operating System	Windows 11

In the study, confusion matrices were used to evaluate the performance of classification algorithms. A separate confusion matrix was created for each classification algorithm [42-44], and performance analyses were conducted using the TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) values in these matrices. During the training of the algorithms, cross-validation was employed to achieve a more accurate classification result. In this method, the k value was set to 10. The average classification accuracies of the AdaBoost, Stacking, and Gradient Boosting methods applied in the study are summarized in Table 5.

***Table 5**. Performance metric results for the applied methods*

	AdaBoost	**Stacking**	**Gradient Boosting**
Accuracy	88.80%	91.50%	91.60%
Precision	87.0%	89.0%	89.0%
Recall	88.80%	91.5%	91.6%
F1-Score	87.80%	89.3%	89.0%

With the AdaBoost algorithm, the classification of heart disease achieved an accuracy rate of 88.80% in 273.46 seconds of training time and 5.02 seconds of testing time. The Stacking algorithm achieved an accuracy of 91.50% with 2777.97 seconds of training time and 6.20 seconds of testing time. The most impressive result was obtained with the Gradient Boosting algorithm, which classified heart disease with an accuracy rate of 91.60%, 291.04 seconds of training time, and 1.15 seconds of testing time. A comparison of the performance times (training time and testing time) of the algorithms is better visualized in Figure 7 and Figure 8.

