HYBRID MACHINE LEARNING TECHNIQUES FOR EFFICIENT HEART DISEASE PREDICTION

Report submitted to the SASTRA Deemed to be University as the requirement for the course

CSE300 / INT300 / ICT300 - MINI PROJECT

Submitted by

CHAYAMMA GARI HARITHA
(Reg. No.: 124157025, Computer Science & Engineering (Cyber Security & BlockChain Technology)
B.V.SAI.GEETHANJALI
(Reg. No.: 124157076, Computer Science & Engineering (Cyber Security & BlockChain Technology)
M. SAHITYA
(Reg. No.: 124157089, Computer Science & Engineering (Cyber Security & BlockChain Technology)

May 2023

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF COMPUTING

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the report titled "**HYBRID MACHINE LEARNING TECHNIQUES FOR EFFICIENT HEART DISEASE PREDICTION"** submitted as a requirement for the course, CSE300 / INT300 / ICT300: **MINI PROJECT** for B.Tech. is a bonafide record of the work done by **Ms. Chayamma Gari Haritha (Reg. No.: 124157025, Computer Science & Engineering (Cyber Security & Blockchain Technology) , Ms. B.V.Sai.Geethanjali (Reg. No.: 124157076, Computer Science & Engineering (Cyber Security & Blockchain Technology) Ms. M.Sahitya(Reg. No.: 124157089, Computer Science & Engineering (Cyber Security & Blockchain Technology)** during the academic year 2022- 23, in the School of Computing, under my supervision.

Signature of Project Supervisor : N. Senthil Selvan

Name with Affiliation : DR. SENTHIL SELVAN N (AP-II/CSE/SOC)

Date : 08-05-22

Mini Project *Viva voce* held on_____

Examiner 1 **Examiner 2**

Acknowledgements

We would like to thank our Honorable Chancellor **Prof. R. Sethuraman** for providing us with an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

We would like to thank our Honorable Vice-Chancellor **Dr. S. Vaidhyasubramaniam** and **Dr. S. Swaminathan,** Dean**,** Planning & Development, for the encouragement and strategic support at every step of our college life.

We extend our sincere thanks to **Dr. R. Chandramouli,** Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

We extend our heartfelt thanks to **Dr. A. Umamakeswari**, Dean, School of Computing, **Dr.S.Gopalakrishnan**,Associate Dean**,** Department of Computer Application, **Dr. B.Santhi,** Associate Dean, Research, **Dr. V. S. Shankar Sriram**, Associate Dean, Department of Computer Science and Engineering, **Dr. R. Muthaiah**, Associate Dean, Department of Information Technology and Information & Communication Technology .

Our guide **Dr. Senthil Selvan N,Assistant Professor**, School of Computing was the driving force behind this whole idea from the start. His deep insight in the field and invaluable suggestions helped us in making progress throughout our project work. We also thank the project review panel members for their valuable comments and insights which made this project better.

We would like to extend our gratitude to all the teaching and non-teaching faculties of the School of Computing who have either directly or indirectly helped us in the completion of the project.

We gratefully acknowledge all the contributions and encouragement from my family and friends resulting in the successful completion of this project. We thank you all for providing me an opportunity to showcase my skills through project.

List of Figures

LIST OF TABLES

Abbreviations

KNN	K-Nearest Neighbour
DT	Decision Tree
SVM	Support Vector Machine
MLP	Multilayer Perceptron
RF	Random Forest
ROC	Receiver Operating Characteristic
TN	True Negative
TP	True Positive
FN	False Negative
FP	False Positive

Abstract

One of the dominant causes of death in the world is cardiovascular heart disease. Cardiovascular disease prediction presents a significant challenge for clinical data analysis. With the use of machine learning (ML), it has been established that it is possible to make predictions and judgments from the large amounts of clinical data generated by the healthcare sector. Additionally, we have observed the employment of ML approaches in recent advancements across several IoT domains (IoT). Only a few researches have looked into using ML to predict cardiac disease. In this study, by applying machine learning approaches to uncover critical traits, we propose a novel method to increase the accuracy of cardiovascular disease prediction.

KEY WORDS: Machine Learning Techniques, Ensemble Model, Cardio Vascular Disease

TABLE OF CONTENTS

CHAPTER 1

SUMMARY OF BASE PAPER

Table 1.1 Base Paper Details

Title	HYBRID MACHINE LEARNING TECHNIQUES FOR EFFICIENT HEART DISEASE PREDICTION
Authors	SENTHILKUMAR MOHAN , CHANDRASEGAR THIRUMALAI, AND GAUTAM SRIVASTAVA
Journal Name	IEEE
Year of Publishing	2019
Link	https://ieeexplore.ieee.org/document/8740989

1.1 Introduction:

Heart disease prediction has always been a great complexity and is one of the dominant causes of mortality. Due to the difficulty in detection and accuracy of tests it is becoming hard to prevent the disease at initial phases. Sometimes it is harder to diagnose because a patient with heart disease can have a wide range of symptoms or none. Some of the challenges faced in heart disease prediction include: 1) Asymptomatic nature 2) Vague symptoms 3)Limited diagnosis tests 4)Lack of standardization 2) Limited access to care. Though not all can be challenges can be defeated using Machine Learning Algorithms but it certainly has its advantages. Fig 1.1 shows the diagnosing of heart disease.

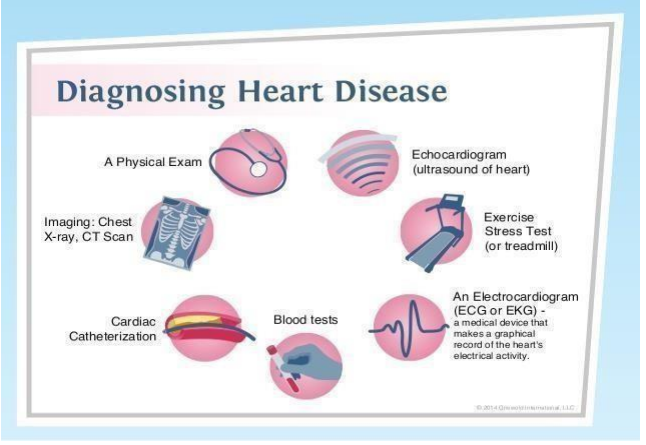

Fig 1.1 Heart disease diagnosis

In this project, we implement the application of a hybrid machine learning strategy that integrates the algorithms of support vector machines, random forests, and logistic regression to predict the outcome. Statistically, the association between one or more predictor factors and a binary outcome, such as the presence which is true (represented as 1) or absence of heart disease (represented as 0), can be modeled using logistic regression. An ensemble learning technique called random forest combines numerous decision trees to increase prediction accuracy. A powerful method called a support vector machine may classify data using a hyper plane in a high-dimensional space. Our findings show that this hybrid approach performs better than individual machine learning algorithms and can be a useful tool for identifying patients at risk for heart disease. This method can assist healthcare practitioners execute targeted preventative and treatment methods and ultimately improve patient outcomes by increasing the accuracy of heart disease prediction.

1.2 Dataset:

The dataset used in this project is already pre-processed and is downloaded from UCI repository. The dataset contains 303 patient records out of which only 297 can be used since the remaining 6 records have missing data values. The dataset is imbalanced as it contains 137 records of patients diagnosed with heart disease and 160 records of patients without absence of heart disease. The dataset contains 13 attributes and 1 decision class label with binary values 1 and 0. Presence of heart disease is indicated by 1 and absence by 0. Among the 13 attributes age are sex of the patients are not considered important since they identify personal profile of the patients and the remaining 11 are vital clinical records. The split ratio of the dataset into train and test is 70 is to 30.

The attributes present in the dataset are:

1) Age
2) Sex
3) Cp
4) Trestbps
5) Chol
6) Fbs
7) Restecg
8) Thalach
9) Exang
10) Oldpeak
11) Slope
12) Ca
13) Th

1.3 Proposed Model:

In this project a total of 7 machine learning algorithms have been implemented. They are:

1) Decision Trees
2) Language Model
3) Support Vector Machine
4) Random Forest
5) Naive Bayes
6) Neural Networks
7) K-Nearest Neighbor

And we arrived at an ensemble model HRFLM in which the outputs of the algorithms Random Forest and Linear Model have been combined to derive the final prediction. The metrics including accuracy, sensitivity, specificity, and area under the ROC curve are used to assess the ensemble model's performance.

The proposed model offers several advantages over individual machine learning algorithms, including improved prediction accuracy, reduced risk of overfitting or underfitting, and greater flexibility in handling complex and heterogeneous datasets. By providing an effective tool for identifying patients at risk for heart disease, this model can help healthcare providers to implement targeted prevention and treatment strategies and ultimately improve patient outcomes. The proposed model workflow is shown in the fig 1.2.

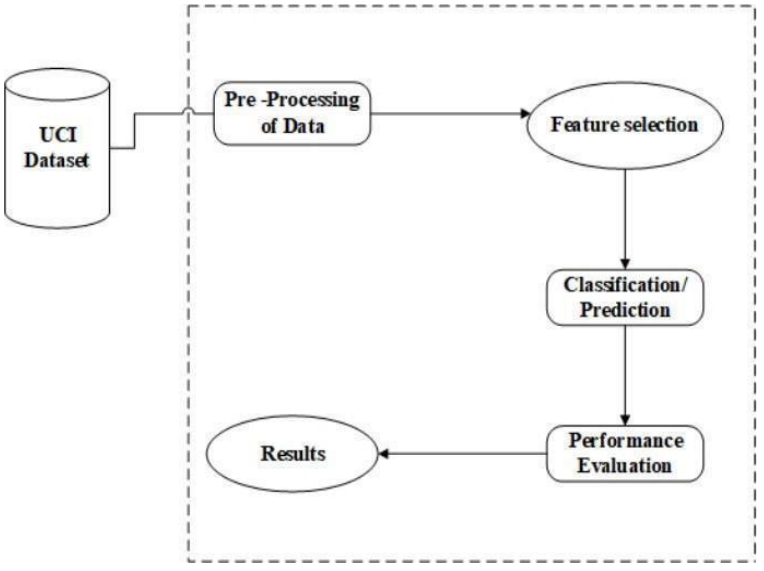

Fig 1.2 Workflow of the experiment

1.4 **Machine Learning Algorithms**:

1.4.1 Decision Tree:

Decision Tree is one of the popular supervised machine learning algorithms which is used for both classification and regression tasks. Decision tree algorithm creates a tree like structure with all the decisions and their potential outcomes. The elements contained by a Tree are nodes, edges, and leaves. Every node in the tree depicts a choice made in correspondence to a feature and threshold value, and every edge shows how the decision of the node turned out, with the possible outcomes of leading to a fresh node or a leaf. The conclusion is represented by a leaf node. Fig 1.3 describes the how decision tree works and Entropy is calculated using the formula represented by (1)

$$Entropy(s) = -P(yes)\log_2 P(yes) - P(no)\log_2 P(no) \qquad (1)$$

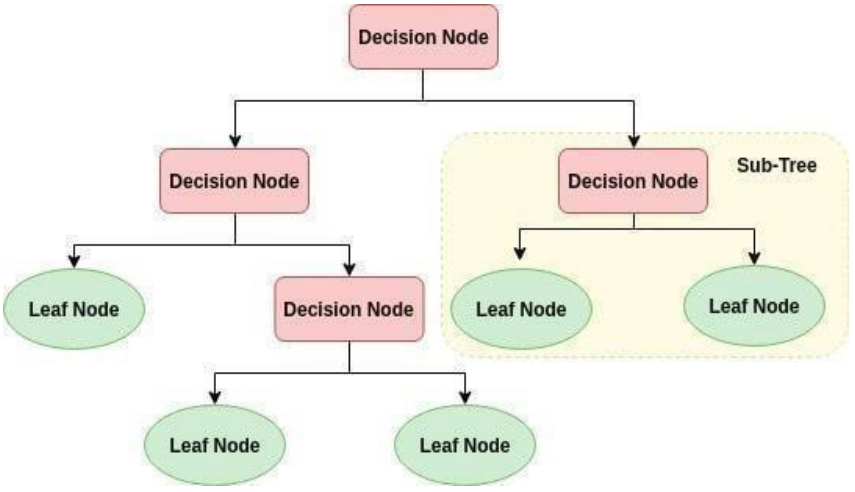

Fig 1.3 Working of Decision Tree algorithm

1.4.2 Language Model:

In binary classification problems, such as determining if a disease exists or not, the objective is to forecast the likelihood that an event will occur. One statistical technique used for these problems is logistic regression. Applying a logistic function, commonly referred to as a sigmoid function, to a linear combination of the independent variables is how the logistic regression algorithm operates. The logistic function produces a number between 0 and 1, which represents the likelihood that the event will occur. To determine the values of the model parameters that best match the data and maximize the likelihood of the observed labels, the program uses an optimization technique like maximum likelihood estimation. The maximum likelihood is found by equation (2) and Fig 1.4 shows an example of how logistic regression works with an input of audio signal.

$$log[p(X) / (1-p(X))] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \qquad (2)$$

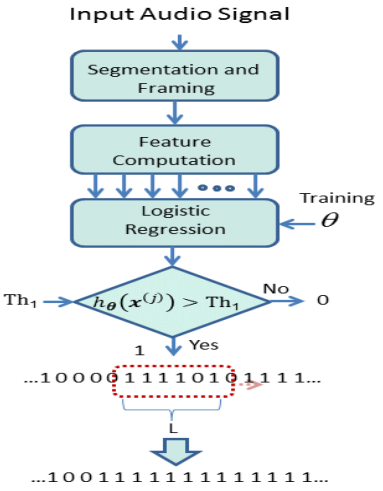

Fig 1.4 Working of Logistic Regression algorithm

1.4.3 Support Vector Machine:

SVMs are support vector machines, which are employed to solve classification and regression issues. The SVM method looks for a hyperplane that splits the data into multiple classes, maximising the distance between the closest data points from each class. The SVM algorithm selects the data points from each class that are closest to the hyperplane based on which hyperplane maximises the margin. The support vectors, or data points closest to the hyperplane, determine the hyperplane's location and direction. The SVM technique may handle both linearly separable and non-linearly separable datasets by using a kernel function to transfer the data to a higher-dimensional feature space where the data may be linearly separable. Minimizing function is calculated using equation (3) and its working is shown in fig 1.5.

