IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 December 2023, accepted 28 December 2023, date of publication 8 January 2024,
date of current version 18 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350996

RESEARCH ARTICLE

An Efficient Computational Risk Prediction Model of Heart Diseases Based on Dual-Stage Stacked Machine Learning Approaches

SUBHASH MONDAL[1,2], (Member, IEEE), RANJAN MAITY[1], (Senior Member, IEEE), YACHANG OMO[3], SOUMADIP GHOSH[4], (Member, IEEE), AND AMITAVA NAG[1], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Central Institute of Technology Kokrajhar, Kokrajhar, Assam 783370, India
[2]Department of Computer Science and Engineering (AI & ML), Dayananda Sagar University, Bengaluru, Karnataka 560078, India
[3]Department of Civil Engineering, Central Institute of Technology Kokrajhar, Kokrajhar, Assam 783370, India
[4]Department of Computer Science and Engineering, Future Institute of Technology, Kolkata, West Bengal 700150, India

Corresponding author: Amitava Nag (amitava.nag@cit.ac.in)

ABSTRACT Cardiovascular diseases (CVDs) continue to be a prominent cause of global mortality, necessitating the development of effective risk prediction models to combat the rise in heart disease (HD) mortality rates. This work presents a novel dual-stage stacked machine learning (ML) based computational risk prediction model for cardiac disorders. Leveraging a dataset that includes eleven significant characteristics from 1190 patients from five distinct sources, five ML classifiers are utilized to create the initial prediction model. To ensure robustness and generalizability, the classifiers are cross-validated ten times. The model performance is optimized by employing two hyperparameter tuning approaches: RandomizedSearchCV and GridSearchCV. These methods aim to find the optimal estimator values. The highest-performing models, specifically Random Forest, Extreme Gradient Boost, and Decision Tree undergo additional refinement using a stacking ensemble technique. The stacking model, which leverages the capabilities of the three models, attains a remarkable accuracy rate of 96%, a recall value of 0.98, and a ROC-AUC score of 0.96. Notably, the rate of false-negative results is below 1%, demonstrating a high level of accuracy and a non-overfitted model. To evaluate the model's stability and repeatability, a comparable dataset consisting of 1000 occurrences is employed. The model consistently achieves an accuracy of 96.88% under identical experimental settings. This highlights the strength and dependability of the suggested computer model for predicting the risk of cardiac illnesses. The outcomes indicate that employing this two-step stacking ML method shows potential for prompt and precise diagnosis, hence aiding the worldwide endeavor to decrease fatalities caused by heart disease.

INDEX TERMS Cardiovascular disease (CVD), extreme gradient boost (XGB), hyper-parameter tuning, heart disease, random forest classifier, stacking ensemble technique.

I. INTRODUCTION

The major essential organ – the heart, whose main function is to move the blood throughout our body, but having a threat to it is a matter of concern that causes several health issues. Heart disease (HD) is contributing the leading cause of death due to sudden strokes and heart attacks in today's world. Every year, 17.9 million people die from some causes related to CVD, and a total of 32% of all deaths are estimated globally [1]. The most common type of HD that contributes to major deaths worldwide is coronary heart disease (CHD). There are various forms of heart disease, namely problems related to heart rhythms, valves, heart muscles, heart infection, blood vessel disease, and congenital heart defects.

The associate editor coordinating the review of this manuscript and approving it for publication was Nuno M. Garcia.

Because of these different forms, several symptoms can be observed: dizziness, fainting, slow heartbeat, racing heartbeat, shortness of breath, etc. While heart disease can be deadly, it can also be prevented by adopting a healthy lifestyle like regular meditation, regular exercise, a nutritious diet, etc.

In identifying modern healthcare-related diseases, machine learning (ML) and deep learning (DL) play an important role, including properly detecting and classifying these kinds of ailments. The cost of curing heart disease is very high worldwide and may exceed $1 trillion by 2035 [2]. In the United States, about $229 billion was spent annually between 2017 and 2018 [3]. Hence, the minimization of the treatment cost for HD has become extremely important. For this purpose, many researchers have applied classical ML approaches [4], [5], and [6] based on characteristic features to determine if a patient can suffer from HD. The dataset [7] used in this research study contains 11 important features, one target outcome indicates whether patients had a risk of HD or not, and 1190 instances with which we trained five well-known classical ML models, namely Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), and Extreme Gradient Boost (XGB), along with an ensemble-based stacking technique to gain higher results in performance metrics.

The conventional ML model is considered for detecting HD and provides a prediction model that is more stable and accurate. We used k-fold cross-validation (CV) to properly split the acquired dataset into training, testing, and validation. Next, we trained the above-mentioned standalone models with and without hyper-parameter tuning by the RandomizedSearchCV (RS-CV) and GridSearchCV (GS-CV) and selected the best three models, namely DT, RF, and XGB, corresponding to their RS-CV technique predicted accuracy, and performed the ensemble stacking operation with the hold-out testing set, and achieved an accuracy (Acc), Precision (Pr), Recall (Re), F1-score (Fs), ROC-AUC (Ra) score, and Cohen-cappa score (Cs) of 96%, 0.96, 0.98, 0.96, 0.96 and 0.92 respectively. To validate the HD risk prediction model, a similar dataset was collected from the Mendeley data portal with 1000 instances and 12 features replicated the above procedure on it, and was used to check the stability and robustness of the deployed model with the accuracy of 96.88% and a lower standard deviation.

The latter sections of this study give a detailed description of all the ML algorithms mentioned earlier, along with their descriptive result analyses. Then, the related study is summarized in Section II, the preliminary study is discussed in Section III, and Section IV illustrates the details of the proposed methodology. Finally, the experimental results and comparison discussion are provided in Section V, and the research study is concluded in Section VI.

II. RELATED STUDY

The related kinds of literature are discussed in this section, considering ML-based prediction or detection models of HD. The entire literature is subdivided into a few sections related to the approaches, like ensemble-based ML prediction models, standalone different ML models, and reduced feature-based HD prediction and classification models.

This section mainly focuses on ensemble-based approaches like the one proposed by the authors in [8] to predict the risk of HD on the Cleveland dataset (CD). A few base models were used for initial prediction, like Naive Bayes (NB), RF, Bayes Net (BN), C4.5, Multi-Layer Perception (MLP), and PART (Projective Adaptive Resonance Theory). Four ensemble approaches were used for the final prediction, including bagging, boosting, and stacking, and results were claimed by the BN, RF, NB, and MLP based on voting. Features were considered in the selected way for each model evaluation. The claimed results were 85.48%, with increased accuracy on chosen features. In [9], the authors deployed the HD prediction model on the famous CD using an AI approach based on the digital history of patients' records. The algorithms used were DT, Artificial Neural Network (ANN), Rough Set (RS), SVM, and NB. The ten-fold CV was considered to calculate the mean accuracy with the RS, which gave an accuracy of 88.1%. The authors then proposed an ensemble method wherein they combined the top 3 performing algorithms, namely RS, NN, and NB, to improve the performance to 89%, outperforming all other classifiers. The authors in [10] chose the CD for the HD prediction model, which contained 76 attributes that were reduced to 13 significant features and one target attribute. Four different algorithms were applied: Stochastic Gradient Descent (SGD), K-Nearest Neighbor (KNN), LR, and RF. Finally, an ensemble method was proposed by combining all four models using hard voting. Amongst the four algorithms, SGD gave the highest accuracy of 88% after hyperparameter tuning, and the model concludes with the best accuracy of 90%.

Also, in [11], the authors presented the classical AI approach to predict heart disease early using three ML algorithms: RF, DT, and a proposed hybrid model. The CD was used to deploy the model, with 70% of the dataset taken for training and the rest, 30% for testing. The proposed hybrid model combined with RF and DT gave an Acc of 88%. In [12], they explored the use of ML and DL classifiers to provide an early detection system for HD. The CD was taken, and LR, SVM, KNN, NB, ANN, DT, the Back Propagation Neural Network (BPNN), and ensemble-based stacking, boosting, and bagging were applied to increase the models' Acc. SVM gave an accuracy of 86% using all features and an accuracy of 88% using selected features. The proposed ensemble algorithm reaches an Acc of 92.30% and a BPNN of 93%. They considered in [13] the various ML techniques for large-scale heart disease prediction based on big data analysis. The widely used Cleveland dataset was used to split in a 0.90:0.10 ratio. Different algorithms were applied: DT, RF, LR KNN, SVM, AdaBoost (AB), Gradient Boost (GB), DT, HRFLM, and HGBDTLR. The proposed ensemble algorithm HGBDTLR gave the best accuracy of 91.8%. In [14], the authors proposed an ensemble technique based on the Classification and Regression Tree (CART) for

HD risk prediction. Two datasets were taken and split into random numbers based on mean values. The different CART models faded into a homogenous ensemble classifier, and accuracy was weighted. Their experimental results showed an accuracy of 93% on the CD. They compared their results with the performances of the SVM, RF, LR, Linear Discriminant Analysis (LDA), GB, and KNN models.

Moreover, the article [15] used the CD to predict heart diseases using various ML algorithms, namely KNN, RF, NB, SVM, XGB, and LR, which were applied, and an observatory analysis was drawn. Several evaluation metrics were taken for the performance acumen evaluation of every model. Among these, a proposed ensemble method based on voting and combining three algorithms, namely KNN, XGB, and LR, gave the best accuracy of 92%. In [4], the author uses three datasets to train a custom-based ensemble classifier consisting of NB, RF, SVM, and XGB with parameter tuning. The data has been trained on a 0.60:0.40 ratio of training to testing, and underfitting is also taken care of by using such a division. The prediction made by the classifier is put to the test on one of the datasets provided with the help of a voting classifier. The highest accuracy achieved by the model is 96.75% with the Mendeley dataset using the proposed model, while the other two datasets also give results of 88.24% and 93.39%, respectively. In [5], the authors extracted features from recording the magnetocardiography (MCG) signal, used 164 features categorized into three subgroups, and developed a fast model to detect ischemic heart disease. The four ML models were used to compare the results, like KNN, DT, SVM, and XGB; after that, finally, the ensemble method was applied, combining two SVM and XGB models to get the best results with 94.03% accuracy and an AUC value of 0.98.

