Performance Evaluation and Comparative Analysis of Different Machine Learning Algorithms in Predicting Cardiovascular Disease

Md. Asfi-Ar-Raihan Asif, Mirza Muntasir Nishat, Fahim Faisal, Rezuanur Rahman Dip, Mahmudul Hasan Udoy, Md. Fahim Shikder and Ragib Ahsan

***Abstract*—This study focuses on investigating the performance of different machine learning algorithms and corresponding comparative analysis in predicting cardiovascular disease. Globally this fatal disease causes a plethora of mortality among mankind and so, machine learning algorithms can play a significant role in early detection which will ensure proper treatment for the patients and reduce severity in many cases. The University of California, Irvine (UCI) data repository is utilized for the training and testing of the model. Twelve machine learning algorithms were studied and the performances were observed for default hyperparameter (DHP), grid search cross validation (GSCV) and random search cross validation (RSCV) method. Moreover, computational time were also calculated for both GSCV and RSCV. An accuracy of 92% has been found in both hard and soft voting ensemble classifiers (EVCH and EVCS). However, it observed that Adaboost algorithm outperforms EVCH and EVCS in terms of precision and specificity . Hence, the overall comparative analyses among all the algorithms are carried out extensively where accuracy, precision, sensitivity, specificity, F1 score, and ROC-AUC are brought into action. Jupyter notebook 6.0.3 is utilized for simulation.**

***Index Terms*—Cardiovascular disease, UCI dataset, Accuracy, Machine Learning Algorithms**

Manuscript received September 09, 2020; revised April 08, 2021.

Md. Asfi-Ar-Raihan Asif is an undergraduate student in the Department of EEE, Islamic University of Technology, Dhaka, Bangladesh (Email: raihanasif@iut-dhaka.edu).

Mirza Muntasir Nishat is a Lecturer in the Department of EEE, Islamic University of Technology, Dhaka, Bangladesh. (Email: mirzamuntasir@iut-dhaka.edu).

Fahim Faisal is an Assistant Professor of Department of EEE, Islamic University of Technology, Dhaka, Bangladesh (Email: faisaleee@iut-dhaka.edu). (Corresponding author)

Rezuanur Rahman Dip is an undergraduate student in the Department of EEE, Islamic University of Technology, Dhaka, Bangladesh (Email: rezuanurrahman@iut-dhaka.edu).

Mahmudul Hasan Udoy is an undergraduate student in the Department of EEE, Islamic University of Technology, Dhaka, Bangladesh (Email: mahmudul81@iut-dhaka.edu).

Md. Fahim Shikder is an undergraduate student in the Department of EEE, Islamic University of Technology, Dhaka, Bangladesh (Email: fahimshikder@iut-dhaka.edu).

Ragib Ahsan is an undergraduate student in the Department of EEE, Islamic University of Technology, Dhaka, Bangladesh (Email: ragibahsan@iut-dhaka.edu).

I. Introduction

CARDIOVASCULAR disease, taking an estimated 17.9 million lives annually is ranked on the top of the chart as the cause of people's demise around the world [1]. High blood pressure, overweight, high cholesterol, family history, smoking, and alcohol are the prime reasons for this fatal disease which are very commonly observed among different ages of people in recent times. However, prompt and accurate diagnosis of the cardiovascular disease seems to be quite challenging for healthcare professionals as this disease is associated with various symptoms. Therefore, a huge amount of data are being collected globally by healthcare industries for discovering the insights and exploring the facts about heart diseases which will eventually assist the healthcare professionals to understand the disease better and ensure proper treatment for the patients. Nonetheless collected data require a lot of screening and processing so that information can be extracted effectively. However, these explorations on huge datasets were previously impossible with traditional statistics. Therefore, Machine learning (ML) has emerged as the most efficient tool in the modern era to process the data and utilize that information for the development of the healthcare sector [2-3].

Among all organs of the human body, the heart is the most crucial one because of its important role in blood pumping. Forecasting the condition of the heart and predicting the disease with the help of machine learning can play a vital role in reducing the mortality rate due to heart diseases [4]. Hence, exploration and extraction of huge datasets to discover hidden knowledge and patterns have drawn the attention of the researchers. And so, the implementation of machine learning algorithms in medical data can help in effective decision making, preventing error in diagnosis and reducing the death rate eventually.

II. Related Works

Many researchers have been trying to predict cardiovascular disease by employing different machine learning techniques. Jaymin et al., focusing on different techniques, executed the J48 Tree Technique, Logistic Model Tree, and RF Algorithm [5]. Among them, the J48 Tree technique was proposed with 56.76% accuracy. On the other hand, Archana et al. implemented four algorithms and

obtained the highest 87% accuracy from KNN [6]. However, Alim et al. used Logistic Regression (LR), Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosting (GB) for the prediction of the heart disease where Random Forest (RF) with a stratified K-fold model was finally proposed with an accuracy of 86.12% [7]. Furthermore, Kannan et al. applied four machine algorithms named Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and Stochastic Gradient Boosting (SGD) to predict & diagnose heart disease [8]. Though they used tenfold cross-validation in SVM and GB, the best performance was found in LR with 87% accuracy. Atallah et al. have introduced multiple machine learning techniques by using the UCI dataset with 14 attributes and obtained the best accuracy of 90% employing hard voting ensemble method [9]. Kohli et al. used Backward Elimination method, a machine learning feature to experiment with heart disease dataset [10]. Among the five algorithms brought into action, Logistic Regression (LR) exhibits the highest accuracy of 87.1% to predict the heart disease. Conversely, Abderrahmane et al. have made an application to process and monitor the data where Spark MLlib along with Spark streaming has been used for data processing. They have obtained an accuracy of 87.5%, sensitivity of 86.66% and specificity of 88.37% and have compared execution time between Spark and traditional framework and found significant improvement in Spark framework [11]. Nevertheless, Srivastava et al. introduced a hybrid machine learning model to predict heart disease where the proposed hybrid random forest with linear model (HRFLM) which is the combination of Random Forest and linear method. The fitness function was deduced with the Genetic Algorithm (GA) for tournament selection, crossover and mutation. The model achieves an overall accuracy of 88.4% [12]. It creates a new path to use neural network in this dimension. Considering the aforementioned works, our research targeted to achieve more accuracy and more satisfactory outcomes in terms of other performance metrics with a view to addressing cardiovascular diseases and detecting it in a more proficient manner.

In our work, various machine learning algorithms have been implemented to predict cardiovascular disease more accurately to assist the doctors for early diagnosis. The methodology and study of the machine learning algorithms are presented in section II and section III respectively. Extensive analyses of the obtained results from the simulation are reported in Section IV where the performances of the algorithms are evaluated by carrying out a comprehensive and comparative study. Lastly, the conclusion of the research is outlined in section V.

