Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Early heart disease prediction using feature engineering and machine learning algorithms

Mohammed Amine Bouqentar [a], Oumaima Terrada [a], Soufiane Hamida [a,b,d], Shawki Saleh [a], Driss Lamrani [a], Bouchaib Cherradi [a,b,c,*], Abdelhadi Raihani [a]

[a] *EEIS Laboratory, ENSET of Mohammedia, Hassan II University of Casablanca, Mohammedia, Morocco*
[b] *2IACS Laboratory, ENSET, University Hassan II of Casablanca, Mohammedia, Morocco*
[c] *STIE Team, CRMEF Casablanca-Settat. Provincial Section of El Jadida, El Jadida, 24000, Morocco*
[d] *GENIUS Laboratory, SupMTI of Rabat, Rabat, Morocco*

ARTICLE INFO

Keywords:
Cardiovascular diseases
Artificial intelligence
Machine learning
Deep learning
Classification
Prediction

ABSTRACT

Heart disease is one of the most widespread global health issues, it is the reason behind around 32 % of deaths worldwide every year. The early prediction and diagnosis of heart diseases are critical for effective treatment and sickness management. Despite the efforts of healthcare professionals, cardiovascular surgeons and cardiologists' misdiagnosis and misinterpretation of test results may happen every day. This study addresses the growing global health challenge raised by Cardiovascular Diseases (CVDs), which account for 32 % of all deaths worldwide, according to the World Health Organization (WHO). With the progress of Machine Learning (ML) and Deep Learning (DL) techniques as part of Artificial Intelligence (AI), these technologies have become crucial for predicting and diagnosing CVDs. This research aims to develop an ML system for the early prediction of cardiovascular diseases by choosing one of the powerful existing ML algorithms after a deep comparative analysis of several. To achieve this work, the Cleveland and Statlog heart datasets from international platforms are used in this study to evaluate and validate the system's performance. The Cleveland dataset is categorized and used to train various ML algorithms, including decision tree, random forest, support vector machine, logistic regression, adaptive boosting, and K-nearest neighbors. The performance of each algorithm is assessed based on accuracy, precision, recall, F1 score, and the Area Under the Curve metrics. Hyperparameter tuning approaches have been employed to find the best hyperparameters that reflect the optimal performance of the used algorithms based on different evaluation approaches including 10-fold cross-validation with a 95 % confidence interval. The study's findings highlight the potential of ML in improving the early prediction and diagnosis of cardiovascular diseases. By comparing and analyzing the performance of the applied algorithms on both the Cleveland and Statlog heart datasets, this research contributes to the advancement of ML techniques in the medical field. The developed ML system offers a valuable tool for healthcare professionals in the early prediction and diagnosis of cardiovascular diseases, with implications for the prediction and diagnosis of other diseases as well.

[*] Corresponding author. EEIS Laboratory, ENSET of Mohammedia, Hassan II University of Casablanca, Mohammedia, Morocco.
E-mail address: bouchaib.cherradi@gmail.com (B. Cherradi).

https://doi.org/10.1016/j.heliyon.2024.e38731
Received 11 January 2024; Received in revised form 28 September 2024; Accepted 28 September 2024
Available online 1 October 2024

1. Introduction

Cardiovascular Diseases (CVDs) have become a major global health issue, with 32 % of all deaths worldwide being attributed to these diseases, resulting in 17.9 million deaths per year according to the World Health Organization (WHO).[1] In this regard, Machine Learning (ML) and Deep Learning (DL) techniques as branches of Artificial Intelligence (AI), have become notably more important tools for scientists and medical professionals in their efforts to predict and diagnose CVDs [1].

Artificial Intelligence is a large term that has several meanings. Its significance has been evolving and can change depending on its application field. In other words, AI can be briefly defined as the use of machines with learning abilities that can be similar to the cognitive functions of human beings [2]. In fact, ML is an essential component of the field of AI. On the supervised learning side, it refers to the use of trained algorithms that allow machines to learn and perform tasks and solve equations independently, based on previously known inputs and outputs [3]. In the case of unsupervised learning, the outputs are unknown. ML and DL have a wide range of applications in several fields of expertise, including data science [4], image analysis [5,6], voice and noise processing [7], urban traffic management [8,9], digital marketing [10], autonomous car driving [11], fraud detection [12], Handwritten recognition [13], etc.

In the applied medicine field, previous researches have shown that ML and DL methods can be applied to predict various diseases, including CVDs [14–16], breast cancer [17,18], pneumonia prediction [19–21], COVID-19 diagnosis [22], diabetic retinopathy detection [23,24], Parkinson disease prediction [25–27], etc. The early detection of cardiovascular diseases is essential to improving patients' daily lives and reducing the mortality rate associated with these diseases [28]. Traditional methods of diagnosis often rely on physical examination, medical history, and various biological tests. However, these methods can take too much time, be highly expensive, and may not provide an accurate diagnosis in all cases [29]. ML has emerged as a powerful tool in the medical field, providing a new approach to the prediction and diagnosis of cardiovascular diseases [30]. By incorporating large amounts of medical data and using advanced algorithms, ML systems can identify patterns and correlations that may not be immediately recognized by the visual inspection of healthcare professionals [31]. This can help in the early detection and prediction of cardiovascular diseases, leading to improved patient outcomes and a reduction in the global burden of these diseases. Indeed, the increasing use of ML techniques in the field of medicine highlights the potential of these methods in addressing complex health problems. In the context of CVDs, the use of ML can provide doctors with valuable information that can aid in the early prediction and diagnosis of these diseases, helping to improve treatment outcomes and overall public health. In light of these factors, there is a clear need for the development of ML systems that work either on-site or remotely based on the Internet of Medical Things (IoMT) [32] for the prediction and diagnosis of cardiovascular diseases. By combining cutting-edge technology with a wealth of medical data, these systems have the potential to revolutionize the way CVDs are diagnosed and treated, ultimately improving public health conditions.

In this study, we developed an accurate system for the early prediction of heart diseases by choosing one of the powerful existing ML algorithms. In order to evaluate and validate different performances of our system, we used the Cleveland dataset stored in international platforms. The process involved in this study consists of several key steps. Firstly, the dataset is collected from international databases and labeled into different categories. This step is crucial as it allows the algorithm to learn from the relevant information in the dataset and make predictions based on this information. Secondly, six ML Algorithms which are decision tree (DT), random forest (RF), support vector machine (SVM), logistic regression (LR), adaptive boosting (AdaBoost), and K-nearest neighbors (KNN) are applied to classify the data. The performance of each algorithm is then evaluated based on metrics such as accuracy, precision, recall, and F1 score. To further improve the performance of the predictions, the hyperparameters of the algorithms are tuned using various hyperparameter tuning approaches. As a novelty of our study compared to previous works, this step aims to efficiently explore the hyperparameter space and identify the optimal combination of parameters that maximizes the model performance. The results of all the algorithms are compared and analyzed to determine the most effective one for heart disease early prediction. Finally, this last is validated using the Statlog Heart dataset and 10-fold cross-validation with a 95 % confidence interval. The optimized pre-trained model will be used to make predictions for future cases of heart disease.

The aim of this study is to contribute to the advancement of ML techniques in the medical field and provide a useful tool for healthcare professionals in the early prediction and diagnosis of cardiovascular diseases. Additionally, this study will provide valuable insights into the potential applications of ML in the prediction and diagnosis of other diseases. Hence, the main points and ideas of the present study can be summarized as follows.

- An accurate early prediction of heart disease risk is of great interest to doctors to prevent patient death.
- A diagnostic decision support system to address cardiologists' misdiagnoses and avoid misinterpretations of test results (that may happen every day) is proposed.
- The Cleveland heart disease dataset is explored using feature engineering techniques and used for training and testing the proposed ML models.
- A selected set of ML algorithms is trained on the Cleveland heart disease dataset and hyperparameters tuning is performed.
- The developed prediction system achieved the best accuracy of 92 % outperforming similar studies and reached the accuracy of 91.18 % using the Statlog heart dataset with performance validated through 10-fold cross-validation with a 95 % confidence interval.

[1] https://www.who.int/health-topics/cardiovascular-diseases.

The rest of this paper is structured as follows: In Section II, we review previous work in the field to provide background information for our study. Section III details the methodology, algorithms, evaluation metrics, approaches, and computational resources used in our research. The results and their discussions are presented in Section IV, where the key findings of the study are emphasized. Finally, in Section V, we summarize the main points and discuss the implications of our research.

2. Related works

Heart disease diagnosis is the first stage of the patient's treatment within the cardiovascular department. However, the prediction of such diseases has become the most preoccupying issue for all physicians and researchers working in the cardiovascular area. Due to the ability of AI techniques, recently, several studies have aimed to improve systems that can help predict heart failures and cardiovascular diseases, with the help of ML tools.

The authors in Ref. [33] proposed a medical diagnostic support system capable of the early prediction of atherosclerosis disease. The authors tested two supervised ML algorithms artificial neural network (ANN) and KNN validated on four different heart datasets which are Cleveland, Hungarian, Switzerland, and Long Beach datasets. The performance evaluation measures used in this paper are sensitivity, specificity, and accuracy. The results of this work showed that the ANN algorithm had the highest performance in all datasets. The authors of [34] proposed an early prediction system for the disease of atherosclerosis. The system was evaluated by comparing two ML algorithms (ANN and AdaBoost) on three datasets (Cleveland, Hungarian, and Z-Alizadeh Sani). The performance of the algorithms was measured using several metrics such as accuracy, sensitivity, specificity, precision, recall, and F1 score, and the results were visualized using a receiver operating characteristic (ROC) plot. The results showed that the ANN algorithm was the most accurate, with the highest recall and F1 score on all three datasets. The AdaBoost algorithm had the highest precision only in one dataset (Z-Alizadeh Sani).

In another research [35], the authors developed a model for predicting heart disease using several ML classification techniques, including KNN, Decision Tree, Logistic Regression, Support Vector Machine, Naive Bayes (NB), a hybrid technique named Vote and Neural Network. These algorithms were applied to the Cleveland dataset, which was obtained from the University of California Irvine (UCI) ML Repository.[2] The initial results showed that the Vote, SVM, and NB classifiers had the highest accuracy and precision rate. The results illustrated that the Vote model, using only 9 of the 14 attributes in the dataset, achieved an accuracy rate of 87.41 % and gave the best prediction.

In [36], the researchers worked on the Statlog heart disease dataset using several ML algorithms for classification: ANN, SVM, NB, LR, KNN, and Classification Trees. The study used ten-fold cross validation for evaluation and extracted eight quality measures, such as accuracy, sensitivity, specificity, precision, negative predictive value, false positive rate, etc. The results showed that the most accurate and sensitive algorithm was Logistic Regression, the most precise and specific was SVM, and the algorithm with the highest rate of misclassification and false positive rate was the Classification Trees.

The authors in Ref. [37], aimed to predict heart disease using two ML algorithms on two different databases. The first algorithm, J48, was applied to the Hungarian dataset, and the second algorithm, Naive Bayes, was run on the echocardiogram dataset. To evaluate the two models, the authors used metrics such as confusion matrix, accuracy, true positive rate, precision, F-measure, and ROC area. The experiments were conducted using the Weka data-mining tool, and two tests were performed on each dataset, one using all attributes and the other using specific attributes. The results showed that accuracy increased when all attributes were used. On the Hungarian dataset, the accuracy rate was 65.64 % with selected attributes and 82.3 % with all features. On the echocardiogram dataset, the accuracy rate was 93.24 % with selected attributes and 98.64 % with all features.

In [38], the results showed that the hybrid algorithm was the most accurate, which suggests that combining multiple algorithms can sometimes lead to improved performance. However, it is important to note that this result may not be generalized to other datasets and tasks, further experimentation would be needed to determine the effectiveness of the hybrid algorithm in other contexts. Moreover, the study [39] demonstrates the importance of careful experimentation and evaluation when choosing an ML algorithm or data mining tool for a given task. The study provides valuable insights into the performance of different data mining tools and ML algorithms in the context of classifying cardiovascular diseases. The comparison of six data mining tools and six ML algorithms applied to the Cleveland dataset highlights the diversity of tools and algorithms available for this task. The performance metrics results showed that Matlab's ANN model had the highest accuracy and sensitivity, while RapidMiner's SVM model had the highest specificity.

Overall, these studies highlight the importance of careful experimentation and evaluation when choosing an ML algorithm or data mining tool for a given task. Nevertheless, despite the advancements in the development of ML algorithms, there are still limitations that need to be addressed. One of the main limitations of these studies is the lack of standardization of the datasets used. Each study uses a different dataset, which makes it difficult to compare the results across different studies. Moreover, the datasets used are usually small in size, which can affect the accuracy of the developed models. This is because the models need a large amount of data to learn and make accurate predictions.

Another limitation of these studies is that they focus primarily on the evaluation of the performance of different ML algorithms. Little attention is given to the optimization of the hyperparameters, which can significantly affect the training phase and the performance of the models. Hyperparameter optimization is a crucial step in the development of ML models, and it requires a considerable number of computational resources and time. Finally, many studies in this field do not address the interpretability of the developed

[2] http://archive.ics.uci.edu/ml/index.php.

models. The lack of interpretability can make it difficult for healthcare professionals to understand how the models make predictions, which can limit the adoption of these models in real-world settings.