The subsection below considered different ML models for HD prediction compared their observable experimental outcomes and concluded their research with a particular ML model. Like in [16], the authors used an open-source dataset for predicting heart diseases. They incorporated five different ML algorithms for training the dataset: KNN, DT, RF, NB, and SVM. Metrics like Acc, specificity, and Re were used to conclude the performance acumen of the model. Amongst these, KNN had the best accuracy, at 85%. The authors in [17] deployed five ML algorithms to classify CVD: SVM, NB, KNN, DT, and LR. An open-source dataset with 77,000 instances was used to make the prediction models. Performance metrics like Re, Pr, Acc, and Fs were measured to analyze the models. LR and SVM observed efficient algorithms for diagnosing CVD anomalies with accuracies of 72.66% and 72.36%, respectively. The article [18] explored using ML models to predict CVD early at low and affordable costs. Various algorithms like NB, DT, RF, KNN, SVM, and LR were incorporated to predict the CVD. The CD was taken for model prediction, concluded with measuring parameters to acumen, and the RF had the highest Acc of 83.52%. In [19], authors had laid down an approach to predict heart diseases with the greatest accuracy and consuming less time for computation so that diagnosis of the disease is at the earliest. The popular CD was used to train and test models like LR, KNN, DT, and RF. Amongst these, RF gave the best accuracy of 88.16%, followed by LR with an accuracy of 84.21%. The authors in the article [20] considered a heart disease dataset to predict and deploy six ML models: LR, AB, KNN, CART, XGB, and RF. After data processing, ten-fold cross-validation was applied with hyperparameter tuning of the above model's parameters. Their claimed results were accurate, and the ROC-AUC values are 84.8% and 0.917 for the RF model. In [21], the authors used different supervised ML models to predict heart diseases. The dataset used in their study contains 14 attributes and 303 instances in total. Algorithms like LR, KNN, and SVM were incorporated, and these models' efficacies were judged on metrics like Re, Pr, specificity, Fs, and Ra. Overall, LR performed well, with an accuracy of 86%, a precision of 0.83, and an AUC score of 0.87. The authors in [22] presented an approach for the future possibility of HD using ML models like RF, NB, SVM, Hoeffding DT, and Logistic Model Tree (LMT). The CD was incorporated for training the models. RF and Gaussian-NB performed considerably well, with accuracies of 95.08% and 93.44%, respectively. In [6], the CART algorithm predicts the HD and extracts the decision rule to find the relationship between the target class and the inputs. It provides an influencing feature-based ML model with an accuracy of 87.25% over the dataset of 1190 instances and 11 features.

Some related works have been observed on the feature selection-based approaches; the authors in [23] presented the KNN model as the base method for predicting cardiovascular diseases. The famous CD dataset was incorporated for this work. However, KNN alone could not achieve significant results, so the authors utilized techniques like standardization, feature selection, and cross-validation to increase the method's accuracy. Fourteen important features out of 75 total attributes were used in the process. The ten-fold CV was used to measure the mean Acc, 89.23%. In [24], the two ML models, LR and ANN, were considered for predicting HD using the CD. Eleven significant features out of 75 were considered for the process. Variable hidden layers and other activation functions were applied for further tuning these models. ANN with one hidden layer and the sigmoid function gave an Acc of 92.31% and an LR of 90.11%. The authors in [25] explored the data mining approach for identifying meaningful patterns from HD datasets. For this purpose, they used the CD, wherein six ML algorithms were applied, namely NB, DT, LR, RF, KNN, and SVM, after applying Information Gain, Chi-Square, Gain Ratio, RELIEF, and One-R. Amongst these, SVM gave the best accuracy of 83.41% after applying the feature selection method. In [26], the authors used AI in the healthcare sector to predict heart diseases and other locomotive disorders. They used the CD, which consisted of 76 features and 303 instances, among which 14 important attributes were chosen for the prediction model. A ten-fold CV was used to measure the mean accuracy

of the deployed models. In addition, performance indicators like Acc, Re, Pr, Fs, and Ra were calculated to measure results acumen. The SVM, LR, NB, DT, KNN, and RF models were deployed, and, among them, KNN gave the highest accuracy of 94.1% using seven significant features.

The above-related literature study is considered mainly for predicting HD using conventional ML models with the specific motivation of resource-constrained devices' perspective use of less memory and a significantly shorter response time. ML models are less complex and more effective for the scenario of real-life human disease detection compared to other approaches that exist in state-of-the-art (SOTA) models based on deep learning methods by using the ECG signal and applying CNN models [27], [28], genetic algorithms [29], fuzzy-based [30], etc.

The existing literature and most relevant articles are summarized in Table 1. All the considered deployed models help decide on model selection for further deployment in this research to find out the research gap and improve the existing SOTA models.

TABLE 1. Summarization of the ML model used in the literature.

ML Model Name	# References used by literature
RF	[8], [16], [10], [11], [18], [19], [20], [25], [15], [22], [13], [14], [26], [4]
DT	[9], [16], [10], [11], [12], [17], [18], [19], [25], [22], [13], [26], [5]
SVM	[9], [16], [12], [17], [18], [25], [15], [21], [22], [13], [14], [26], [5], [4]
LR	[10], [12], [17], [18], [24], [19], [20], [25], [15], [21], [22], [13], [14], [26]
KNN	[16], [10], [12], [17], [18], [23], [19], [20], [25], [15], [21], [13], [14], [26], [5]
XGB	[20], [15], [5], [4]
NB	[8], [9], [16], [12], [17], [18], [25], [15], [22], [4]
ANN	[9] , [12], [24]
CART	[20], [14], [6]
BN	[8], [26]
AB	[20] , [13]

The above-related articles, particularly those on selected ML models for HD risk prediction, have noted the number of occurrences of each used model in the entire literature for the model selection.

Table 1 helps to select the appropriate ML model for further deployment. The RF, DT, LR, and SVM classifiers are selected for this study due to their frequent usage and corresponding moderate-to-high-performance results ranging from 83% to 95% accuracy, as observed by many researchers in the literature section. The KNN and NB declared results are not so good compared to others, implying they were not considered in this study. The high accuracy obtained by the articles in the literature on the XGB boosting-based model is deemed to produce better prediction results. So finally, five ML models were selected to get a more accurate model prediction result for HD detection.

In summary, two key points are drawn based on the above review of the existing literature. First, no existing work has addressed overfitting or underfitting scenarios of ML models, and second hyper-parameter tuning is necessary to deploy prediction models based on the best-performing boosting ML classifier. These two factors have an impact on how well a classifier performs.

The prime contribution of this research study is formulated and noted down as follows:

- The most frequently used ML models are selected for initial deployment, with boosting-based meta-ensemble stacking proposed at the final stage for precise HD illness prediction.
- The k-fold CV is used to eliminate the overfitting problem on low-data instances in this case.
- Two hyperparameter tuning techniques, RandomizedSearchCV and GridSearchCV, are added contributions to the search for the best hyperparameter values of each model and are executed to get the best possible results.
- A hyperparameter-tuned model-based two-stage stacking ensemble prediction approach is introduced in this research study.

III. PRELIMINARIES AND BACKGROUND STUDY

This section demonstrates a brief overview of the few ML models that are considered for training and testing to predict the risk of HD. The performance matrix is elaborated to evaluate the model's performance using the parameters mentioned in the earlier section.

A. LOGISTIC REGRESSION (LR)

This ML algorithm is used for grouping data points into certain labels. It is used to predict new input datasets and classify them based on the labels it has been trained upon. In this case, the output achieved is within the range of 0 and 1, as the classes mentioned are in binary.

$$h_w(X) = \frac{1}{1 + e^{-wX}} \tag{1}$$

As the formula (1) suggests, parameter X is the input dataset and w is to be trained as a parameter while also optimizing the output. Optimizing the classification task properly requires a loss function to refine the prediction and apply the log-likelihood function defined below [31].

$$J(w) = -\frac{1}{m}\sum_{m}^{i=1} (y^i log\,(p^i) + (1 - y^i) log(1 - p^i)) \tag{2}$$

Here in (2), m is the quantity of samples for training the model. Whereas y^i is the class of the ith sample and p^i is the predicted value in the ith sample. $J(w)$ is the quadratic cost function, smaller values indicate the model fitted better with the dataset. The gradient descent function is used to optimize the loss function in the global minimum.

B. SUPPORT VECTOR MACHINE (SVM)

SVM is a very popular supervised learning algorithm used for regression and classification of data while also developing patterns. The SVM models use a hyperplane for separating the data values of two classes, and for multiple labels, numerous hyperplanes are used. This hyperplane is placed at the maximum distance between the two or multiple data points of different classes. It is an algorithm that is used for a small to medium-sized dataset. In this case, linear SVM is used as parameter tuning because the dataset used is categorical in nature. As we know, the equation of a line is $y = ax + b$. The equation of a hyperplane is defined in (3).

$$w^t x = y - ax - b \; and \; w^t X + b = 0 \tag{3}$$

Here, w indicates the weight vector, and b is the bias. To optimize the results, we need to maximize the difference between the two data points and the hyperplane. The loss function that helps in doing this is referred to as ''hinge loss'' and is given below in (4) [31].

$$J(w) = \sum_{i=1} max\left(0, 1 - y^i\left[w^t x_i + b\right]\right) + \lambda \, ||w||_2^2$$
$$\lambda = \frac{1}{C} \tag{4}$$

C = Regularization Coefficient
$\lambda \, ||w||_2^2$ = Regularization
$max\left(0, 1 - y_i\left[w^t x_i + b\right]\right)$ = Loss Function

C. DECISION TREE (DT)

It is a supervised ML algorithm for solving problems involving regression and classification. It is a tree-like model that uses the decision rule to form structures for training on a dataset and predict the target class. A single root or parent node is corresponding to an attribute, and there could be multiple branch nodes for decisions taken using the other attributes' values. The algorithm uses the sum of the product architecture and a top-down, greedy search approach with selections based on the type of dependent variable. Some algorithms necessary for decision-making are ID3, C4.5, CART, etc. ID3 (Iterative Dichotomiser-3) is the most popular and essential algorithm for generating decision trees. The primary challenge in forming the tree is the selection of the correct attribute as a root node and the branch nodes according to some criteria like, Entropy (E), Information Gain (IG), Gini Index, Gain Ratio, Reduction in Variance, and Chi-Square values.

In this case, as the model works on a dataset with a classification problem, only E and IG are necessary. The formulas used for selecting an appropriate attribute for multiple labels available for selection are given below in (5), (6), and (7) [32].

$$E(S) = -\sum_{i=1}^{c} p^i log_2 p^i \; for \; 1 \; attribute \tag{5}$$

$$E\,(T, X) = \sum_{C \in X} P\,(c)\, E\,(c) \; for \; two \; attributes \tag{6}$$

$$IG(T, X) = E(T) - E(T, X) \tag{7}$$

Here, S is the current state for the tree while p^i is the probability of an event, i whereas, T is the current state of the tree and X is the selected attribute.