$$Minimize : \phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \epsilon_i \qquad (3)$$

$$\Rightarrow Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j x_i^T x_j$$

$$where\ 0 \leq \alpha_i \leq C\ \forall\ i$$

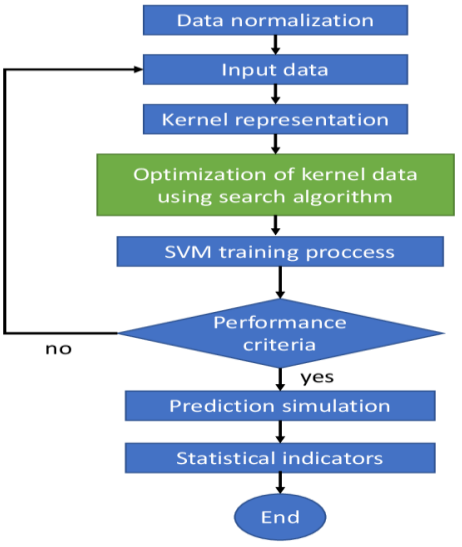

Fig 1.5 Working of Support Vector Machine algorithm

1.4.4 Random Forest:

It is a supervised Machine Learning algorithm used for both classification and regression tasks. The bootstrapping technique used by the random forest algorithm during the training phase produces several decision trees by randomly picking samples from the initial dataset. At each split, a random subset of characteristics and a portion of the samples are used to train each decision tree, reducing overfitting, and boosting tree diversity. The random forest algorithm combines the forecasts of all the decision trees in the forest to arrive at a prediction. Regression issues use the mean or median of the expected values, whereas classification problems use the mode of the predicted class labels as the final prediction. Calculation of Gini index can be done by using the equation (4) and its working mechanism is shown in fig 1.6

$$Gini\ Index = 1 - \sum_{i=1}^{n}\left(P_i\right)^2 \qquad (4)$$

$$= 1 - [(P_+)^2 + (P_-)^2]$$

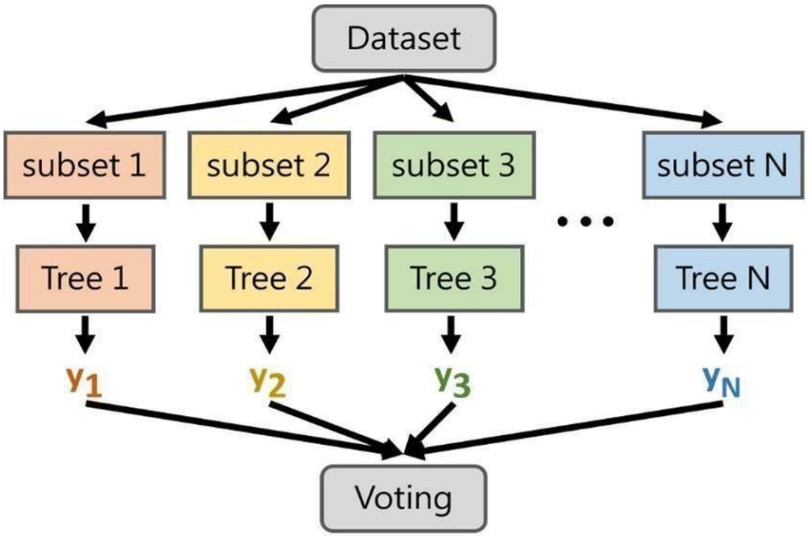

Fig 1.6: Working of Random Forest algorithm

1.4.5 Naive Bayes:

A probabilistic algorithm for categorization issues is called Naive Bayes. It is grounded on the Bayes theorem, which estimates the likelihood of an event based on knowledge of potential confounding factors in the past. By integrating the prior probability and the likelihood using Bayes' theorem, the procedure determines the posterior probability of each class given the input features and is calculated from the equation (5). The projected class for the instance is chosen based on the class with the highest posterior probability. The Naive Bayes algorithm is quick and easy to use, and it works well for text classification and spam filtering. It can manage data with several dimensions. Fig 1.7 shows the working of this technique.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \qquad (5)$$

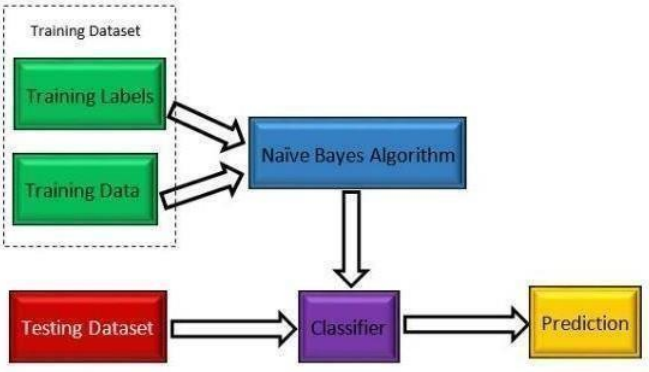

Fig 1.7: Working of Naïve Bayes algorithm

1.4.6 Neural Networks:

To learn hierarchical data representations, deep learning uses artificial neural networks with several layers. Through a succession of interconnected layers, the neural network architecture used in deep learning is intended to learn and extract features from the incoming data. Each layer is made up of a group of neurons that take input from the layer below and transform it in a non-linear way before sending it to the layer above. Large volumes of labeled data are used to train deep learning models, and during the learning process, the weights and biases of the neurons are changed to reduce the loss function, which measures the discrepancy between the expected output and the actual output. Usually, gradient descent optimization techniques are used for this, which update the weights and biases according to the gradient of the loss function with respect to the parameters. MSE is calculated from the formula mentioned in 6.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2. \quad (6)$$

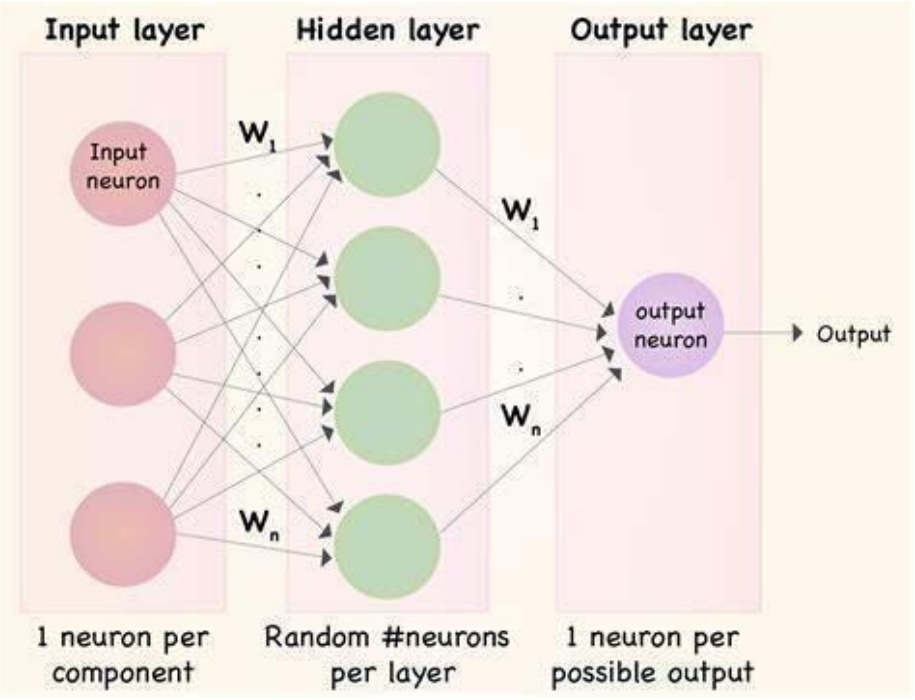

Fig 1.8 Working of Neural networks

1.4.7 K-Nearest Neighbor:

For problems involving classification and regression, K-Nearest Neighbours (KNN) is employed. By locating the K closest data points in the training set and using their class or regression values to forecast the value of the new data point, the KNN method classifies a new data point. K's value is a hyperparameter that the user selects. KNN estimates the distance between each new data point and every other data point in the training set, often using the Euclidean distance, in order to determine the K nearest neighbours. Then, based on how far they are from the newly added data point, the K nearest neighbours are chosen. For classification tasks, the method allocates the new data point to the class that appears the most frequently among the K closest neighbours. In order to forecast the value for a new data point in regression tasks, the method computes the average or weighted average of the values of the K nearest neighbours. Distance can be found using equation 7. Fig 1.9 shows the data points scattering in KNN.

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \qquad (7)$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

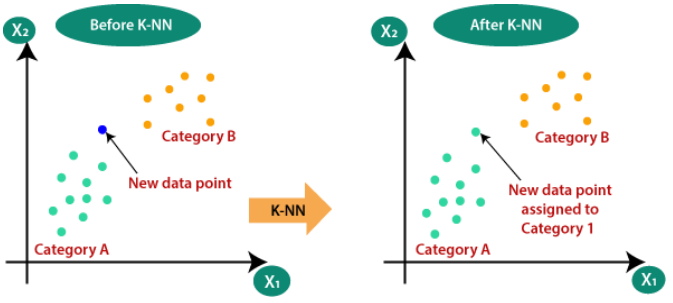

Fig 1.9 Working of K-Nearest Neighbour algorithm

CHAPTER 2

MERITS AND DEMERITS OF THE BASE PAPER

2.1 Literature Survey

REFERENCE	WORK	METHODS	LIMITATIONS
[1]	To carefully select features, train and evaluate model, and validate its accuracy before deploying it in a clinical records	Classification technique such as Decision Trees and Random Forest has been used in predicting the accuracy	There may be certain limitations to the model's ability to predict coronary heart disease effectively. For example, it might not account for genetic or individual variations in risk factors, or it might not include all aspect of the condition that influences it.
[2]	A comprehensive approach to analyse a neural network - based heart prediction system,from data collection and preprocessing to deployment	K-Nearest Neighbours Classifier, SVM, RF, GNB, MLP,Bagging, Gradient Boosting Correlation, RFE- LR, RFI-ECT, SelectKBest Rapid Minor	Rapid Minor performs better but this cannot be general rule since the performance depends on the nature of the dataset, sampling and pre-processing steps

[3]	To evaluate the better performance of neural network algorithm and to which features play a role in prediction of heart disease and using matlab for statistical calculations. Simulations were performed on the Intel Pentium 4 2.33	Error back propagation technique,multi- layer perceptron. Neuron activation functions-sigmoidal and tansig	backpropagation neural networks can be useful for predicting heart disease, they may have limitations related to data quality, overfitting, black box nature, computational complexity, and model limitations.
[4]	The creation of weighted fuzzy rules using an automated method, as well as the creation of a fuzzy rule-based decision support system. To create the weighted fuzzy rules in the first phase, we employed the mining approach, attribute selection, and attribute weightage method.	a clinical decision support system (CDSS) based on weighted fuzzy rules Neural Networks based approach	Lack of external validation and limited scope. Fuzzy rules may not be interpretable.
[5]	The suggested technique use PPCA to extract high impact characteristics. The highest covariance projection vectors are extracted using PPCA and utilised to further reduce feature dimension. Using Parallel Analysis, projection vectors are chosen.	PPCA,PA	The study makes the unfounded assumption that the various types of cardiac disease can be linearly distinguished from one another. For an appropriate diagnosis of heart disease, a more complex strategy that takes into consideration the non-linear correlations between characteristics and the diagnosis may be required.

TITLE	WORK	METHODS	LIMITATIONS
[6]	describes a method for identifying heart disease that uses test data as input and extracts a subset of reduced dimensional features.	Support vector machines (SVM), adial basis function (RBF)	The proposed method was not contrasted with other available techniques for diagnosing heart disease in the publication. The suggested strategy was not validated in the study on an external dataset, which is required to guarantee the generalizability of the results.
[7]	Based on their symptoms and risk factors, individuals with heart disease are categorised using a number of different algorithms.	Artificial neural networks(ANNs), decision trees and fuzzy logic.	The ANN component needs a lot of data to be trained. The procedure could depend on the weighting formula chosen to mix the results from the various components.