3. Materials and methods

3.1. General layout of the proposed prediction system

In Fig. 1, the layout of the proposed system is depicted, illustrating the systematic arrangement of the proposed heart disease prediction system. The diagnostic process starts by collecting the patient's data and inserting these informative data into a user interface to submit them to a pre-trained model to predict whether the user has heart disease or not. Furthermore, this system has been designed to be easy to use and to support the decisions of healthcare providers.

3.2. Global overview of the proposed methodology

Our proposed methodology focuses on building a heart disease prediction system by evaluating the performance of six supervised ML algorithms. The Heart Disease Cleveland dataset, a widely used dataset in heart disease prediction studies, was employed in our study [40]. The dataset contains various health-related factors that are utilized to predict the presence of heart disease. To achieve our objective, we adopted the ML Pipeline process, which is an organized and systematic approach to ML modeling. The ML Pipeline encompasses the entire process from raw data inputs to the final predicted outputs, including feature engineering, hyperparameter tuning, and model selection. Fig. 2 illustrates a diagrammatic representation of the ML Pipeline process.

Pipelining is an important concept in computer architecture that allows the processor to start executing a new instruction without waiting for the previous instruction to complete. This can result in significant improvement in performance. In the context of building a predictive system for heart disease, we followed several steps to take advantage of this concept.

Our investigation commenced with an exhaustive examination and evaluation of the Cleveland database, a renowned dataset that has been extensively utilized in studies concerning the prediction of heart disease. The database comprises an extensive variety of health-related characteristics that are recognized to have an impact on the prognosis of cardiovascular disease. The principal objective of our study was to identify the critical characteristics among these variables and classify them as the primary risk factors that contribute to the precise prognosis of heart disease.

Following this, six unique ML algorithms were implemented to determine the most effective classification framework for our particular problem domain. A diverse range of algorithms were utilized, such as the AdaBoost algorithm, Support Vector Machine, Logistic Regression, K-Nearest Neighbors, Random Forest, and Decision Tree. The aim of our study was to conduct an exhaustive comparison for these algorithms' performance in order to determine the most appropriate algorithm that produces the most favorable results for our specific situation.

A comprehensive set of performance metrics was employed to meticulously assess the models' performance, including accuracy, sensitivity (recall), precision, F1 score, and the Area Under the Curve (AUC). By conducting a thorough evaluation, we were able to compare the results of the models in great detail, which allowed us to determine which algorithm was most suitable for the problem at hand. Furthermore, in order to enhance the predictive capabilities of the models, we implemented a hyperparameter optimization strategy for finding the optimal hyperparameters due to their importance and impact on the training and prediction phases.

The medical predictive diagnosis process, as illustrated in Fig. 3, is a comprehensive flowchart that visually represents the utilization of six distinct ML algorithms. The flowchart functions as a graphical representation, outlining the consecutive procedures entailed in developing a resilient predictive system specifically designed for the diagnosis of cardiovascular disease. The flowchart begins with an initial phase that is specifically designed for the investigation and evaluation of the Cleveland database. This is a critical step that seeks to gain a thorough comprehension of the structure and contents of the dataset. During this stage, the data is preprocessed, features are selected, and exploratory data analysis is conducted in order to extract significant insights and detect pertinent patterns in the dataset.

After the exploration of the database, the flowchart illustrates the consecutive procedures entailed in constructing the model. This involves the implementation of six machine learning algorithms that each contribute to the development of the predictive system. The flowchart additionally depicts the phase of model evaluation, during which a comprehensive assessment of the performance of each algorithm is conducted by employing a predetermined set of performance evaluation metrics. The metrics comprised of accuracy, recall, precision, F1 score, and AUC collectively offer a comprehensive assessment of the predictive capabilities of the models.

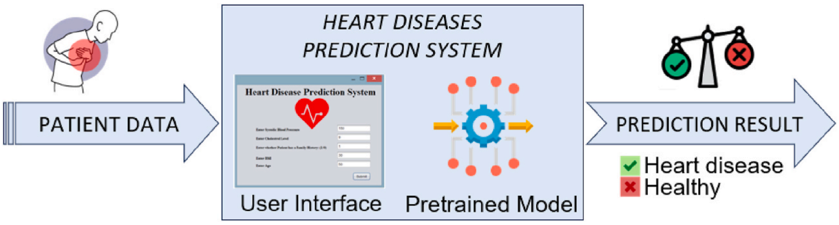


Fig. 1. Layout of the proposed system.

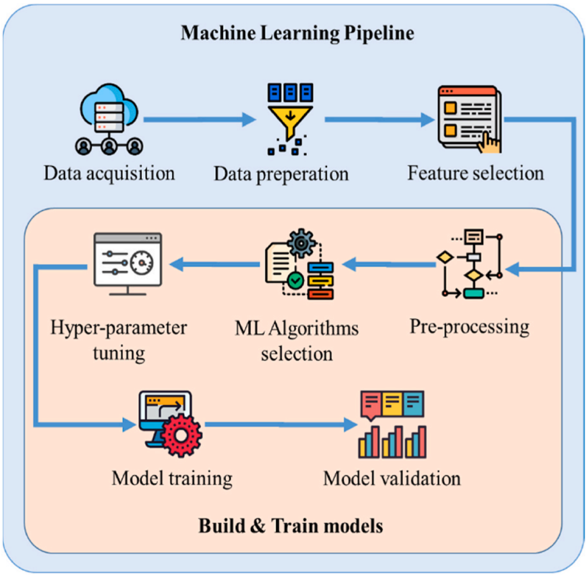


Fig. 2. ML Pipeline process.

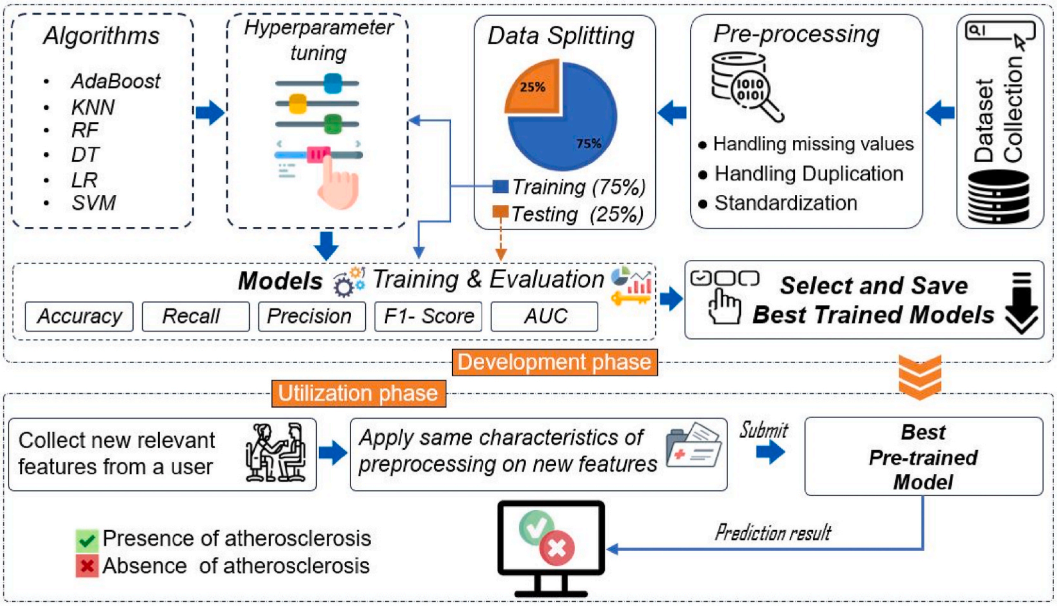


Fig. 3. Flowchart of the medical predictive diagnosis using six ML algorithms.

Indeed, Fig. 3 provides a comprehensive and organized synopsis of the proposed medical predictive diagnosis process. Moreover, this flowchart consists of two phases. The first is the development phase which represents the dataset collection, preprocessing, shuffle and splitting, building of classifiers and tuning their hyperparameters, the process of training and evaluation, and selecting of the best model to be the kernel of the proposed system in the utilization phase. On the other hand, the utilization phase starts by collecting the patient's data, applying the same characteristics of the processing step to the newly collected features, and submitting these features to the model to predict if the user has heart disease.

3.3. Exploratory data analysis (EDA)

The first step in the pipeline for classification is to understand the data. Thus, we used the Cleveland heart disease database based on Exploratory Data Analysis (EDA).

3.3.1. Cleveland dataset description

The Cleveland Clinic Foundation dataset, collected by David Aha, is widely used in the heart disease prediction field. It was collected in 1987 and contains 303 samples with 76 features. According to the study [40], we used the most relevant features which consist of 14 features (13 attributes and 1 target variable). The target variable represents the presence or absence of heart disease, comprising 165 sick patients represented by "1" and 138 healthy samples represented by "0", whereas the remaining 13 attributes represent various health-related factors such as age, sex, chest pain type, blood pressure, cholesterol levels, and others [40].Table I outlines and describes the 14 attributes in the Heart Disease Cleveland dataset. The variables include demographic information such as age and gender, medical information such as chest pain type and cholesterol levels, and test results such as resting blood pressure and electrocardiographic results. Each variable has a specific meaning and is used to determine the presence or absence of heart disease.

3.3.2. Data preprocessing

First, we started our preprocessing phase by detecting the duplicates and missing values, which led us to delete and remove 16 out of the 303 samples of the dataset. The next step is to check and handle the outliers due to the significant effect of outliers on the results of statistical and ML models. The aim of this process is to identify and handle data points that significantly deviate from the majority of the dataset. There are various methods for identifying outliers, such as the Z-score method, the Interquartile Range (IQR) method, and the modified Z-score method. The appropriate method depends on the nature of the data, so it is important to choose the right method for identifying outliers. After handling outliers, it is important to verify the data types of the attributes. Incorrect data types can lead to errors during analysis and modeling, so it is important to convert the attributes to the correct data type if necessary.

In fact, all the preprocessing stages are performed to ensure that there are no missing or null values in the dataset. This is important because missing or null values can cause problems when building an ML model and can lead to inaccurate results. To ensure that the dataset does not contain any missing or null values, one can check for missing or null values using appropriate programming tools and methods. Once the dataset has been cleaned of any missing or null values, it is important to perform outlier detection. Outliers are observations that lie significantly away from other values in the dataset and can have a significant impact on the results of an ML model. They can cause the model to overfit or underfit the data and can bias the results if they are not handled appropriately.

One common way to detect outliers is by plotting box plots for all features in the dataset. Box plots are a graphical representation of the distribution of a dataset and provide a clear visual representation of the presence of outliers. By examining the box plots, we can identify any potential outliers in the data and determine if they should be removed or handled in some other way. Fig. 4 plots box plots

Table 1
The attributes of The Cleveland Dataset along with their Definitions.

ID	Attributes	Definitions	Description
1	Age	This feature provides information on how old a patient is, which is an important factor to consider in heart disease prediction as the risk of heart disease increases with age.	29 to 71 (Mean: 54.37 ± SD 9.1)
2	GD	Refers to the biological sex of a person, which can be either "Woman" or "Man".	Woman: 96 Man: 207
3	CP	Chest pain is classified into four types: Typical angina, Atypical angina, Non-angina pain, and Asymptomatic.	Typical angina (0): 143, Atypical angina (1): 50, Non-angina pain (2): 87, Asymptomatic (3): 18.
4	Trestbps	The "resting blood pressure" attribute refers to the patient's blood pressure when they are in a relaxed state (mm Hg)	94 to 200 (Mean: 131.62 ± SD 17.54)
5	Chol	Refers to the levels of cholesterol in a patient's blood. (mg/dl)	126 to 564 (Mean: 246.26 ± SD 51.83)
6	Fbs	Diabetes is a binary variable that indicates whether a patient has been diagnosed with diabetes.	True (1): 45 False (0): 258
7	restecg	The electrocardiographic results attribute is a categorical variable that describes the results of an electrocardiogram (ECG) test performed on a patient.	0: 147 1: 152 2: 4
8	Thalach	The Heart rate attribute refers to the number of times the heart beats per minute.	71 to 202 (Mean: 149.65 ± SD 22.91)
9	Exang	The Angina attribute is a categorical variable that indicates whether a patient has angina, a type of chest pain that occurs when there is not enough blood flow to the heart.	Yes (1): 99 No (0): 204
10	Oldpeak	Refers to the ST depression induced by exercise relative to rest.	0 to 6.2 (Mean: 1.04 ± SD 1.16)
11	Slope	Refers to the slope of the peak exercise ST segment, which is a measure of the electrical activity of the heart.	Upsloping (1): 21 Flat (2): 140 Downsloping (3): 142
12	ca	Number of major vessels (0–3) attribute refers to the number of major blood vessels (0–3) that are visible by fluoroscopy (a type of X-ray that uses a continuous X-ray beam to produce real-time images).	0: 175 1: 65 2: 38 3: 25
13	Thal	The thallium heart scan attribute refers to the results of a thallium heart scan, which is a type of nuclear imaging test used to evaluate blood flow to the heart muscle.	Normal (0): 20, Fixed defect (1): 166, Reversable defect (2): 117.
14	Target	The target attribute is the target variable or the dependent variable that represents the presence or absence of heart disease in a patient.	Presence (1): 165, Absence (0): 138.