Entropy is used for computing the stochasticity in the data being processed. High entropy leads to worse model training and vice-versa. Information gain is necessary as a statistical tool for estimating the quality of the information given in relation to the target feature and the training attribute. Both entropy and information gain are very important techniques for understanding the amount of information provided by training the model with the given attribute and the relation with the target feature in sight. A high IG and low entropy are necessary for a model to have good accuracy.

D. RANDOM FOREST (RF)

It is a popular ensemble learning algorithm with the decision tree at its core for classification of the data into multiple classes. It is a subset of data and the subset of features obtained from the multiple decision trees used, as well as the averages of the scores obtained from multiple decision trees.

Hence, the random forest is known as an ensemble technique. The use of the Gini importance formula given below is necessary to calculate the necessity of each node.

This is done by presuming two child nodes, thus forming a binary tree using the below-mentioned formula in (8) [33].

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \tag{8}$$

In this case, ni_j = the importance of node j, w_j = a weighted number of samples reaching node j, C_j = impurity value of node j, left(j) = child node from the left split on node j, and right(j) = child node from the right split on node j.

Then the importance for each node of a decision tree is further computed using the formula (9), where fi_i is the importance of the feature i.

$$fi_i = \frac{\sum j : node \; j \; spilts \; on \; feature \; i \; ni_j}{\sum_{k \in \, all \; nodes} ni_k} \tag{9}$$

Now it can be standardized with the help of the formula given (10) and brought into a range of values between 0 and 1 with the help of the denominator, i.e., the sum of all attribute importance values.

$$fi_i = \frac{fi_i}{\sum_{j \in all \; features} features \, fi_j} \tag{10}$$

The last level has the random forest's values which average the total values by dividing them the total number of trees T given as the divisor in (11).

$$fi_i = \frac{\sum_{j \in all \;\; trees} norms \, fi_{ij}}{T} \tag{11}$$

Here, $normsfi_{ij}$ is defined as the standardized attribute importance of node i in tree j, and fi_i is the final output of the RF.

The importance of every attribute is measured as the node impurity decreases with the probability of reaching the node. The node probability could be measured by the number of

samples averaged over the number of samples. A higher value indicates that the feature is more important.

Hyperparameters are necessary for the problem given, and using them gives great accuracy and increases the efficiency of the algorithm. Some of the parameters used increase the algorithm's predictive power, whereas others increase the speed of prediction and the time taken by the model.

E. EXTREME GRADIENT BOOSTING (XGB)

XGBoost is a popular non-proprietary ensemble machine learning algorithm used to implement distributed gradient boosting decision trees. It supports multiple tree boosting at the same time and uses this algorithm for multiple problems such as ranking, categorization, and regression. It is based on the gradient boosting algorithm, which uses an approximation-based loss function and different regularization techniques to optimize the learner's outcome. The individual decision tree scores are added to get the final prediction score using the following equation (12) [33].

$$\hat{y}_i = \Sigma_{k=1}^{k} f_x(x_i), f_x \in F \tag{12}$$

Here, K is the total number of trees over the possible decision tree space F and small $f(x)$ is the functional space over the data values. XGB uses an objective function using the formula (13) for the final prediction classes.

$$equal\mathcal{L}^{(t)} = \sum_{i=1}^{n} l[y_i, (\hat{y_i}^{(t-1)} + f_t\,(x_i)] + \Omega(f_t) \tag{13}$$

It looks like an $f(x + \Delta x)$, where $x = \hat{y}_i^{(t-1)}$ and y_i is a real value. The first part of the equation represents the loss function and the second part is the regularization parameter.

XGBoost uses regularization for correcting the complex models with the help of L1 (Lasso) and L2 (Ridge regression) to manage the overfitting problem of the trees. L1 decreases the unnecessary features coefficient to minimum values, which helps in the removal of underfitting, and L2 decreases the chances of the model being overfitted. The important difference between them is the technique of distinguishing penalty terms, which uses the following cost function formulas (14) and (15) with the difference of the coefficient of parameter β_j.

$$L1 = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{14}$$

$$L2 = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{15}$$

GB uses the first derivative and tries to decrease the loss by finding the most suitable dimensions. Whereas XGB sums weak base learners and estimates a multiplex equation comprising an anti-gradient and negative second derivative to calculate an accurate method to decrease loss. The results obtained are more accurate with the use of both first and second derivatives by XGB.

XGB uses different hyperparameters to improve model accuracy by tuning parameters like max_depth, learning rate, subsample, gamma, etc.

F. MODEL PERFORMANCE MEASUREMENT

The various performance metrics are used to evaluate the deployed model's efficiency using a confusion matrix and calculating measures like accuracy, recall, precision, and F1 score values. Also, the k-fold mean accuracy, Cohen Kappa score, model standard deviation, and ROC-AUC values are considered for the model evaluation.

A confusion matrix is a method of computing used for finding different statistical estimates of the predictions made by the ML classification algorithm to measure the performance of a classification problem. For a confusion matrix to work successfully, it requires a model trained on a dataset with similar features and a good quantity of data for accurate predictions on the test dataset.

The confusion matrix is used for computing metrics such as Precision, Accuracy, Sensitivity or recall values, and the Specificity of a custom user-made model built using multiple ML classification models is represented in Table 2, and the performance metrics are described in Table 3.

TABLE 2. The confusion matrix in the binary classification of HD.

Actual target level	Prediction target level	
	HD = 1 (Positive)	Normal = 0 (Negative)
HD = 1 (True)	True Positive (TP)	True Negative (TN)
Normal = 0 (False)	False Positive (FP)	False Negative (FN)

In standard deviation, the value of the ith element of xi and the mean (μ) among the N number of elements.

In Ks, the po is derived by adding the sum of the diagonal values of the confusion matrix divided by the other values; it is the actual and predicted value agreement, and the p_e is the probability values of the by chance agreement between the true and false values.

Mean accuracy n indicates the total observations, yi and $f(xi)$ is the actual and predicted value of the iteration. The mean square error is calculated in each iteration concerning each fold data split.

IV. PROPOSED METHODOLOGY

The below segment illustrates the collected dataset descriptions, various data pre-processing techniques used in the model creation, and the proposed stacked model. The entire flow of this research is presented in a diagram in Fig. 1.

Fig. 1 depicts the detailed insight, which consists of dataset acquisition followed by exploratory data analysis and dataset splitting, the next model training using default and hyperparameter tuning, and finally, the proposed stacking model.

A. DATASET DESCRIPTION

This study uses a dataset collected from an open-source online portal, IEEE Data Port [7]. The dataset was associated with five other datasets: the Hungarian dataset, the Cleveland dataset, the Switzerland dataset, the Long Beach dataset, and the Statlog dataset. This combined dataset (DF1) contains

TABLE 3. Performance measurement metrics of the ML models.

Performance Metrics	Description	Mathematical Formula
Accuracy (Acc)	Calculates the exact correct prediction by the classification models.	$\frac{TP+TN}{TP+FP+FN+TN}$
Precision (Pr)	Measures the correct positive prediction among all the positive predictions by the model.	$\frac{TP}{TP+FP}$
Recall (Re)	Measure the actual positive prediction among all the correct predictions by the models. FN is a major concern in the medical disease detection model.	$\frac{TP}{TP+FN}$
F1-Score (Fs)	It is a harmonic mean or Pr and Re, where both are important to predict the trends of a model.	$\frac{2 \times Pr \times Re}{Pr+Re}$
Standard Deviation (Sd)	Indicates the errors in the predicted values and measures the difference between the incorrect and the actual score to be predicted. Less Sd means a more stable model.	$\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{N}}$
Cohen's Kappa Score (Cs)	The model's reliability and validity are measured by this metric and indicate the two raters' agreement for the prediction model.	$k = \frac{p_0-p_e}{1-p_e}$ $k = 1-\frac{1-p_0}{1-p_e}$
ROC-AUC Score (Ra)	AUC score tells the correct separation of classes and measures the efficiency of the prediction model. Higher values indicate the goodness of fit. It is a good choice for the imbalanced dataset. The TPR and FPR plot the entire area under the curve with different threshold values.	$TPR = \frac{TP}{TP+FN}$ $FPR = \frac{FP}{FP+TN}$
Mean Accuracy (Ma)	The k-fold cross-validation is used to average the performance metrics of the values predicted by the model by training and testing on the different number of k-split datasets.	$MSE = (1/n) * \Sigma(yi - f(xi))2$ $Test\ MSE = (1/k) * \Sigma MSEi$

1190 instances, a total of 11 important features, and a target outcome describing whether a person has a heart risk, which is all collected from the above five datasets. The description of the datasets with their respective instances is given below in Table 4.

The detailed analysis of the features, their definitions, datatypes, and the number of null values, is given in Table 5.

Also, another dataset is collected from the Mendeley Data portal [34], named DF2, like the IEEE Data Port, with fewer instances of about 1000 records and 12 related features of the heart disease risk of a person. Only one additional feature named ''number of major vessels'' was there. The DF2 is used to validate the performance of the deployed model with similar features but more diverse variance of the databases.

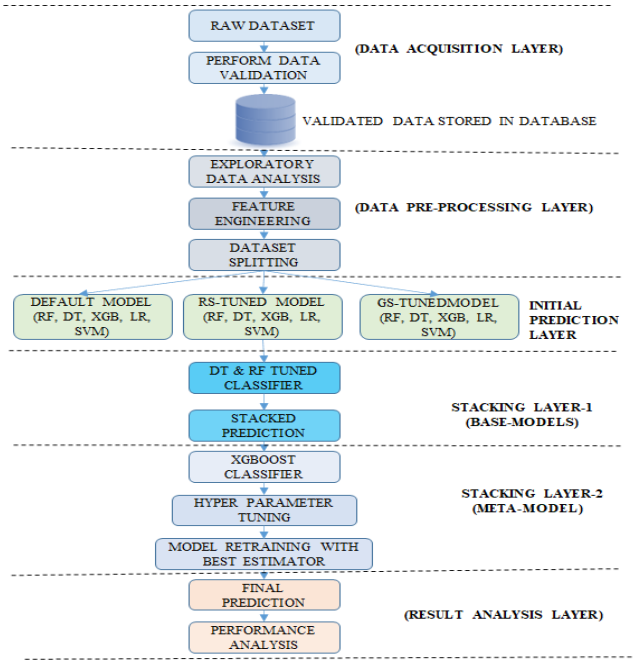

FIGURE 1. The workflow diagram of the proposed framework of the stacked model.

TABLE 4. Description of the considered dataset (DF1) features.