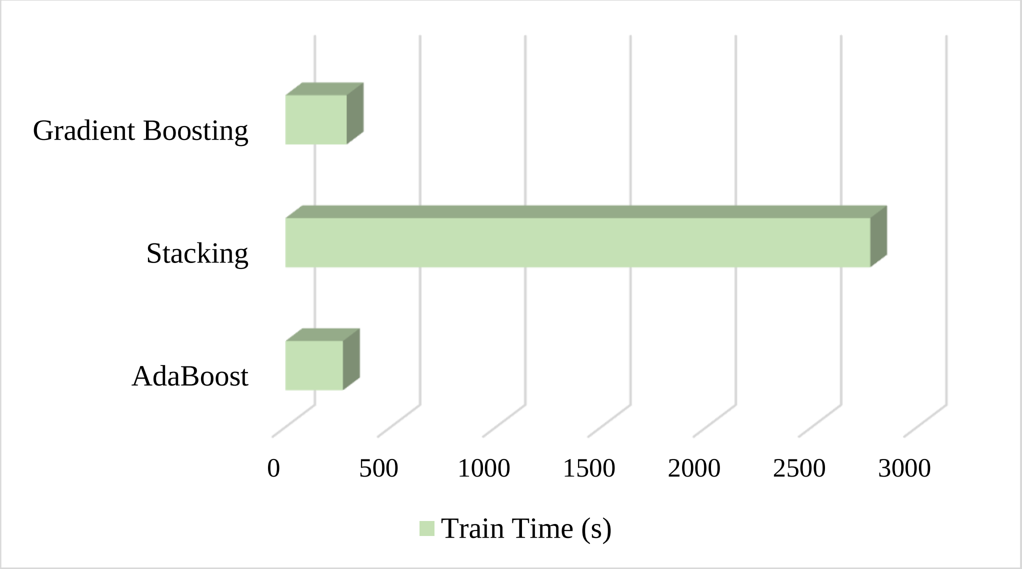


***Figure 7**. Training time graph for all machine learning algorithms*

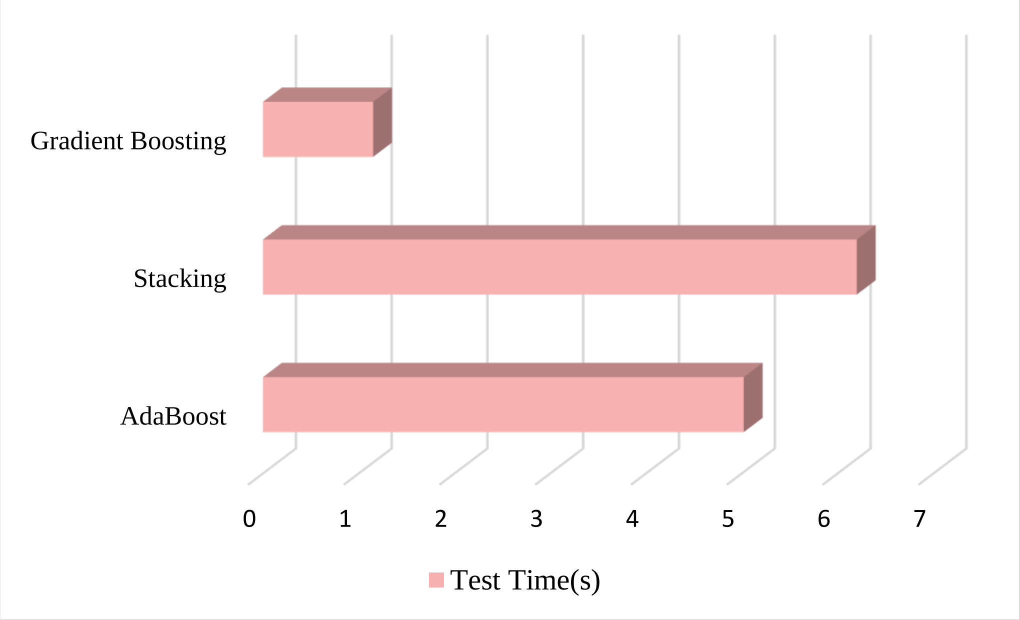
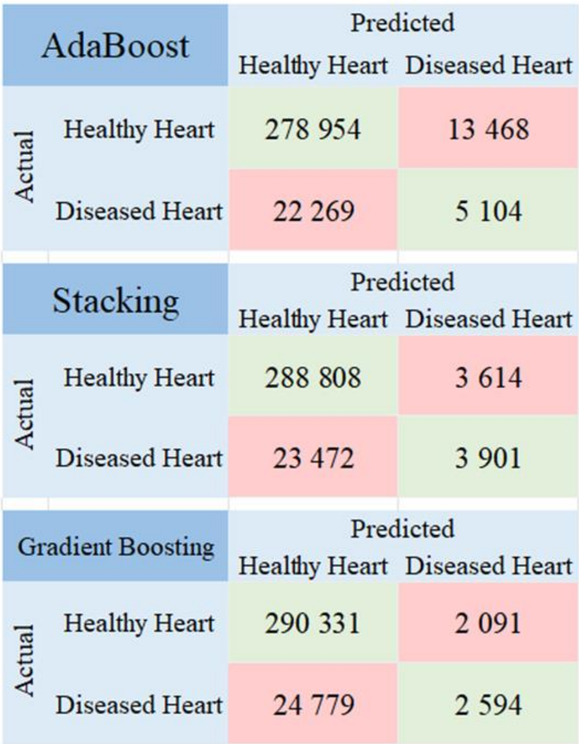


Figure 8. *Testing time graph for all machine learning algorithms*

Figure 9 includes the confusion matrix for machine learning algorithms. Without any feature extraction, AdaBoost achieved an accuracy of 88.80%, Stacking 91.50%, and Gradient Boosting 91.60%. Based on the confusion matrix in Figure 9, when classifying the diagnosis of heart disease, it correctly classified 290331 records as a healthy heart and predicted them as a healthy heart while using Gradient Boosting. At the same time, using Gradient Boosting, it correctly identified 2594 images as a diseased heart. The Gradient Boosting model misclassified 2091 images from healthy heart images and 24779 images from diseased heart images, as shown in Figure 9.



AdaBoost		Predicted	
		Healthy Heart	Diseased Heart
Actual	Healthy Heart	278 954	13 468
	Diseased Heart	22 269	5 104

Stacking		Predicted	
		Healthy Heart	Diseased Heart
Actual	Healthy Heart	288 808	3 614
	Diseased Heart	23 472	3 901

Gradient Boosting		Predicted	
		Healthy Heart	Diseased Heart
Actual	Healthy Heart	290 331	2 091
	Diseased Heart	24 779	2 594

Figure 9*. Confusion matrix for AdaBoost, Stacking, and Gradient Boosting algorithms*

Despite not finding any articles for comparison on the same dataset, we discovered Kaggle code executions using the same dataset. The dataset's link, as provided on Kaggle, was included in the data availability section. Notably, the machine learning results obtained in our study are higher than those reported on Kaggle.