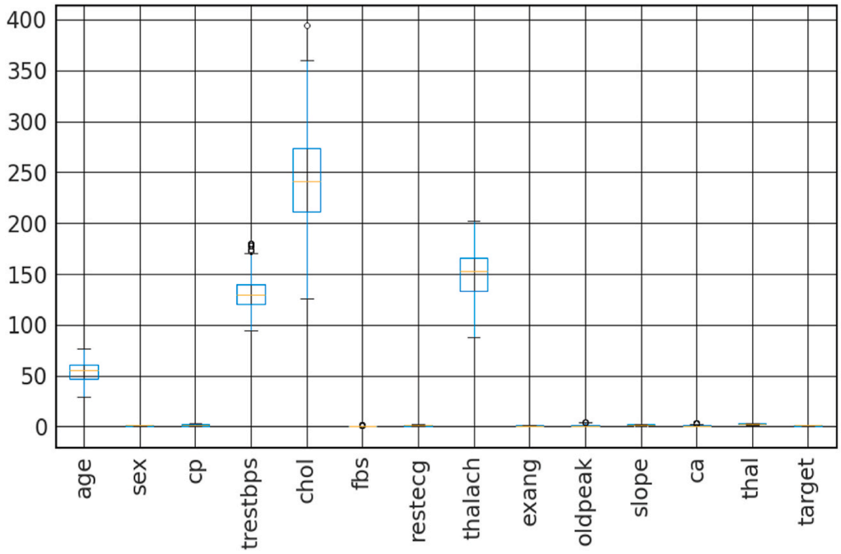


Fig. 4. box plots for Cleveland database.

for all Cleveland database features.

In order to perform univariate analysis on the Cleveland database, histograms were utilized for each of the features. This approach was taken because all features in the dataset were encoded as numerical values, which saved time compared to the process of categorical encoding that is typically carried out during the feature-engineering phase [41]. Histograms, clustered bar plots, and box plots are all EDA techniques that are well-suited for use with classification ML algorithms. These techniques allow for the visualization of the distribution of individual features and can provide valuable insights into the shape, center, and spread of the data. In particular, histograms provide a concise representation of the distribution of a single feature. The range of the data is divided into bins, and the frequency of observations for each bin is plotted. This allows for the identification of any potential outliers and a general understanding of the distribution of the data. Fig. 5 represents the Cleveland dataset features histogram matrix, which displays key features including age (Fig. 5A), sex (Fig. 5B), chest pain (Fig. 5C), resting blood pressure (Fig. 5D), cholesterol (Fig. 5E), diabetes (Fig. 5F), electrocardiographic results (Fig. 5G), heart rate (Fig. 5H), angina (Fig. 5I), ST depression (Fig. 5J), slope (Fig. 5K), number of major vessels (Fig. 5L), and thallium heart scan (Fig. 5M).

The importance of categorical values in determining target values can be illustrated using a clustered bar chart. This type of chart provides a simple representation of the distribution of target values for different categories of a categorical feature. For example, if the target variable in the dataset is binary, a clustered bar chart can be used to compare the distribution of target values for two different categories of a categorical feature. For example, if the categorical feature is "sex," the chart could compare the distribution of target values for "sex = 1″ and "sex = 0." If the distribution of target values for these two categories is different, it suggests that the "sex" feature may play an important role in predicting the target variable. This type of analysis can be valuable for identifying which features are most strongly associated with the target variable and may be important predictors. It is important to note that this type of analysis is based on the assumption of independence between the features and the target variable and that any correlations found may be subject to confounding factors.

On the other hand, if the distributions of the target variable are the same for different categories of a categorical feature, it suggests that the feature and target variable are uncorrelated. In such cases, it is important to further examine the relationship between the feature and the target variable to determine if the feature is still important for prediction. To visualize the relationship between discrete features and the target variable, a count plot can be used. This type of plot displays the frequency of observations for different categories of a feature, with the target variable displayed as different colors or markers. For continuous features, histograms and Kernel Density Estimation (KDE) plots can be used to visualize the relationship with the target variable [42]. A histogram shows the distribution of a single feature, with the target variable displayed as different colors or markers. KDE plots provide a smooth estimate of the underlying density function for each feature, with the target variable displayed in different colors or markers, as shown in Fig. 6, which illustrates the distribution of instances according to features including sex (Fig. 6A), chest pain (Fig. 6B), diabetes (Fig. 6C), electrocardiographic results (Fig. 6D), angina (Fig. 6E), slope (Fig. 6F), number of major vessels (Fig. 6G), and thallium heart scan (Fig. 6H).

In these cases, we applied Z-score to detect the outliers and normalize (scaling) the dataset. In several works such as [43], the Z-score represents the difference between the value and the standard deviations indicating how far is away from the mean. We used the following Eq. (1) to calculate a Z-score:

$$Z - score = \frac{(x - \mu)}{\sigma} \tag{1}$$

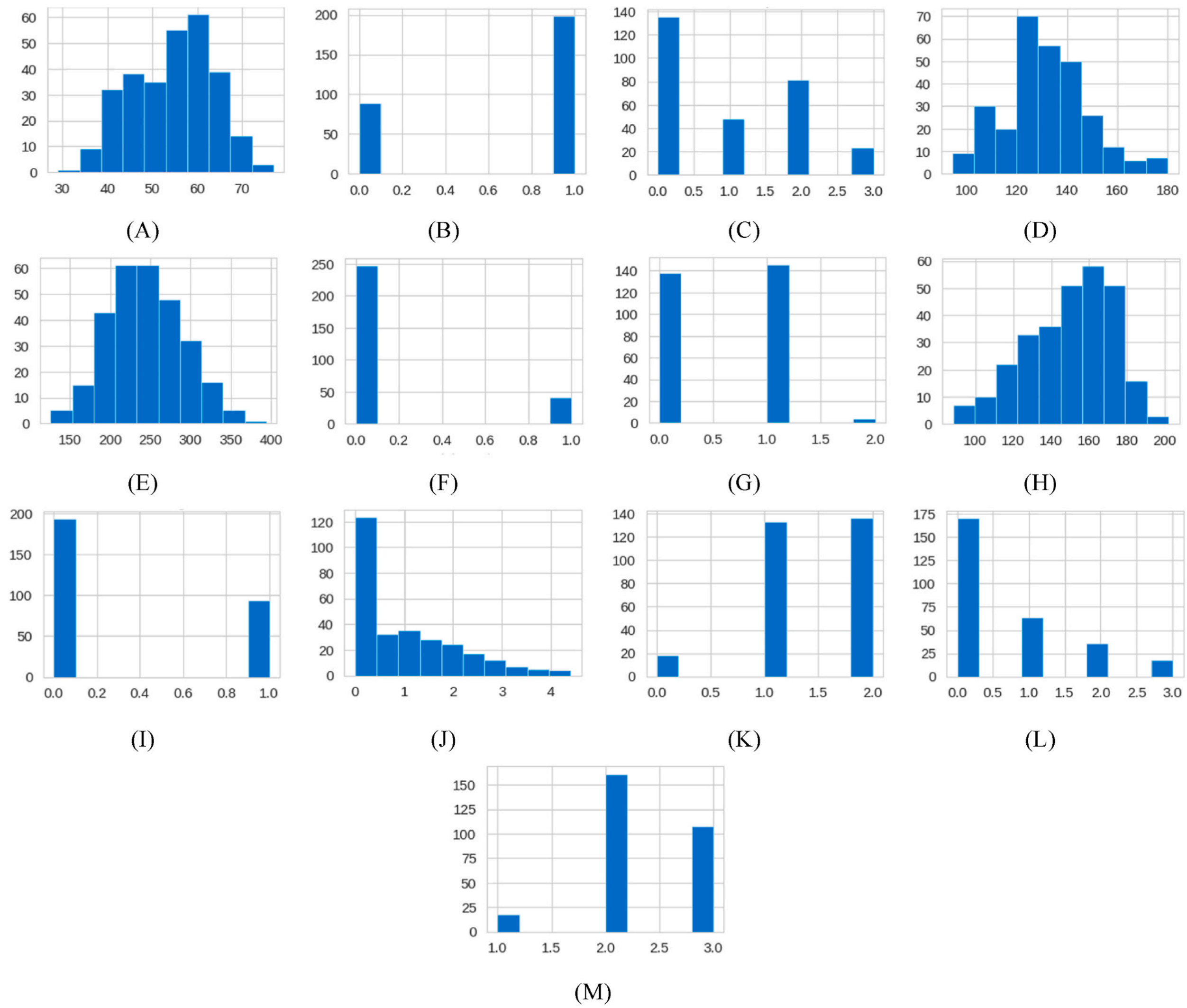


Fig. 5. Features histogram matrix. (A) Age. (B) Sex. (C) Chest pain. (D) Resting blood pressure. (E) Cholesterol. (F) Diabetes. (G) Electrocardiographic results. (H) Heart rate. (I) Angina. (J) ST depression. (K) Slope. (L) Number of major vessels. (M) Thallium heart scan.

Where.

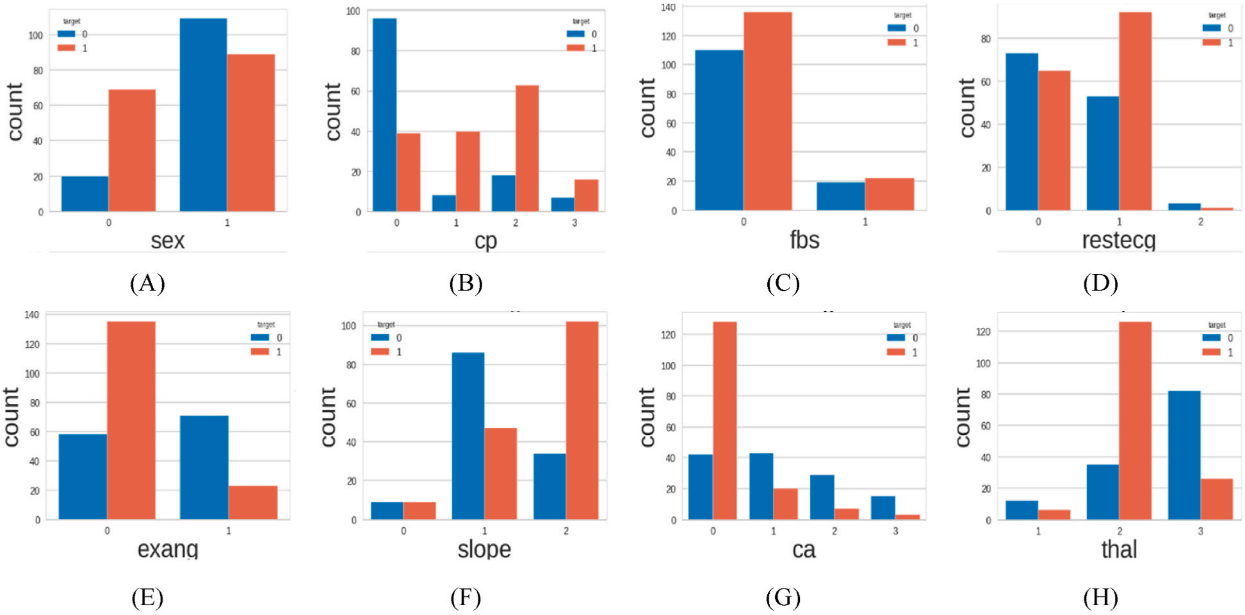
- x represents the raw data value,
- μ represents the population mean,
- σ represents the population standard deviation.

After removing outliers using Z-score and duplicate rows, we obtained a new dataset that contains 287 records with 14 features including the target.

Before implementing any ML algorithm, we divide the Cleveland dataset into a training set and a testing set. This step is requisite to build up the predictive models. We used the Pipeline process to perceive the best method. Therefore, preprocessing is necessary to clean and to make an exploratory analysis of data. Thus, we divided the Cleveland Database into two partitions. The first partition used 75 % of the data for training and the remaining 25 % used for testing. In Fig. 7, we depict the continuous features with the target variable using histogram and KDE results, including age (Fig. 7A), Resting blood pressure (Fig. 7B), Cholesterol (Fig. 7C) Heart rate (Fig. 7D) ST depression (Fig. 7E)

3.4. Machine learning algorithms

In this work, we choose six ML algorithms that are commonly used in the previous studies due to their boundary of classification,



(A) (B) (C) (D)

(E) (F) (G) (H)

Fig. 6. Distribution of instances according to features using count plot. (A) Sex attribute. (B) Chest pain attribute. (C) Diabetes attribute. (D) Electrocardiographic results attribute. (E) The Angina attribute. (F) Slope attribute. (G) The number of major vessels attribute. (H) The Thallium heart scan attribute.

versatility, robustness and effectiveness across various types of datasets. These algorithms have provided strong performance and are often considered baseline models in many ML tasks. Additionally, they provide many approaches, enabling better coverage of the problem space and more robust model evaluation.