Dataset Name	Instances taken
Statlog (Heart) Dataset	270
Switzerland	123
Hungarian	294
Cleveland	303
Long Beach VA	200
Total	1190

B. DATA PRE-PROCESSING

Data pre-processing, or feature engineering, is an important tool in ML models for preparing the raw data, making it suitable for model training and testing. The features are simplified and reshaped to achieve the goal of getting a better result. We have applied a few data pre-processing processes to train the used dataset properly. The same pre-processing techniques were applied to both the datasets DF1 and DF2 due to their similar characteristics of each feature.

1) NULL VALUES REMOVAL

Null values represent missing values in a dataset. After proper analysis, we found that the feature 'cholesterol' in DF1 has approximately 172 null values, and in DF2, the feature 'serumcholestrol' has 53 null values. So, to fill those null values, the customized imputation technique is applied. we first replaced them with zeroes and reinstated them with the median value of the concern feature column. Due to the lower number of instances of both datasets and to intact their originality we decided to do customized imputation approaches.

TABLE 5. Describing the dataset names along with their instances taken.

Feature Name	Datatype Integer (I) Float (F)	#NULL values	Description
Age	I	0	Age of patients in years
Sex	I	0	Describing gender male (1) and female (0)
Chest pain type	I	0	1 describes typical chest pain, 2 describes typical angina, 3 describes non-anginal pain, and 4 is asymptomatic
Resting BP	I	0	Rest mode blood pressure level
Cholesterol	I	172	Concentration of Cholesterol
Fasting blood sugar	I	0	Fasting Blood sugar> 120 mg/dl represents 1 for true and 0 as false
Resting ECG	I	0	ECG test result during rest is categorized as 0 for normal and 1 for abnormality
Max heart Rate	I	0	Maximum heart rate measured
Exercise Angina	I	0	0 depicting no angina induced and 1 describes angina induced
Old Peak	F	0	Exercise-induced ST-depression rate/ rest state.
ST slope	I	0	ST slope measured during exercise 0: Normal 1: Unsloping 2: Flat 3: Down sloping.
Target	I	0	1 (HD) and 0 (Normal)

2) FINDING THE CORRELATION

Correlation means finding the relationship between two or more variables. In this case, we need to find how the features correspond with the target variable so that the training can be done with highly correlated features. This can be easily done using the correlation feature heatmap of the dataset. the positive and negative correlation of each feature with the target features is represented in dark and cool colors respectively. The heatmap of the DF1 and DF2 is shown in Fig. 2 and 3.

3) FEATURE DENSITY

Feature density can easily find the distribution of the features through the entire dataset, which thereby helps to find any outliers present or not. The probability distribution of each feature is represented by a histogram. The features are in a dense nature, and most of the values are viewed as non-zero. So, the ML classifiers can easily handle the dense feature vectors. Each feature space value is relatively small, so for HD prediction purposes features should play an important role. The feature influence-based HD prediction model should be the other alternative approach. The feature density curves of all the features for both DF1 and DF2 were derived but for authenticity, only DF1 density curves are given below in Fig. 4.

4) SCALING THE VALUES

Scaling generally means bringing down the values of all the features except the target outcome of the dataset to the same scale between 0 to 1; it is also useful for removing the

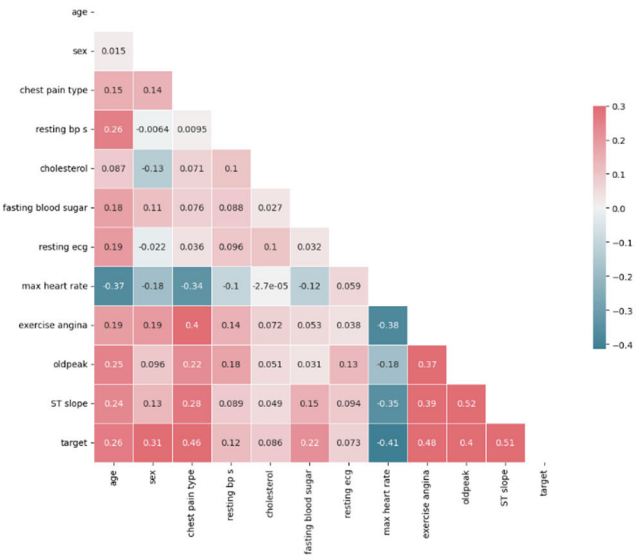

FIGURE 2. The feature correlation heatmap of the used dataset DF1.

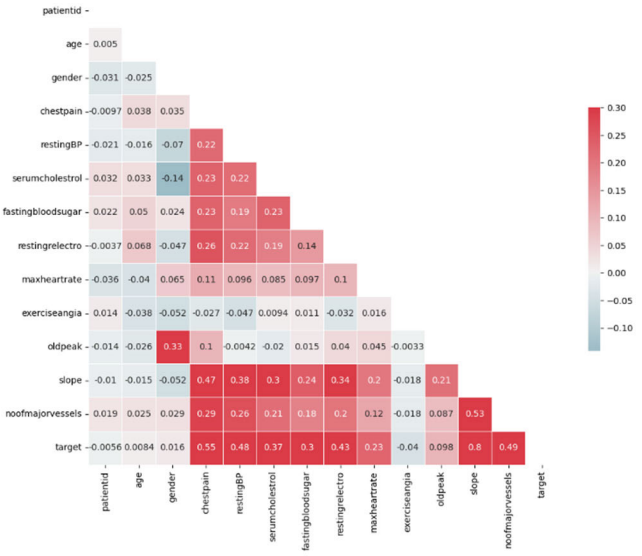

FIGURE 3. The feature correlation heatmap of the used dataset DF2.

outliers of each feature of the dataset. We have applied the *StandardScaler* library from Sci-Kit Learn to standardize and scalarize the ranges of values in both DF1 and DF2 datasets.

5) BALANCING THE DATASET

After examining the dataset properly, we found that the target variable, which depicts whether a patient has HD or not, was almost balanced, so no further oversampling techniques were applied to make it balanced. In disease prediction, it is undesirable to create the synthetic values of the target column which may create biases in the ML model prediction outcomes. The probability graph of the target variable of DF1 and DF2 is given below in Fig. 5 and 6.

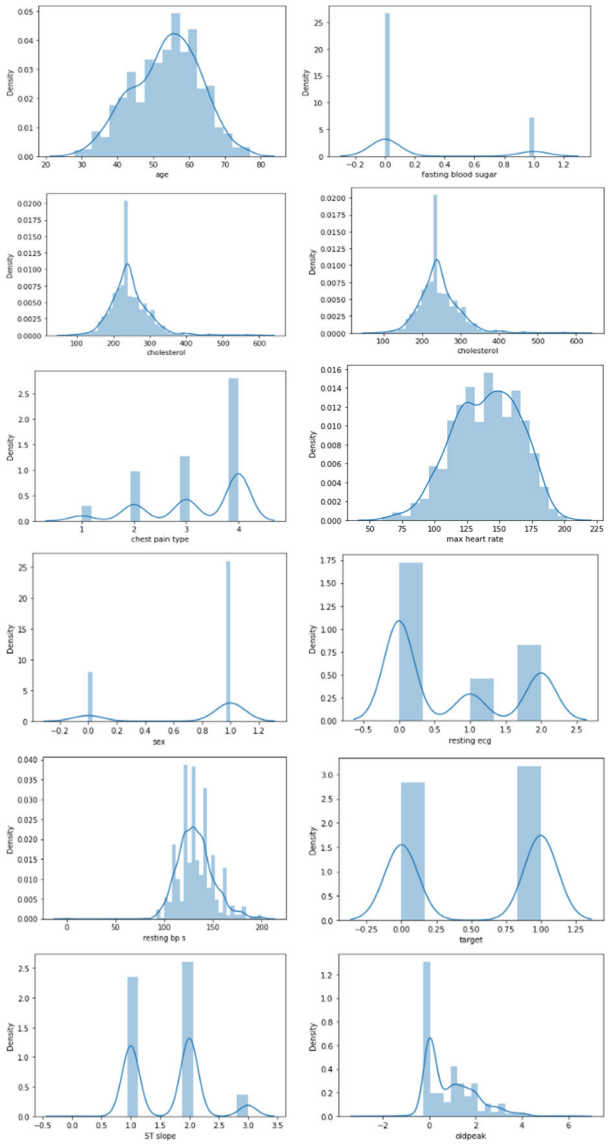

FIGURE 4. The density distribution curve for each feature present in the dataset of DF1.

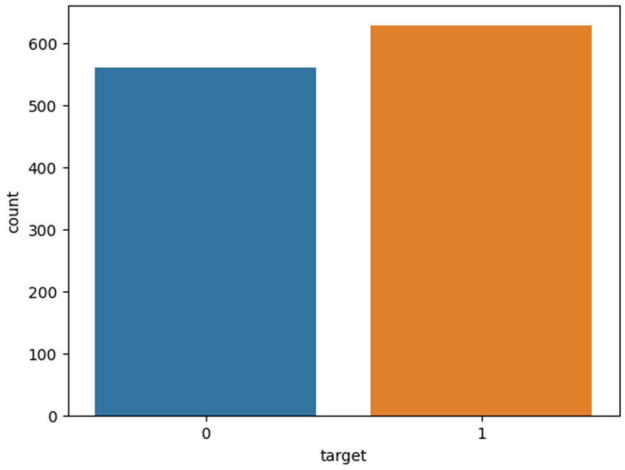

FIGURE 5. Target outcome instance distribution of DF1.

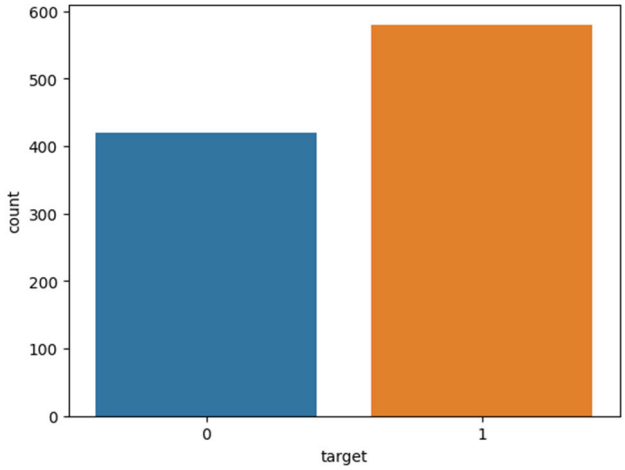

FIGURE 6. Target outcome instance distribution of DF2.