III. Methodology

For any machine learning model, the dataset plays the most significant role. In this paper, we used the UCI dataset for training and testing of our models. According to many researchers and authorities of the UCI, this dataset is very well known, balanced as well as verified. Originally this dataset contains 303 instances and 76 attributes [13]. One of the famous data science toolkits, Jupyter notebook 6.0.3 from Anaconda navigator was used as a coding platform in our research. A visualization of the distribution of feature attributes was performed and Gaussian distribution is found as kernel density estimation. However, standardization or variance scaling performs better in feature scaling when the distribution of attributes is Gaussian [14-15]. This method scales data according to the following formula,

$$X' = \frac{X - \bar{X}}{\sigma}$$

Where $\bar{X}$ stands for mean & σ symbolizes standard deviation of the feature values. After that, a correlation heat map was plotted to find the paramount features from data that were efficacious for the prediction. Fig.1 illustrates the correlation heat map.

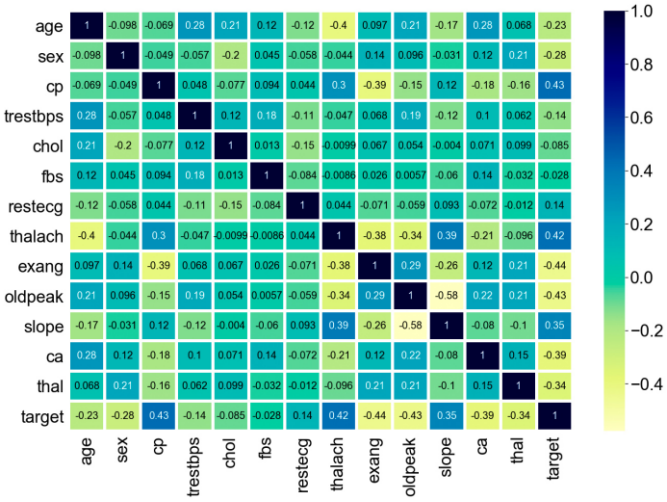


Fig.1 Correlation heatmap

According to the correlation heat map Chest Pain (CP), Thalach, slope, resting electrocardiographic result (Restecg) were highly correlated with the target variable. Table I denotes a brief description of these highly correlated features.

TABLE I
Description of Highly Correlated Attributes

Attribute	Type	Description
Chest Pain (CP)	Discrete	Chest Pain type: (a) Typical Angina (b) Atypical Angina (c) Non-anginal Pain
Thalach	Continuous	Maximum heart rate achieved
Slope	Discrete	Peak exercise slope segment: 1. Up Sloping 2. Flat 3. Down Sloping
Resting Electrocardiographic Result (Restecg)	Discrete	Resting electrocardiographic result: 0 = Normal 1 = Having ST-T Abnormality 2 = Probable Left Ventricular Hypertrophy

After data visualization & analysis, data were split into a training set & testing set. Various ratios of the train-test split were performed and it was observed that 80% of the training set and 20% of the testing set of total data was the most efficient as it depicted low bias & low variance for

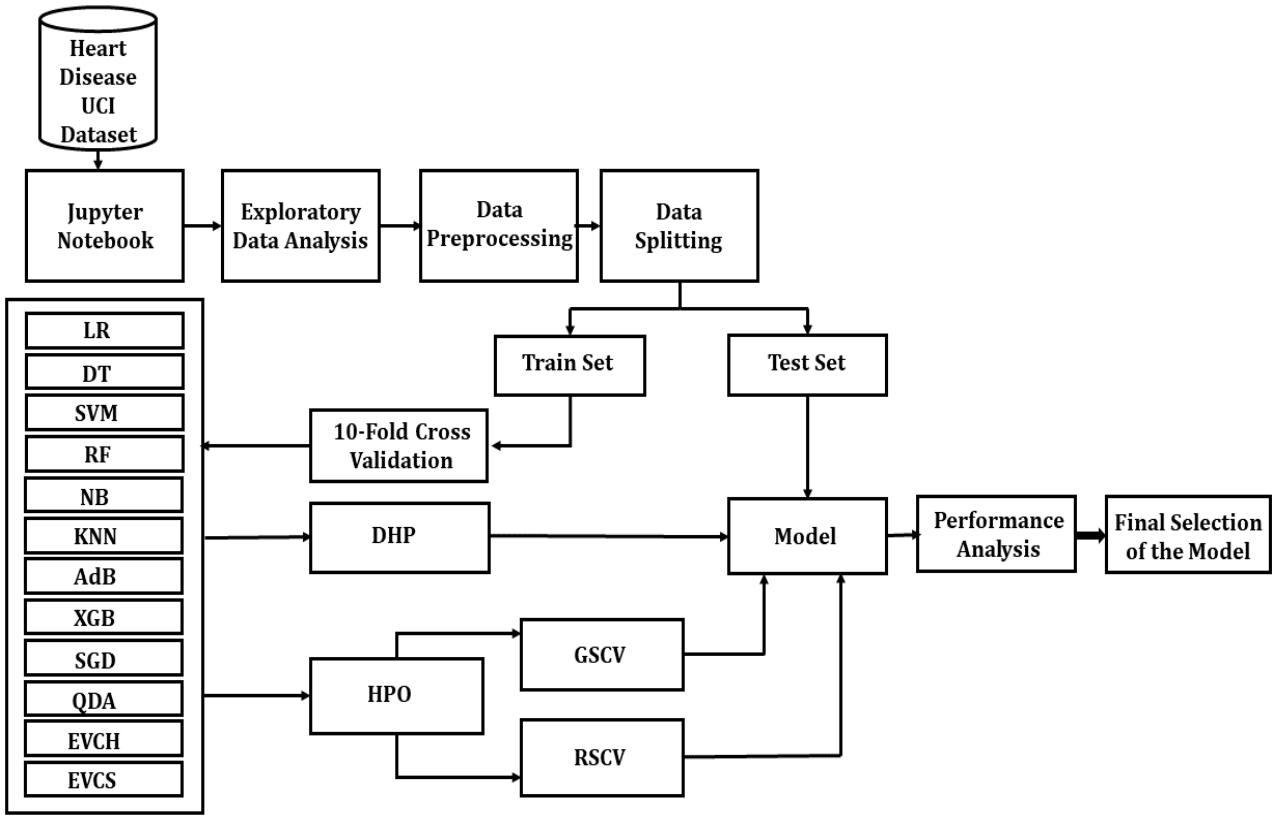


Fig. 2 Overall workflow diagram

machine learning algorithms we implemented. Twelve machine learning algorithms were investigated and ten-fold cross-validation was performed over the training data. Default hyperparameter (DHP) and Hyperparameter Optimization (HPO) were carried out so that more enhanced results can be achieved in terms of the performance metrics. Hence, quantitative and qualitative analyses are presented so that the most efficient model can be proposed finally. The overall workflow diagram is depicted in Fig. 2

IV. Study of Machine Learning (ML) Algorithms

In this section, the machine learning algorithms were studied in order to implement them in the aforementioned dataset. The performance of each algorithm was observed and tabulated in respect of the accuracy, precision, sensitivity, specificity, F1 score, and area under the ROC curve. Each algorithm with a brief description is presented below:

A. Logistic Regression (LR)

Logistic Regression (LR) is generally used for the evaluation of the probability where an instance fits a certain class [16]. Logistic regression describes the relationship between a categorical outcome quantity with one or more categorical predictor quantity by using the logit transformation to the dependent variable where the logistic model forecasts the logit of dependent variable from independent variable. The logit function and the equation of the probability is given below:

$$\hat{p} = h_\theta(x) = \sigma(x^T\theta)$$

Where,

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

However, Logistic Regression (LR) has the advantage of providing the final classification based on probability. Moreover, it can face the complete class separation problem [17].