3.4.1. Logistic regression

Logistic Regression is a simple and widely used statistical technique for binary classification problems, such as the prediction of heart disease. It models the relationship between the dependent variable (heart disease) and a set of independent variables (features) using a logistic function [44].

For binary LR with a single predictor, the Eq. (2) is a statistical model given by:

$$\left(\frac{p}{1-p}\right) = b_0 + b_1 x_1 \tag{2}$$

Where.

- p is the probability that the dependent variable Y = 1 given the value of the independent variable x,
- b is the parameters of the model.

3.4.2. Support vector machine

Support Vector Machines are a type of ML algorithm that can be used for classification and regression problems. They work by finding the hyperplane that best separates the data into two classes [45]. This algorithm aims to create a line representing a boundary that separates two classes of features initially mapped into n-dimensional space, and each class contains many features. This line is called a hyperplane. Thus, every new data is put in the right class.

The Eq. (3) of the hyperplane H [46] can be written as:

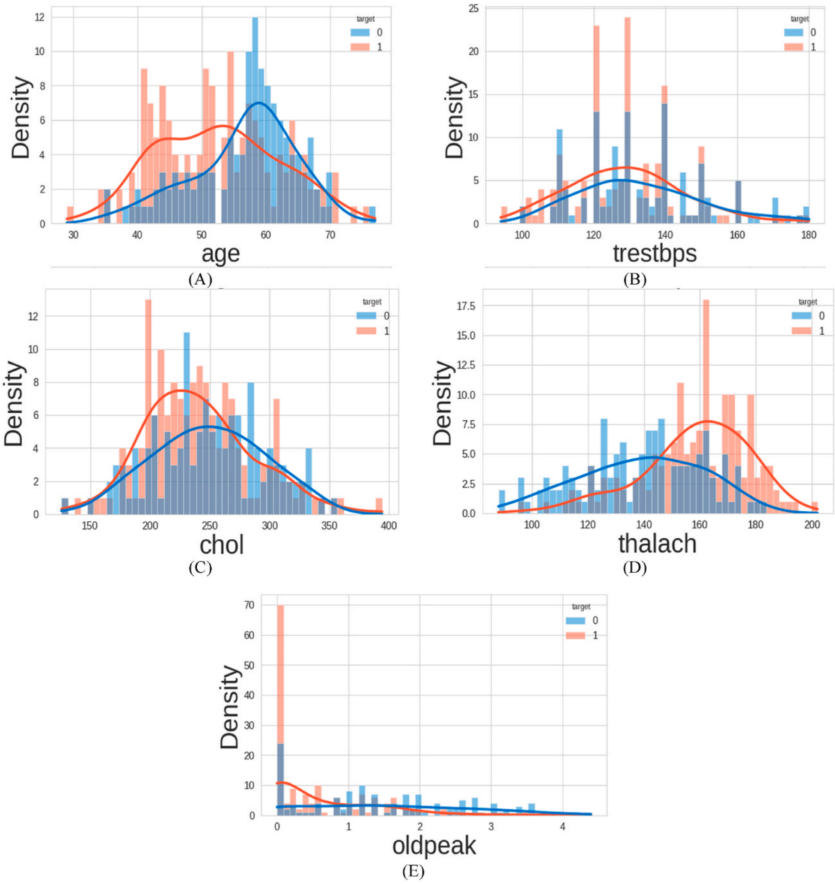
$$H : w^T(x) + b = 0 \tag{3}$$

Where b is the Intercept and bias term of the hyperplane equation.

The hyperplane is always a D-1 operator in a D dimensional space. For example, in a 2D space, the hyperplane is a 1D line. The distance of a hyperplane's equation (3) from a given point vector $\phi(x_0)$ is set up with the following Eq. (4):

$$d_H(\phi(x_0)) = \frac{|w^T(\phi(x_0)) + b|}{\|w\|_2} \tag{4}$$

Where $\|w\|_2$ is the Euclidean norm for the length of w given by Eq. (5):



(A)

(B)

(C)

(D)

(E)

Fig. 7. Continuous features with target variable using histogram and KDE. (A) Age attribute. (B) Resting blood pressure attribute. (C) Cholesterol attribute. (D) Heart rate attribute. (E) ST depression attribute.

$$\|w\|_2 = \sqrt{w_1^2 + w_2^2 + w_3^2 + \ldots + w_n^2} \tag{5}$$

3.4.3. *Adaptive boosting*

Adaptive boost shortly AdaBoost is an ML method applied to classification and regression problems [47]. AdaBoost was first set by Yoav Freund and Robert Schapire [34]. The algorithm builds iteratively a final weighted classifier by generating a set of classifiers at each iteration. The Eq. (6) used to calculate the weighted classification error for each learner t is the following [34]:

$$e_t = \sum_{n=1}^{N} d_n^{(t)} I(y_n \neq h_t(x_n)) \tag{6}$$

Where.

- x_n is the predictor values for observation n;
- y_n is the true class label;
- h_t is the hypothesis (learner predictor);
- I is the indicator function;
- $d_n^{(t)}$ is the observation weight in step t.

For the prediction, the Eq. (7) used to compute is the following:

$$f(x) = \sum_{t=1}^{T} \propto_t h_t(x) \tag{7}$$

With the Eq. (8) of $\propto_t$:

$$\propto_t = \frac{1}{2}\, ln\ ln\ \frac{1-\varepsilon_t}{\varepsilon_t} \tag{8}$$

Where $\propto_t$ are the weak hypothesis weights in the set.

The part training of the AdaBoost algorithm can assimilated to the minimization of exponential loss with following Eq. (9):

$$\sum_{n=1}^{N} w_n\ exp^{\left(-y_n f(x_n)\right)} \tag{9}$$

With.

- $y_n\ \epsilon\ \{-1,1\}$ is the true class;
- w_n present the observation weights normalized;
- $f(x_n)\ \epsilon\ (-\infty,+\infty)$ present the predicted classification.

3.4.4. Decision tree

The decision tree algorithm is one of the most common supervised ML algorithms. It has a structure like a tree [45], where the root node is the decision node, the internal nodes correspond to the tests on the feature, the branches represent the outcomes and the terminal nodes or leaves are the class labels. The decision tree algorithm could be used for classification as well as prediction. In the case of multiple features, there is a need to choose the feature with the most information that is in the first stage as a root node. For that, a metric called 'information gain' gives the purity of the feature [48]. This metric is based on the measure of the entropy that gives the impurity or randomness of the dataset calculated as follows (Eq. (10)) [49]:

$$Entropy = \sum_{i=1}^{C} -p_i\ {*}log_2(p_i) \tag{10}$$

Where p_i is the ratio of the sample number of the subset and the i^{th} attribute value.

Thus, the gain given by the Eq. (11):

$$Gain\ (S,A) = \sum_{v\in V(A)} \frac{S_v}{S}\ Entropy(S_v) \tag{11}$$

Where $V(A)$ is the range of attribute A, and S_v is a subset of set S equal to the value of attribute v.

3.4.5. Random forest

As a forest is the sum of many trees, the RF algorithm is based on an ensemble of DTs [45]. For The RF algorithm, the outcomes are taken from many different trees. Therefore, the classification decision comes from each tree and the forest chooses the one having the majority of votes. In case of regression,[3] the decision is based on the average of all trees with the following Eq. (12) [45].

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{12}$$

here, pi represents the relative frequency of the observed class in the dataset and c represents the number of classes. In our cases, the C equals two. This formula used to determine the Gini of each branch on a node and could define which of the branches is more probable to occur [50]. We can use entropy to define how many nodes are in each decision tree as shown in Eq. (10).

3.4.6. K-nearest neighbors

The K Nearest Neighbors algorithm is a robust supervised ML algorithm [51]. The algorithm consists of measuring the distance between similar data types and classifying them in shapes holding the nearby ones. The KNN algorithm's main goal is to create a model f that predicts a class label y' for an unidentified pattern x' [33]. KNN allocates the majority label class of K-Nearest patterns in data space and offers the nearest patterns to a target x'. The Hamming distance has been used to define a similarity metric in data space following the Eq. (13).

$$D(x', c_j) = \frac{1}{p}\sum_{i=1}^{p} l\{x' \neq c_{j,i}\} \tag{13}$$

in the binary scenario, p = 2. This distance is used to calculate the distance between query points and a data set (testing dataset).

The KNN is calculated as follows in Eq. (14):

[3] https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

$$f_{KNN}(x') = \begin{cases} 1 \ if \sum_{i \in N_k} (x') y_i \ \geq 0 \\ -1 \ if \sum_{i \in N_k} (x') y_i \ < 0 \end{cases} \tag{14}$$

k is the neighborhood size, and $N_k(x')$ is the set of nearest pattern indices, with label set y = {-1, 1}. For an unknown pattern x_0 in multi-class classification, the KNN approach predicted the majority class label of the K-Nearest patterns in data space as follows in Eq. (15):

$$f_{KNN}(x') = \sum_{i \in N_k(x')} L(y_i = y) \tag{15}$$

With L is an indicator function where:

$$L(y_i = y) = \begin{cases} 1 & if\ argument\ is\ true \\ 0 & otherwise \end{cases} \tag{16}$$

Where y_i in the Eq. (16) is the output sample's i^{th} case and y is the projected output. Odd numbers are commonly used as k values. The k value was found by determining the classification error rate using the testing set and experimenting with different k values. The use of large k values can result in better decision areas and provide good probabilistic information. Larger values of k, on the other hand, are destructive and reduce estimation precision even further.

3.5. Performance evaluation measures

The classification based on supervised ML techniques should be split into a minimum of two subsets. Furthermore, the model was trained on the training set and then examined using the test set. The performance measures used to evaluate the ML algorithms are based on the values of the confusion matrix instances. A confusion matrix is a useful tool for evaluating the performance of a classifier. It provides a summary of the correct and incorrect predictions made by a classifier and can help to further interpret the performance of the model. In a confusion matrix, the rows represent the actual class labels, and the columns represent the predicted class labels. The entries in the matrix represent the number of instances that have a certain actual class label and were predicted to have a certain predicted class label. For example, in a binary classification problem, the entries might be the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions [52]. Using the information in the confusion matrix, we can calculate various metrics to evaluate the performance of the classifier. For example, accuracy can be calculated using Eq. (17). It represents the ratio of the total number of correct predictions to the total number of predictions. The Eq. (18) identifies the recall (sensitivity) as the ratio of the number of true positive predictions to the number of actual positive instances. Furthermore, the Precision can be calculated by Eq. (19) as the ratio of the number of true positive predictions to the number of positive predictions, WhileF1score and AUC can be calculated as determined in Eqs. (20) and (21), respectively. [53].

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$PRS = \frac{TP}{TP + FP} \tag{19}$$

$$F1_{score} = \frac{2*TP}{2*TP + FP + FN} \tag{20}$$

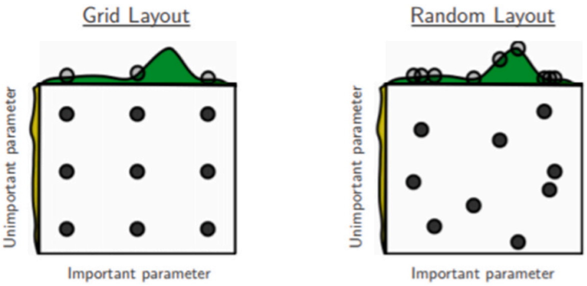


Fig. 8. Comparison between Grid and Random search with nine trials.

$$AUC = \int_{x=0}^{1} TPR\left(FPR^{-1}(x)\right)dx \tag{21}$$

3.6. Hyperparameters tuning

Hyperparameter tuning is a procedure that permits the optimization of the model prediction performances by reducing the error and maximizing the accuracy of the models [54]. It consists on the selection of the best parameters of each algorithm and then setting them to train and test the ML algorithms. There are several hyperparameters approaches. In this work, we used Grid and Random search. These approaches consist on the evaluation of hyperparameters that impacted the algorithm's performance metrics [55]. Then, the method estimates the combination of hyperparameters. As a final step, the best result is extracted and applied [54]. While the Grid search looks for the best combination of hyperparameters, the Random search approach is based on the test of random combinations with a defined number of iterations based on available resources and time. In Fig. 8, a comparison between Grid and Random search with nine trials is presented, illustrating the performance of these hyperparameter optimization techniques. This comparison is crucial for evaluating the effectiveness of different search strategies in finding the optimal hyperparameters for machine learning models.

Within the domain of machine learning, the concept of "grid layout" pertains to the systematic arrangement of hyperparameters in the form of a grid. Each element in this grid corresponds to a distinct combination of hyperparameter values. The purpose of this architecture is to establish the search space for hyperparameter tuning, enabling a methodical and comprehensive investigation of different configurations of hyperparameters [55].