Receiver Operating Characteristic (ROC) curve is used to evaluate the performance of classification models by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings, with the area under the ROC curve (AUC) serving as a measure of the model's ability to distinguish between classes. AdaBoost combines multiple weak learners to create a strong classifier, with each subsequent model attempting to correct the errors of the previous models. Gradient Boosting operates similarly to AdaBoost but builds models sequentially, training each new model to correct the errors made by its predecessors. Stacking involves training multiple models, such as AdaBoost and Gradient Boosting, and then combining their predictions using another model to improve overall performance. The ROC curve is shown in Figure 10.

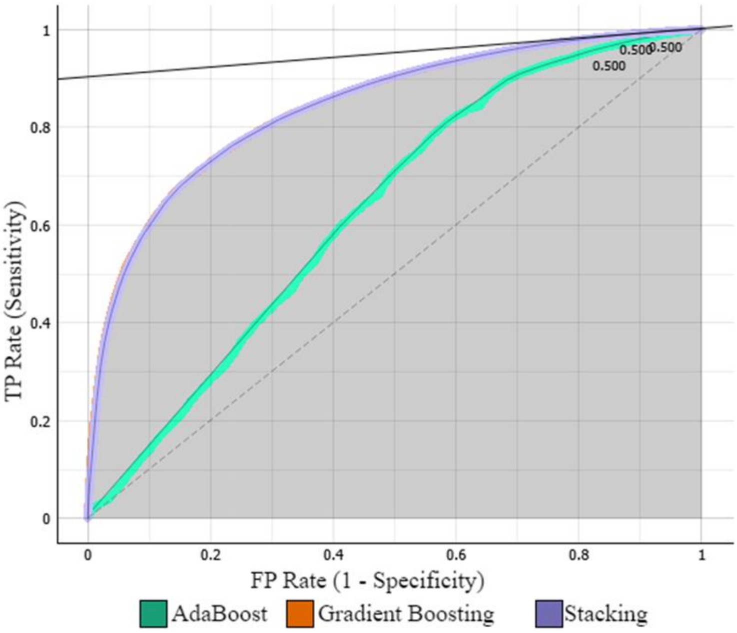


Figure 10. *ROC curve for AdaBoost, Stacking, and Gradient Boosting algorithms*

The ROC results focus on identifying the 'No' class, indicating individuals without heart disease. Both false positive and false negative prediction errors have an associated cost of 500. With a target probability threshold of 91.0%, models predicting a 91.0% chance or higher of an individual not having heart disease classify them as 'No.' A higher Area Under the Curve (AUC) value suggests a better-performing model. Comparing AUC values for AdaBoost, Gradient Boosting, and the stacking ensemble helps identify which model best distinguishes between individuals with and without heart disease. Since false positive and false negative costs are equal, balancing sensitivity (True Positive Rate) and specificity (1 - False Positive Rate) is essential. The 91.0% threshold reflects a conservative approach to avoid false negatives.

Conclusion

The present study assessed the efficacy of three ensemble machine learning algorithms—AdaBoost, Stacking, and Gradient Boosting—by analyzing a dataset consisting of 319795 records with 18 variables pertaining to heart disease. The efficacy of these algorithms was evaluated by doing statistical analysis on the confusion matrices obtained from their classification outcomes.

Of the algorithms assessed, Gradient Boosting proved to be the most effective, attaining an accuracy rate of 91.6%. It had a training time of 291.04 seconds and a testing time of 1.15 seconds. The performance of the algorithm highlights its ability process extensive datasets and a multitude of attributes efficiently. The Stacking approach achieved a high accuracy rate of 91.5%, but it necessitated a somewhat longer training duration. Although AdaBoost achieved an accuracy of 88.8%, it outperformed the other two algorithms.

The study emphasizes the potential of ensemble approaches in the early detection of cardiac disease. The exceptional efficacy of Gradient Boosting, specifically, indicates its appropriateness for medical applications of this nature. Subsequent investigations may delve into the incorporation of supplementary data mining methodologies and the creation of more intricate models to augment the predicted precision for diagnosing heart disease.