2.2 Evaluation Metrics

Algorithms	Accuracy	Precision	Sensitivity	F1-Score	Specificity
KNN	84.21	93.33	73.68	82.35	94.73
Decision tree	76.31	81.25	68.42	74.28	84.21
Random Forest	87.52	93.54	76.31	84.05	94.73
Multi-layer perceptron	77.12	84.78	73.58	78.79	93.64
SupportVectorClassifier	86.95	92.85	72.22	81.25	96.42
Naïve Bayes	86.42	87.09	81.81	84.37	90.69
Logistic Regression	85.52	93.54	76.31	84.05	94.73

The seven ml models applied assessment measures, including accuracy, precision, sensitivity, f1 score, and specificity, are listed in table 2.2. The most accurate model is the multilayer perceptron, which is closely followed by the other models. Additionally, the bar graph in Figure 2.1 was created using the data from Table 2.2

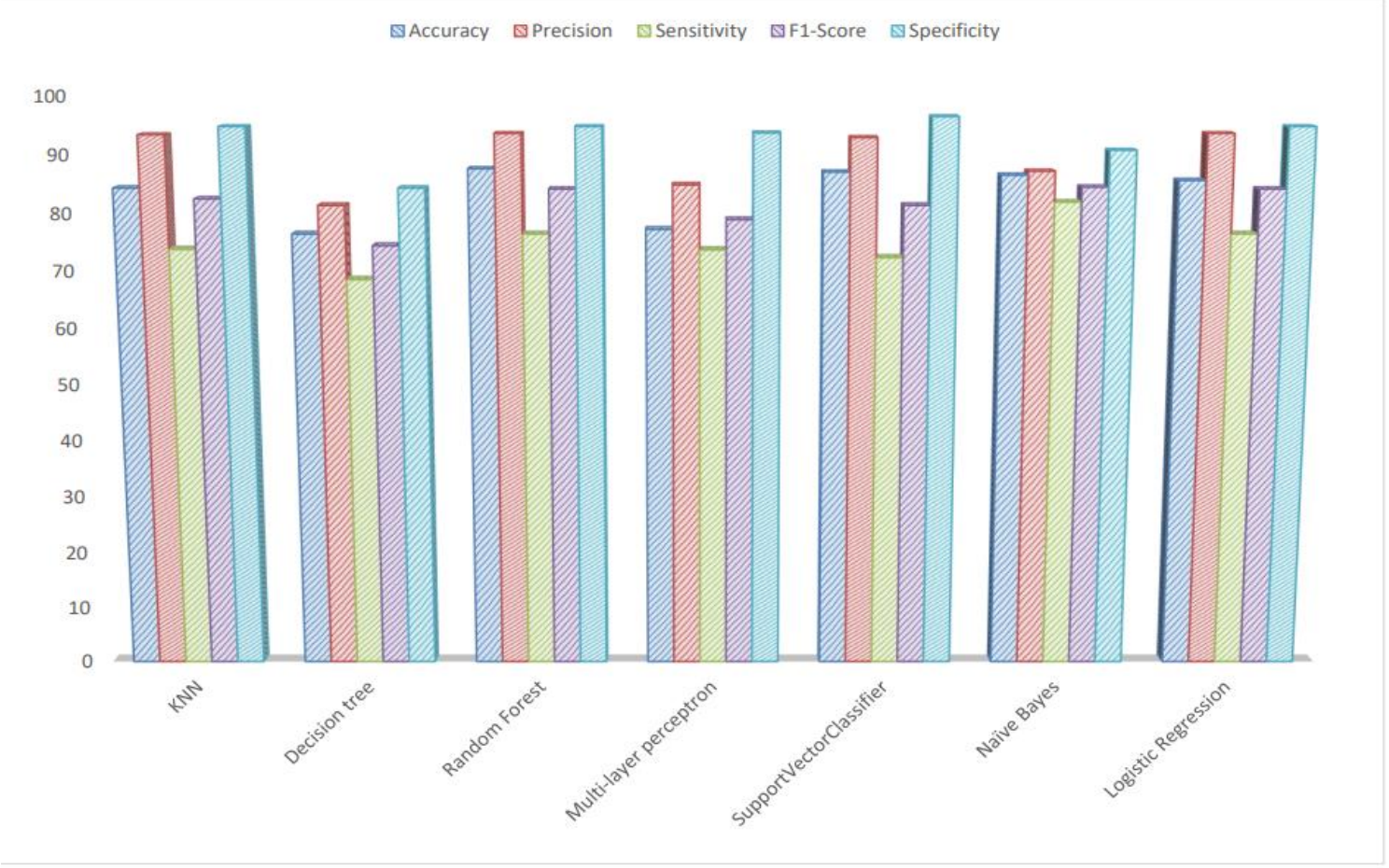

Fig 2.1 Graphical Representation of Performance Metrics

2.3 Merits

1.5 Implementation of Ensemble learning which creates multiple processes using combination of one or more models that gives accurate results

1.6 As the ensemble learning is a metadata approach for machine learning ,it gives best predictive performance by combining the predictions of multiple models.

1.7 While linear models are better suited to capture linear correlations, random forests are known for their capacity to handle complicated and nonlinear relationships in data. Combining the two may result in more accuracy than utilising either technique separately.

1.8 When compared to tree-based models like random forests, linear models are frequently easier to understand because they generate coefficients that show the degree and direction of each input variable's influence on the result. While still gaining the advantages of the complexity of the random forest, hybrid models can retain some of this interpretability.

1.9 Although linear models may be considerably more effective in this situation, random forests may manage missing data to some extent. Combining the two may result in a more reliable model that is less vulnerable to missing data.

1.10 Overall, using a hybrid random forest-linear technique can be computationally effective, balance interpretability and accuracy, and handle missing data well. However, the precise benefits will rely on the available data and the issue at hand, therefore the technique selection should always be based on empirical analysis.

2.4 Demerits

1) Hybrid methods can be more complex than individual methods, as they require combining and integrating different models and algorithms. This complexity can make the model more difficult to understand, interpret, and implement.

2) A hybrid model can still be prone to overfitting, just like any other machine learning model. The complexity of a hybrid model can make it more susceptible to overfitting if not properly regularized or validated**.**

3) The hybrid method involves multiple models and algorithms, each with its own set of hyperparameters. This can make hyperparameter tuning more challenging and time-consuming, requiring careful selection and optimization of hyperparameters for each component model.

CHAPTER 3

3.1 Source Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('cleve.csv')
X = dataset.iloc[:,:-1].values
y = dataset.iloc[:, 13].values

from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan ,strategy='mean')
imputer=imputer.fit(X[:,11:13])
X[:,11:13]=imputer.transform(X[:,11:13])

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.25, random_state = 42)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

dataset.num.value_counts()

from sklearn.neural_network import MLPClassifier
classifier =
MLPClassifier(hidden_layer_sizes=(8,8,8),activation='logistic',solver=
'adam',max_iter=500)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
from sklearn.metrics import plot_confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plot_confusion_matrix(classifier, X_test, y_test, cmap=plt.cm.Blues)
plt.title('Confusion matrix FOR DEEP LEARNING')
plt.show()
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
```

```
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from imblearn.metrics import specificity_score
precision = precision_score(y_test, y_pred)
print('Precision: %f' % precision)
recall = recall_score(y_test,y_pred)
print('Recall: %f' % recall)
f1 = f1_score(y_test, y_pred)
print('F1 score: %f' % f1)
accuracy_score(y_test,y_pred)
print('Acuuracy:',accuracy_score(y_test,y_pred))
s1=specificity_score(y_test,y_pred)
print('Specificity score: %f' % s1)

from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, classifier.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test,
classifier.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='MLP (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```

```
from sklearn.linear_model import LogisticRegression
classifier =LogisticRegression()
classifier.fit(X_train,Y_train)
```

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(X_train, y_train)
```

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

```
from sklearn.ensemble import RandomForestClassifier
classifier =RandomForestClassifier(n_estimators=20)
classifier.fit(X_train, y_train)
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

classifier = DecisionTreeClassifier(criterion = 'entropy',
random_state = 8)
classifier.fit(X_train, y_train)
```

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0
,probability=True)
classifier.fit(X_train, Y_train)
```

```
import numpy as np
from copy import deepcopy

class ModelTree(object):

    def __init__(self, model, max_depth=5, min_samples_leaf=10):

        self.model = model
        self.max_depth = max_depth
        self.min_samples_leaf = min_samples_leaf
        self.tree = None



    def fit(self, X, y, verbose=False):

        # Settings
        model = self.model
        min_samples_leaf = self.min_samples_leaf
        max_depth = self.max_depth


        if verbose:
            print(" max_depth={},
min_samples_leaf={}...".format(max_depth, min_samples_leaf))

        def _build_tree(X, y):

            global index_node_global

            def _create_node(X, y, depth, container):
                loss_node, model_node = _fit_model(X, y, model)
                node = {"name": "node",
                        "index": container["index_node_global"],
                        "loss": loss_node,
                        "model": model_node,
                        "data": (X, y),
                        "n_samples": len(X),
                        "j_feature": None,
                        "threshold": None,
                        "children": {"left": None, "right": None},
                        "depth": depth}
```

```
                container["index_node_global"] += 1
                return node

            # Recursively split node + traverse node until a terminal
node is reached
            def _split_traverse_node(node, container):

                # Perform split and collect result
                result = _splitter(node, model,
max_depth=max_depth,min_samples_leaf=min_samples_leaf)

                # Return terminal node if split is not advised
                if not result["did_split"]:
                    if verbose:
                        depth_spacing_str = " ".join([" "] *
node["depth"])
                        print(" {}*leaf {} @ depth {}: loss={:.6f},
N={}".format(depth_spacing_str, node["index"], node["depth"],
node["loss"], result["N"]))
                    return

                # Update node information based on splitting result
                node["j_feature"] = result["j_feature"]
                node["threshold"] = result["threshold"]
                del node["data"]  # delete node stored data

                # Extract splitting results
                (X_left, y_left), (X_right, y_right) = result["data"]
                model_left, model_right = result["models"]

                # Report created node to user
                if verbose:
                    depth_spacing_str = " ".join([" "] *
node["depth"])
                    print(" {}node {} @ depth {}: loss={:.6f},
j_feature={}, threshold={:.6f}, N=({},{})".format(depth_spacing_str,
node["index"], node["depth"], node["loss"], node["j_feature"],
node["threshold"], len(X_left), len(X_right)))

                # Create children nodes
                node["children"]["left"] = _create_node(X_left,
y_left, node["depth"]+1, container)
```

```
                node["children"]["right"] = _create_node(X_right,
y_right, node["depth"]+1, container)
                node["children"]["left"]["model"] = model_left
                node["children"]["right"]["model"] = model_right

                # Split nodes
                _split_traverse_node(node["children"]["left"],
container)
                _split_traverse_node(node["children"]["right"],
container)

            container = {"index_node_global": 0}  # mutatable
container
            root = _create_node(X, y, 0, container)  # depth 0 root
node
            _split_traverse_node(root, container)  # split and
traverse root node

            return root

        # Construct tree
        self.tree = _build_tree(X, y)
        return self.tree

    # ====================
    # Predict
    # ====================
    def predict(self, X):
        assert self.tree is not None
        def _predict(node, x):
            no_children = node["children"]["left"] is None and \
                          node["children"]["right"] is None
            if no_children:
                y_pred_x = node["model"].predict([x])[0]
                return y_pred_x
            else:
                if x[node["j_feature"]] <= node["threshold"]:  # x[j]
< threshold
                    return _predict(node["children"]["left"], x)
                else:  # x[j] > threshold
                    return _predict(node["children"]["right"], x)
        y_pred = np.array([_predict(self.tree, x) for x in X])
```

```
        return y_pred



    # ====================
    # Loss
    # ====================
    def loss(self, X, y, y_pred):
        loss = self.model.loss(X, y, y_pred)
        return loss

def _splitter(node, model,max_depth=5, min_samples_leaf=10):

    # Extract data
    X, y = node["data"]
    depth = node["depth"]
    N, d = X.shape

    # Find feature splits that might improve loss
    did_split = False
    loss_best = node["loss"]
    data_best = None
    models_best = None
    j_feature_best = None
    threshold_best = None

    # Perform threshold split search only if node has not hit max
depth
    if (depth >= 0) and (depth < max_depth):

        for j_feature in range(d):



            threshold_search = []

            for i in range(N):
                threshold_search.append(X[i, j_feature])


            # Perform threshold split search on j_feature
            for threshold in threshold_search:
```

```
                # Split data based on threshold
                (X_left, y_left), (X_right, y_right) =
_split_data(j_feature, threshold, X, y)
                N_left, N_right = len(X_left), len(X_right)

                # Splitting conditions
                split_conditions = [N_left >= min_samples_leaf,
                                    N_right >= min_samples_leaf]

                # Do not attempt to split if split conditions not
satisfied
                if not all(split_conditions):
                    continue

                # Compute weight loss function
                loss_left, model_left = _fit_model(X_left, y_left,
model)
                loss_right, model_right = _fit_model(X_right, y_right,
model)
                loss_split = (N_left*loss_left + N_right*loss_right) /
N

                # Update best parameters if loss is lower
                if loss_split < loss_best:
                    did_split = True
                    loss_best = loss_split
                    models_best = [model_left, model_right]
                    data_best = [(X_left, y_left), (X_right, y_right)]
                    j_feature_best = j_feature
                    threshold_best = threshold

    # Return the best result
    result = {"did_split": did_split,
              "loss": loss_best,
              "models": models_best,
              "data": data_best,
              "j_feature": j_feature_best,
              "threshold": threshold_best,
              "N": N}

    return result
```

```
def _fit_model(X, y, model):
    model_copy = deepcopy(model)  # must deepcopy the model!
    model_copy.fit(X,y)
    y_pred = model_copy.predict(X)
    loss = model_copy.loss(X, y, y_pred)
    assert loss >= 0.0
    return loss, model_copy

def _split_data(j_feature, threshold, X, y):
    idx_left = np.where(X[:, j_feature] <= threshold)[0]
    idx_right = np.delete(np.arange(0, len(X)), idx_left)

    assert len(idx_left) + len(idx_right) == len(X)
    return (X[idx_left], y[idx_left]), (X[idx_right], y[idx_right])
```

```
import matplotlib.pyplot as plt
import pandas as pd
from ModelTree import ModelTree
import numpy as np
from sklearn.metrics import mean_squared_error

class logistic_regr:

    def __init__(self):
        from sklearn.linear_model import LogisticRegression
        self.model =
LogisticRegression(penalty="l2",solver='liblinear')
        self.flag = False
        self.flag_y_pred = None

    def fit(self, X, y):
        y_unique = list(set(y))
        if len(y_unique) == 1:
            self.flag = True
            self.flag_y_pred = y_unique[0]
        else:
            self.model.fit(X, y)

    def predict(self, X):
```

```
        if self.flag:
            return self.flag_y_pred * np.ones((len(X),), dtype=int)
        else:
            return self.model.predict(X)

    def loss(self, X, y, y_pred):
        return mean_squared_error(y, y_pred)

    def predict_proba(self,X):
        return self.model.predict_proba(X)
dataset = pd.read_csv('cleve.csv')
X = dataset.iloc[:,:-1].values
y = dataset.iloc[:, 13].values

from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan ,strategy='mean')
imputer=imputer.fit(X[:,11:13])
X[:,11:13]=imputer.transform(X[:,11:13])
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.25, random_state =9)


from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)


dataset.num.value_counts()
esitmators=5
y_pred=[]
n_train_split=int(len(X_train)/esitmators)
inital_train=0
final_train=0


yy_pred=[]
classifier=None

for i in range(1,esitmators+1):

    classifier =logistic_regr()
```

```
    final_train=i*n_train_split
    temp_X_train=X_train[inital_train:final_train]
    temp_y_train=y_train[inital_train:final_train]


    L=ModelTree(classifier,max_depth=20, min_samples_leaf=10)

    node=L.fit(temp_X_train,temp_y_train,verbose=False)
    classifier=node["model"]

    y_pred_temp=L.predict(X_test)
    yy_pred.append(y_pred_temp)

for j in range(len(yy_pred[0])):
    curr=[]
    for  i  in  range(len(yy_pred)):
        curr.append(yy_pred[i][j])
    a=curr.count(0)
    b=curr.count(1)
    if a>b:
        y_pred.append(0)
    else:
        y_pred.append(1)


from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)


from sklearn.metrics import classification_report
print(classification_report(y_test,  y_pred))
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, y_pred)
fpr, tpr, thresholds = roc_curve(y_test,
classifier.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='HRFLM (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
```

```
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.show()
```