B. Decision Tree (DT)

A tree looking model is used in the decision tree algorithm to identify possible consequences including event outcomes [18]. A discrete set of values can be taken by the target variables in the tree model. However, in tree structures, leaves signify class labels and branches symbolize feature joins that represent class labels. The equation of entropy is provided underneath,

$$E = -\sum_{b=1}^{n} p_{ab} \log_2 p_{ab} .$$

For the visualization of the interaction of the variables, the tree structure is well suited having different nodes and edges. Decision tree works well when there is monotonic transformation among the features. Nevertheless, decision tree cannot handle linear relationship and sometimes the trees are unstable. If the terminal nodes of decision tree are more, it is very much difficult to interpret the whole tree.

C. Support Vector Machine (SVM)

Support vector machines (SVM) categorize data by judging the hyperplane that increases the boundary line between the classes in the training data [19]. Hyperplane can be formulated as follows,

$$f(x) = a^T x + c$$

Where, a = dimensional coefficient, c = offset.

However, SVM has the advantage of selecting various kernels. With the help of various kernels, a much complex structured data set can be utilized. Moreover, it has also less overfitting problems. Though the kernel is the strength of the

support vector machine, it is not easy to select a certain kernel. On the other hand, it takes lots of computational time whenever the data set is larger [20].

D. Random Forest (RF)

Random Forest (RF) is a commune of the Decision Tree (DT) algorithm [21]. Decision trees consist of high variance and low bias and the variance component of the model is minimized by averaging decision trees. By averaging the prediction, the unknown samples can be made,

$$I = \frac{1}{N}\sum_{n=1}^{N} f(x)$$

where uncertainty is,

$$\sigma = \sqrt{\frac{\sum_{n=1}^{N}(f(x)-\hat{f})^2}{N-1}}$$

Random Forest (RF) algorithm uses various decision trees on data, collecting prediction from each of them and finds the best possible way of solution. It is also based on an ensemble learning technique which is based on bagging algorithm and can handle missing values of data [22].

E. Naïve Bayes Classifier (NBC)

Naive Bayes Classifier (NBC) is a popularly used classifier algorithm which follows Bayes' theorem mathematically [23].

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Above Bayes' theorem asserts an interconnection of provided class variable *y* as well as dependent feature vector x_1 through $\mathbf{x}_j$.

$$P(y \mid x_1,....x_j) = \frac{P(x_1,....x_j \mid y)P(y)}{P(x_1,....x_j)}$$

The most advantageous part of NBC is that it requires less computational time comparing with other machine learning algorithms. It can handle categorical input variables well than numerical input variables. Moreover, it conjectures all the features as independent variable which makes it difficult to implement practically [24].

F. K-Nearest Neighbor (KNN)

In K-Nearest Neighbor (KNN), the algorithm explores K instances of the dataset which is near to the observation. After that, the algorithm itself will utilize its output to evaluate the variable *y* of the inspection that should be predicted [25]. For calculating the distance of two observations, Euclidean distance is used, and the equation is as follows:

$$d(x_i, y_i) = \sqrt{(x_{i,1}-y_{i,1})^2 + ...+ (x_{i,m}-y_{i,m})^2}$$

K nearest neighbor requires very less computational time because it does not need training initially and basically learns from data set in the times of making prediction. This algorithm can easily be implemented as it requires just two values: (i) The value of K and (ii) The value of distance function. However, it faces problems whenever the data set is large and does not work well whenever there are high dimensions in data [26].

G. AdaBoost (AdB)

AdaBoost is an algorithm that iteratively builds a classifier where it summons a base learner in each iteration that returns a classifier and the weight coefficient is also allocated to it [27]. A weighted 'vote' of the base classifiers will be the deciding factor for final classification. If the error of the base classifier is lesser, its weight in the final vote will be larger. Basically, Adaboost algorithm changes the distribution of the data with respect to the classification correction of the samples of the training set. Then it sends the updated weights from the modified data to the lower classifier and all the training classifiers are merged in the end. The function AdaBoost uses to calculate its final output is provided below:

$$C(x) = sign(\sum_{n=0}^{N} \alpha_n W_n(x))$$

where, $\alpha_n = 0.5\ln(\frac{1-\varepsilon_n}{\varepsilon_n})$, ε_t = total error

$W_n(x)$ = output from weak classifiers

Adaboost is less prone to overfitting but faces difficulties in case of noisy data and data with outliers.

H. XGBoost (XGB)

XGBoost (Extreme Gradient Boost), developed by Tianqi Chen in 2014 is a new arrival in the ML algorithm family. It is established based on gradient boosting principles. It encompasses both the domain of optimization and machine learning [28-29]. Mathematically in XGBoost algorithm, the objective function is:

$$O(t) = \sum_{i=0}^{n} Q(y_i, y'^{t-1} + f_t(x_i)) + K$$

Then normalization function:

$$Nor(f_t) = \kappa T + 0.5\lambda \sum_{i=0}^{T} W_j^2$$

Where κ = Controlling factor for the Leaf node number

T = Leaf node number

W_j = Weightage of the j leaf nodes

λ = Over-fitting controlling factor

K = Constant

XGBoost works well for both small data and big data but it faces difficulties if there are more categorical values in the dataset.

I. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a streamlined classifier to fit linear classifiers under convex loss functions [30]. It commences from an arbitrary point on a function and moves down its slope in steps until it grasps the lowest point of that function. For binary classification, the regularized training error is given by,

$$E(\omega, b) = \frac{1}{n}\sum_{k=1}^{n} L(y_k, f(x_k)) + \alpha R(\omega)$$

Where L =loss function that measures model misfit

R = regularization term.

Stochastic gradient descent is computationally faster and easily fits into the memory. Also, it converges faster for large datasets. However, it faces difficulties of convergence whenever the data is noisy.

J. Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA) is another version of a linear classifier that utilizes a quadratic decision surface to classify two or multiple classes [31]. It is different from linear discriminant analysis in that sense that there is no hard and fast rule like covariance of each of the attributes should be alike. Generally, the surface that separates the surface is of conical shape like parabola, hyperbola etc. [32-33]. Mathematically the discriminant function is:

$$Q_k(X) = (X-\mu_k)^T \sum_k (X-\mu_k) + \log\left|\sum_k\right| - 2\log\pi_k$$

The discriminant rule is:

$$Q_{\hat{k}}(X) = \min_{1\le k\le K} Q_k(X) \Leftrightarrow \max_{1\le k\le K} r(\frac{k}{x})$$

Where, k = class

μ_k = mean vector

π_k = prior probability

$\sum_k$ = covariance matrix

$r(\frac{k}{x})$ = posterior distribution

In QDA, the computational time is faster but it has a complex metrics calculation which introducs difficulties in practical understanding.

K. Ensemble Voting Classifiers (EVC)

In ensemble voting classifiers, many machine learning models are agglomerated for training and based on the majority of the voting, the highest probable class is chosen as output [34-36]. There are two types of voting classifiers:

i. Ensemble Voting Classifier-Hard (EVCH): In this case, each classifier votes for the output class and thus majority voted class is chosen. Mathematically,

$$H_v = \text{mod}\{C_1(x), C_2(x), C_3(x), ..., C_n(x)\}$$

Where, $C_n(x)$ = Output class from classifier

ii. Ensemble Voting Classifier-Soft (EVCS): Here, each classifier gives the probability about the output class and based on the classifiers' importance the probabilities are weighted and summed up and finally, the best probable class is chosen. From the mathematical point of view,

$$S_v = \arg\max_i \sum_{j=1}^{n} H_j P_{ij} \quad i \in \{0,1\}, [\, j = 1,2,3,.....,n]$$

Where, H_j= heap up to j[th] classifier

P_{ij}= probability from the classifiers

Ensemble methods provide a better understanding of the data and less bias and variance in case of prediction for the most of the case. Sometimes ensemble methods are difficult to interpret as there are many classifiers in it.

V. Results and Analysis

After studying different machine learning algorithms, simulations have been performed extensively and following that, results are obtained which can predict patients with heart diseases or without heart diseases. Different performance metrics like accuracy, precision, sensitivity, specificity, F1-score, and area under the receiver operating characteristic (ROC-AUC) curve are calculated and evaluated. The corresponding mathematical expressions for each metric are demonstrated below where TP, TN, FP, and FN denote *True Positive*, *True Negative*, *False Positive*, and *False Negative* respectively.

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

$$Precision = \frac{TP}{TP+FP}$$

$$Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP}$$

$$F1-Score = \frac{2\times Precision\times Sensitivity}{Precision+Sensitivity}$$

To acquire efficient prediction, hyperparameter optimization (HPO) of the machine learning model is obligatory. There are two popular ways of hyperparameter optimization: (a) *RandomizedsearchCV* and (b) *GridsearchCV*. In our proposed model, we have proposed *RandomizedsearchCV* for hyperparameter tuning as *GridsearchCV* is a little bit inefficient when there is a large amount of data [15]. To examine this phenomenon, hyperparameters were also tuned by *GridsearchCV* for comparative understanding. Hence, Table II illustrates the comparison of the computational time consumed by each of the algorithms deploying both grid and random search method for hyperparameter tuning. This analysis was performed on a computer with Intel Core i5 9[th] generation processor with 16 GB RAM.

TABLE II
Comparison of Computational Time

Name of the algorithm	Computational Time (Grid Search)	Computational Time (Random Search)
LR	0.9 sec	0.4 sec
DT	5.7 sec	1.7 sec
SVM	0.8 sec	0.2 sec
RF	3.6 min	27.2 sec
NBC	1.6 sec	0.1 sec
KNN	2.5 min	1.7 sec
AdB	2.3 min	1.1 sec
XGB	9.8 min	0.6 sec
SGD	0.9 sec	0.1 sec
QDA	0.6 sec	0.1 sec
EVCH	18.19 min	18.85 sec
EVCS	18.20 min	18.92 sec

After tuning the hyperparameters with our proposed method, the machine learning models were trained to keep the bias as less as possible to avoid overfitting. Subsequently, they were tested with cross-validation so that there is no chance of data leakage keeping the variance as less as possible. Confusion matrices for all the classifiers are presented in Table III. However, the values of all the

performance metrics for default hyperparameter (DHP) tuning, grid-search cross-validation (GSCV), and random search cross-validation (RSCV) are presented in Table IV, Table V, and Table VI respectively for all the ML algorithms. Moreover, Table VII depicts the best performing algorithms in terms of these three hyperparameter tuning method.

TABLE III
CONFUSION MATRICES FOR ALL THE CLASSIFIERS

Confusion Matrix (LR)		Predicted	
		True	False
Actual	True	27	5
	False	4	25

Confusion Matrix (DT)		Predicted	
		True	False
Actual	True	26	6
	False	6	23

Confusion Matrix (SVM)		Predicted	
		True	False
Actual	True	29	3
	False	3	26

Confusion Matrix (RF)		Predicted	
		True	False
Actual	True	28	4
	False	5	24

Confusion Matrix (NBC)		Predicted	
		True	False
Actual	True	28	4
	False	3	26

Confusion Matrix (KNN)		Predicted	
		True	False
Actual	True	29	3
	False	3	26

Confusion Matrix (AdB)		Predicted	
		True	False
Actual	True	30	2
	False	4	25

Confusion Matrix (XGB)		Predicted	
		True	False
Actual	True	28	4
	False	4	25

Confusion Matrix (SGD)		Predicted	
		True	False
Actual	True	29	3
	False	4	25

Confusion Matrix (QDA)		Predicted	
		True	False
Actual	True	24	8
	False	2	27

Confusion Matrix (EVCH)		Predicted	
		True	False
Actual	True	29	3
	False	2	27

Confusion Matrix (EVCS)		Predicted	
		True	False
Actual	True	29	3
	False	2	27

From Table VII, it is evident that both ensemble voting classifier-hard (EVCH) and ensemble voting classifier-soft (EVCS) display the highest accuracy of 92% with RSCV. Nonetheless, EVCS with GSCV also displays an accuracy of 92%. In terms of precision, Adaboost (AdB) triumphs all other algorithms for both grid and random search cross-validation. Furthermore, EVCH and EVCS perform better in case of sensitivity with a value of 0.936 each. However, AdaBoost (AdB) showcases the highest specificity of 0.926 among all other algorithms.