Conversely, Grid search operates as a method of optimizing hyperparameters by methodically traversing a predetermined grid of hyperparameter values. The cross-validation process is employed to train and assess the machine learning model for every possible combination of hyperparameters in the grid layout. Cross-validation entails dividing the dataset into several subsets, with one subset being used to train the model and the other being utilized to evaluate its performance. Each subset undergoes this procedure in turn, and the mean performance is employed as the criterion for assessing every hyperparameter combination.

Depending on the nature of the problem, grid search attempts to identify the hyperparameter combination that yields the best model performance, which is typically evaluated using metrics including accuracy, precision, recall, and F1 score. The objective of grid search is to identify the optimal hyperparameters that maximize the performance of the model on the provided dataset through an exhaustive exploration of the grid layout.

As it offers a methodical and exhaustive approach to exploring the hyperparameter space, grid search is an extensively employed technique in machine learning for hyperparameter optimization. It can, however, be computationally intensive, particularly when dealing with expansive search spaces or intricate models. When this occurs, more sophisticated methods such as Bayesian optimization or random search may be implemented to search for optimal hyperparameters in an efficient manner.

3.7. Equipment and implementation

This work has been performed using the Standard Google Colab provided by Google as a cloud-based Integrated Development Environment (IDE), with hardware resources an Intel Xeon(R) CPU @ 2.20 GHz with 2 vCPUs (2 virtual CPUs) and 13 GB RAM. For the development of our system, we implement several Python libraries such as Scikit-learn, NumPy, and SciPy.

4. Results and discussion

To choose the best algorithm. The six ML methods applied separately on the Cleveland dataset and evaluated using six different performance measures. In this work, we used 287 patients and 14 features that illustrated the atherosclerosis symptoms. We obtained this patient's number after many processing steps using Z-score. To improve the system and find the best configuration of each ML algorithms, we summarized all the experimental results in tables and graphs. Then, we compared it with further results from the previous works.

4.1. Preprocessing

The first step of preprocessing is cleaning the data. That means we have to focus only on deleting all the missing values, implementing outlier detection, outlier treatment, training models, and choosing an appropriate model. After loading the database and importing the essential libraries, we divided the used data into two important parts. 75 % of the total data was used in training data and 25 % in test data. Therefore, all these steps are used before implementing any classification algorithm. Thus, we optimized our proposed atherosclerosis system using many performance evaluation metrics. The best-selected model just used after testing data set

Table 2
Details of database before classification.

Database	Total instances (100 %)	Training data (75 %)			Testing data (25 %)		
		Healthy (0)	**Sick (1)**	**Total**	**Healthy (0)**	**Sick (1)**	**Total**
Cleveland	287	97	118	215	32	40	72

and match the classification prediction results with the actual data. Table II explains the details of each split database. Thus, the last step of the preprocessing is to implement the performance metrics in order to evaluate the prediction results. This evaluation defines patients without or with atherosclerosis and demonstrates the effectiveness of the proposed computer-aided diagnosis system.

4.2. Implementation and training result

4.2.1. ML algorithms' best configuration

For The experimental default parameter, the LR algorithm fitted and tested with the "lbfgs" as the solver and a "1.0" as the value of C. The DT classifier executed with "gini" as a criterion and best splitter. The RF parameters were a "gini" as criterion and "100" as n_estimator. The SVM classifier was fitted with gamma: "scale", kernel: "rbf" and C: "1.0". The parameters of the KNN model, we reduced the error until finding the best K n_neighbors. The AdaBoost classifier has a "SAMME.R" algorithm and "50" as n_estimator. The best configuration of each algorithm is shown in Table III.

By applying the training parameters and configurations of each algorithms using pipeline method, we could find the best hyper-parameters tuning. According to the obtained results, our proposed system shows the high accuracy and precision. The red line between each blue point and the prediction line are the errors. Each error is the distance from the point to its predicted point.

Thus, we found the best K value in the MSE = 0.097 after calculating the error for K values between 1 and 25 which the k value obtained is 23. Fig. 9 shows the MSE graph and the best K value.

In this process, the best way to obtain the K value of KNN algorithm when applied the error reduction. In the following Eq. (22), we can calculate the Mean Squared Error (MSE):

$$MSE=\frac{1}{n}\sum_{i=1}^{n}(x_i-\widetilde{x}_i)^2 \tag{22}$$

Most of the ML algorithms have many hyperparameters that we could adjust. We implemented pipeline techniques to set them before the training phase. Hyperparameters need some special steps to be applied until building accurate and robust models. These steps can include selecting the right model, reviewing the list of the initial parameters, choosing the adequate hyperparameters tuning

Table 3
The best used hyperparameters to train algorithms.

Methods	Parameters	Values
LR	Tol (Tolerance for stopping criteria)	0.01
	Verbosity	0
	C (Inverse of regularization strength)	1.0
	Random state	1234
	Fit intercept	True
	Intercept scaling	2
	Max iteration	1000
	Solver	liblinear
KNN	Number of neighbors	23
	Metric	canberra
	P (Power parameter for the Minkowski metric)	2
	Weights	uniform
	Algorithm	brute
	n_jobs	−1
AdaBoost	Algorithm	SAMME.R
	Random state	42
	Learning rate	0.01
	Number of estimators	1100
DT	Max depth (The maximum depth of the tree)	5
	Criterion	Entropy
	Random state	40
RF	Number of trees in the forest.	500
	Criterion	gini
	Max depth	8
	Minimum number of samples required to split an internal node	10
SVM	C (Regularization parameter)	2
	Kernel coefficient	linear
	Tol	0.01
	Decision function of shape	one-vs-rest
	Gamma	Auto
	Probability estimates	True
	Cache size (Specify the size of the kernel cache (in MB))	100
	Verbosity	False
	class_weight	False
	max_iter	−1
	break_ties	False

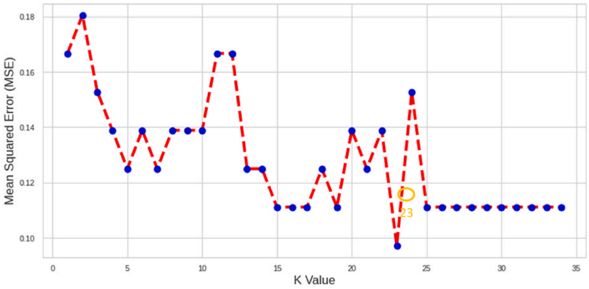
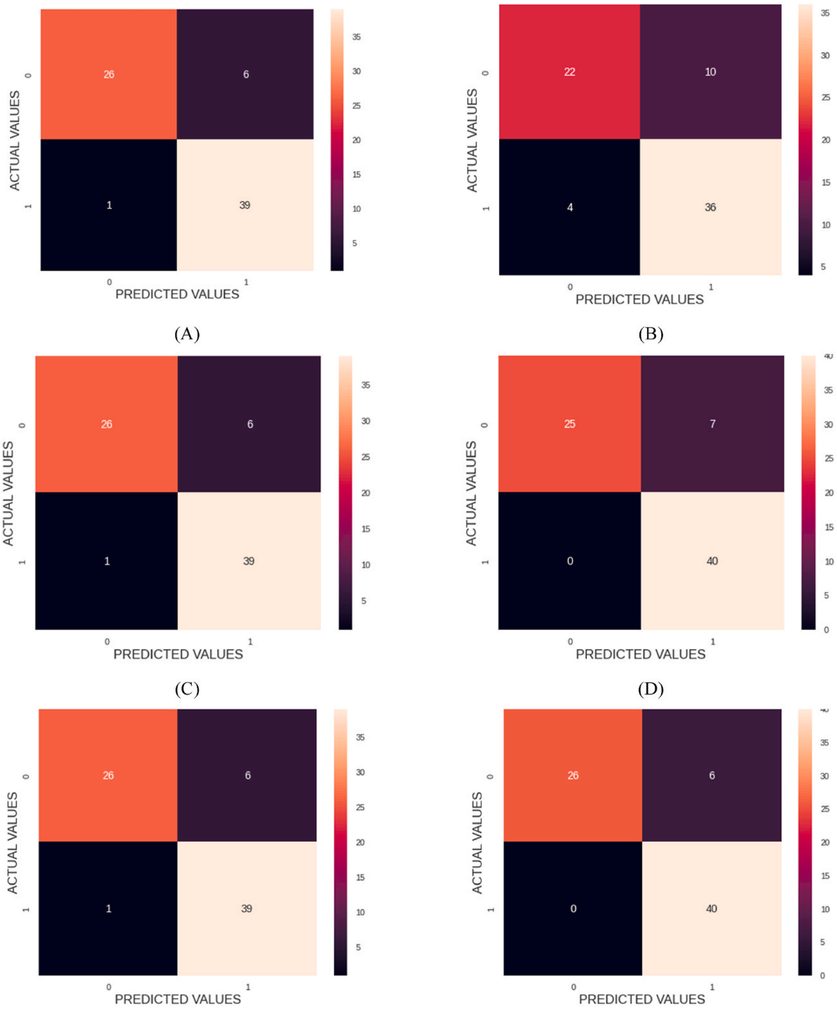


Fig. 9. Graph illustrating K and MSE values.

method (grid search, random search, …) and applying the cross-validation approach.

4.2.2. Performance evaluation

To improve the usefulness of our proposed system, we made our experimentations on the Cleveland database of heart disease using six ML algorithms. As stated above, all of these were evaluated through various performance evaluation metrics: accuracy (ACC), precision (PRS), sensitivity (SS), specificity (SP), F1 score, and area under the curve (AUC), which we will have presented in a receiver



(A) (B)

(C) (D)

(E) (F)

Fig. 10. The classification provided by the studied models using six ML algorithms. (A) LR algorithm. (B) Decision Tree algorithm. (C) KNN algorithm. (D) RF algorithm. (E) AdaBoost algorithm. (F) SVM algorithm.

operating characteristic (ROC) graph. This evaluation allowed us to have different perspectives on the performance of each model and identify any trade-offs between accuracy and other factors, such as false positive or false negative rates. We trained multiple ML algorithms and compared their performance to select the best model. By comparing the performance of each algorithm on the same dataset, using the same evaluation metrics, we were able to determine which algorithm performed the best. We carefully chose and implemented each algorithm, as well as chose and set the hyperparameters, to ensure that our results were representative and reliable. The results of our comparison depend on these factors. Fig. 10 represents the confusion matrix results for each of the ML methods, which were obtained using the testing partition of the data, including LR (Fig. 10A), Decision Tree (Fig. 10B), KNN (Fig. 10C), RF (Fig. 10D), AdaBoost (Fig. 10E), and SVM (Fig. 10F).

The goal of this phase was to evaluate the performance of the proposed predictive model in classifying patients with or without atherosclerosis disease. The results were then evaluated using important classification performance metrics, such as accuracy, recall, precision, and F1 score. These metrics provide different perspectives on the performance of the classifier, including how well it is able to accurately classify instances, how sensitive it is to detecting instances of the positive class, how precise it is in its positive class predictions, and a balance between precision and recall.

When developing and evaluating machine learning models, it is crucial to assess their performance on the training set to understand how well the models have learned from the dataset for identifying issues such as overfitting, where a model performs exceptionally well on the training data but fails to generalize to new instances (or unseen data). By comparing the performance metrics of the training set to those of the testing set, we can gain insights into the model's generalization capability and its potential robustness in real-world applications. Table IV shows the evaluation metrics obtained from the training set, including accuracy, precision, recall, and F1 score for each algorithm.

Moreover, Table V illustrates the classification outcomes and the performance metric using a testing set (unseen data). By analyzing these metrics side by side with those obtained from the training set, we aim to ensure that our models not only perform well on known data but also maintain high performance on unseen data, thereby demonstrating their reliability and effectiveness in practical scenarios.

In order to give more details about the behavior of the models in the classification task and the obtained matrices using our imbalanced dataset, it is crucial to provide comprehensive metrics that accurately reflect the model's performance across different classes. As shown in Table VI, Macro and weighted averages offer different perspectives on the model's performance in multiclass classification tasks. The macro average computes the unweighted mean of each metric for each class, this statistic offers a fair picture of the model's performance across all classes. These are treated similarly, irrespective of their size or significance. The model's capacity to generalize across different classes and to spot potential performance flaws across minority classes may both be evaluated using the macro average. The weighted average takes into account the class distribution by calculating the average metric weighted by the number of true instances in each class. This means that the classes with more instances contribute more to the overall average, reflecting their importance in the dataset. The weighted average is valuable for evaluating the models' performance in real-world scenarios where a class imbalance is prevalent, ensuring that the evaluation reflects the practical impact of the models.

Furthermore, we have compared the performance of the six models by looking at their precision, recall, and F1 score. Precision measures the accuracy of the positive predictions, recall measures the proportion of actual positive instances that are correctly predicted as positive, and the F1 score is the harmonic mean of precision and recall. Table VI presents further evaluation performance metrics.

The SVM model has a high precision and recall, with a precision of 100 % for the healthy group and 86 % for the sick group, and a recall of 81.25 % for the healthy group and 100 % for the sick group. The weighted average precision is 93 %, recall is 92 %, and F1 score is 92 %. The macro average precision is 93.48 %, recall is 91 %, and F1 score is 91.34 %.