CHAPTER 4

4.1 SNAPSHOTS

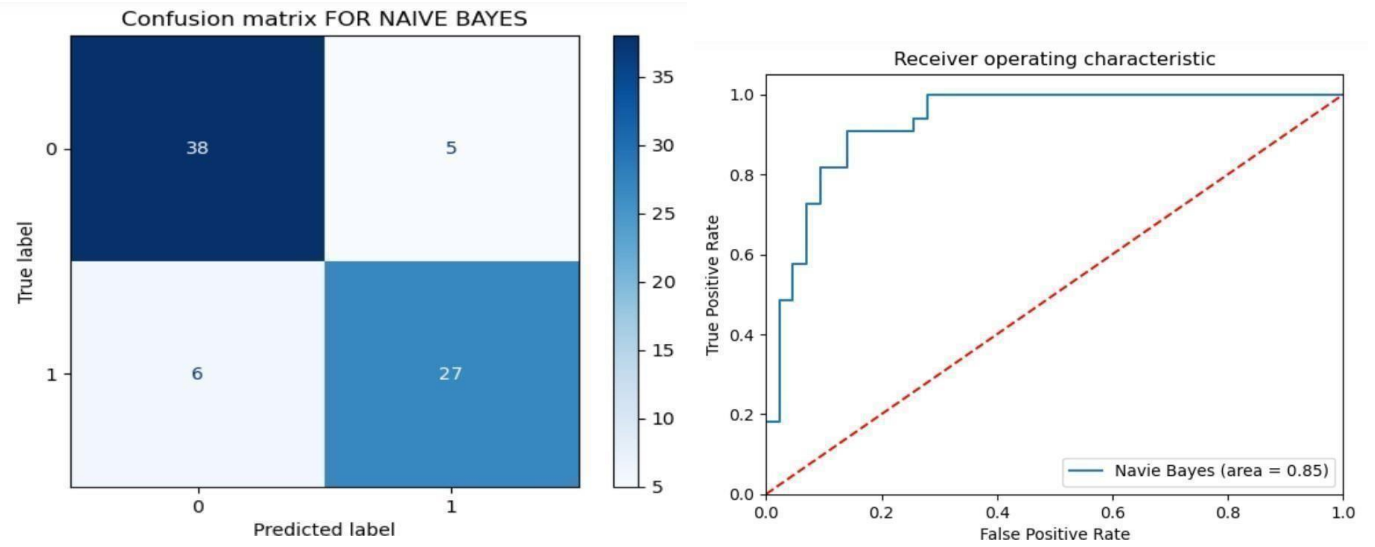

Fig 4.1 Confusion matrix and roc curve for naïve bayes

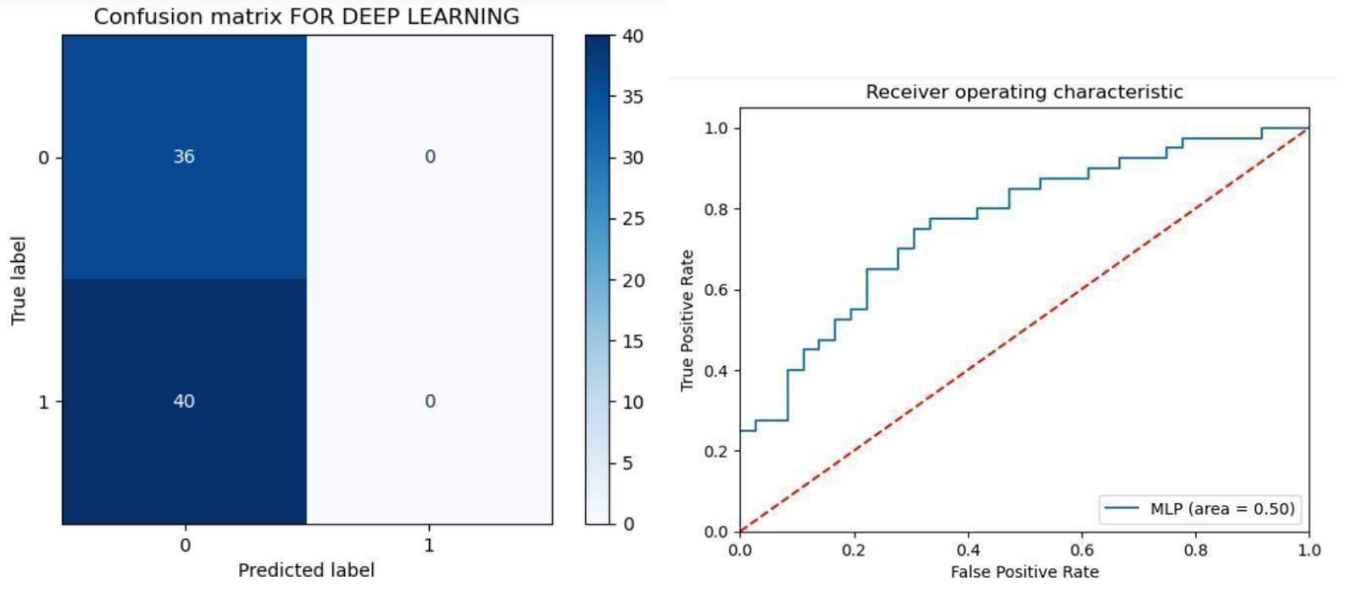

Fig 4.2 Confusion matrix and roc curve for multi-layer perceptron

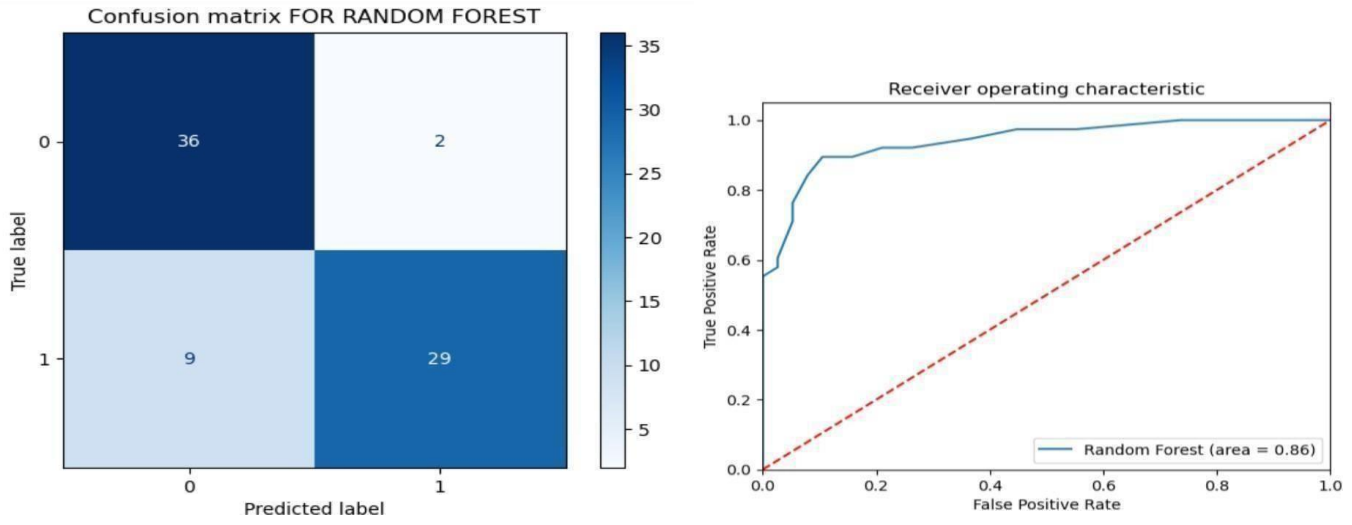

Fig 4.3 Confusion matrix and roc curve for random forest

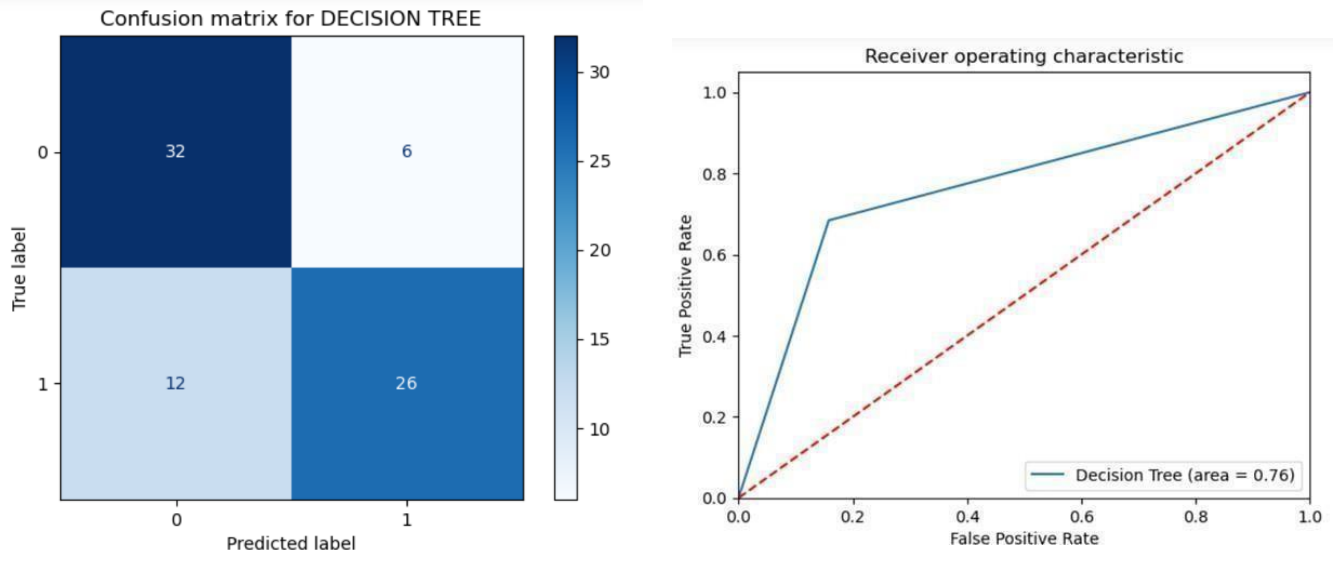

Fig 4.4 Confusion matrix and roc curve for decision tree

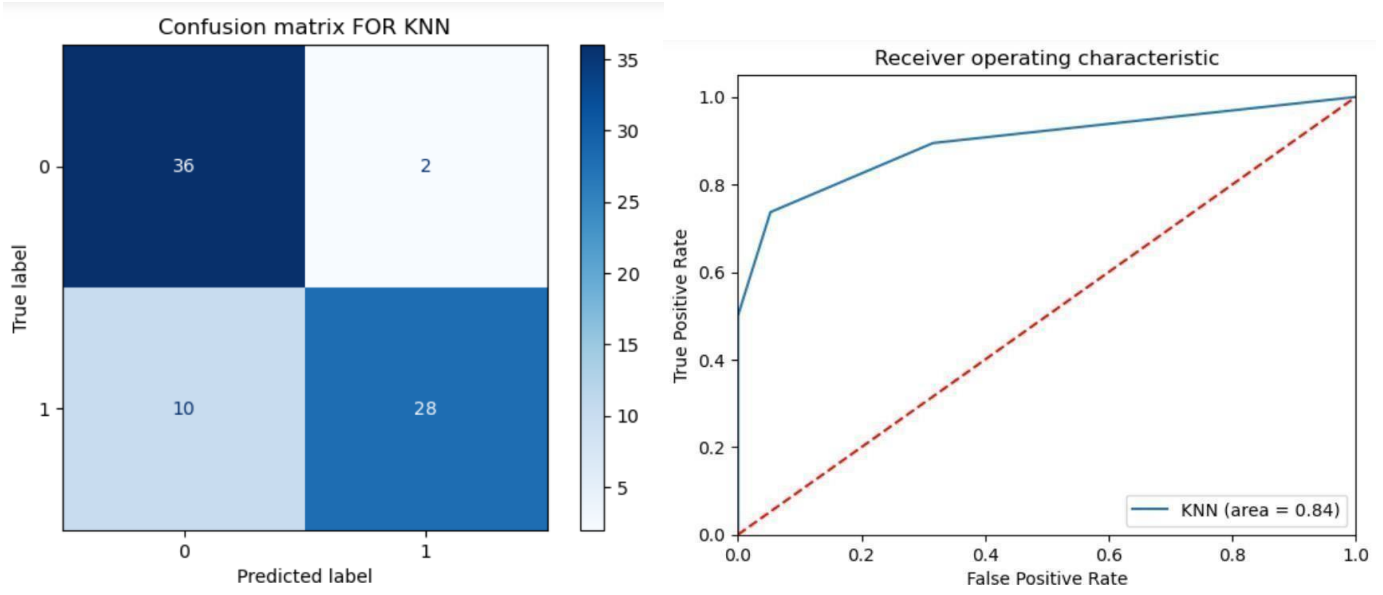

Fig 4.5 Confusion matrix and roc curve for knn

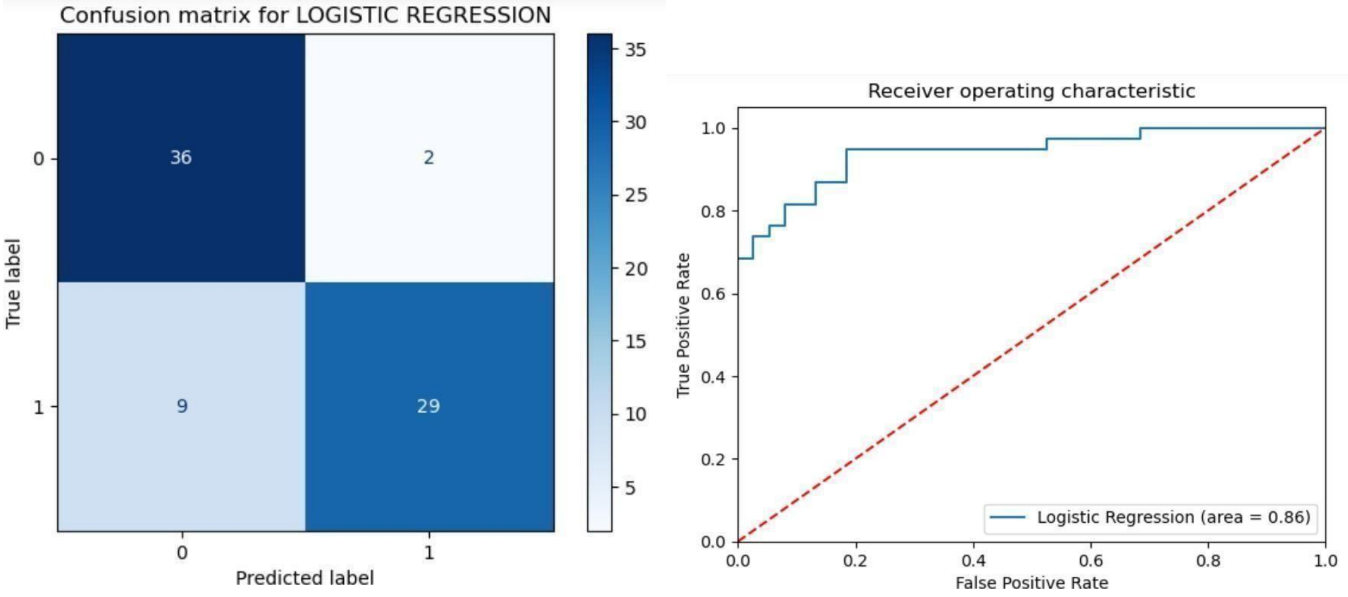

Fig 4.6 Confusion matrix and roc curve for logistic regression

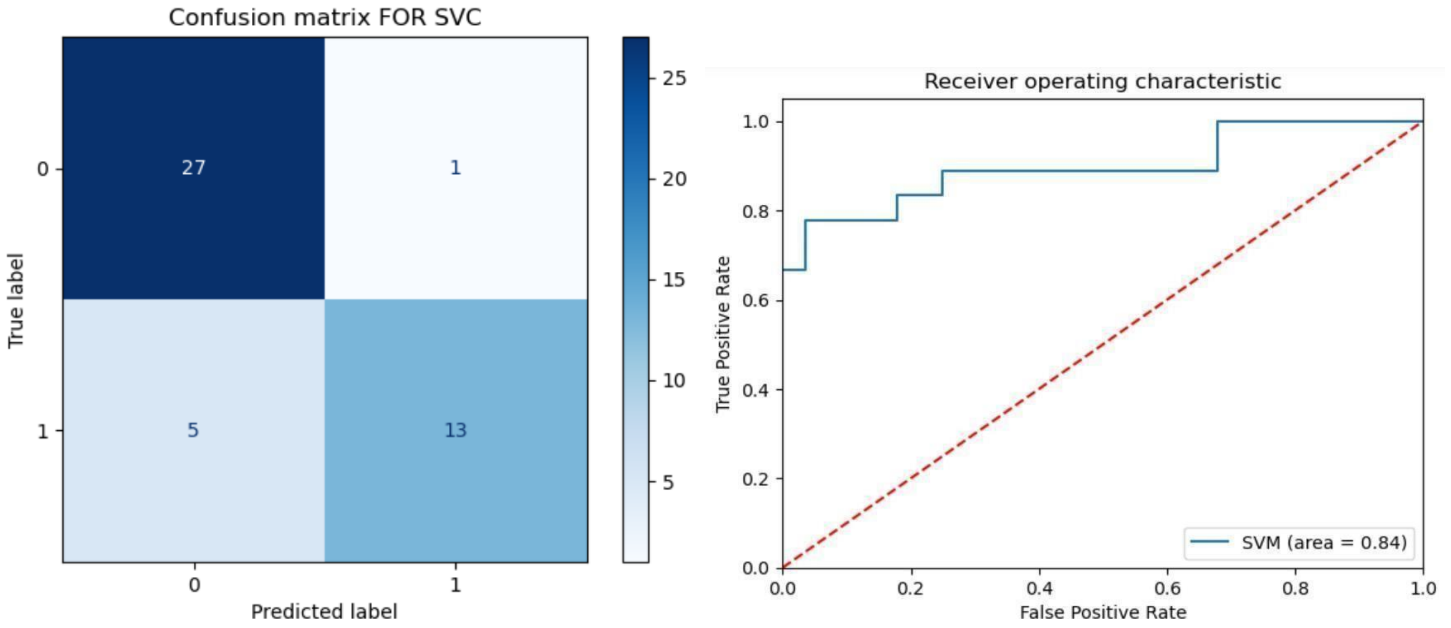

Fig 4.7 Confusion matrix and roc curve for svc

The confusion matrix and ROC curve of the employed algorithms are shown in figure no 4.1 to 4.7. True positives, false positives, true negatives, and false negatives are all entries in the confusion matrix. The roc curve is one of the greatest ways to show sensitivity and specificity, and the receiver operating characteristic is a plot between true positives and false positives. Fig 4.8 and Fig 4.9 represent the performance and roc curve of the HRFLM model

	precision	recall	f1-score	support
0	0.91	0.87	0.89	47
1	0.81	0.86	0.83	29
accuracy			0.87	76
macro avg	0.86	0.87	0.86	76
weighted avg	0.87	0.87	0.87	76

Fig 4.8 Classification report for HRFLM

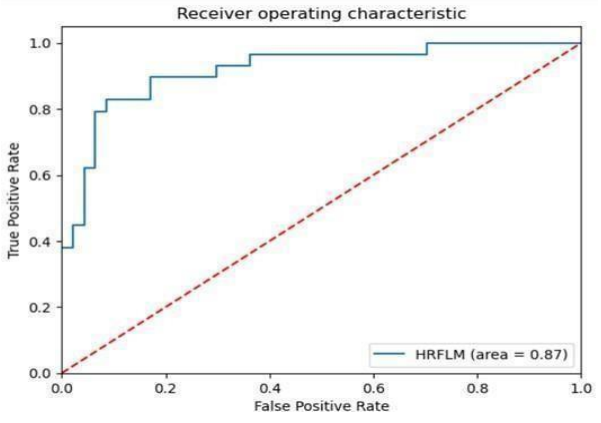

Fig 4.9 Roc curve for HRFLM

4.2 CONCLUSIONS & FUTURE PLANS

The identification of the processing of raw healthcare data of cardiac information would enable the long-term preservation of human life and the early diagnosis of anomalies in heart diseases. Machine learning techniques were used in this study to analyse the raw data and provide a new and unique perspective on cardiac disease. In the medical field, heart disease prediction is challenging and important. However, the death rate can be considerably decreased if the disease is identified in its early stages and preventative measures are put in place as soon as is practical. Further improvement of this study is highly desirable in order to concentrate the research on real-world datasets rather than merely theoretical frameworks and simulations.

The Linear Method (LM) and Random Forest (RF) approaches are combined in the proposed hybrid HRFLM strategy to maximise their benefits. A pretty good prediction of heart disease using HRFLM was found. Various combinations of machine learning techniques can be used to enhance prediction approaches in the research's future path. New feature selection techniques can also be created to get a better understanding of the important aspects in order to increase the accuracy of heart disease prediction.

The hybrid machine learning model may be compared to other machine learning models like support vector machines (SVMs), random forests, and decision trees to see which model is more effective in predicting cardiac illness. Investigating the model's explicability and interpretability could lead to further research in this area. This could boost confidence in the model by enabling medical practitioners to comprehend how it makes predictions. By examining various hybrid machine learning methods, including deep learning and reinforcement learning, the study may be furthered.

4.3 REFERENCES

1) S. Abdullah and R. R. Rajalaxmi, ‘‘A data mining model for predicting the coronary heart disease using random forest classifier,’’ in Proc. Int. Conf. Recent Trends Comput. Methods, Commun. Controls, Apr. 2012, pp. 22–25
2) A. Devi, S. P. Rajamhoana, K. Umamaheswari, R. Kiruba, K. Karunya, and R. Deepika, ‘‘Analysis of neural networks based heart disease prediction system,’’ in Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI), Gdansk, Poland, Jul. 2018, pp. 233–239.
3) N. Al-milli, ‘‘Backpropogation neural network for prediction of heart disease,’’ J. Theor. Appl.Inf. Technol., vol. 56, no. 1, pp. 131–135, 2013.
4) P. K. Anooj, ‘‘Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules,’’ J. King Saud Univ.-Comput. Inf. Sci., vol. 24, no. 1, pp. 27–40, Jan. 2012. doi: 10.1016/j.jksuci.2011.09.002.
5) S. M. S. Shah, S. Batool, I. Khan, M. U. Ashraf, S. H. Abbas, and S. A. Hussain, ‘‘Feature extraction through parallel probabilistic principal component analysis for heart disease diagnosis,’’ Phys. A, Stat. Mech. Appl., vol. 482, pp. 796–807, 2017. doi: 10.1016/j.physa.2017.04.113
6) S. M. S. Shah, S. Batool, I. Khan, M. U. Ashraf, S. H. Abbas, and S. A. Hussain, ‘‘Feature extraction through parallel probabilistic principal component analysis for heart disease diagnosis,’’ Phys. A, Stat. Mech. Appl., vol. 482, pp. 796–807, 2017. doi: 10.1016/j.physa.2017.04.113.
7) Y. E. Shao, C.-D. Hou, and C.-C. Chiu, ‘‘Hybrid intelligent modeling schemes for heart disease classification,’’ Appl. Soft Comput. J., vol. 14, pp. 47–52, Jan. 2014. doi: 10.1016/j.asoc.2013.09.020

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON SMART CACHING, COMMUNICATIONS, COMPUTING AND CYBERSECURITY FOR INFORMATION-CENTRIC INTERNET OF THINGS