TABLE IV
PERFORMANCE METRICS OF DIFFERENT ML ALGORITHMS USING DEFAULT HYPERPARAMETER (DHP)

Name of the algorithm	Accuracy (%)	Precision	Sensitivity	Specificity	F1 Score	ROC-AUC
LR	76.44	0.800	0.750	0.790	0.77	0.813
DT	72.10	0.719	0.741	0.700	0.73	0.789
SVM	83.87	0.794	0.900	0.774	0.84	0.878
RF	80	0.844	0.794	0.814	0.80	0.851
NBC	78	0.781	0.806	0.767	0.78	0.825
KNN	81	0.765	0.897	0.750	0.81	0.876
AdB	84	0.875	0.824	0.852	0.84	0.901
XGB	82	0.867	0.788	0.857	0.82	0.883
SGD	83.56	0.844	0.844	0.827	0.83	0.898
QDA	82.26	0.800	0.827	0.812	0.81	0.872
EVCH	87	0.849	0.903	0.839	0.87	0.908
EVCS	86.77	0.848	0.903	0.833	0.88	0.907

TABLE V
PERFORMANCE METRICS OF DIFFERENT ML ALGORITHMS USING GRID SEARCH CROSS-VALIDATION (GSCV)

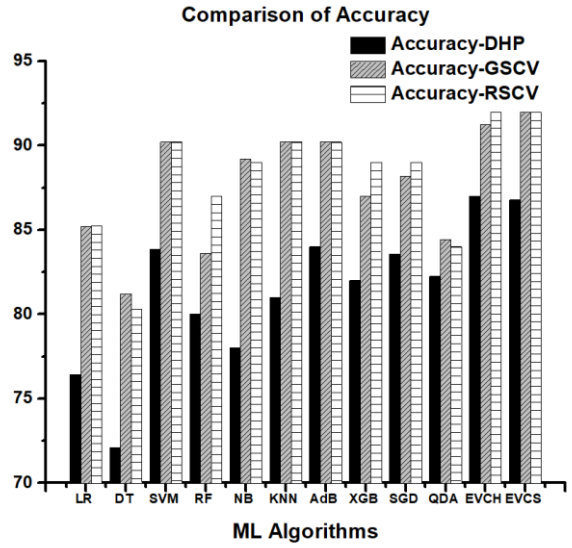
Name of the algorithm	Accuracy (%)	Precision	Sensitivity	Specificity	F1 Score	ROC-AUC
LR	85.20	0.841	0.870	0.833	0.850	0.916
DT	81.20	0.800	0.828	0.793	0.814	0.864
SVM	90.20	0.875	0.848	0.857	0.877	0.916
RF	83.61	0.871	0.814	0.857	0.844	0.892
NBC	89.20	0.876	0.904	0.869	0.893	0.917
KNN	90.20	0.906	0.906	0.896	0.900	0.910
AdB	90.20	0.938	0.882	0.926	0.901	0.912
XGB	87	0.871	0.872	0.866	0.869	0.909
SGD	88.20	0.902	0.873	0.891	0.897	0.917
QDA	84.43	0.753	0.925	0.770	0.872	0.883
EVCH	91.25	0.902	0.933	0.890	0.921	0.926
EVCS	92	0.906	0.936	0.900	0.920	0.927

TABLE VI
PERFORMANCE METRICS OF DIFFERENT ML ALGORITHMS USING RANDOMIZED SEARCH CROSS-VALIDATION (RSCV)

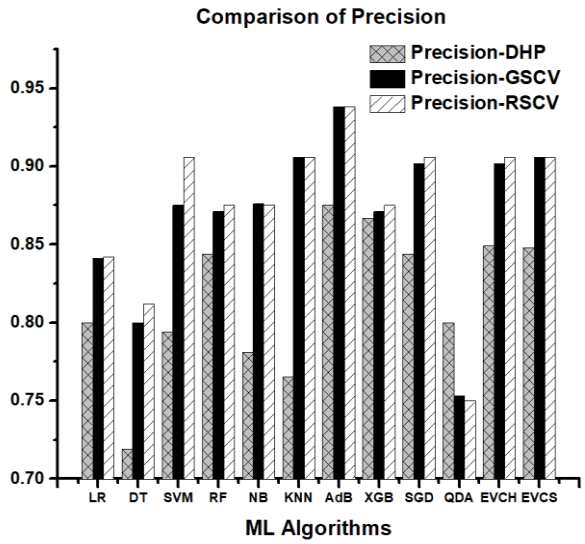
Name of the algorithm	Accuracy (%)	Precision	Sensitivity	Specificity	F1 Score	ROC-AUC
LR	85.25	0.842	0.871	0.833	0.85	0.916
DT	80.32	0.812	0.812	0.793	0.80	0.864
SVM	90.20	0.906	0.906	0.896	0.90	0.912
RF	87	0.875	0.848	0.857	0.87	0.916
NBC	89	0.875	0.903	0.867	0.89	0.916
KNN	90.20	0.906	0.906	0.896	0.90	0.912
AdB	90.20	0.938	0.882	0.926	0.90	0.912
XGB	89	0.875	0.875	0.862	0.88	0.912
SGD	89	0.906	0.879	0.893	0.90	0.919
QDA	84	0.750	0.923	0.771	0.87	0.882
EVCH	92	0.906	0.936	0.900	0.92	0.927
EVCS	92	0.906	0.936	0.900	0.92	0.927

TABLE VII
BEST PERFORMANCE METRICS FOR DIFFERENT HYPERPARAMETER OPTIMIZATION METHOD

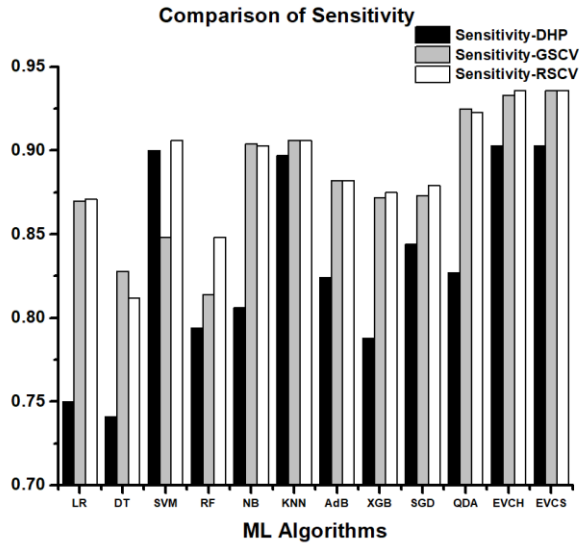
Hyper Parameter Tuning	Performance Metrics					
	Accuracy (%)	Precision	Sensitivity	Specificity	F1 Score	ROC-AUC
Default Hyper Parameter (DHP)	EVCH (87)	AdB (0.875)	EVCH & EVCS (0.903)	XGB (0.857)	EVCS (0.88)	EVCH (0.908)
Grid Search Cross-Validation (GSCV)	EVCS (92)	AdB (0.938)	EVCS (0.936)	AdB (0.926)	EVCH (0.921)	EVCS (0.927)
Random Search Cross-Validation (RSCV)	EVCH & EVCS (92)	AdB (0.938)	EVCH & EVCS (0.936)	AdB (0.926)	EVCH & EVCS (0.92)	EVCH & EVCS (0.927)