In its entirety, the research highlights the effectiveness of ensemble machine learning algorithms in identifying heart illness, particularly Gradient Boosting, which stands out for its quick and precise performance. This study establishes the foundation for future research endeavors focused on enhancing early detection and treatment approaches using sophisticated machine learning methods.

Data Availability

The dataset can be accessed using the links provided:

https://www.kaggle.com/datasets/abubakarsiddiquemahi/heart-disease-dataset

https://www.kaggle.com/code/sumitkumarprasad/heart-disease-prediction-with-gradio-deployment

Acknowledgement We would like to thank the Scientific Research Coordinator of Selcuk University for their support with the project titled "Diagnosis and Classification of Heart Disease with Artificial Intelligence Techniques" numbered 23401163

Funding/Financial Disclosure The authors declare that they have no financial interests or relationships pertaining to the publication of this article

Ethics Committee Approval and Permissions The study does not require ethics committee approval or any special permission.

Conflicts of Interest The authors declared no conflict of interest.

Authors Contribution All authors read and approved the final manuscript.

References

[1] Erdem, K., & Duman, A. (2023). Pulmonary artery pressures and right ventricular dimensions of post-COVID-19 patients without previous significant cardiovascular pathology. *Heart & Lung*, *57*, 75-79. https://doi.org/10.1016/j.hrtlng.2022.08.023

[2] Erdem, K., Kobat, M. A., Bilen, M. N., Balik, Y., Alkan, S., Cavlak, F., Poyraz, A. K., Barua, P. D., Tuncer, I., & Dogan, S. (2023). Hybrid-Patch-Alex: A new patch division and deep feature extraction-based image classification model to detect COVID-19, heart failure, and other lung conditions using medical images. *International Journal of Imaging Systems and Technology*, *33*(4), 1144-1159. https://doi.org/10.1002/ima.22914

[3] Kavitha, M., Gnaneswar, G., Dinesh, R., Sai, Y. R., & Suraj, R. S. (2021). Heart disease prediction using hybrid machine learning model. 2021 6th international conference on inventive computation technologies (ICICT). Coimbatore, India, 1329-1333. https://doi.org/10.1109/ICICT50816.2021.9358597.

[4] Buber, M., Fadime, S., Bulut, I., & Kursun, R. (2015). Cloud computing environments which can be used in health education. *International Journal of Intelligent Systems and Applications in Engineering*, *3*(4), 124-126. https://doi.org/10.18201/ijisae.92756

[5] Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access*, *7*, 81542-81554. https://doi.org/10.1109/ACCESS.2019.2923707

[6] Repaka, A. N., Ravikanti, S. D., & Franklin, R. G. (2019). Design and implementing heart disease prediction using naives Bayesian. 2019 3rd International conference on trends in electronics and informatics (ICOEI). Tirunelveli, India, 292-297, https://doi.org/10.1109/ICOEI.2019.8862604

[7] Anitha, S., & Sridevi, N. (2019). Heart disease prediction using data mining techniques. *Journal of Analysis and Computation*, *7*(2), 48-55.

[8] Shah, D., Patel, S., & Bharti, S. K. (2020). Heart disease prediction using machine learning techniques. *SN Computer Science*, *1*, 1-6. https://doi.org/10.1007/s42979-020-00365-y

[9] Motarwar, P., Duraphe, A., Suganya, G., & Premalatha, M. (2020). Cognitive approach for heart disease prediction using machine learning. 2020 international conference on emerging trends in information technology and engineering (ic-ETITE). Vellore, India, 1-5, https://doi.org/10.1109/ic-ETITE47903.2020.242

[10] Junaid, M. J. A., & Kumar, R. (2020). Data science and its application in heart disease prediction. 2020 International Conference on Intelligent Engineering and Management (ICIEM). London, UK, 396-400, https://doi.org/10.1109/ICIEM48762.2020.9160056

[11] Sharma, S., & Parmar, M. (2020). Heart diseases prediction using deep learning neural network model. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *9*(3), 2244-2248. https://doi.org/10.35940/ijitee.C9009.019320.

[12] Anbuselvan, P. (2020). Heart disease prediction using machine learning techniques. *International Journal of Engineering Research & Technolog*, *9*(11), 515-518.