4.2.3. ROC performance

The ROC analysis allows us to evaluate and compare the performance of the ML algorithms in terms of their ability to discriminate between positive and negative instances, providing valuable insights into the predictive capabilities of the models. The results of this analysis are crucial for understanding the strengths and weaknesses of each algorithm in the context of the specific predictive task at hand. In Fig. 11 the ROC curves depict the results obtained from the analysis of the six studied machine learning algorithms: LR algorithm (Fig. 11A), DT algorithm (Fig. 11B), KNN algorithm (Fig. 11C), RF algorithm (Fig. 11D), AdaBoost algorithm (Fig. 11E), and SVM algorithm (Fig. 11F). These curves illustrate the performance of each algorithm in terms of its ability to discriminate between positive and negative instances, providing a comprehensive overview of their predictive capabilities.

Table 4
Details of models' performance using the training set.

Models	Performance metrics of Training set				
	ACC	PRS	SS	$F1_{\text{Score}}$	AUC
LR	85.12 %	84.13 %	89.83 %	0.87	0.85
DT	93.49 %	93.33 %	94.92 %	0.94	0.93
KNN	100 %	100 %	100 %	1.00	1.00
RF	94.42 %	93.44 %	96.61 %	0.95	0.94
AdaBoost	85.12 %	85.83 %	87.29 %	0.87	0.94
SVM	86.05 %	83.85 %	92.37 %	0.88	0.85

Table 5
Details of models' performance using the testing set.

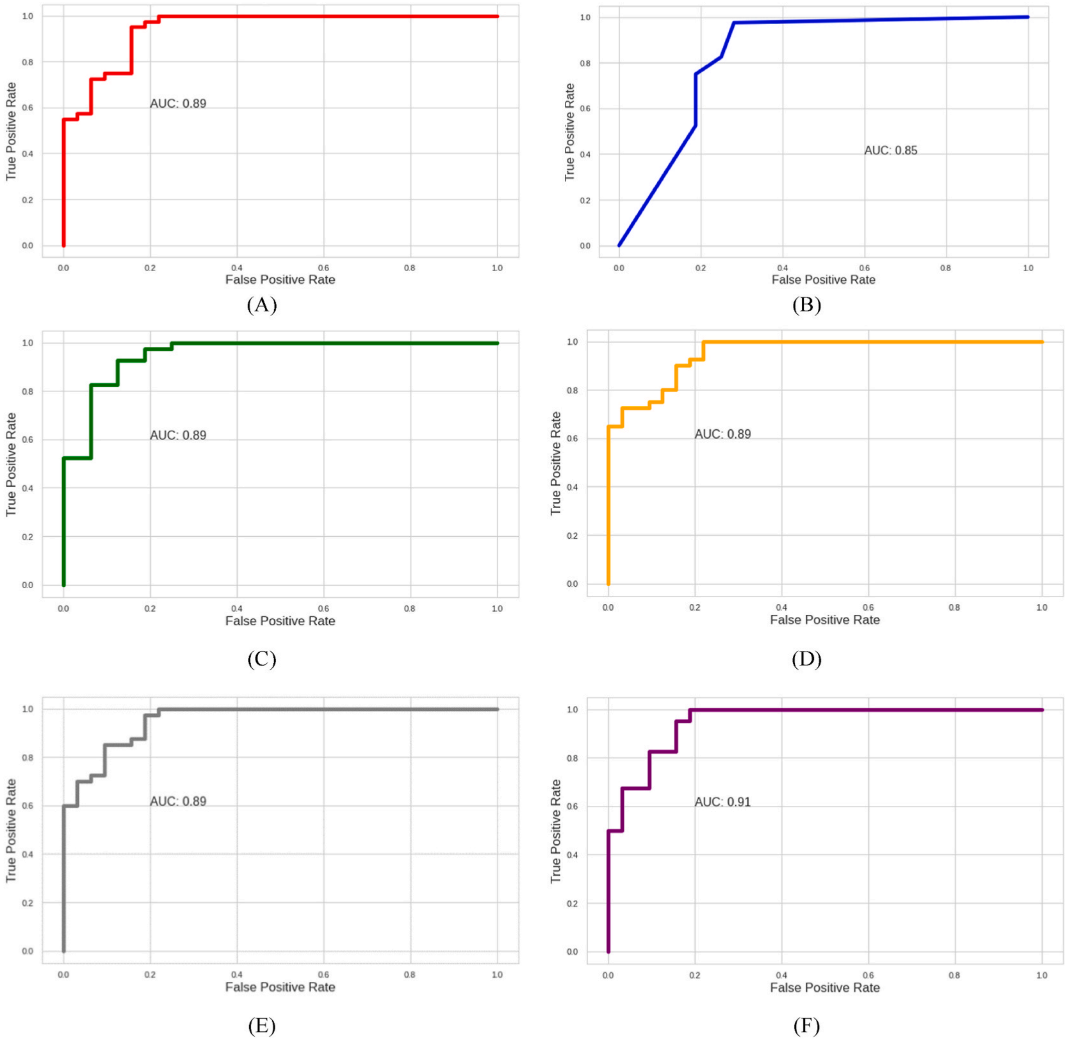
Models	Confusion matrix				Performance metrics using testing set				
	TP	TN	FN	FP	ACC	PRS	SS	F1 $_{\text{Score}}$	AUC
LR	26	39	1	6	90.28 %	86.67 %	89.38 %	0.92	0.89
DT	22	36	4	10	81.94 %	81.25 %	84.69 %	0.89	0.85
KNN	26	39	1	6	83.33 %	86.67 %	89.38 %	0.92	0.89
RF	25	40	0	7	90.28 %	85.11 %	89.06 %	0.92	0.89
AdaBoost	26	39	1	6	90.28 %	86.67 %	89.38 %	0.92	0.89
SVM	26	40	0	6	92.00 %	86.96 %	90.62 %	0.93	0.91

Table 6
Evaluation performance metrics comparison.

Models	Class	PRS (%)	SS (%)	F1 score (%)	Number of patients
LR	Healthy (0)	96.30	81.25	88.14	32
	Sick (1)	87.00	98.00	92.00	40
	Macro average	91.50	89.38	90.00	72
	Weighted average	91.00	90.28	90.15	72
DT	Healthy (0)	85.00	67.00	76.00	32
	Sick (1)	78.26	90.00	84.00	40
	Macro average	81.50	79.37	78.00	72
	Weighted average	81.09	81.00	80.23	72
KNN	Healthy (0)	96.30	81.25	88.13	32
	Sick (1)	87.00	98.00	92.00	40
	Macro average	91.50	89.37	90.00	72
	Weighted average	91.00	90.27	90.15	72
RF	Healthy (0)	100.0	78.12	88.00	32
	Sick (1)	85.11	100.0	92.00	40
	Macro average	93.00	89.06	90.00	72
	Weighted average	92.00	90.28	90.07	72
AdaBoost	Healthy (0)	96.30	81.25	88.14	32
	Sick (1)	87.00	98.00	92.00	40
	Macro average	91.50	89.38	90.00	72
	Weighted average	91.00	90.28	90.18	72
SVM	Healthy (0)	100.0	81.25	90.00	32
	Sick (1)	86.00	100.0	93.02	40
	Macro average	93.48	91.00	91.34	72
	Weighted average	93.00	92.00	92.00	72

The AUC provides a single number summary of the performance of a classifier across all possible threshold settings. A value of 1 indicates perfect performance, while a value of 0.5 indicates random performance. From Fig. 11, the AUC values for six different ML models were calculated and compared, including Logistic Regression, Decision Tree, KNN, RF, AdaBoost, and SVM. The LR model achieved an AUC value of 0.89, which indicates a good overall performance in terms of separating positive and negative classes. This value suggests that the LR classifier is able to correctly identify 89 % of positive instances and 89 % of negative instances. The Decision Tree model achieved an AUC value of 0.85, which is slightly weaker compared to the LR and some of the other models. The AUC value of 0.85 suggests that the Decision Tree classifier is able to correctly identify 85 % of positive instances and 85 % of negative instances. The KNN model achieved an AUC value of 0.89, which is similar to the LR model. This suggests that the KNN classifier is able to correctly identify 89 % of positive instances and 89 % of negative instances. The RF model also achieved an AUC value of 0.89, similar to the LR and KNN models. This suggests that the RF classifier is able to correctly identify 89 % of positive instances and 89 % of negative instances. The AdaBoost model achieved an AUC value of 0.89, similar to the Logistic Regression, KNN, and RF models. This suggests that the AdaBoost classifier is able to correctly identify 89 % of positive instances and 89 % of negative instances. Finally, the SVM model achieved the highest AUC value of 0.91, which indicates the best overall performance among the models considered. The AUC value of 0.91 suggests that the SVM classifier is able to correctly identify 91 % of positive instances and 91 % of negative instances.

The confusion matrix, combined with the results from the ROC graph and the evaluation metrics, provides a comprehensive view of the performance of the classifiers. By carefully analyzing these results, we can determine which classifier is the most appropriate for our problem, and make informed decisions about how to set the hyperparameters to achieve the desired performance. Therefore, the results of this study suggest that the SVM model achieved the best performance among the models considered, with an AUC value of 0.91. The Logistic Regression, KNN, RF, AdaBoost, and Decision Tree models all performed similarly, with AUC values ranging from 0.85 to 0.89. These results suggest that the choice of model may not have a large impact on the overall performance, but it is still important to carefully evaluate the performance of each model and make informed decisions about which one to use for a given problem.



(A) (B)

(C) (D)

(E) (F)

Fig. 11. ROC curves results given by the studied models based on the six algorithms. (A) LR algorithm. (B) Decision Tree algorithm. (C) KNN algorithm. (D) RF algorithm. (E) AdaBoost algorithm. (F) SVM algorithm.

4.3. Model validation using statlog heart dataset

In order to validate the performances of the proposed model on new unseen samples, a new dataset has been used. This dataset is the Statlog heart dataset which consists of 14 attributes including the target and 270 instances [56]. The 14 attributes are depicted as follow: the ‘age’ with a range from 29 to 77, the ‘sex’ that represents the gender of the patients (0 for female and 1 for male), the ‘chest’ that stands for the chest pain type, the ‘resting blood pressure’ in mmHG with a range of 94-200, the ‘serum cholesterol’ in mg/dl from 126 to 564, the ‘fasting blood sugar’ which is a binary feature (1 for a level >120 mg/dl and 0 for a level <120 mg/dl), the ‘resting electrocardiographic results’ with a range 0–2, the ‘maximum heart rate achieved’ from 71 to 202, the ‘exercise induced angina’ that represents the existence of angina pain in a binary type (1: yes, 0: no), the ‘oldpeak’ describes the ST depression induced by exercise, the ‘slope’ that has 3 values (upsloping: 1, flat: 2, downsloping: 3), the ‘the number of major vessels’ that are colored by fluoroscopy which is a categorical feature from 0 to 3, the ‘thal’ indicates the type of thalassemia (normal: 3, fixed defect: 6, reversible defect: 7) and the last feature is the ‘target’ which describes the absence (0) or the presence (1) of CVD.

For the validation phase of the SVM based model as the best experimented algorithm, the same hyperparameters (Table III) were used such as the penalty of misclassification (C = 2), a “linear” kernel coefficient, and a tolerance value of 0.01. As a result, the model achieved approximatively the same accuracy comparing with the performances obtained using the Cleveland dataset. It provided an accuracy of 91,18 %, a precision of 90,48 %, a recall of 95 %, an F1 score of 92,69 %, and an AUC of 90,36 %. These metrics were calculated from the confusion matrix in Fig. 12.

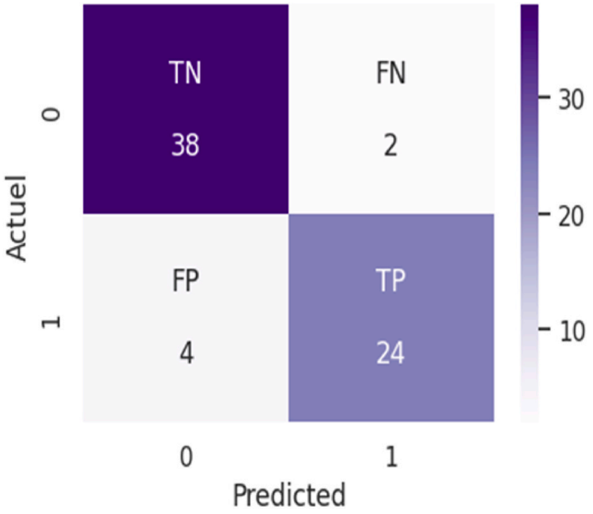


Fig. 12. The key indicators of the confusion matrix of the proposed model using the Statlog heart dataset.

4.4. 10-Fold cross-validation

To provide a more robust evaluation and reduce the variance of the model performance, using 10-fold cross-validation with a 95 % confidence interval, may better reflect the level of the true performance and how reliable the system will react with new data by averaging results across multiple splits. For this purpose, we split our two datasets into 10 folds. The 95 % confidence interval using the Cleveland dataset is comprised between 86.59 % and 92.17 % while the interval obtained using Statlog heart dataset ranges from 86.93 % to 91.58 %.