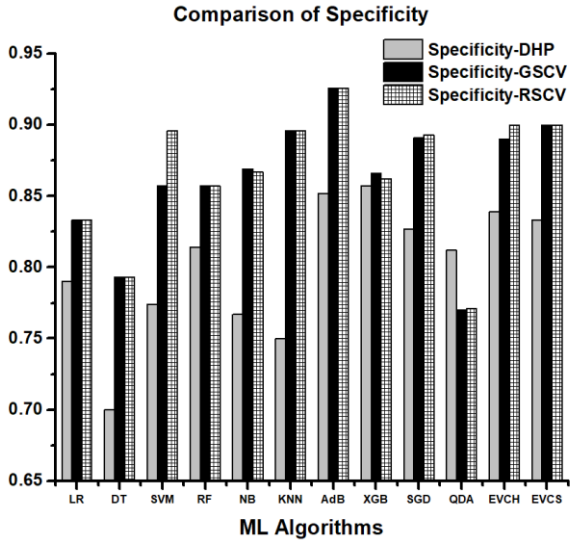
(a)



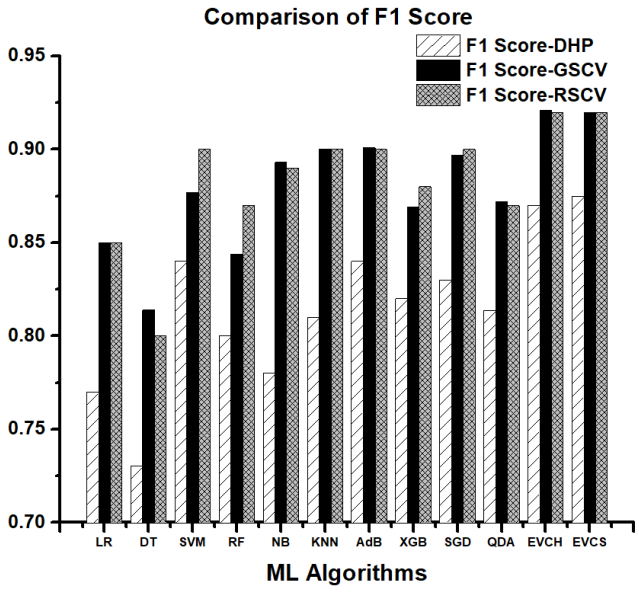
(b)



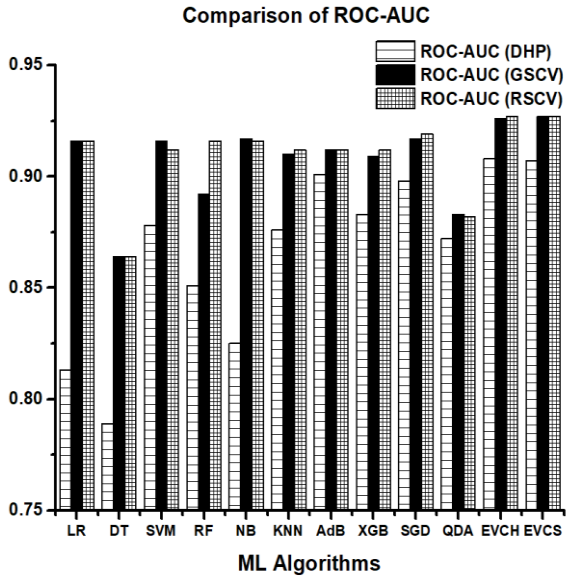
(c)



(d)



(e)



(f)

Fig. 3 Comparative analysis of the ML Algorithms based on for (a) Accuracy (b) Precision (c) Sensitivity (d) Specificity (e) F1 score (f) ROC-AUC

TABLE VIII
COMPARISON OF PERFORMANCE OF OUR PROPOSED MODEL WITH OTHER REFERENCES

Work	Model Used	Type of data used	Measures Reported	Best Performance Reported (%)
[5]	J48 Tree Technique	Balanced	Accuracy	56.76
[6]	K-Nearest Neighbor (KNN)	Balanced	Accuracy	78
[7]	Random Forest (RF) with 199 Samples	Balanced	Accuracy Precision Sensitivity	86.94 86.42 100
[8]	Logistic Regression (LR)	Balanced	Accuracy	86.51
[9]	Ensemble Voting Classifier - Hard (EVCH)	Balanced	Accuracy	90
[10]	Logistic Regression (LR)	Balanced	Accuracy	87.1
[11]	Random Forest (RF)	Balanced	Accuracy Sensitivity Specificity	87.5 86.66 88.37
[12]	Random Forest and Linear method	Balanced	Accuracy	88.4
Proposed	Ensemble Voting Classifier – Hard and Soft (EVCS)	Balanced	Accuracy Precision Sensitivity	92 90.6 93.6

After calculating the F1 score and area under the ROC curve, it is observed that EVCH and EVCS perform a lot better than other machine learning models. The comparative graphical representation for accuracy, precision, sensitivity, specificity, F1 Score and ROC-AUC are reported in Fig. 3 (a), 3 (b), 3 (c), 3 (d), 3 (e), and 3 (f) respectively. Moreover, Table VII is illustrated graphically through Fig. 4 where the comparative analysis among all the best performing models is portrayed. At the end of our analysis, we have presented the comparative analysis in Table VIII where we have shown our proposed algorithms (EVCH and EVCS) show more accuracy (92%) than other algorithms. Moreover, these two methods have also displayed better results in terms of other performance metrics. Therefore, these classifiers can be implemented practically to predict patients with cardiovascular disease to ease the diagnosis process and reduce human-made error.