[13] Rani, P., Kumar, R., Ahmed, N. M. S., & Jain, A. (2021). A decision support system for heart disease prediction based upon machine learning. *Journal of Reliable Intelligent Environments*, *7*(3), 263-275. https://doi.org/10.1007/s40860-021-00133-6

[14] Jindal, H., Agrawal, S., Khera, R., Jain, R., & Nagrath, P. (2021). Heart disease prediction using machine learning algorithms. *IOP Conference Series: Materials Science and Engineering*, *1022*(1), 01-10. https://doi.org/ 10.1088/1757-899X/1022/1/012072

[15] Goel, R. (2021). Heart disease prediction using various algorithms of machine learning. Proceedings of the International Conference on Innovative Computing & Communication (ICICC). Delhi, India, https://dx.doi.org/10.2139/ssrn.3884968

[16] Boukhatem, C., Youssef, H. Y., & Nassif, A. B. (2022). Heart disease prediction using machine learning. 2022 Advances in Science and Engineering Technology International Conferences (ASET). Dubai, United Arab Emirates, 1-6, https://doi.org/10.1109/ASET53988.2022.9734880

[17] Sugendran, G., & Sujatha, S. (2023). Earlier identification of heart disease using enhanced genetic algorithm and fuzzy weight based support vector machine algorithm. *Measurement: Sensors*, 100814. https://doi.org/10.1016/j.measen.2023.100814.

[18] Erdem, K., Yildiz, M. B., Yasin, E. T., & Koklu, M. (2023). A Detailed Analysis of Detecting Heart Diseases Using Artificial Intelligence Methods. *Intelligent Methods in Engineering Sciences*, *2*(4), 115-124. https://doi.org/10.58190/imiens.2023.4

[19] Mahi, A. B. S. (2023). Heart disease dataset (Version 1) [Dataset]. Kaggle. https://www.kaggle.com/datasets/abubakarsiddiquemahi/heart-disease-dataset, Community Data License Agreement – Sharing, Version 1.0

[20] Ozkan, I. A., & Koklu, M. (2017). Skin lesion classification using machine learning algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, *5*(4), 285-289.

[21] Ozkan, I. A., Koklu, M., & Sert, I. U. (2018). Diagnosis of urinary tract infection based on artificial intelligence methods. *Computer Methods and Programs in Biomedicine*, *166*, 51-59. https://doi.org/10.1016/j.cmpb.2018.10.007

[22] Koklu, M., & Unal, Y. (2013). Analysis of a population of diabetic patients databases with classifiers. *International Journal of Biomedical and Biological Engineering*, *7*(8), 481-483.

[23] Tunc, A., Tasdemir, S., Koklu, M., & Cinar, A. C. (2022). Age group and gender classification using convolutional neural networks with a fuzzy logic-based filter method for noise reduction. *Journal of Intelligent & Fuzzy Systems*, *42*(1), 491-501. https://doi.org/10.3233/JIFS-219206.

[24] Prasad S. k. (2022). Heart disease prediction with gradio deployment (Version 1) [Dataset]. Kaggle. https://www.kaggle.com/code/sumitkumarprasad/heart-disease-prediction-with-gradio-deployment/notebook

[25] Butuner, R., Cinar, I., Taspinar, Y. S., Kursun, R., Calp, M. H., & Koklu, M. (2023). Classification of deep image features of lentil varieties with machine learning techniques. *European Food Research and Technology*, *249*, 1303–1316. https://doi.org/10.1007/s00217-023-04214-z

[26] Taspinar, Y. S., Koklu, M., & Altin, M. (2021). Fire Detection in Images Using Framework Based on Image Processing, Motion Detection and Convolutional Neural Network. *International Journal of Intelligent Systems and Applications in Engineering*, *9*(4), 171-177. https://doi.org/10.18201/ijisae.2021473636

[27] Yasin, E. T., & Koklu, M. (2023, April 28-30). Classification of Organic and Recyclable Waste based on Feature Extraction and Machine Learning Algorithms. International Conference on Intelligent Systems and New Applications (ICISNA’23). Liverpool, United Kingdom. 59-65.