Received May 13, 2019, accepted June 9, 2019, date of publication June 19, 2019, date of current version July 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923707

Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques

SENTHILKUMAR MOHAN[1], CHANDRASEGAR THIRUMALAI[1], AND GAUTAM SRIVASTAVA [2,3], (Member, IEEE)

[1] School of Information Technology and Engineering, VIT University, Vellore 632015, India
[2] Department of Mathematics and Computer Science, Brandon University, Brandon, MB R7A 6A9, Canada
[3] Research Center for Interneural Computing, China Medical University, Taichung 40402, Taiwan

Corresponding authors: Senthilkumar Mohan (senthilkumar.mohan@vit.ac.in) and Gautam Srivastava (srivastavag@brandonu.ca)

This work was supported by the China Medical University.

ABSTRACT Heart disease is one of the most significant causes of mortality in the world today. Prediction of cardiovascular disease is a critical challenge in the area of clinical data analysis. Machine learning (ML) has been shown to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry. We have also seen ML techniques being used in recent developments in different areas of the Internet of Things (IoT). Various studies give only a glimpse into predicting heart disease with ML techniques. In this paper, we propose a novel method that aims at finding significant features by applying machine learning techniques resulting in improving the accuracy in the prediction of cardiovascular disease. The prediction model is introduced with different combinations of features and several known classification techniques. We produce an enhanced performance level with an accuracy level of 88.7% through the prediction model for heart disease with the hybrid random forest with a linear model (HRFLM).

INDEX TERMS Machine learning, heart disease prediction, feature selection, prediction model, classification algorithms, cardiovascular disease (CVD).

I. INTRODUCTION

It is difficult to identify heart disease because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, abnormal pulse rate and many other factors. Various techniques in data mining and neural networks have been employed to find out the severity of heart disease among humans. The severity of the disease is classified based on various methods like *K* -Nearest Neighbor Algorithm (KNN), Decision Trees (DT), Genetic algorithm (GA), and Naive Bayes (NB) [11], [13]. The nature of heart disease is complex and hence, the disease must be handled carefully. Not doing so may affect the heart or cause premature death. The perspective of medical science and data mining are used for discovering various sorts of metabolic syndromes. Data mining with classification plays a significant role in the prediction of heart disease and data investigation.

We have also seen decision trees be used in predict- ing the accuracy of events related to heart disease [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Jun Wu.

Various methods have been used for knowledge abstraction by using known methods of data mining for prediction of heart disease. In this work, numerous readings have been carried out to produce a prediction model using not only distinct techniques but also by relating two or more techniques. These amalgamated new techniques are commonly known as hybrid methods [14]. We introduce neural networks using heart rate time series. This method uses various clinical records for prediction such as Left bundle branch block (LBBB), Right bundle branch block (RBBB), Atrial fibrillation (AFIB), Normal Sinus Rhythm (NSR), Sinus bradycardia (SBR), Atrial flutter (AFL), Premature Ventricular Contraction (PVC)), and Second degree block (BII) to find out the exact condition of the patient in relation to heart disease. The dataset with a radial basis function network (RBFN) is used for classification, where 70% of the data is used for training and the remaining 30% is used for classification [4], [15].

We also introduce Computer Aided Decision Support System (CADSS) in the field of medicine and research. In previous work, the usage of data mining techniques in the healthcare industry has been shown to take less time for

the prediction of disease with more accurate results [16]. We propose the diagnosis of heart disease using the GA. This method uses effective association rules inferred with the GA for tournament selection, crossover and the mutation which results in the new proposed fitness function. For experimental validation, we use the well-known Cleveland dataset which is collected from a UCI machine learning repository. We will see later on how our results prove to be prominent when compared to some of the known supervised learning techniques [5], [17]. The most powerful evolutionary algorithm Particle Swarm Optimization (PSO) is introduced and some rules are generated for heart disease. The rules have been applied randomly with encoding techniques which result in improvement of the accuracy overall [2]. Heart disease is predicted based on symptoms namely, pulse rate, sex, age, and many others. The ML algorithm with Neural Networks is introduced, whose results are more accurate and reliable as we have seen in [8], [12].