C. MODEL TRAINING & EXPERIMENTAL RESULTS

Machine Learning is the process of training and teaching machines to achieve human intelligence. In this proposed research study, we used different classical ML classifiers to build the prediction models of HD risk. After applying the preprocessing methods on the collected two datasets, now the DF1 and DF2 are suitable to deploy the ML models. As the dataset has fewer instances, next to decide how the models will be trained with how many instances as training data. So, we must split the dataset properly before training the models with the desired datasets. For this purpose, we used k-fold cross-validation, which is a technique where a proper splitting point is discovered in the dataset and split into train and test ratios, respectively. This is mainly used to estimate the prediction model skill to test every time on unseen data. The different reference values of k were applied to choose the exact value of k-fold to generalize the model prediction during training and testing. To trade off the bias-variance of the deployed models as well as the considered collected data, the proper value of k was chosen before each model deployment. The 10-fold cross-validation was used to get more accurate results with a lower chance of overfitting the deployed models.

The next step was ML classifier selection and training the models properly. As discussed previously, we selected five well-known ML models from the literature survey: LR, SVM, DT, RF, and XGB. Researchers in the literature most frequently used those models for HD risk prediction in a more accurate and precise way.

The above models were trained and tested using the best configurable experimental simulation platform defined by the above architecture, which is replicated programmatically to deploy the models to find the experimental results in the Google Colab Python 3.2 compute engine with the GPU set to execute the models. To execute the hyperparameter tuning to search for the best parameter values of each model by

GridSearchCV and RandomizedSearchCV, the cloud engine DGX station is equipped with a Tesla PCIe V100 GPU, system RAM of 128 GB, and graphics RAM of 16 GB, which is used to get the deployed models' results to deliver high-speed processing and more accurate performances. The experimental results are stored in the classification metrics: Acc, Pr, Re, Fs, Cs, Ra. Also, each model's standard deviation (Sd) was calculated by using the accuracy metric.

The detailed analysis of the above-mentioned default models' experimental results under test conditions with default parameters set as random_state equal to 'zero' is noted in Table 6 of DF1 and DF2.

TABLE 6. The experimental results with the default parameter of deployed models.

Model	Acc (%)	Pr	Re	Fs	Cs	Sd	Ra
			DF1				
XGB	91.60	0.92	0.92	0.92	0.83	1.60	0.92
RF	94.96	0.95	0.95	0.95	0.82	0.34	0.95
DT	88.66	0.89	0.89	0.89	0.77	0.74	0.89
LR	84.03	0.84	0.84	0.84	0.68	1.43	0.84
SVM	70.59	0.71	0.71	0.71	0.41	1.85	0.71
			DF2				
XGB	95.65	0.96	0.95	0.96	0.92	1.52	0.95
RF	96.00	0.96	0.96	0.96	0.92	0.36	0.96
DT	93.00	0.92	0.92	0.92	0.84	0.65	0.92
LR	87.00	0.87	0.86	0.86	0.73	1.65	0.86
SVM	78.00	0.77	0.78	0.77	0.55	2.03	0.78

The outcome was not up to par after observing the results of the different performance metrics in the above table. Only the XGB, RF, and DT models with default parameters performed in both datasets above 90% with mean accuracy after applying a 10-fold CV. Also, the stability of the three models was observed in healthy positions. The experimental observations on the dataset DF2 performed better for DF1 due to low instances. So, we decided to apply hyper-parameter search tuning techniques: RandimizedSearchCV and GridSearchCV to each ML model. Fine-tuning each model's best parameters with their estimator values might be the best solution for the HD prediction.

D. RANDOMIZED SEARCH CV

Randomized Search (RS) is an ML technique where the different hyperparameters are used in a random combination with the related training parameters of a model to find the best estimators' values of the corresponding parameters from the models. All the above-mentioned five models were trained with RS-CV, and the best parameter values were noted after executing the tuning method with the help of those parameters. The models were again trained by passing the best parameter values to achieve a better outcome under the test condition. The outcomes of the RS-CV models with 10-fold cross-validation are mentioned below in Table 7 on both the DF1 and DF2 under the testing environment.

TABLE 7. Experimental results of the different metrics of the deployed models with RS CV.

Model	Acc (%)	Pr	Re	Fs	Cs	Sd	Ra
			DF1				
XGB-RS	95.38	0.95	0.95	0.95	0.91	1.63	0.95
RF-RS	95.00	0.95	0.95	0.95	0.82	0.32	0.91
DT-RS	94.53	0.95	0.95	0.95	0.89	0.72	0.95
SVM-RS	84.87	0.85	0.85	0.85	0.70	1.89	0.85
LR-RS	84.45	0.84	0.84	0.84	0.69	1.47	0.85
			DF2				
XGB-RS	97.00	0.97	0.97	0.96	0.92	1.59	0.93
RF-RS	96.65	0.95	0.94	0.95	0.92	0.43	0.94
DT-RS	95.85	0.95	0.95	0.96	0.91	0.82	0.94
LR-RS	84.50	0.84	0.83	0.84	0.67	2.17	0.83
SVM-RS	87.50	0.87	0.87	0.87	0.74	2.01	0.87

The experimental outcome of RS-CV models corresponding to XGB, RF, and DT on both datasets is increased. Performing the RS-CV on each model to find the best estimator values did not take more time and was less complex.

E. GRID SEARCH CV

Grid Search (GS) is also a hyper-parameter tuning method that is used to search through every possible combination of the best parameter values from the given set of parameters. With the help of GS CV, we noted the best parameters of each model mentioned in Table 10. After that, we trained the models with those best parameters, and the executed results obtained after testing the model using the test dataset of 10-fold CV are mentioned below in Table 8.

TABLE 8. Experimental results of the different metrics of the deployed models with GS CV.

Model	Acc (%)	Pr	Re	Fs	Cs	Sd	Ra
			DF1				
XGB-GS	93.20	0.93	0.93	0.93	0.86	1.72	0.93
RF-GS	91.00	0.91	0.91	0.91	0.82	0.41	0.91
DT-GS	85.29	0.85	0.85	0.85	0.71	0.77	0.85
LR-GS	85.29	0.85	0.85	0.85	0.71	1.53	0.85
SVM-GS	84.03	0.84	0.84	0.84	0.68	1.93	0.84
			DF2				
XGB-GS	96.00	0.96	0.95	0.95	0.91	1.98	0.95
RF-GS	95.00	0.95	0.94	0.95	0.89	0.60	0.94
DT-GS	88.00	0.88	0.88	0.88	0.75	0.71	0.87
LR-GS	84.50	0.84	0.83	0.84	0.67	1.41	0.83
SVM-GS	88.50	0.89	0.87	0.88	0.76	1.88	0.87

The parameter selection of each model was the main challenge faced during the GS-CV process. GS usually takes more time to execute due to considering each combination

TABLE 9. The best estimator of the deployed models with RS CV.

Model	RS-CV Best Estimators value	RS-CV searching Estimators
XGB	learning_rate='0.1', min_child_weight=7, subsample=0.7, max_depth=30, n_estimators=400	"n_estimators": [start=100, stop=1200] "learning_rate": ['0.05', '0.1', '0.2', '0.3','0.5', '0.6'], "max_depth": [5 to 30], "subsample": [0.7, 0.6, 0.8], "min_child_weight": [3, 4, 5, 6, 7]
LR	C=22.54434690031882, max_iter=100, multi_class='auto', intercept_scaling=1, penalty='l2', tol=0.0001, solver='lbfgs'	'C': array ([1.00e-03, 1.47e-01, 2.15e+01, 3.16e+03, 4.64e+05, 6.81e+07, 1.00e+10]), 'penalty': ['l2', 'l1', 'elasticnet'], max_iter=(100 to 500), multi_class='auto', intercept_scaling=1, tol=0.0001, solver='lbfgs'
SVM	C=1300, cache_size=200, decision_function_shape='ovr', tol=0.0001, degree=3, kernel='rbf', gamma=0.0001, max_iter=-1, tol=0.001	'C': [0.1, 1, 10, 100, 1000, 1300], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf'], cache_size=200, kernel='rbf', degree=3, decision_function_shape='ovr', max_iter=-1, tol=0.001
RF	criterion='entropy', max_features='log2', max_depth=20, n_estimators=600, min_samples_leaf=2,	"n_estimators": [(start=100, stop=1200, num=12)], "criterion": ["gini", "entropy"], "max_depth": [(5, 30, num=6)], "min_samples_split": [2,5,10,15, 100], "min_samples_leaf": [1,2,5,10], "max_features": ["auto", "sqrt", "log2"]
DT	criterion='entropy', min_samples_split=5, max_depth=25, min_samples_leaf=2, max_features='log2',	"criterion": ["gini", "entropy"], "max_depth": [(5, 30, num=6)], "min_samples_split": [2,5,10,15, 100], "min_samples_leaf": [1, 2, 5, 10], "max_features": ["auto", "sqrt", "log2"]

TABLE 10. The best estimator of the deployed models with GS CV.

Model	GS-CV Best Estimators Value	GS-CV Searching Estimators
XGB	max_depth=6, learning_rate='0.05', n_estimators=180, min_child_weight=7, subsample=0.8	'max_depth': range (2, 10, 1), 'n_estimators': range (60, 220, 40), 'learning_rate': [0.1, 0.01, 0.05], subsample= [0.6, 0.7, 0.8], "min_child_weight": [3, 4, 5, 6, 7]
LR	C=21.54434690031882, max_iter=100, multi_class='auto', intercept_scaling=1, penalty='l2', tol=0.0001, solver='lbfgs'	'C': array ([1.00e-03, 1.47e-01, 2.15e+01, 3.16e+03, 4.64e+05, 6.81e+07, 1.00e+10]), 'penalty': ['l2', 'l1', 'elasticnet'], max_iter=(100 to 500), multi_class='auto', intercept_scaling=1, tol=0.0001, solver='lbfgs'
SVM	C=1000, cache_size=200, kernel='rbf', degree=3, decision_function_shape='ovr', gamma=0.0001, max_iter=-1, kernel='rbf', tol=0.001	'C': [0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf'], cache_size=200, kernel='rbf', degree=3, decision_function_shape='ovr', max_iter=-1, tol=0.001
RF	criterion='entropy', min_samples_leaf=2, max_depth=8, n_estimators=200 min_samples_split=5	'n_estimators': [200, 500], 'max_features': ['auto', 'sqrt', 'log2'], 'max_depth': [4,5,6,7,8], 'criterion' :['gini', 'entropy'], min_samples_split=[5, 10, 15]
DT	criterion='gini', splitter='best', max_depth=15, min_samples_split=2, max_features='sqrt', min_samples_leaf=5	'criterion': ['gini', 'entropy'], 'max_depth': [5, 10, 15, 20, 25, 30], 'max_features': ['auto', 'sqrt', 'log2'], 'min_samples_leaf': [1, 2, 5, 10], 'min_samples_split': [2, 5, 10, 15, 100]

of parameters. So, it is better to choose a few parameters but the most prominent parameters selection will produce the best results. Also, it requires more system resources to perform the entire process. Despite facing the challenges, we performed the parameter searching by providing sufficient resources. However, the experimental results were not satisfactory compared to default and RS-CV model outcomes. Only the boosting model performed better but tree-based and other linear models did not produce suitable outcomes due to their inherent nature.