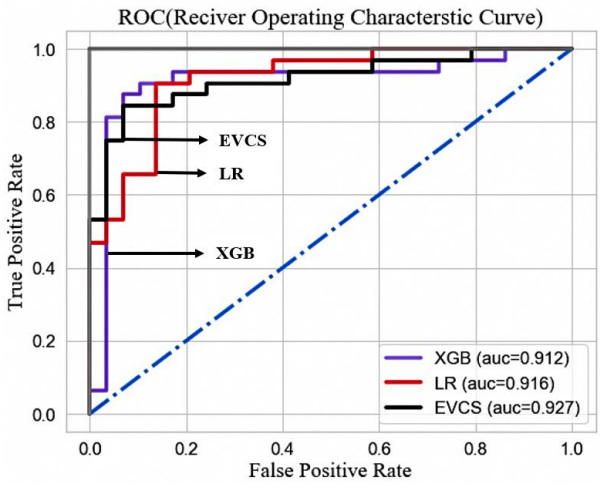


Fig. 5 ROC curve for XGB, LR and EVCS model

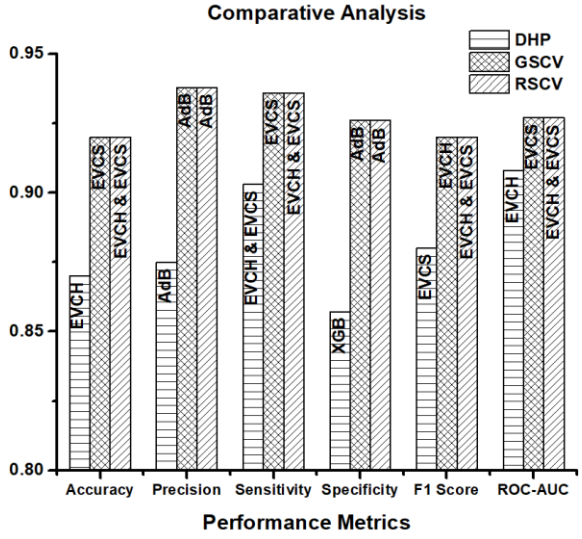


Fig. 4 Overall comparative analysis

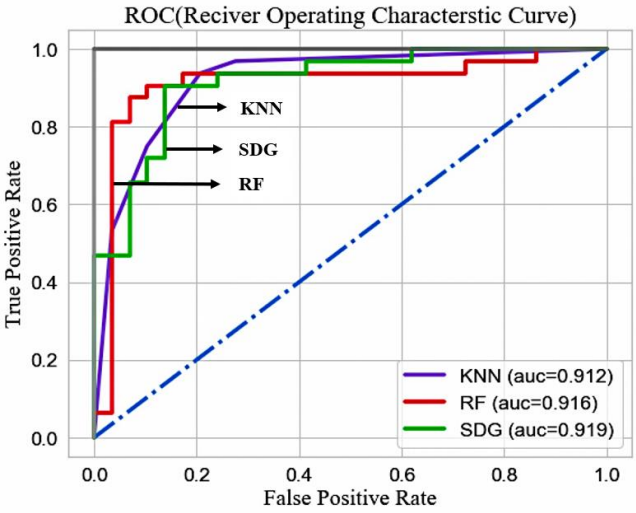


Fig. 6 ROC curve for KNN, RF and SDG model

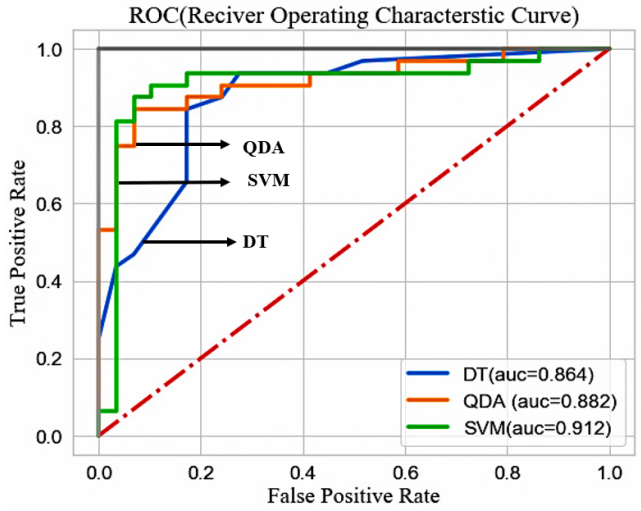


Fig. 7 ROC curve for DT, QDA and SVM model

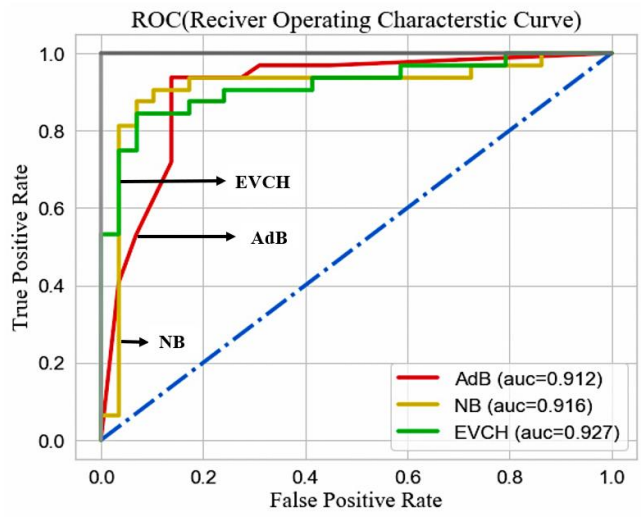


Fig. 8 ROC curve for AdB, NB and EVCH model

In addition, ROC (Receiving Operator Characteristics) curves have been plotted for further investigation for each machine learning model and have been displayed in Fig. 5, Fig. 6, Fig. 7 and Fig. 8. These curves provide the visualization of the performances of classifiers and the tradeoff between true positive rate and false positive rate in all possible classification threshold.The area under the curve (AUC) of ROC signifies the capability of distinguishing the classes of the machine learning model where the value ranges from zero to one [40]. The more it is near to one, the more it is capable to make the separation of the classes.

VI. Conclusion

Being an essential organ of the human body, heart disease is a serious concern among mass people. Hence, applying machine learning algorithms on data regarding cardiovascular disease will assist in predicting this fatal disease and saving a lot of human lives. Moreover, early detection of any kind of abnormalities in heart condition can also help researchers to study more effectively which will carry a positive impact in healthcare in the long run. In this paper, we have used several machine learning algorithms in predicting heart disease and a detailed comparative analysis is brought into the display where the highest accuracy (92%) is obtained in hard and soft voting ensemble classifiers. Moreover, Adaboost algorithm appears to be another promising approach among all other models where it shows highest precision of 0.938 and specificity of 0.926. Furthermore, the computational time for RSCV is lesser than GSCV which proves that RSCV method is capable of providing faster and more accurate results than other tuning methods. Thus these approaches can play a noteworthy role in combatting heart disease and assisting healthcare professionals in the process of diagnosis and ensuring proper treatment for the patients.

References

[1] Jagannathan, Ram, Shivani A. Patel, Mohammed K. Ali, and KM Venkat Narayan, "Global updates on cardiovascular disease mortality trends and attribution of traditional risk factors," Current diabetes reports, vol. 19, no. 7, pp 1-12, 2019

[2] Krittanawong, Chayakrit, HongJu Zhang, Zhen Wang, Mehmet Aydar, and Takeshi Kitai, "Artificial intelligence in precision cardiovascular medicine," *Journal of the American College of Cardiology*, vol. 69, no. 21, pp 2657-2664, 2017

[3] Rajkomar, Alvin, Jeffrey Dean, and Isaac Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp 1347-1358, 2019

[4] Yusuf, S., Joseph, P., Rangarajan, S., Islam, S., Mente, A., Hystad, P., Brauer, M., Kutty, V.R., Gupta, R., Wielgosz, A. and AlHabib, K.F., "Modifiable risk factors, cardiovascular disease, and mortality in 155 722 individuals from 21 high-income, middle-income, and low-income countries (PURE): a prospective cohort study," The Lancet, vol. 395, no. 10226, pp 795-808, 2020