Neural networks are generally regarded as the best tool for prediction of diseases like heart disease and brain disease. The proposed method which we use has 13 attributes for heart disease prediction. The results show an enhanced level of performance compared to the existing methods in works like [3]. The Carotid Artery Stenting (CAS) has also become a prevalent treatment mode in the medical field during these recent years. The CAS prompts the occurrence of major adverse cardiovascular events (MACE) of heart disease patients that are elderly. Their evaluation becomes very important. We generate results using a Artificial Neural Network ANN, which produces good performance in the prediction of heart disease [6], [18]. Neural network methods are introduced, which combine not only posterior probabilities but also predicted values from multiple predecessor techniques. This model achieves an accuracy level of up to 89.01% which is a strong results compared to previous works. For all experiments, the Cleveland heart dataset is used with a Neural Network NN to improve the performance of heart disease as we have seen previously in [9], [19].

We have also seen recent developments in machine learning ML techniques used for Internet of Things (IoT) as well [43]. ML algorithms on network traffic data has been shown to provide accurate identification of IoT devices con- nected to a network. Meidan *et al.* collected and labeled network traffic data from nine distinct IoT devices, PCs and smartphones. Using supervised learning, they trained a multi-stage meta classifier. In the first stage, the classi- fier can distinguish between traffic generated by IoT andnon-IoT devices. In the second stage, each IoT device is associated with a specific IoT device class. Deep learning is a promising approach for extracting accurate information from raw sensor data from IoT devices deployed in complex environments [44]–[47]. Because of its multilayer structure, deep learning is also appropriate for the edge computing environment [48], [49].

In this work, we introduce a technique we call the Hybrid Random Forest with Linear Model (HRFLM). The main objective of this research is to improve the performance accuracy of heart disease prediction. Many studies have been conducted that results in restrictions of feature selection for algorithmic use. In contrast, the HRFLM method uses all features without any restrictions of feature selection. Here we conduct experiments used to identify the features of a machine learning algorithm with a hybrid method. The experiment results show that our proposed hybrid method has stronger capability to predict heart disease compared to existing methods.

The rest of the paper is organized as follows, SectionIIdiscusses heart related works, existing methods and techniques available. We also provide an overview of our results in SectionIII. SectionIVdiscusses HRFLM Data pre-processing followed by feature selection, classification modeling and performance measure. SectionVgives the algorithms used and the experimental setup. SectionVIshows the evaluation of datasets and experimental setup. It also shows how the experiment was conducted and the results that were achieved. SectionVIIcontains a discussion about the HRFLM method results and benchmarking of the proposed model. Finally, SectionVIIIends with a conclusion of current work and some notes on future enhancement.

II. RELATED WORK

There is ample related work in the fields directly related to this paper. ANN has been introduced to produce the high- est accuracy prediction in the medical field [6]. The back propagation multilayer perception (MLP) of ANN is used to predict heart disease. The obtained results are compared with the results of existing models within the same domain and found to be improved [10]. The data of heart disease patients collected from the UCI laboratory is used to discover patterns with NN, DT, Support Vector machines SVM, and Naive Bayes. The results are compared for performance and accuracy with these algorithms. The proposed hybrid method returns results of 86.8% for F -measure, competing with the other existing methods [7]. The classification without segmentation of Convolutional Neural Networks (CNN) is introduced. This method considers the heart cycles with various start positions from the Electrocardiogram (ECG) signals in the training phase. CNN is able to generate features with various positions in the testing phase of the patient [22], [41]. A large amount of data generated by the medical industry has not been used effectively previously. The new approaches presented here decrease the cost and improve the prediction of heart disease in an easy and effective way. The various different research techniques considered in this work for prediction and classification of heart disease using ML and deep learning (DL) techniques are highly accurate in establishing the efficacy of these methods [27], [42].

III. OVERVIEW OF METHOD ANDRESULTS

In HRFLM, we use a computational approach with the three association rules of mining namely, apriori, predictive and Tertius to find the factors of heart disease on the UCI

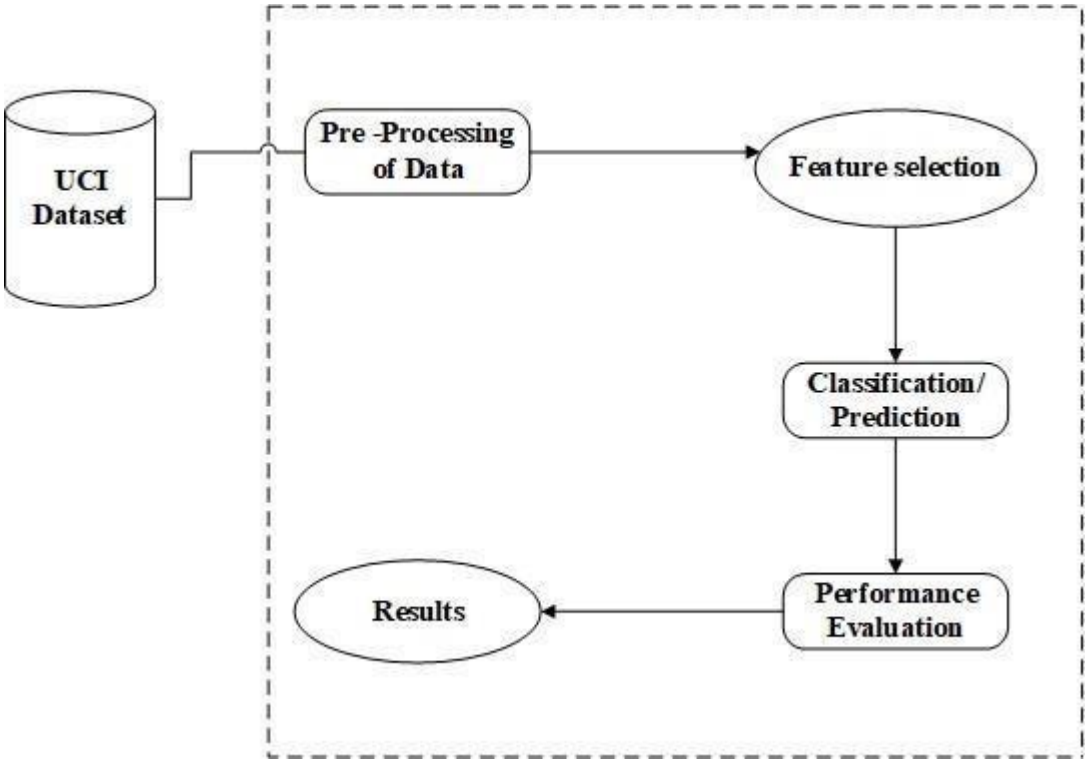

FIGURE 1. Experiment workflow with UCI dataset.

Cleveland dataset. The available information points to the deduction that females have less of a chance for heart disease compared to males. In heart diseases, accurate diagnosis is primary. But, the traditional approaches are inadequate for accurate prediction and diagnosis.

HRFLM makes use of ANN with back propagation along with 13 clinical features as the input. The obtained results are comparatively analyzed against traditional methods [20], [23]. The risk levels become very high and a number of attributes are used for accuracy in the diagnosis of the disease [24]. The nature and complexity of heart disease require an efficacious treatment plan. Data mining methods help in remedial situations in the medical field. The data mining methods are further used considering DT, NN, SVM, and KNN. Among several employed methods, the results from SVM prove to be useful in enhancing accuracy in the prediction of disease [25]. The nonlinear method with a module for monitoring heart function is introduced to detect the arrhythmias like bradycardia, tachycardia, atrial, atrial-ventricular flutters, and many others. The performance efficacy of this method can be estimated from the accuracy in the outcome results based on ECG data. ANN training is used for the accurate diagnosis of disease and the prediction of possible abnormalities in the patient [26], [34].

Diverse data mining approaches and prediction methods, such as KNN, LR, SVM, NN, and Votehave been rather popular lately to identify and predict heart disease [23]. The novel method Vote in conjunction with a hybrid approach using LR and NB is proposed in this paper. The UCI dataset is used for conducting the experiments of the proposed method, which resulted in 87.4% accuracies in the prediction of heart disease [28], [36]. The Probabilistic Principal Component Analysis (PPCA) method is proposed for evaluation, based on three data sets of Cleveland, Switzerland, and Hungarian in UCI respectively. The method extracts the vectors with high covariance and vector projection used for minimiz- ing the feature dimension. The feature selection with minimizing dimension is provided to a radial basis function, which supports kernel-based SVM. The results of the methods are 82.18%, 85.82% and 91.30% of UCI data sets of Cleveland, Switzerland and Hungarian respectively [29]. The hybrid method combining Linear regression (LR), Multivariate Adaptive Regression Splines (MARS) and ANN is introduced with rough set techniques and is the main novel contribution of this paper. The proposed method effectively reduced the set of critical attributes. The remainingattributes are input for ANN subsequently. The heart disease datasets are used to demonstrate the efficacy of the development of the hybrid approach [30], [38]. The heart disease prediction with multilayer perception of NN is proposed. This method uses 13 clinical attribute features as the input and trained by back propagation are very accurate results in identifying whether the patient has heart disease or not [39].

We also introduce the Apriori algorithm with SVM and compare it with nine other classification methods to predict heart disease more accurately. The results of the classification

method have proved a higher degree of accuracy and performance in the prediction of heart disease compared to the other existing methods [32]. The feature selection plays a prominent role in the prediction of heart disease. ANN with back propagation is proposed for better prediction of the disease. The results obtained from the application of ANN are highly accurate and very precise [33]. The genetic algorithm with fuzzy NN known as Recurrent Fuzzy Neural Network (RFNN) is introduced for the diagnosis of heart disease.

In the UCI data set 297 instances of patient records, in total, are considered of which 252 records are used for training and the remaining for testing. The results have been located to be satisfying based on the assessment [35]. Heart disease prediction with SVM and ANN is proposed. In this approach, two methods are used for the premise of the accuracy and time of testing. The proposed model arranges the data records into two classes in SVM as well as ANN for further anal- ysis as shown in [37]. The Back Propagation Neural Net- work (BPNN) with classification method is introduced, where the hypertension gene sequence is generated and then, thereafter the exact gene sequence. The performance of the BPNN techniques has been measured in the training phase as well as the testing phase with the various numbers of samples. The accuracy of this technique has improved in correspondence to the number of records [40].

IV. PROPOSED METHOD HRFLM

In this study, we have used an *R* studio rattle to perform heart disease classification of the Cleveland UCI repository. It provides an easy-to-use visual representation of the dataset, working environment and building the predictive analytics. ML process starts from a pre-processing data phase followed by feature selection based on DT entropy, classification of modeling performance evaluation, and the results with improved accuracy. The feature selection and modeling keep on repeating for various combinations of attributes. Table1shows the UCI dataset detailed information with attributes used. Table2shows the data type and range of values. The performance of each model generated based on 13 features and ML techniques used for each iteration and performance are recorded. Section A summarizes the data pre-processing, Section B discusses the feature selection using entropy, Section C explains the classification with ML techniques and Section D presented for the performance of the results.

A. DATA PRE-PROCESSING

Heart disease data is pre-processed after collection of various records. The dataset contains a total of 303 patient records, where 6 records are with some missing values. Those 6 records have been removed from the dataset and the remaining 297 patient records are used in pre-processing. The multi-class variable and binary classification are introduced for the attributes of the given dataset. The multi-class variable is used to check the presence or absence of heart disease. In the instance of the patient having heart disease, the value is set to 1, else the value is set to 0 indicating the absence of heart disease in the patient. The pre-processing of data is carried out by converting medical records into diagnosis values. The results of data pre-processing for 297 patient records indicate that 137 records show the value of 1 establishing the presence of heart disease while the remaining 160 reflected the value of 0 indicating the absence of heart disease.

TABLE 1. UCI dataset attributes detailed information.

Attribute	Description	Type
Age	Patient's age in completed years	Numeric
Sex	Patient's Gender (male represented as1 and female as 0)	Nominal
Cp	The type of Chest pain categorized into 4 values: 1. typical angina, 2. atypical angina, 3. non-anginal pain and 4. asymptomatic	Nominal
Trestbps	Level of blood pressure at resting mode (in mm/Hg at the time of admitting in the hospital)	Numeric
Chol	Serum cholesterol in mg/dl	Numeric
FBS	Blood sugar levels on fasting > 120 mg/dl; represented as 1 in case of true, and 0 in case of false	Nominal
Resting	Results of electrocardiogram while at rest are represented in 3 distinct values: Normal state is represented as Value 0, Abnormality in ST-T wave as Value 1, (which may include inversions of T-wave and/or depression or elevation of ST of > 0.05 mV) and any probability or certainty of LV hypertrophy by Estes' criteria as Value 2	Nominal
Thali	The accomplishment of the maximum rate of heart	Numeric
Exang	Angina induced by exercise. (0 depicting 'no' and 1 depicting 'yes')	Nominal
Oldpeak	Exercise-induced ST depression in comparison with the state of rest	Numeric
Slope	ST segment measured in terms of the slope during peak exercise depicted in three values: 1. unsloping, 2. flat and 3. downsloping	Nominal
Ca	Fluoroscopy coloured major vessels numbered from 0 to 3	Numeric
Thal	Status of the heart illustrated through three distinctly numbered values. Normal numbered as 3, fixed defect as 6 and reversible defect as 7.	Nominal
Num	Heart disease diagnosis represented in 5 values, with 0 indicating total absence and 1 to 4 representing the presence in different degrees.	Nominal

TABLE 2. UCI dataset range and datatype.