F. EXPERIMENTAL ANALYSIS

The experimental results analysis regarding the k-fold mean accuracy and ROC-AUC score of the three types of prediction models, as the default model, and with two hyper-parameter tuning models are discussed in this section. Also, the selected best parameters of the tuning models corresponding to the best estimator values of RS-CV and GS-CV are represented here. The best estimator parameter values of all the deployed models were obtained after executing the RS-CV and GS-CV by considering all possible tuning parameters of each model; the searching estimators and the best searching parameter values are represented in Tables 9 and 10 for the DF1 respectively. The same parameter and the same estimator values of each ML model were considered for the deployment by using the DF2 under an identical experimental environment. To generalize the model hyperparameter same estimator values were taken regardless of the different dataset.

The accuracy of the models with a standalone classifier with default parameter values, RS-CV, and GS-CV classifiers' comparative results are depicted graphically in Fig. 7 for the DF1 and DF2. From the diagram, the randomized search CV model's performance is better concerning others.

The ROC-AUC curve of all the five deployed models with Standalone default parameters, RS-CV, and GS-CV plotted with their probabilistic values is depicted in Fig. 8 to 12, respectively concerning DF1.

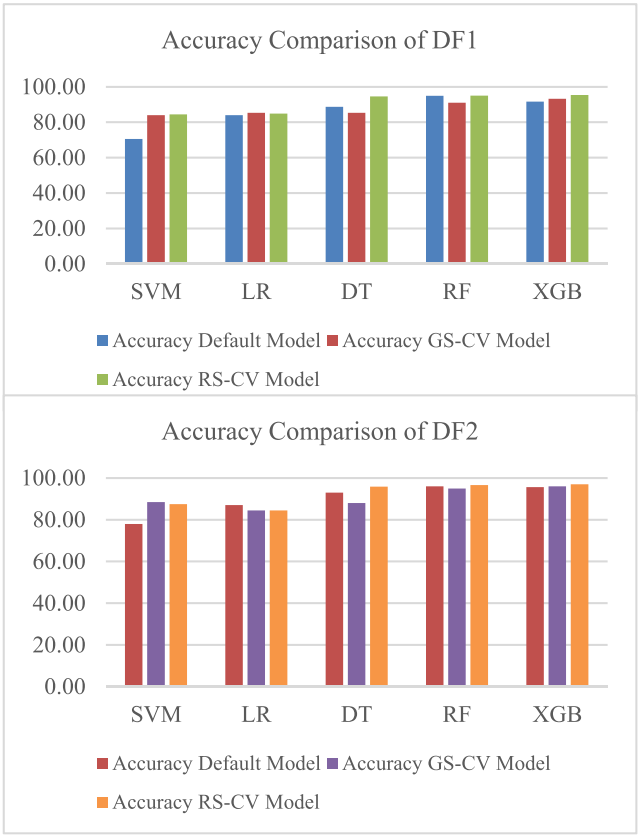

FIGURE 7. Comparative accuracy of the deployed models for DF1 & DF2.

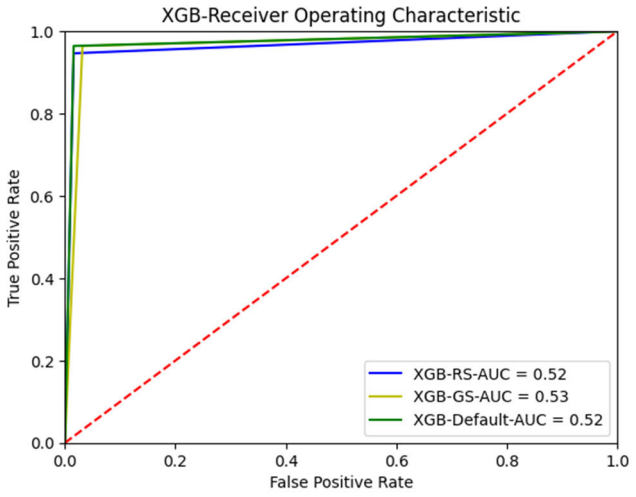

FIGURE 8. The ROC-AUC probabilistic curve of the XGB model concerning Default, RS-CV, & GS-CV.

G. STACKING ENSEMBLE MODEL

After proper analysis and observation, we found that the results shown by the standalone models were not up to the mark. So that is why we incorporated the concept of stacking, a widely used ensemble learning approach. The different heterogeneous weak learners are stacked as a base prediction, and usually, strong learners are pushed as a meta-predictor,

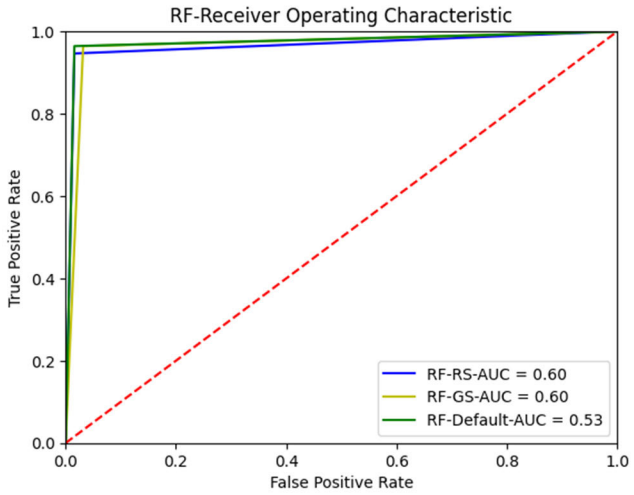

FIGURE 9. The ROC-AUC probabilistic curve of the RF model concerning Default, RS-CV, & GS-C.

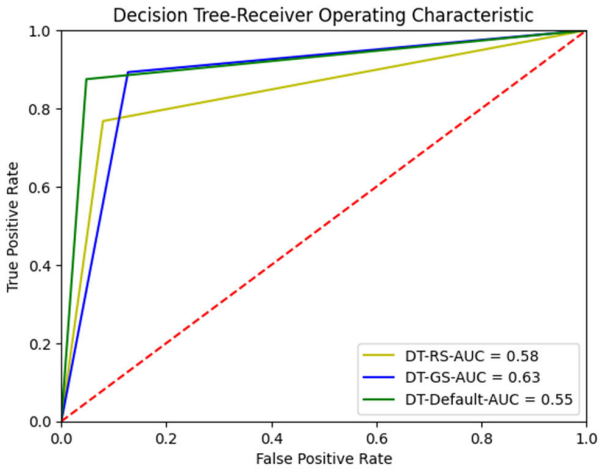

FIGURE 10. The ROC-AUC probabilistic curve of the DT model concerning Default, RS-CV, & GS-CV.

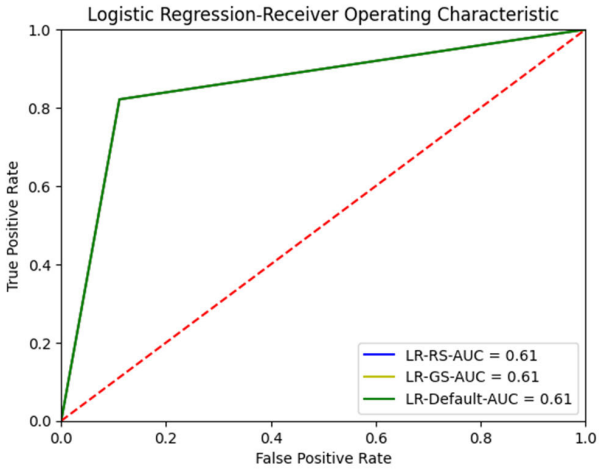

FIGURE 11. The ROC-AUC probabilistic curve of the LR model concerning Default, RS-CV, & GS-CV.

producing much better results. On analyzing the model results from the tables of both RS-CV and GS-CV, it was noticed

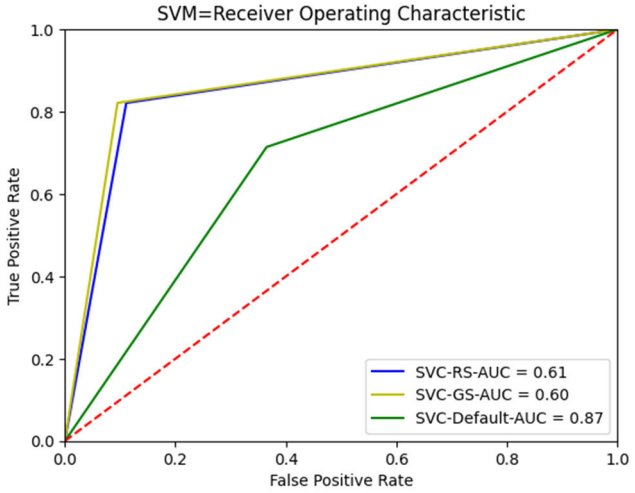

FIGURE 12. The ROC-AUC probabilistic curve of the SVC model concerning Default, RS-CV, & GS-CV.

TABLE 11. The staking models' experimental results.

Stack Model (RF, DT, XGB) - CV-10							
# Dataset	Acc (%)	Pr	Re	Fs	Cs	Ra	Sd
DF1	96.00	0.96	0.98	0.96	0.92	0.96	2.83
DF2	96.88	0.96	0.97	9.97	0.93	0.97	2.64

that three standalone models, namely RF, DT, and XGB, gave the best results for the RS-CV model. we decided to build the stacked model using these three. However, we faced difficulties in deciding whether a model should be kept as a meta-learner or base-level learner. So, we tried different permutations and combinations and concluded that selecting DT and RF base learners and XGB meta-learners would yield the best result regarding considered performance metrics. The tested proposed stack model with the holdout dataset after applying a 10-fold CV was kept independent, and the results are shown in Table 11. The ROC-AUC curve, prediction curve, and confusion matrix are given below in Fig. 13 and Fig.14 for DF1 and DF2.

The model considered ten-fold cross-validation and the standard deviation (Sd) to overcome the overfitting phenomenon. The results indicate that no such situation arises, and the actual and validation results show that the proposed model is more accurate.