4.5. Discussion

The study aimed to evaluate the performance of six ML algorithms for diagnosing heart disease based on the Cleveland database. The models were evaluated using four evaluation metrics: precision, recall, F1 score, and number of patients. The models were logistic regression, decision tree, KNN, RF, AdaBoost, and SVM. The results showed that the SVM and RF models had the best overall performance, with weighted F1 scores of 92 % and 90.07 % respectively. Both models had 100 % precision in detecting healthy patients, while the SVM model had a 100 % recall rate in detecting sick patients and the RF model had a recall rate of 100 % for healthy patients and 85.11 % for sick patients. The LR and AdaBoost models performed similarly, with weighted F1 scores of 90.15 % and 90.18 % respectively. The LR model had a precision of 96.30 % in detecting healthy patients and 87 % in detecting sick patients, while the AdaBoost model had a precision of 96.30 % for healthy patients and 87 % for sick patients. The decision tree model had a lower overall performance, with a weighted F1 score of 80.23 %. The decision tree model had a precision of 85 % in detecting healthy patients and 78.26 % in detecting sick patients. The KNN model had similar performance to the LR model, with a weighted F1 score of 90.15 %.

Based on the performance metrics provided, it seems that the obtained results in this study exceed the existing studies. The SVM model, for example, has a higher macro average F1 score (91.34 %) and weighted average F1 score (92 %) compared to other models. Additionally, it has the highest precision for both healthy (100 %) and sick (86 %) patients, and the highest recall for sick patients (100 %). The high precision and recall scores indicate that the models in this study have a good ability to accurately predict the status of patients as healthy or sick, which is a critical aspect of medical diagnosis. The high F1 scores show that the models have a good balance between precision and recall, which is an important consideration when evaluating the performance of ML models. The novelty of our approach is the use of a hyperparameter tuning technique that allow us to find the ideal combination of parameters that enhances the accuracy and overall performance of our model. Additionally, the performances of the model reached by using a new dataset (Statlog Heart dataset) are approximatively close to the obtained results from the training and testing sets of the Cleveland dataset. Thus, the model can perform well using other datasets. As a final evaluation step, 10-fold cross-validation with a 95 % confidence interval was conducted on the two datasets. Table VII summarizes the reviewed studies, detailing the methods used and the best results achieved as determined by the evaluation metrics employed.

While the results of this study seem promising, it is important to acknowledge the limitations of these models and the data used to evaluate their performance. One limitation is the limited sample size of 72 patients used to evaluate the models. A larger sample size could provide a more robust evaluation of the models and account for any potential biases in the data.

5. Conclusion and perspectives

A common approach in ML is to conduct a comparative analysis of different algorithms in order to determine the best performing

Table 7
Comparison of the proposed performance and previous studies.

Study	Dataset	Method	Algorithm	Accuracy
[35]	Cleveland dataset	Feature selection and 10-fold cross-validation. The best accuracy is obtained by the vote method (hybrid technique with NB and LR) with 9 significant features.	Naïve Bayes	84.81 %
			SVM	85.19 %
			Vote	87.41 %
[39]	Cleveland dataset	Six data mining tools (Orange, Weka, RapidMiner, Knime, Matlab, and Scikit-learn). Six algorithms are trained and evaluated. 10-folds cross-validation for sampling the dataset.	Naïve Bayes	83.16–84.15 %
			Random Forest	74.41–84.48 %
			LR	82.49–83.84 %
			SVM	81.14–83.84 %
			ANN	71.04–85.86 %
			KNN	63.64–81.48 %
[33]	Cleveland dataset	Train-test split used to train and evaluate the model (training: 70 %, validation: 15 %, testing: 15 %).	KNN	88.00 %
[34]	Cleveland dataset	Features selection and train-test split used to train and evaluate the model (training: 80 %, testing: 20 %).	ANN	91.41 %
			AdaBoost	72.22 %
This work	Cleveland dataset	Hyperparameter tuning for six ML algorithms was trained and evaluated using a train-test split (training: 75 %, testing: 25 %).	SVM	**92 %**
		Then evaluated using 10-folds cross-validation		**86.59–92.17 %**
	Statlog heart dataset	The model was validated by 25 % of the dataset's samples.		**91.18 %**
		Then evaluated using 10-folds cross-validation		**86.93–91.58 %**

one for a specific problem. In this study, we compared the performance of several commonly utilized supervised ML algorithms for classification, including Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Decision Tree, Random Forest, and Adaptive Boosting. These algorithms were evaluated on the Heart Disease Cleveland dataset in terms of accuracy, precision, recall, and F1 score, to determine the best performing one for our final model. To optimize the model parameters, we utilized a hyperparameter tuning approach based on pipeline techniques. After preprocessing, which involved reducing the number of patients from 303 to 287 samples, selecting 13/76 features, removing missing data, and normalizing the dataset using the Z-score approach, the experimental results showed that the SVM model outperformed the other five ML algorithms in terms of accuracy. Further, the performance evaluation metrics, including the ROC plot, were used to enhance the efficiency of the predictive system. Furthermore, a comparative analysis was made between our results and those from several available literature works. The study showed that the proposed predictive system for atherosclerosis had a main accuracy rate of 92 % during the testing phase and an accuracy of 91,18 % in the validation phase performed using the Statlog heart dataset. In addition, the results of 10-fold cross-validation with 95 % confidence interval confirmed the reliability of the model, with a range between 86.59 % and 92.17 % for the Cleveland dataset and from 86.93 % to 91.58 % obtained using Statlog heart dataset. Therefore, developing an efficient and robust ML model for predicting heart disease has the potential to save lives and improve public health.

Nevertheless, it is important to acknowledge some limitations of our study. Firstly, while the Cleveland and the Statlog datasets contain only clinical features, cardiovascular diseases can be approached by various ways, one of them is to focus on the tissues of the heart by analyzing radiomics features such as the pericoronaric adipose tissue [57] and the Epicardial and thoracic subcutaneous fat texture [58]. Additionally, the first feature of the dataset is age which is a demographic feature that has a range between 29 and 71, Despite that being a large population, the limitation is due to the reduced number of samples. Furthermore, the ethnic factor is not taken into consideration, there is no feature describing the ethnicity of each patient. It would be more interesting to focus on a wide representativeness of the population with different ranges of age [59] and more diversified ethnicity [60], in order to prove the applicability of our model on different populations.

In terms of future works, several directions can be pursued to further improve the performance of the proposed predictive system for heart disease. Firstly, incorporating other related data sources, such as demographic information, lifestyle factors [61], genetic data [62], and medical history, could provide additional insight into the risk of heart disease and enhance the model's performance. Furthermore, employing different datasets including real-world data provides generalizability and more precision of our models to predict different heart diseases based on bioinformatics analysis approaches [63,64]. Secondly, exploring more advanced ML techniques, such as deep learning, could potentially lead to better performance results. Additionally, incorporating more extensive feature selection and dimensionality reduction methods could reduce the noise in the data and improve the robustness of the model. Finally, conducting a large-scale validation study on a diverse population with varying demographic and lifestyle characteristics would further strengthen the generalizability of the model and its potential real-world applications.

Data availability statement

The data used in this paper is publicly available and referenced in the article.

CRediT authorship contribution statement

Mohammed Amine Bouqentar: Writing – original draft, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Oumaima Terrada:** Writing – review & editing, Software, Resources, Investigation, Formal analysis, Data curation, Conceptualization. **Soufiane Hamida:** Writing – review & editing, Methodology, Investigation. **Shawki Saleh:** Writing – review & editing, Visualization. **Driss Lamrani:** Writing – review & editing, Visualization. **Bouchaib Cherradi:** Writing – review & editing, Validation, Supervision, Project administration, Investigation. **Abdelhadi Raihani:** Writing – review & editing, Validation, Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H.V. Denysyuk, R.J. Pinto, P.M. Silva, R.P. Duarte, F.A. Marinho, L. Pimenta, A.J. Gouveia, N.J. Gonçalves, P.J. Coelho, E. Zdravevski, P. Lameski, V. Leithardt, N.M. Garcia, I.M. Pires, Algorithms for automated diagnosis of cardiovascular diseases based on ECG data: a comprehensive systematic review, Heliyon 9 (2023) e13601, https://doi.org/10.1016/j.heliyon.2023.e13601.

[2] C. Collins, D. Dennehy, K. Conboy, P. Mikalef, Artificial intelligence in information systems research: a systematic literature review and research agenda, Int. J. Inf. Manag. 60 (2021) 102383, https://doi.org/10.1016/j.ijinfomgt.2021.102383.

[3] P.P. Shinde, S. Shah, A review of machine learning and deep learning applications, in: 2018 Fourth Int. Conf. Comput. Commun. Control Autom. ICCUBEA, 2018, pp. 1–6, https://doi.org/10.1109/ICCUBEA.2018.8697857.

[4] A. Basak, K.M. Schmidt, O.J. Mengshoel, From data to interpretable models: machine learning for soil moisture forecasting, Int. J. Data Sci. Anal. 15 (2023) 9–32, https://doi.org/10.1007/s41060-022-00347-8.

[5] S. Zhang, H. Zhou, L. Zhang, Recent machine learning progress in image analysis and understanding, Adv. Multimed. 2018 (2018) 1–2, https://doi.org/10.1155/2018/1685890.

[6] N. Ait Ali, B. Cherradi, A. El Abbassi, O. Bouattane, M. Youssfi, GPU fuzzy c-means algorithm implementations: performance analysis on medical image segmentation, Multimed. Tools Appl. 77 (2018) 21221–21243, https://doi.org/10.1007/s11042-017-5589-6.

[7] T. Wang, H. Guo, Z. Ge, Q. Zhang, Z. Yang, An MMSE graph spectral magnitude estimator for speech signals residing on an undirected multiple graph, EURASIP, J. Audio Speech Music Process 2023 (2023) 7, https://doi.org/10.1186/s13636-023-00272-z.

[8] N. Tang, C. Do, T.B. Dinh, T.B. Dinh, Urban traffic monitoring system, in: D.-S. Huang, Y. Gan, P. Gupta, M.M. Gromiha (Eds.), Adv. Intell. Comput. Theor. Appl. Asp. Artif. Intell., Springer, Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 573–580, https://doi.org/10.1007/978-3-642-25944-9_74.

[9] A. Ait Ouallane, A. Bakali, A. Bahnasse, S. Broumi, M. Talea, Fusion of engineering insights and emerging trends: intelligent urban traffic management system, Inf. Fusion 88 (2022) 218–248, https://doi.org/10.1016/j.inffus.2022.07.020.

[10] P.K. Kannan, H. "Alice" Li, Digital marketing: a framework, review and research agenda, Int. J. Res. Mark. 34 (2017) 22–45, https://doi.org/10.1016/j.ijresmar.2016.11.006.

[11] M.R. Bachute, J.M. Subhedar, Autonomous driving architectures: insights of machine learning and deep learning algorithms, Mach. Learn. Appl. 6 (2021) 100164, https://doi.org/10.1016/j.mlwa.2021.100164.

[12] V.N. Dornadula, S. Geetha, Credit card fraud detection using machine learning algorithms, Procedia Comput. Sci. 165 (2019) 631–641, https://doi.org/10.1016/j.procs.2020.01.057.

[13] U. Tiwari, M. Jain, S. Mehfuz, Handwritten character recognition—an analysis, in: S.N. Singh, F. Wen, M. Jain (Eds.), Adv. Syst. Optim. Control, Springer Singapore, Singapore, 2019, pp. 207–212, https://doi.org/10.1007/978-981-13-0665-5_18.

[14] V. Shorewala, Early detection of coronary heart disease using ensemble techniques, Inform. Med. Unlocked 26 (2021) 100655, https://doi.org/10.1016/j.imu.2021.100655.

[15] S.P. Patro, G.S. Nayak, N. Padhy, Heart disease prediction by using novel optimization algorithm: a supervised learning prospective, Inform. Med. Unlocked 26 (2021) 100696, https://doi.org/10.1016/j.imu.2021.100696.

[16] I.D. Mienye, Y. Sun, Z. Wang, An improved ensemble learning approach for the prediction of heart disease risk, Inform. Med. Unlocked 20 (2020) 100402, https://doi.org/10.1016/j.imu.2020.100402.

[17] S. Huang, J.T. Xu, M. Yang, Review: predictive approaches to breast cancer risk, Heliyon 9 (2023) e21344, https://doi.org/10.1016/j.heliyon.2023.e21344.

[18] Md.M. Islam, Md.R. Haque, H. Iqbal, Md.M. Hasan, M. Hasan, M.N. Kabir, Breast cancer prediction: a comparative study using machine learning techniques, SN Comput. Sci. 1 (2020) 290, https://doi.org/10.1007/s42979-020-00305-w.

[19] C. Kishor Kumar Reddy, P.R. Anisha, K. Apoorva, Early prediction of pneumonia using convolutional neural network and X-ray images, in: S.C. Satapathy, V. Bhateja, M.N. Favorskaya, T. Adilakshmi (Eds.), Smart Comput. Tech. Appl., Springer Singapore, Singapore, 2021, pp. 673–681, https://doi.org/10.1007/978-981-16-1502-3_67.