AGE	Numeric [29 to 77;unique=41;mean=54.4;median=56]
SEX	Numeric [0 to 1;unique=2;mean=0.68;median=1]
CP	Numeric [1 to 4;unique=4;mean=3.16;median=3]
TESTBPS	Numeric [94 to 200;unique=50;mean=131.69;median=130]
CHOL	Numeric [126 to 564;unique=152;mean=246.69;median=241]
FBS	Numeric [0 to 1;unique=2;mean=0.15;median=0]
RESTECG	Numeric [0 to 2;unique=3;mean=0.99;median=1]
THALACH	Numeric [71 to 202;unique=91;mean=149.61;median=153]
EXANG	Numeric [0 to 1;unique=2;mean=0.33;median=0.00]
OLPEAK	Numeric [0 to 6.20;unique=40;mean=1.04;median=0.80]
SLOPE	Numeric [1 to 3;unique=3;mean=1.60;median=2]
CA	Categorical [5 levels]
THAL	Categorical [4 levels]
TARGET	Numeric [0.00 to 4.00;unique=5;mean=0.94;median=0.00]

B. FEATURE SELECTION AND REDUCTION

From among the 13 attributes of the data set, two attributes pertaining to age and sex are used to identify the personal

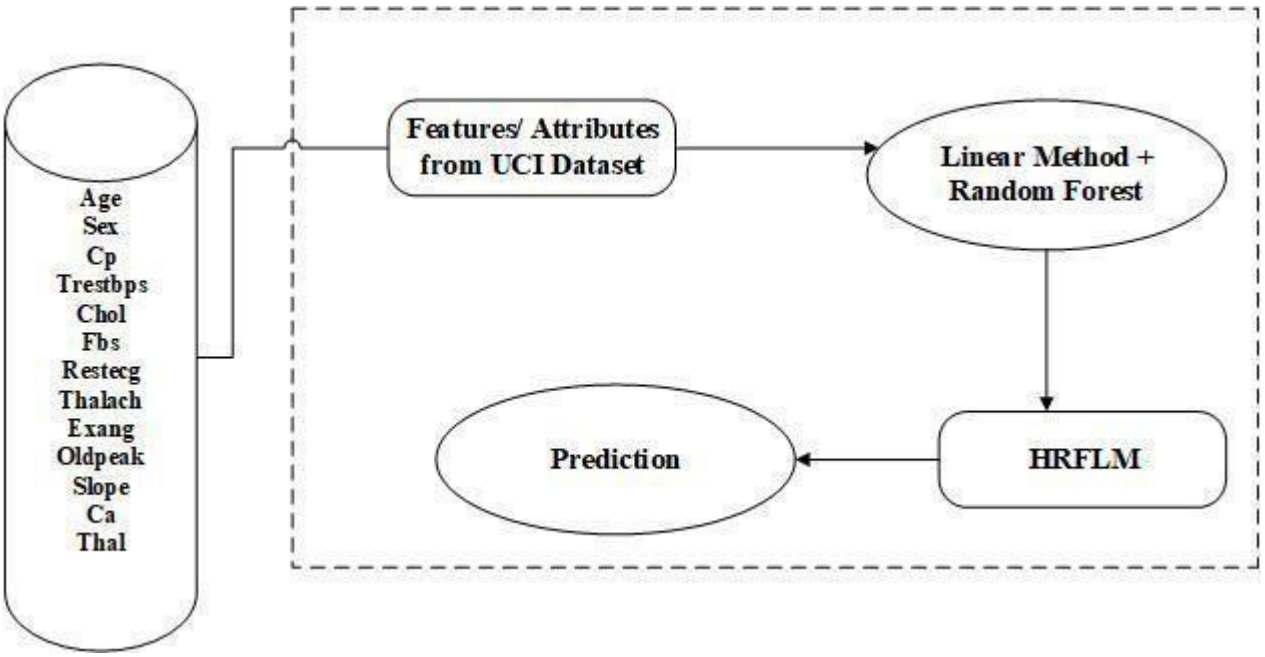

FIGURE 2. Prediction of heart disease with HRFLM.

information of the patient. The remaining 11 attributes are considered important as they contain vital clinical records. Clinical records are vital to diagnosis and learning the severity of heart disease. As previously mentioned in this experiment, several (ML) techniques are used namely, NB, GLM, LR, DL, DT, RF, GBT and SVM. The experiment was repeated with all the ML techniques using all 13 attributes. Figure2shows the prediction method of HRFLM.

C. CLASSIFICATION MODELLING

The clustering of datasets is done on the basis of the variables and criteria of Decision Tree (DT) features. Then, the classifiers are applied to each clustered dataset in order to estimate its performance. The best performing models are identified from the above results based on their low rate of error. The performance is further optimized by choos- ing the DT cluster with a high rate of error and extraction of its corresponding classifier features. The performance of the classifier is evaluated for error optimization on this data set.

1) DECISION TREES

For training samples of data D, the trees are constructed based on high entropy inputs. These trees are simple andfast constructed in a top down recursive divide and conquer (DAC) approach. Tree pruning is performed to remove the irrelevant samples on D.

$$Entropy = -\sum_{j=1} p_{ij} \log_2 p_{ij} \tag{1}$$

2) LANGUAGE MODEL

For given input features x_i, y_i with input vector x_i of data D the linear form of solution $f(x) = mx + b$ is solved by subsequent parameters:

$$m = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} \tag{2}$$

$b = \bar{y} - m\bar{x}$ where $\bar{x}$, $\bar{y}$ are the means.

3) SUPPORT VECTOR MACHINE

Let the training samples having dataset $Data = \{y_i, x_i\}$; $i = 1, 2, \ldots, n$ where $x_i \in R^n$ represent the i^{th} vector and $y_i \in R^n$ represent the target item. The linear SVM finds the optimal hyperplane of the form $f(x) = w^T x + b$ where w is a dimensional coefficient vector and b is a offset. This is done by solving the subsequent optimization problem:

$$\begin{aligned} &Min_{w,b,\xi_i} \frac{1}{2} w^2 + C \sum_{i=1} \xi_i \\ &s.t.\ y_i \left(w^T x_i + b \right) \geq 1 - \xi_i, \xi_i \geq 0, \quad \forall_i \in \{1, 2, \ldots, m\} \end{aligned} \tag{3}$$

4) RANDOM FOREST

This ensemble classifier builds several decision trees and incorporates them to get the best result. For tree learning, it mainly applies bootstrap aggregating or bagging. For agiven data, $X = \{x_1, x_2, x_3, \ldots, x_n\}$ with responses $Y = \{x_1, x_2, x_3, \ldots, x_n\}$which repeats the bagging from $b = 1$ to B.

TABLE 3. Result of various models with proposed model.

Models	Accuracy	Classification error	Precision	F-measure	Sensitivity	Specificity
Naive Bayes	75.8	24.2	90.5	84.5	79.8	60.0
Generalized Linear Model	85.1	14.9	88.8	91.6	94.9	20.0
Logistic Regression	82.9	17.1	89.6	90.2	91.1	25.0
Deep Learning	87.4	12.6	90.7	92.6	95	33.3
Decision Tree	85	15.0	86	91.8	98.8	0.0
Random Forest	86.1	13.9	87.1	92.4	98.8	10.0
Gradient Boosted Trees	78.3	21.7	94.1	86.8	80.7	60.0
Support Vector Machine	86.1	13.9	86.1	92.5	100	0.0
VOTE	87.41	12.59	90.2	84.4	-	-
HRFLM (proposed)	**88.4**	**11.6**	**90.1**	**90**	**92.8**	**82.6**

The unseen samples x^r is made by averaging the predictions $\sum_{b=1}^{B} fb(x^r)$ from every individual trees on x^r:

$$j = \frac{1}{B}\sum_{b=1}^{B} fb\,x^r \tag{4}$$

The uncertainty of prediction on these tree is made through its standard deviation,

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B} fb\,(x^r) - \hat{f}}{B-1}} \tag{5}$$

5) NAIVE BAYES

This learning model applies Bayes rules through independent features. Every instance of data D is allotted to the class of highest subsequent probability. The model is trained through the Gaussian function with prior probability $P\,X_f = priority \in (0:1)$

$$P\,X_{f\,1}, X_{f\,2}, \ldots, X_{fn}\,|c = \prod_{i=1}^{n} P\,X_{fi}|c$$

$$P\,X_f|c_i = \frac{P\,c_i|X \quad P\,X_f}{f} \qquad c \in \{benign, malignant\} \tag{6}$$

$$P\,(c_i)$$

At last, the testing data is categorized based on the probability of association:

$$c_{nb} = \arg\max P\,(c_k) \prod_{i=1}^{n} P\;X_{fi}|c_k\;, \quad for\ k = 1, 2$$

6) NEURAL NETWORKS

Algorithm 1 Decision Tree-Based Partition

Require: Input: D dataset – features with a target class
for ∀features **do**
  for Each sample **do**
    Execute the Decision Tree algorithm
  end for
  Identify the feature space $f_1, f_2, \ldots, f_x$ of dataset UCI.
end for (9)
Obtain the total number of leaf nodes $l_1, l_2, l_3, \ldots, l_n$ with its constraints (10)
Split the dataset D into $d_1, d_2, d_3, \ldots, d_n$ based on the leaf nodes constraints. (11)
Output: Partition datasets $d_1, d_2, d_3, \ldots, d_n$

Algorithm 2 Apply ML to Find Less Error Rate
Require: Input: Datasets with partition – $d_1, d_2, d_3, \ldots, d_n$
for Apply the rules **do**
  On the dataset R(d1,d2,d3, ...dn)
end for
Classify the dataset based on the rules C(R(d_1),R(d_2) . . . . . R(d_n)) (12)
Output: Classified datasets with rules C(R(d_1),R(d_2) ..... R(d_n))

7) K-NEAREST NEIGHBOUR

It extract the knowledge based on the samples Euclidean distance function $d\,x_i, x_j$ and the majority of k-nearest neighbors.

$$d\,x_{i,x_i,} = \sqrt{x_{i,1} - x_{j,1} + \ldots + x_{i,m} - x_{j,m}} \tag{8}$$

The neuron components includes inputs x_i, hidden layers and output y

function like sigmoid and a bias constant b.

$$f\left(b + \sum_{i=1}^{\Sigma} x_i u_i\right) \quad (7)$$

D. PERFORMANCE MEASURES

Several standard performance metrics such as accuracy, precision and error in classification have been considered for the computation of performance efficacy of this model. Accuracy in the current context would mean the percentage of instances correctly predicting from among all the available instances. Precision is defined as the percentage of corrective prediction

TABLE 4. Classification rules for HRFLM.

Rule No	Attributes	Values	Range
1.	root	212 97 0	(0.54 0.18 0.13 0.12 0.033)
2.	CA=0	123 29 0	(0.76 0.15 0.049 0.024 0.0081)
3.	CA=1,2,3	89 67 3	(0.24 0.21 0.24 0.25 0.067)
4.	EXANG	< 0.5 97 11 0	(0.89 0.1 0 0.01 0)
5.	EXANG	>=0.5 26 17 1	(0.31 0.35 0.23 0.077 0.038)
6.	SEX	< 0.5 24 11 0	(0.54 0 0.17 0.25 0.042)
7.	SEX	>=0.5 65 46 1	(0.12 0.29 0.26 0.25 0.077)
8.	THAL	=3 11 3 0	(0.73 0.18 0.091 0 0)
9.	THAL	=3,6,7 15 8 1	(0 0.47 0.33 0.13 0.067)
10.	THAL	=3 15 2 0	(0.87 0 0.067 0.067 0)
11.	THAL	=6,7 9 4 3	(0 0 0.33 0.56 0.11)
12.	THALACH	>=139.5 33 20 1	(0.15 0.39 0.33 0.061 0.061)
13.	RESTECG	< 1 17 10 2	(0.29 0.29 0.41 0 0)
14.	RESTECG	>=1 16 8 1	(0 0.5 0.25 0.12 0.12)
15.	THALACH	< 139.5 32 18 3	(0.094 0.19 0.19 0.44 0.094)

TABLE 5. Result of various models with proposed model.

	Overall error rate			Best Model	Overall classification error rate			Best Model
Data Split	DT	RF	LM		DT	RF	LM	
1	14.9	4	6.7	RF	14.9	14.9	16.2	DT /RF
2	34.9	12.2	22.6	RF	39.6	37.7	38.7	RF
3	50	11.1	16.6	RF	50	27.8	50	RF
4	62.5	20.9	29.2	RF	62.5	54.1	54.2	RF
5	60	13.3	13.3	RF/ LM	60	53.4	53.3	LM
6	54.6	12	18.1	RF	60.6	57.6	54.6	LM
7	57.1	0	28.5	RF	57.1	28.5	42.8	RF
8	36.4	18.2	9.1	LM	36.4	27.3	27.3	RF/ LM

Algorithm 3 Feature Extraction Using Less Error Classifier

Require: Input: Classified datasets C(R(d_1),R(d_2). .....R(d_n))

for Find out min error rate from the input **do**
Min(C(R(d_1),R(d_2) .....R(d_n))) (13)

end for

Find out max(min) error rate from the classifier.

Output: Features with classified attributes F(d_1, d_2, d_3, . . . , d_n)

Algorithm 4 Apply Classifier on Extracted Features

Apply the hybrid method based on the error rate

$$\sum_{n}^{0} F(n) = d + m_1 x_1 + m_2 x_2 + \ldots + m_n x_n \quad (14)$$

$$F(0) = Gain + \sum_{0}^{n} w_i x_i \quad (15)$$

in the positive class of the instances. Classification error is defined as the percentage of accuracy missing or error available in the instances. To identify the significant features of heart disease, three performance metrics are used which will help in better understanding the behavior of the various

TABLE 6. Results generated based on HRFLM.

Data Split	Overall rate RF	LM		Overall classification error rate RF	LM	
1	4	6.7	RF	14.9	16.2	DT /RF
2	12.2	22.6	RF	37.7	38.7	RF
3	11.1	16.6	RF	27.8	50	RF
4	20.9	29.2	RF	54.1	54.2	RF
5	13.3	13.3	RF/ LM	53.4	53.3	LM
6	12	18.1	RF	57.6	54.6	LM
7	0	28.5	RF	28.5	42.8	RF
8	18.2	9.1	LM	27.3	27.3	RF/ LM

combinations of the feature-selection. ML technique focuses on the best performing model compared to the existing models. We introduce HRFLM, which produces high accuracy and less classification error in the prediction of heart disease. The performance of every classifier is evaluated individually and all results are adequately recorded for further investigation.

E. HRFLM ALGORITHMS

See Algorithms 1–4.

TABLE 7. Comparison of various models with the proposed model.