During hyperparameter tuning of each model, we faced major difficulties and challenges in searching the best estimators' values under the GridSearchCV and RandomizedSearchCV method executions because it takes more time to search each parameter's best estimator values. Also, the tuning method takes up a lot of computing resources and uses a cloud engine to execute each model. So, fitting the five models with a proper parameter was a crucial challenge in this research study.

This study particularly focused on using the conventional ML classification model to predict HD from the perspective

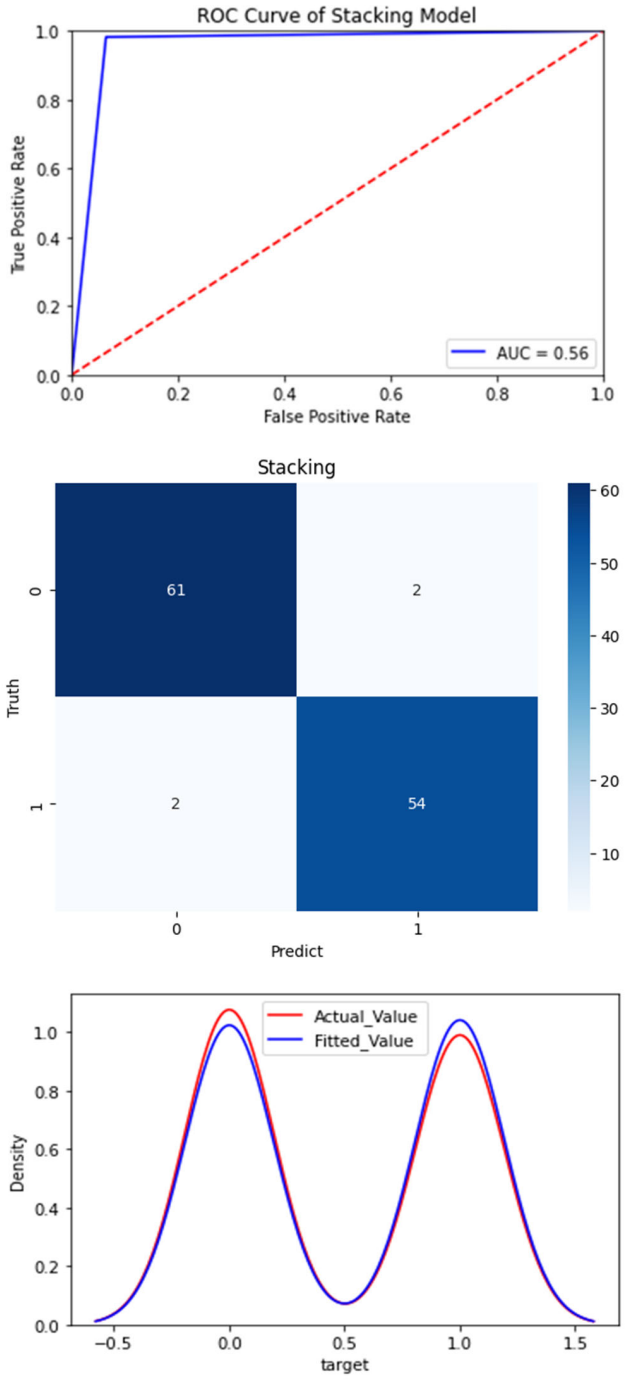

FIGURE 13. Dual-stage stacked model experimental outcome as ROC-AUC curve, CM, and Prediction curve for DF1.

of resource-constrained devices with the specific aim of a quick response disease detection model. The prediction accuracy may increase using other approaches like deep learning-based ANN models.

V. COMPARATIVE RESULTS ANALYSIS

This chapter represents a detailed investigation of the experimental outcomes of this study with the related literature observable results under their explained test conditions using

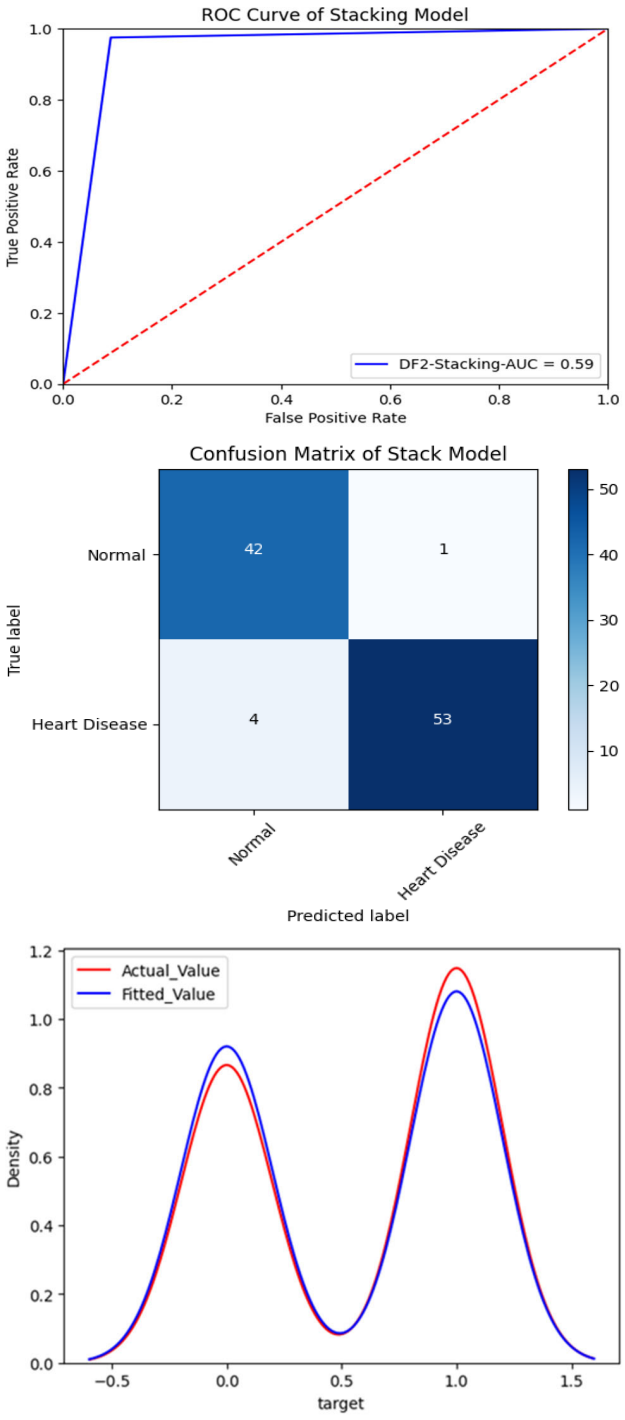

FIGURE 14. Dual-stage stacked model experimental outcome as ROC-AUC curve, CM, and Prediction curve for DF2.

the IEEE comprehensive HD dataset [7] and Mendeley dataset [34] and considering the well-known performance metrics mentioned earlier. Many researchers presented their results only considering the Acc, Pr, Re, and Fs, but few mentioned the ROC-AUC values. This study considers all these parameters to analyze the performance of this proposed model.

TABLE 12. The claimed literature outcomes analysis with the proposed model.

#Ref	Acc (%)	Pr	Re	Fs	Ra	Dataset used # Instances	
						DF1 1190	DF2 1000
[4]	93.39	0.99	0.88	0.90	-	√	
[4]	96.75	0.98	0.96	0.97	-		√
[6]	87.25	0.88	0.89	-		√	
Proposed Study	**96.88**	**0.96**	**0.97**	**0.97**	**0.97**		√
	96.00	**0.96**	**0.98**	**0.96**	**0.96**	√	

Compared with the literature [4] and [6], the stacked model is more accurate, and the recall and ROC-AUC values are better than all the literature. The results of the proposed stacking model analysis with all other researchers' studies are represented in Table 12. The blank cell indicates that the authors have not provided any results corresponding to metrics.

The same experiment, considering all the parameters and the test conditions, was replicated over the dataset [34] after applying the same data preprocessing steps and deploying the models. The experimental outcome observed with an accuracy of 96.88% and the ROC-AUC values of 0.97 are far better than the literature in the above table. The above two datasets are the most current and appropriate heart disease-related risk prediction open-source datasets that are used for research purposes.

VI. CONCLUSION & FUTURE WORK

This research study concludes a two-stage stacking-based ensemble prediction model for the HD risk using three best-performing ML models: RF, DT, and XGB. The RandomizedSearchCV and GridSearchCV were used for the best parameter searching of each model, and the results of the RandomizedSearchCV of the above three deployed models were stacked to get the average accurate prediction results for the final model selection. The model's performance creates a new benchmark with an average accuracy of 96%, which is better than the other researcher's observable model results. The proposed model also defeats the overfitting problem with less standard deviation and uses the k-fold CV technique. The recall value is very attractive in this disease prediction model, with negligible False Negative values of 0.84% under test conditions. The outcome simulation results will work effectively on resource-constrained devices in the case of lower dataset instances because disease detection has a limitation on input data. The other low-resource dataset demonstrates the prediction model's stability and robustness, with an excellent accuracy of 96.88% and a low standard deviation for the proposed HD risk prediction model.

This research study will be further extended with feature selection algorithms and deployed on reduced features to reach a new milestone finding. Also, a deep learning-based

ANN model can be incorporated to compare with these findings, and a more generalized model should be proposed using the larger instances of a dataset on the diversified domain of human disease prediction.

CONFLICT OF INTEREST

The authors do not have any financial or non-financial interest in this research study to disclose.

ACKNOWLEDGMENT

This study is solely conducted by the authors who prepared the manuscript with their research interests and capabilities, and the research was conducted at the Central Institute of Technology Kokrajhar (CITK), Kokrajhar, Assam, India. The Department of Computer Science and Engineering,, CITK, provided the research resource.

REFERENCES

[1] D. T. Khan. *Cardiovascular Diseases*. World Health Organization. Accessed: Oct. 20, 2022. [Online]. Available: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

[2] D. Thompson. (Feb. 2017). *Heart Disease May Cost $1 Trillion Yearly By 2035*. WebMD. Accessed: Oct. 20, 2022. [Online]. Available: https://www.webmd.com/heart-disease/news/20170214/heart-disease-could-cost-us-1-trillion-per-year-by-2035-report

[3] U.S. Department of Health & Human Services. (Oct. 14, 2022). *Heart Disease Facts*. Accessed: Oct. 20, 2022. [Online]. Available: https://www.cdc.gov/heartdisease/facts.htm

[4] B. P. Doppala, D. Bhattacharyya, M. Janarthanan, and N. Baik, "A reliable machine intelligence model for accurate identification of cardiovascular diseases using ensemble techniques," *J. Healthcare Eng.*, vol. 2022, pp. 1–13, Mar. 2022.