[28] Yasin, E. T., Ozkan, I. A., & Koklu, M. (2023). Detection of fish freshness using artificial intelligence methods. *European Food Research and Technology, 249,* 1979-1990. https://doi.org/10.1007/s00217-023-04271-4

[29] Koklu, M., & Sabanci, K. (2015). The classification of eye state by using kNN and MLP classification models according to the EEG signals. *International Journal of Intelligent Systems and Applications in Engineering*, *3*(4), 127-130.

[30] Cinar, I., & Koklu, M. (2021). Determination of effective and specific physical features of rice varieties by computer vision in exterior quality inspection. *Selcuk Journal of Agriculture and Food Sciences, 35*(3), 229-243.

[31] Al Bataineh, A., & Manacek, S. (2022). MLP-PSO hybrid algorithm for heart disease prediction. *Journal of Personalized Medicine*, *12*(8), 1208. https://doi.org/10.3390/jpm12081208

[32] Cinar, I., Taspinar, Y. S., Kursun, R., & Koklu, M. (2022). Identification of Corneal Ulcers with Pre-Trained AlexNet Based on Transfer Learning. 2022 11th Mediterranean Conference on Embedded Computing (MECO). Budva, Montenegro, 1-4. https://doi.org/10.1109/MECO55406.2022.9797218

[33] Tutuncu, K., Cinar, I., Kursun, R., & Koklu, M. (2022). Edible and poisonous mushrooms classification by machine learning algorithms. 2022 11th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 1-4. https://doi.org/10.1109/MECO55406.2022.9797212

[34] Mahesh, T., Dhilip Kumar, V., Vinoth Kumar, V., Asghar, J., Geman, O., Arulkumaran, G., & Arun, N. (2022). AdaBoost ensemble methods using K-fold cross validation for survivability with the early detection of heart disease. *Computational intelligence and neuroscience*, 2022, Article ID 9005278, https://doi.org/10.1155/2022/9005278.

[35] Cui, S., Yin, Y., Wang, D., Li, Z., & Wang, Y. (2021). A stacking-based ensemble learning method for earthquake casualty prediction. *Applied Soft Computing, 101,* 107038. https://doi.org/10.1016/j.asoc.2020.107038

[36] Taspinar, Y. S., Cinar, I., & Koklu, M. (2022). Classification by a stacking model using CNN features for COVID-19 infection diagnosis. *Journal of X-ray Science and Technology, 30*(1), 73-88.

[37] Chiu, C.-C., Wu, C.-M., Chien, T.-N., Kao, L.-J., Li, C., & Jiang, H.-L. (2022). Applying an improved stacking ensemble model to predict the mortality of ICU patients with heart failure. *Journal of Clinical Medicine*, *11*(21), 6460. https://doi.org/10.3390/jcm11216460

[38] Papouskova, M., & Hajek, P. (2019). Two-stage consumer credit risk modelling using heterogeneous ensemble learning. *Decision Support Systems*, *118,* 33-45. https://doi.org/10.1016/j.dss.2019.01.002.

[39] Jiang, M., Liu, J., Zhang, L., & Liu, C. (2020). An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms. *Physica A: Statistical Mechanics and its Applications*, *541*, 122272. https://doi.org/10.1016/j.physa.2019.122272

[40] Dong, Y., Zhang, H., Wang, C., & Zhou, X. (2021). Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm. *Neurocomputing*, *462*, 169-184. https://doi.org/10.1016/j.neucom.2021.07.084

[41] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review, 54*, 1937-1967. https://doi.org/10.1007/s10462-020-09896-5

[42] Koklu, M., Kahramanli, H., & Allahverdi, N. (2014). A new accurate and efficient approach to extract classification rules. *Journal of the Faculty of Engineering and Architecture of Gazi University, 29*(3), 477-486.

[43] Koklu, M., Kahramanli, H., & Allahverdi, N. (2012). A new approach to classification rule extraction problem by the real value coding. *International Journal of Innovative Computing, Information and Control, 8*(9), 6303-6315

[44] Koklu, M., Kahramanli, H., & Allahverdi, N. (2015. May 27-29). Applications of rule based classification techniques for thoracic surgery. Managing Intellectual Capital and Innovation for Sustainable and Inclusive Society: Managing Intellectual Capital and Innovation; Proceedings of the MakeLearn and TIIM Joint International Conference 2. Bari, Italy. 1991-1998.