[5] Patel, Jaymin, Dr TejalUpadhyay, and Samir Patel, "Heart disease prediction using machine learning and data mining technique," *Heart Disease*, vol. 7, no.1, pp 129-137, 2015

[6] Singh, Archana, and Rakesh Kumar, "Heart Disease Prediction Using Machine Learning Algorithms," 2020 *International Conference on Electrical and Electronics Engineering* (ICE3). IEEE, 2020

[7] Alim, Muhammad Affan, Shamsheela Habib, Yumna Farooq, and Abdul Rafay, "Robust Heart Disease Prediction: A Novel Approach based on Significant Feature and Ensemble learning Model," In 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp. 1-5. IEEE, 2020

[8] Kannan, R., and V. Vasanthi, "Machine learning algorithms with ROC curve for predicting and diagnosing the heart disease," *Soft Computing and Medical Bioinformatics*, Springer, Singapore, 2019, pp 63-72

[9] Atallah, Rahma, and Amjed Al-Mousa, "Heart Disease Detection Using Machine Learning Majority Voting Ensemble Method," 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), IEEE, 2019

[10] Kohli, Pahulpreet Singh, and Shriya Arora, "Application of machine learning in disease prediction," In 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1-4. IEEE, 2018

[11] Ed-Daoudy, Abderrahmane, and Khalil Maalmi, "Real-time machine learning for early detection of heart disease using big data approach," In 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), pp. 1-5, IEEE, 2019

[12] Mohan, Senthilkumar, Chandrasegar Thirumalai, and Gautam Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," IEEE Access, vol. 7, pp 81542-81554, 2019

[13] Dua, D., and Graff, C. (2017). UCI Machine Learning Repository. Retrieved from https://archive.ics.uci.edu/ml/datasets/heart+Disease

[14] Zheng, Alice, and Amanda Casari. Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media, Inc.", 2018.

[15] Géron, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019.

[16] Khan, Muhammad Umar, Sumair Aziz, Muhammad Bilal, and Muhammad Bilal Aamir, "Classification of EMG Signals for Assessment of Neuromuscular Disorder using Empirical Mode Decomposition and Logistic Regression," In 2019 International Conference on Applied and Engineering Mathematics (ICAEM), pp. 237-243. IEEE, 2019

[17] Molnar, Christoph, "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2020.
[18] Ghiasi, Mohammad M., Sohrab Zendehboudi, and Ali Asghar Mohsenipour, "Decision tree-based diagnosis of coronary artery disease: CART model," Computer Methods and Programs in Biomedicine, vol. 192, pp 105400, 2020
[19] Qinghe Zheng, Xinyu Tian, Mingqiang Yang, and Huake Su, "The Email Author Identification System Based on Support Vector Machine (SVM) and Analytic Hierarchy Process (AHP)," IAENG International Journal of Computer Science, vol. 46, no. 2, pp 178-191, 2019
[20] Ingrid Nurtanio, Eha Renwi Astuti, I Ketut Eddy Purnama, Mochamad Hariadi, and Mauridhi Hery Purnomo, "Classifying cyst and tumor lesion using support vector machine based on dental panoramic images texture features," IAENG International Journal of Computer Science, vol. 40, no. 1, pp 29-37, 2013
[21] Nan Jiang, Fei Fu, Hua Zuo, Xiuping Zheng, and Qinghe Zheng, "A Municipal PM2.5 Forecasting Method Based on Random Forest and WRF Model," Engineering Letters, vol. 28, no. 2, pp312-321, 2020
[22] Liu, Yanli, Yourong Wang, and Jian Zhang, "New machine learning algorithm: Random forest," International Conference on Information Computing and Applications, Springer, Berlin, Heidelberg, pp 246-252, 2012
[23] Bin Ning, Wu Junwei, and Hu Feng, "Spam Message Classification Based on the Naïve Bayes Classification Algorithm," IAENG International Journal of Computer Science, vol. 46, no. 1, pp 46-53, 2019
[24] Cong-Cuong Le, P. W. C. Prasad, Abeer Alsadoon, L. Pham, and A. Elchouemi, "Text Classification: Naive Bayes Classifier with Sentiment Lexicon," IAENG International Journal of Computer Science, vol. 46, no. 2, pp 141-148, 2019
[25] Yanying Li, Youlong Yang, Jinxing Che, and Long Zhang, "Predicting the Number of Nearest Neighbor for KNN Classifier," IAENG International Journal of Computer Science, vol. 46, no. 4, pp662-669, 2019
[26] Md. Yasir Arafat, Sabera Hoque, Shuxing Xu, and Dewan Md Farid, "Machine learning for Mining Imbalanced Data," IAENG International Journal of Computer Science 46, no. 2, pp 332-348, 2019
[27] X. Shu and P. Wang, "An Improved Adaboost Algorithm Based on Uncertain Functions," *2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, Wuhan, pp 136-139, 2015
[28] Li, Shenglong, and Xiaojing Zhang, "Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm," Neural Computing and Applications, pp 1-9, 2019
[29] Lai, Cing-Han, Chao-Tung Yang, Endah Kristiani, Jung-Chun Liu, and Yu-Wei Chan, "Using XGBoost for Cyberattack Detection and Analysis in a Network Log System with ELK Stack," In International Conference on Frontier Computing, pp. 302-311. Springer, Singapore, 2019
[30] Deepa, N., B. Prabadevi, Praveen Kumar Maddikunta, Thippa Reddy Gadekallu, Thar Baker, M. Ajmal Khan, and Usman Tariq, "An AI-based intelligent system for healthcare analysis using Ridge-Adaline Stochastic Gradient Descent Classifier," Journal of Supercomputing, pp 1-20, 2020
[31] Zhuang, Chen, Hongbo Zhao, Chao Sun, and Wenquan Feng, "Detection and Classification of GNSS Signal Distortions Based on Quadratic Discriminant Analysis," IEEE Access, vol. 8, pp 25221-25236, 2020
[32] S. Bhattacharyya, A. Khasnobish, S. Chatterjee, A. Konar and D. N. Tibarewala, "Performance analysis of LDA, QDA and KNN algorithms in left-right limb movement classification from EEG data," 2010 International Conference on Systems in Medicine and Biology, Kharagpur, 2010, pp 126-131
[33] Starzacher, Andreas, and Bernhard Rinner, "Evaluating KNN, LDA and QDA classification for embedded online feature fusion," In 2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp 85-90. IEEE, 2008.
[34] Zhou, Yuyang, Guang Cheng, Shanqing Jiang, and Mian Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks (2020): 107247.
[35] Mahabub, Atik, "A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers," SN Applied Sciences, vol. 2, no. 4, pp 1-9, 2020
[36] Leon, Florin, Sabina-Adriana Floria, and Costin Bădică, "Evaluating the effect of voting methods on ensemble-based classification," In 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1-6. IEEE, 2017.