[5] R. Tao, S. Zhang, X. Huang, M. Tao, J. Ma, S. Ma, C. Zhang, T. Zhang, F. Tang, J. Lu, C. Shen, and X. Xie, "Magnetocardiography-based ischemic heart disease detection and localization using machine learning methods," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 6, pp. 1658–1667, Jun. 2019.

[6] M. Ozcan and S. Peker, "A classification and regression tree algorithm for heart disease modeling and prediction," *Healthcare Analytics*, vol. 3, Nov. 2023, Art. no. 100130.

[7] M. Siddhartha. (Jun. 2020). *Heart Disease Dataset (Comprehensive)*. IEEEDataPort. Accessed: Jul. 15, 2022. [Online]. Available: https://ieee-dataport.org/open-access/heart-disease-dataset-comprehensive

[8] C. B. C. Latha and S. C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Informat. Med. Unlocked*, vol. 16, Jan. 2019, Art. no. 100203.

[9] H. A. Esfahani and M. Ghazanfari, "Cardiovascular disease detection using a new ensemble classifier," in *Proc. IEEE 4th Int. Conf. Knowl.-Based Eng. Innov. (KBEI)*, Tehran, Iran, Dec. 2017, pp. 1011–1014.

[10] R. Atallah and A. Al-Mousa, "Heart disease detection using machine learning majority voting ensemble method," in *Proc. 2nd Int. Conf. new Trends Comput. Sci. (ICTCS)*, Amman, Jordan, Oct. 2019, pp. 1–6.

[11] M. Kavitha, G. Gnaneswar, R. Dinesh, Y. R. Sai, and R. S. Suraj, "Heart disease prediction using hybrid machine learning model," in *Proc. 6th Int. Conf. Inventive Comput. Technol. (ICICT)*, Coimbatore, India, Jan. 2021, pp. 1329–1333.

[12] A. U. Haq, J. Li, J. Khan, M. H. Memon, S. Parveen, M. F. Raji, W. Akbar, T. Ahmad, S. Ullah, L. Shoista, and H. N. Monday, "Identifying the predictive capability of machine learning classifiers for designing heart disease detection system," in *Proc. 16th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, Chengdu, China, Dec. 2019, pp. 130–138.

[13] K. Yuan, L. Yang, Y. Huang, and Z. Li, "Heart disease prediction algorithm based on ensemble learning," in *Proc. 7th Int. Conf. Dependable Syst. Appl. (DSA)*, Xi'an, China, Nov. 2020, pp. 293–298.

[14] I. D. Mienye, Y. Sun, and Z. Wang, "An improved ensemble learning approach for the prediction of heart disease risk," *Informat. Med. Unlocked*, vol. 20, Mar. 2020, Art. no. 100402.

[15] S. Asif, Y. Wenhui, Y. Tao, S. Jinhai, and H. Jin, "An ensemble machine learning method for the prediction of heart disease," in *Proc. 4th Int. Conf. Artif. Intell. Big Data (ICAIBD)*, Chengdu, China, May 2021, pp. 98–103.

[16] N. Basha and P. Venkatesh, "Early detection of heart syndrome using machine learning technique," in *Proc. 4th Int. Conf. Electr., Electron., Commun., Comput. Technol. Optim. Techn. (ICEECCOT)*, Mysuru, India, Dec. 2019, pp. 387–391.

[17] W. M. Jinjri, P. Keikhosrokiani, and N. L. Abdullah, "Machine learning algorithms for the classification of cardiovascular disease—A comparative study," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Amman, Jordan, Jul. 2021, pp. 132–138.

[18] P. Sujatha and K. Mahalakshmi, "Performance evaluation of supervised machine learning algorithms in prediction of heart disease," in *Proc. IEEE Int. Conf. Innov. Technol. (INOCON)*, Bangluru, India, Nov. 2020, pp. 1–7.

[19] K. Battula, R. Durgadinesh, K. Suryapratap, and G. Vinaykumar, "Use of machine learning techniques in the prediction of heart disease," in *Proc. Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME)*, Mauritius, Mauritius, Oct. 2021, pp. 1–5.

[20] Y. Lin, "Prediction and analysis of heart disease using machine learning," in *Proc. IEEE Int. Conf. Robot., Autom. Artif. Intell. (RAAI)*, Apr. 2021, pp. 53–58.

[21] S. Hameetha Begum and S. N. Nisha Rani, "Model evaluation of various supervised machine learning algorithm for heart disease prediction," in *Proc. Int. Conf. Softw. Eng. Comput. Syst. 4th Int. Conf. Comput. Sci. Inf. Manage. (ICSECS-ICOCSIM)*, Pekan, Malaysia, Aug. 2021, pp. 119–123.

[22] P. Motarwar, A. Duraphe, G. Suganya, and M. Premalatha, "Cognitive approach for heart disease prediction using machine learning," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng.*, Vellore, India, Feb. 2020, pp. 1–5.

[23] D. Rahmat, A. A. Putra, and A. W. Setiawan, "Heart disease prediction using K-nearest neighbor," in *Proc. Int. Conf. Electr. Eng. Informat. (ICEEI)*, Kuala Terengganu, Malaysia, Oct. 2021, pp. 1–6.

[24] W. A. H. W. Azizan, A. A. A. Rahim, S. L. M. Hassan, I. S. A. Halim, and N. E. Abdullah, "A comparative study of two machine learning algorithms for heart disease prediction system," in *Proc. IEEE 12th Control Syst. Graduate Res. Colloq. (ICSGRC)*, Shah Alam, Malaysia, Aug. 2021, pp. 132–137.

[25] P. Khurana, S. Sharma, and A. Goyal, "Heart disease diagnosis: Performance evaluation of supervised machine learning and feature selection techniques," in *Proc. 8th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Noida, India, Aug. 2021, pp. 510–515.

[26] R. Tr, U. K. Lilhore, S. Simaiya, A. Kaur, and M. Hamdi, "Predictive analysis of heart diseases with machine learning approaches," *Malaysian J. Comput. Sci.*, vol. 2022, pp. 132–148, Mar. 2022.

[27] M. B. Abubaker and B. Babayigit, "Detection of cardiovascular diseases in ECG images using machine learning and deep learning methods," *IEEE Trans. Artif. Intell.*, vol. 4, no. 2, pp. 373–382, Apr. 2023.

[28] P. Pal and M. Mahadevappa, "Adaptive multi-dimensional dual attentive DCNN for detecting cardiac morbidities using fused ECG-PPG signals," *IEEE Trans. Artif. Intell.*, 2022.

[29] K. Uyar and A. Ilhan, "Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks," *Proc. Comput. Sci.*, vol. 120, pp. 588–593, Jan. 2017.

[30] M. W. Nadeem, H. G. Goh, M. A. Khan, M. Hussain, M. F. Mushtaq, and V. Ponnusamy, "Fusion-based machine learning architecture for heart disease prediction," *Comput., Mater. Continua*, vol. 67, no. 2, pp. 2481–2496, 2021.

[31] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106229.

[32] A. Alsirhani, S. Sampalli, and P. Bodorik, "DDoS detection system: Using a set of classification algorithms controlled by fuzzy logic system in apache spark," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 936–949, Sep. 2019.

[33] X. Hu, H. Jia, Y. Zhang, and Y. Deng, "An open-circuit faults diagnosis method for MMC based on extreme gradient boosting," *IEEE Trans. Ind. Electron.*, vol. 70, no. 6, pp. 6239–6249, Jun. 2023.

[34] B. P. Doppala and D. Bhattacharyya. (Apr. 16, 2021). *Cardiovascular_Disease_Dataset*. Mendeley Data. Accessed: Feb. 15, 2023. [Online]. Available: https://data.mendeley.com/datasets/dzz48mvjht

SUBHASH MONDAL (Member, IEEE) received the B.Tech. and M.Tech. degrees in computer science and engineering from the University of Calcutta, Kolkata, India, in 2005 and 2007, respectively. He is currently pursuing the Ph.D. degree in CSE with the Central Institute of Technology Kokrajhar, Kokrajhar, Assam, India.

He is an Assistant Professor with the Department of Computer Science and Engineering (AI & ML), Dayananda Sagar University, Bengaluru, Karnataka, India. He has more than 17 years of teaching experience. He has published more than 27 research publications. His research interests include machine learning, deep learning, and natural language processing.

Mr. Mondal is a professional member of ACM.

RANJAN MAITY (Senior Member, IEEE) received the B.Tech. degree from the University of Kalyani, Kalyani, West Bengal, India, in 2002, the M.S. degree from the Indian Institute of Technology Kharagpur, India, in 2007, and the Ph.D. degree from the Indian Institute of Technology Guwahati, Guwahati, Assam, India, in 2019.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Central Institute of Technology Kokrajhar, Assam, India. His research interests include HCI, machine learning, and deep learning.

Dr. Maity is a Senior Member of ACM.

YACHANG OMO received the B.E. degree in civil engineering from the University B. D. T. College of Engineering, Davangere, Karnataka, and the M.Tech. and Ph.D. degrees from the North Eastern Regional Institute of Science and Technology, Arunachal Pradesh, in 2020.

He is currently an Assistant Professor with the Department of Civil Engineering, Central Institute of Technology Kokrajhar, Assam, India. His research interests include AI and ML for geo-environmental engineering and environmental engineering for sustainable development.

SOUMADIP GHOSH (Member, IEEE) received the B.Tech. degree in computer science technology (CST) from the University of Kalyani, Kalyani, West Bengal, India, in 2002, the M.Tech. degree in computer science and engineering (CSE) from the University of Calcutta, Kolkata, West Bengal, in 2005, and the Ph.D. degree in engineering from the University of Kalyani, in 2017.

He is currently a Professor with the Department of Computer Science and Engineering, Future Institute of Technology, Kolkata. He has contributed to numerous research articles in various journals, book chapters, and conferences of repute.

Dr. Ghosh is a member of IEI and ACM. He has served as a reviewer for reputed international journals, such as ACM, IEEE, Springer, and Elsevier.

AMITAVA NAG (Senior Member, IEEE) received the B.Tech. degree from the University of Kalyani, Kalyani, West Bengal, India, in 2003, the M.Tech. degree in information technology from the University of Calcutta, Kolkata, India, in 2005, and the Ph.D. degree in engineering from the University of Kalyani.

He is currently a Professor of computer science and engineering with the Central Institute of Technology Kokrajhar, Kokrajhar, Assam, India. He has more than 60 research publications in various international journals and conference proceedings. His research interests include the IoT, information security, machine learning, deep learning, and NLP.

Dr. Nag is a Senior Member of ACM and a fellow of IEI.