[20] H. Moujahid, B. Cherradi, O.E. Gannour, L. Bahatti, O. Terrada, S. Hamida, Convolutional neural network based classification of patients with pneumonia using X-ray lung images, Adv. Sci. Technol. Eng. Syst. J. 5 (2020) 167–175, https://doi.org/10.25046/aj050522.

[21] R. Jain, P. Nagrath, G. Kataria, V. Sirish Kaushik, D. Jude Hemanth, Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning, Measurement 165 (2020) 108046, https://doi.org/10.1016/j.measurement.2020.108046.

[22] H. Moujahid, B. Cherradi, M. Al-Sarem, L. Bahatti, Diagnosis of COVID-19 disease using convolutional neural network models based transfer learning, in: F. Saeed, F. Mohammed, A. Al-Nahari (Eds.), Innov. Syst. Intell. Health Inform., Springer International Publishing, Cham, 2021, pp. 148–159, https://doi.org/10.1007/978-3-030-70713-2_16.

[23] O. Daanouni, B. Cherradi, A. Tmiri, NSL-MHA-CNN: a novel CNN architecture for robust diabetic retinopathy prediction against adversarial attacks, IEEE Access 10 (2022) 103987–103999, https://doi.org/10.1109/ACCESS.2022.3210179.

[24] W.L. Alyoubi, W.M. Shalash, M.F. Abulkhair, Diabetic retinopathy detection through deep learning techniques: a review, Inform. Med. Unlocked 20 (2020) 100377, https://doi.org/10.1016/j.imu.2020.100377.

[25] W. Liu, J. Liu, T. Peng, G. Wang, V.E. Balas, O. Geman, H.-W. Chiu, Prediction of Parkinson's disease based on artificial neural networks using speech datasets, J. Ambient Intell. Humaniz. Comput (2022), https://doi.org/10.1007/s12652-022-03825-w.

[26] A. Ouhmida, A. Raihani, B. Cherradi, O. Terrada, A novel approach for Parkinson's disease detection based on voice classification and features selection techniques, Int. J. Online Biomed. Eng. IJOE 17 (2021) 111, https://doi.org/10.3991/ijoe.v17i10.24499.

[27] A. Ouhmida, O. Terrada, A. Raihani, B. Cherradi, S. Hamida, Voice-based deep learning medical diagnosis system for Parkinson's disease prediction, in: 2021 Int. Congr. Adv. Technol. Eng. ICOTEN, 2021, pp. 1–5, https://doi.org/10.1109/ICOTEN52080.2021.9493456.

[28] A.M. Rahmani, E. Yousefpoor, M.S. Yousefpoor, Z. Mehmood, A. Haider, M. Hosseinzadeh, R. Ali Naqvi, Machine learning (ML) in medicine: review, applications, and challenges, Mathematics 9 (2021) 2970, https://doi.org/10.3390/math9222970.

[29] T. Thomas, A.N. Kurian, Artificial intelligence of things for early detection of cardiac diseases, in: F. Al-Turjman, A. Nayyar (Eds.), Mach. Learn. Crit. Internet Med. Things, Springer International Publishing, Cham, 2022, pp. 81–102, https://doi.org/10.1007/978-3-030-80928-7_4.

[30] I. El Naqa, M.J. Murphy, What is machine learning? in: I. El Naqa, R. Li, M.J. Murphy (Eds.), Mach. Learn. Radiat. Oncol. Theory Appl. Springer International Publishing, Cham, 2015, pp. 3–11, https://doi.org/10.1007/978-3-319-18305-3_1.

[31] F. Ali, S. El-Sappagh, S.M.R. Islam, D. Kwak, A. Ali, M. Imran, K.-S. Kwak, A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion, Inf. Fusion 63 (2020) 208–222, https://doi.org/10.1016/j.inffus.2020.06.008.

[32] K.M. Zobair, L. Houghton, D. Tjondronegoro, L. Sanzogni, M.Z. Islam, T. Sarker, M.J. Islam, Systematic review of Internet of medical things for cardiovascular disease prevention among Australian first nations, Heliyon 9 (2023) e22420, https://doi.org/10.1016/j.heliyon.2023.e22420.

[33] O. Terrada, B. Cherradi, A. Raihani, O. Bouattane, Atherosclerosis disease prediction using supervised machine learning techniques, in: 2020 1st Int. Conf. Innov. Res. Appl. Sci. Eng. Technol. IRASET, 2020, pp. 1–5, https://doi.org/10.1109/IRASET48871.2020.9092082.

[34] O. Terrada, B. Cherradi, S. Hamida, A. Raihani, H. Moujahid, O. Bouattane, Prediction of patients with heart disease using artificial neural network and adaptive boosting techniques, in: 2020 3rd Int. Conf. Adv. Commun. Technol. Netw. CommNet, 2020, pp. 1–6, https://doi.org/10.1109/CommNet49926.2020.9199620.

[35] M.S. Amin, Y.K. Chiam, K.D. Varathan, Identification of significant features and data mining techniques in predicting heart disease, Telemat. Inform. 36 (2019) 82–93, https://doi.org/10.1016/j.tele.2018.11.007.

[36] A.K. Dwivedi, Performance evaluation of different machine learning techniques for prediction of heart disease, Neural Comput. Appl. 29 (2018) 685–693, https://doi.org/10.1007/s00521-016-2604-1.

[37] A. Bhatt, S.K. Dubey, A.K. Bhatt, M. Joshi, Data mining approach to predict and analyze the cardiovascular disease, in: S.C. Satapathy, V. Bhateja, S.K. Udgata, P.K. Pattnaik (Eds.), Proc. 5th Int. Conf. Front. Intell. Comput. Theory Appl., Springer, Singapore, 2017, pp. 117–126, https://doi.org/10.1007/978-981-10-3153-3_12.

[38] A.K. Garate Escamilla, A. Hajjam El Hassani, E. Andres, A comparison of machine learning techniques to predict the risk of heart failure, in: G.A. Tsihrintzis, M. Virvou, E. Sakkopoulos, L.C. Jain (Eds.), Mach. Learn. Paradig. Appl. Learn. Anal. Intell. Syst., Springer International Publishing, Cham, 2019, pp. 9–26, https://doi.org/10.1007/978-3-030-15628-2_2.

[39] I. Tougui, A. Jilbab, J. El Mhamdi, Heart disease classification using data mining tools and machine learning techniques, Health Technol. 10 (2020) 1137–1144, https://doi.org/10.1007/s12553-020-00438-1.

[40] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K.H. Guppy, S. Lee, V. Froelicher, International application of a new probability algorithm for the diagnosis of coronary artery disease, Am. J. Cardiol. 64 (1989) 304–310, https://doi.org/10.1016/0002-9149(89)90524-9.

[41] B. Cherradi, O. Terrada, A. Ouhmida, S. Hamida, A. Raihani, O. Bouattane, Computer-aided diagnosis system for early prediction of atherosclerosis using machine learning and K-fold cross-validation, in: 2021 Int. Congr. Adv. Technol. Eng. ICOTEN, 2021, pp. 1–9, https://doi.org/10.1109/ICOTEN52080.2021.9493524.

[42] R. Indrakumari, T. Poongodi, S.R. Jena, Heart disease prediction using exploratory data analysis, Procedia Comput. Sci. 173 (2020) 130–139, https://doi.org/10.1016/j.procs.2020.06.017.

[43] A. Curtis, T. Smith, B. Ziganshin, J. Elefteriades, The mystery of the Z-score, AORTA 04 (2016) 124–130, https://doi.org/10.12945/j.aorta.2016.16.014.

[44] P. Schober, T.R. Vetter, Logistic regression in medical research, Anesth. Analg. 132 (2021) 365–366, https://doi.org/10.1213/ANE.0000000000005247.

[45] S. Uddin, A. Khan, M.E. Hossain, M.A. Moni, Comparing different supervised machine learning algorithms for disease prediction, BMC Med. Inform. Decis. Mak. 19 (2019) 281, https://doi.org/10.1186/s12911-019-1004-8.

[46] Y.-H. Shao, W.-J. Chen, N.-Y. Deng, Nonparallel hyperplane support vector machine for binary classification problems, Inf. Sci. 263 (2014) 22–35, https://doi.org/10.1016/j.ins.2013.11.003.

[47] J. Bjurgert, P.E. Valenzuela, C.R. Rojas, On adaptive boosting for system identification, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 4510–4514, https://doi.org/10.1109/TNNLS.2017.2754319.

[48] B. Charbuty, A. Abdulazeez, Classification based on decision tree algorithm for machine learning, J. Appl. Sci. Technol. Trends 2 (2021) 20–28, https://doi.org/10.38094/jastt20165.

[49] X. Chen, Z. Yang, W. Lou, Fault diagnosis of rolling bearing based on the permutation entropy of VMD and decision tree, in: 2019 3rd Int. Conf. Electron. Inf. Technol. Comput. Eng. EITCE, 2019, pp. 1911–1915, https://doi.org/10.1109/EITCE47263.2019.9095187.

[50] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, S. Homayouni, Support vector machine versus random forest for remote sensing image classification: a meta-analysis and systematic review, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 6308–6325, https://doi.org/10.1109/JSTARS.2020.3026724.

[51] R. Ehsani, F. Drabløs, Robust distance measures for kNN classification of cancer data, Cancer Inform 19 (2020) 1176935120965542, https://doi.org/10.1177/1176935120965542.

[52] W. Xia, R. Zhang, X. Zhang, M. Usman, A novel method for diagnosing Alzheimer's disease using deep pyramid CNN based on EEG signals, Heliyon 9 (2023) e14858, https://doi.org/10.1016/j.heliyon.2023.e14858.

[53] M.R. Hassan, S. Al-Insaif, M.I. Hossain, J. Kamruzzaman, A machine learning approach for prediction of pregnancy outcome following IVF treatment, Neural Comput. Appl. 32 (2020) 2283–2297, https://doi.org/10.1007/s00521-018-3693-9.

[54] R. Valarmathi, T. Sheela, Heart disease prediction using hyper parameter optimization (HPO) tuning, Biomed. Signal Process Control 70 (2021) 103033, https://doi.org/10.1016/j.bspc.2021.103033.

[55] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[56] H. Dr, R. ur, Nabila Kausar, A comparative analysis on Cleveland and statlog heart disease datasets using data mining techniques. https://doi.org/10.5281/ZENODO.5149764, 2021.

[57] C. Militello, F. Prinzi, G. Sollami, L. Rundo, L. La Grutta, S. Vitabile, CT radiomic features and clinical biomarkers for predicting coronary artery disease, Cogn. Comput. 15 (2023) 238–253, https://doi.org/10.1007/s12559-023-10118-7.

[58] M. Agnese, P. Toia, G. Sollami, C. Militello, L. Rundo, S. Vitabile, E. Maffei, F. Agnello, C. Gagliardo, E. Grassedonio, M. Galia, F. Cademartiri, M. Midiri, L. L. Grutta, Epicardial and thoracic subcutaneous fat texture analysis in patients undergoing cardiac CT, Heliyon 9 (2023) e15984, https://doi.org/10.1016/j.heliyon.2023.e15984.

[59] M.N. Islam, K.R. Raiyan, S. Mitra, M.M.R. Mannan, T. Tasnim, A.O. Putul, A.B. Mandol, Predictis: an IoT and machine learning-based system to predict risk level of cardio-vascular diseases, BMC Health Serv. Res. 23 (2023) 171, https://doi.org/10.1186/s12913-023-09104-4.

[60] F.M. Calisto, N. Nunes, J.C. Nascimento, Modeling adoption of intelligent agents in medical imaging, Int. J. Hum. Comput. Stud. 168 (2022) 102922, https://doi.org/10.1016/j.ijhcs.2022.102922.

[61] N. Pamarthi, P.T. Satyanarayana Murty, A.S. Mallesh, P.K. Sree, S.R. Dangeti, B. Maram, A research study of heart health monitoring using deep learning and IoT, in: 2023 1st DMIHER Int. Conf. Artif. Intell. Educ. Ind. 40 IDICAIEI, IEEE, Wardha, India, 2023, pp. 1–6, https://doi.org/10.1109/IDICAIEI58380.2023.10406326.

[62] J. Abrantes, External validation of a deep learning model for breast density classification. https://doi.org/10.26044/ECR2023/C-16014, 2023.

[63] K.-K. Huang, H.-L. Zheng, S. Li, Z.-Y. Zeng, Identification of hub genes and their correlation with immune infiltration in coronary artery disease through bioinformatics and machine learning methods, J. Thorac. Dis. 14 (2022) 2621–2634, https://doi.org/10.21037/jtd-22-632.

[64] T. Wang, Y. Sun, Y. Zhao, J. Huang, Y. Huang, Identification of hub genes in heart failure by integrated bioinformatics analysis and machine learning, Front. Cardiovasc. Med. 10 (2023) 1332287, https://doi.org/10.3389/fcvm.2023.1332287.