Source	Sex	cp	Fbs	restecg	exang	Oldpeak	Slope	Ca	thal
Bhatla & Jyoti (2012	0	1	0	0	1	1	0	1	1
Nahar, Imam, Tickle & Chen (2013)	0	1	1	1	1	0	0	0	0
Sen, Patel & Shukla (2013)	1	1	1	1	1	0	0	0	0
Chaurasia & Pal (2013)						0		0	0
Tomar & Agarwal (2014)	0	1	1	1	0	1	1	1	0
Nahato, Harichandran & Arputharaj (2015)	0	1		1	0	1	0	1	1
Paul, Shill, Rabin & Akhand (2016)	1	1	0	1	1	1	1	1	1
Dey, Singh & Singh (2016)	1	1	1	1		0	1	0	0
Wiharto et.al (2017)									
Liu, Wang, et.al (2017)									
Total					1				
	1	1	0	0	1	1	1	0	0
	0	1	0	1	0	0	1	1	1
	3	10	3	8	7	3	6	3	4

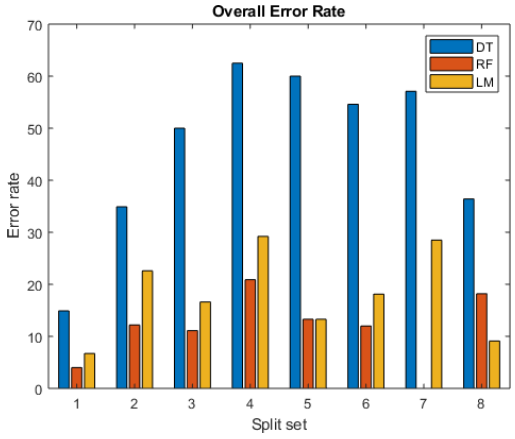

FIGURE 3. Overall error rate of the dataset.

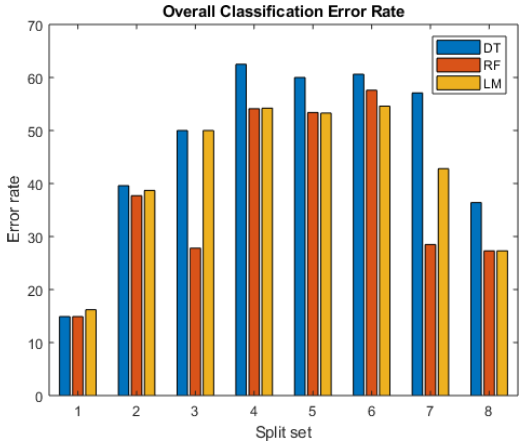

FIGURE 4. Overall classification error rate of the dataset.

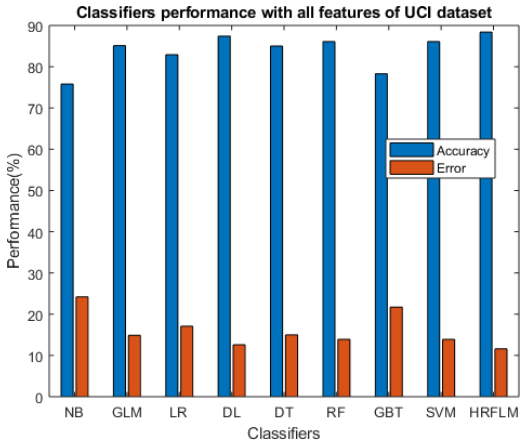

FIGURE 5. Performance comparison with various models.

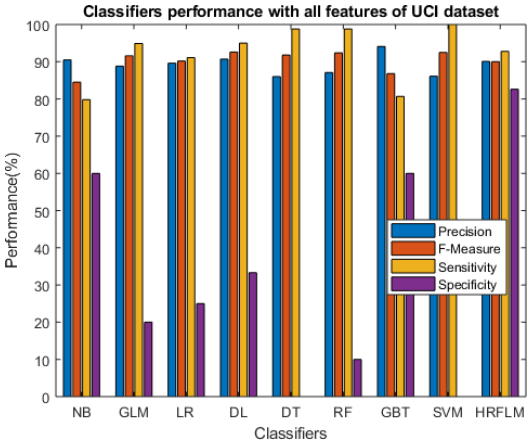

FIGURE 6. Performance comparison with various models.

V. EXPERIMENTATIONAL ENVIRONMENT

A. DATASETS

Heart disease data was collected from the UCI machine learning repository. There are four databases (i.e. Cleveland, Hungary, Switzerland, and the VA Long Beach). The Cleveland database was selected for this research because it is a commonly used database for ML researchers with comprehensive and complete records. The dataset contains 303 records. Although the Cleveland dataset has 76 attributes, the data set provided in the repository furnishes information for a subset of only 14 attributes. The data source of the Cleveland dataset is the Cleveland Clinic Foundation.

TABLE 8. Data split based on DT.

Dataset	DT						RF						LM					
	Confusion matrix					Error	Confusion matrix					Error	Confusion matrix					Error
Original	150	8	1	1	0	6.2	159	1	0	0	0	0.6	152	5	2	0	1	5
	24	13	7	10	0	75.9	10	39	4	1	0	27.8	23	20	6	5	0	63
	8	5	15	7	0	57.1	5	1	28	1	0	20	8	8	11	6	2	68.6
	4	9	5	17	0	51.4	5	4	2	24	0	31.4	3	10	11	11	0	68.6
	2	2	3	6	0	100	1	0	1	2	9	30.8	2	3	1	5	2	84.6
1	63	0	0	0	0	0	63	0	0	0	0	0	61	0	1	0	1	3.2
	3	0	0	0	0	100	2	1	0	0	0	66.7	2	1	0	0	0	66.7
	2	0	0	0	0	100	0	0	2	0	0	0	0	0	2	0	0	0
	4	0	0	0	0	100	1	0	0	3	0	25	0	1	0	3	0	25
	2	0	0	0	0	100	0	0	0	0	2	0	0	0	0	0	2	0
2	20	0	0	0	0	0	20	0	0	0	0	0	22	2	2	0	0	15.4
	5	0	0	0	0	100	3	2	0	0	0	60	3	2	0	0	0	60
	2	0	0	0	0	100	0	0	2	0	0	0	0	0	2	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	3	0	0	0	100	2	1	0	0	0	33.3	3	0	0	0	0	0
	0	6	0	0	0	0	0	6	0	0	0	0	0	6	0	0	0	0
	0	4	0	0	0	100	0	2	2	0	0	50	0	1	3	0	0	25
	0	5	0	0	0	100	0	2	0	3	0	40	0	1	0	4	0	20
	0	2	0	0	0	100	0	0	0	0	2	0	0	0	0	0	2	0
4	16	0	0	0	0	0	16	0	0	0	0	0	13	1	2	0	0	18.8
	6	0	0	0	0	100	1	5	0	0	0	16.7	1	5	0	0	0	16.7
	4	0	0	0	0	100	1	0	3	0	0	25	1	0	3	0	0	25
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	5	0	0	0	100	2	3	0	0	0	60	2	2	0	1	0	60
	0	7	0	0	0	0	0	7	0	0	0	0	0	6	0	1	0	14.3
	0	1	0	0	0	100	0	0	1	0	0	0	0	0	1	0	0	0
	0	4	0	0	0	100	0	2	0	2	0	50	0	0	2	2	0	50
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	8	0	2	0	0	20
	10	0	0	0	0	0	10	0	0	0	0	0	5	5	0	0	0	50
	2	0	0	0	0	100	1	1	0	0	0	50	1	1	0	0	0	50
	1	0	0	0	0	100	0	0	1	0	0	0	0	0	1	0	0	0
	3	0	0	0	0	100	1	0	0	2	0	33.3	1	0	0	2	0	33.3
	2	0	0	0	0	100	1	0	0	0	1	50	0	0	1	0	1	50
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	7	0	0	100	0	4	3	0	0	42.9	0	4	3	0	0	42.1
	0	0	15	0	0	0	0	0	15	0	0	0	0	1	13	0	1	13.3
	0	0	5	0	0	100	0	0	2	2	1	60	0	0	2	3	0	40
	0	0	3	0	0	100	0	0	0	0	3	0	0	0	0	0	3	0
8	0	1	0	0	0	100	1	0	0	0	0	0	1	0	0	0	0	0
	0	6	0	3	1	40	0	6	0	3	1	40	1	4	0	3	2	60
	0	1	0	3	3	100	0	2	5	0	0	28.6	0	0	6	1	0	14.3
	0	5	0	11	1	35.3	0	1	0	14	2	17.6	0	4	0	12	1	29.4
	0	0	0	2	4	33.3	0	0	0	0	6	0	0	0	0	0	6	0

where only one attribute serves as the output or the predicted attribute to the presence of heart disease in a patient.

The Cleveland dataset contains an attribute named `num`to show the diagnosis of heart disease in patients on different scales, from 0 to 4. In this scenario, 0 represents the absence of heart disease and all the values from 1 to 4 represent patients with heart disease, where the scaling refers to the severity of the disease (4 being the highest). Figure1shows the distribution of the `num`attribute among the identified 303 records.

B. EXPERIMENTAL SETUP FOR EVALUATION

We have used an *R* studio rattle to perform the classification of heart disease from Cleveland `UCI` repository. Figure1 depicts the evaluation of the experiment by step-by-step stages. In the first step, the UCI dataset is loaded and the data becomes ready for pre-processing. The subset of 13 attributes (Age, sex, cp, treetops, chol, FBS, restecg, thalach, exang, olpeak, slope, ca, that, target) is selected from the pre-processed data set of heart disease. The three existing models for heart disease prediction (DT, RM, LM) are used to develop the classification. The evaluation of the model is performed with the confusion matrix. Totally, four outcomes are generated by confusion matrix, namely `TP` (**True Positive**), `TN` (**True Negative**), `FP` (**False Pos- itive**) and `FN` (**False Negative**). The following measures are used for the calculation of the accuracy, sensitivity, specificity.

Accuracy = (TN+TP) / (TN+TP+FN+FP)
=105+155/295 = 0.8847

Sensitivity = (TP/TP+FN) =155/155+12 = 92.8

Specificity = (TN/TN+FP) =105/105+22 = 82.6

TABLE 9. Feature extraction from LM.

	DT						RF						LM					
data	Confusion matrix					Error	Confusion matrix					Error	Confusion matrix					Error
Origina	150	8	1	1	0	6.2	158	2	0	0	0	1.2	152	5	2	0	1	5
	24	13	7	10	0	75.9	11	40	2	1	0	25.9	22	21	6	5	0	61.1
	8	5	15	7	0	57.1	5	1	29	0	0	17.1	8	8	12	5	2	65.7
	4	9	5	17	0	51.4	5	5	2	23	0	34.3	3	10	11	11	0	68.6
	2	2	3	6	0	100	0	1	2	1	9	30.8	2	3	1	5	2	84.6
1	98	0	0	0	0	0	98	0	0	0	0	0	96	1	0	1	0	2
	11	0	0	0	0	100	5	6	0	0	0	45.5	8	3	0	0	0	72.7
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	100	0	0	0	1	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	26	0	0	0	0	0	26	0	0	0	0	0	23	1	2	0	0	11.5
	5	0	0	0	0	100	3	2	0	0	0	60	1	3	1	0	0	40
	2	0	0	0	0	100	0	0	2	0	0	0	0	0	2	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	3	0	0	0	100	2	1	0	0	0	33.3	2	0	1	0	0	33.3
	0	6	0	0	0	0	0	6	0	0	0	0	1	5	0	0	0	16.7
	0	4	0	0	0	100	0	2	2	0	0	50	0	1	3	0	0	25
	0	5	0	0	0	100	0	2	0	3	0	40	0	1	0	4	0	20
	0	2	0	0	0	100	0	0	0	0	2	0	0	0	0	2	0	0
4	16	0	0	0	0	0	15	0	1	0	0	6.2	13	3	0	0	0	18.8
	6	0	0	0	0	100	1	5	0	0	0	16.7	1	5	0	0	0	16.7
	4	0	0	0	0	100	1	0	3	0	0	25	1	0	3	0	0	25
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	5	0	0	0	100	2	2	0	1	0	60	2	2	0	1	0	60
	0	7	0	0	0	0	0	7	0	0	0	0	0	6	1	0	0	14.3
	0	1	0	0	0	100	0	0	1	0	0	0	0	0	1	0	0	0
	0	4	0	0	0	100	0	2	0	2	0	50	0	0	1	3	0	25
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	10	0	0	0	0	0	8	1	0	1	0	20	8	0	2	0	0	20
	2	0	0	0	0	100	1	1	0	0	0	50	1	1	0	0	0	50
	1	0	0	0	0	100	0	0	1	0	0	0	0	0	1	0	0	0
	3	0	0	0	0	100	1	0	0	2	0	33.3	0	0	0	2	1	33.3
	2	0	0	0	0	100	1	0	0	0	1	50	0	0	1	0	1	50
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	7	0	0	100	0	4	3	0	0	42.9	0	4	3	0	0	42.9
	0	0	15	0	0	0	0	0	15	0	0	0	0	0	14	0	1	6.7
	0	0	5	0	0	100	0	0	3	2	0	60	0	0	3	2	0	60
	0	0	3	0	0	100	0	0	0	0	3	0	0	0	0	0	3	0
8	0	1	0	0	0	100	1	0	0	0	0	0	1	0	0	0	0	0
	0	6	0	3	1	40	0	6	0	3	1	40	0	6	0	3	1	40
	0	1	0	3	3	100	0	1	5	1	0	28.6	0	1	5	1	0	28.6
	0	5	0	11	1	35.3	0	1	1	14	1	17.6	0	1	1	13	2	23.5
	0	0	0	2	4	33.3	0	0	0	0	6	0	0	0	0	0	6	0

Precision = TP /TP+FP = 155/155+22==87.5

F-Measure = 2TP/ 2TP+FP+FN = 310 /310+22+12 = 0.90

VI. EVALUATION RESULTS

The prediction models are developed using 13 features and the accuracy is calculated for modeling techniques. The best classification methods are given below in Table3. This table compares the accuracy, classification error, precision, F-measure, sensitivity and specificity. The highest accuracy is achieved by HRFLM classification method in comparison with existing methods.

VII. DISCUSSION OF HRFLM TOIMPROVE THE RESULTS

The UCI dataset is further classified into 8 types of datasets based on classification rules. The classification rules are listed in Table4. Each dataset is further classified and processed by **R Studio Rattle**. The results are generated by applying the classification rule for the dataset.

The classification rules generated based on the rule after data pre-processing is done. After pre-processing, the data's three best ML techniques are chosen and the results are generated. The various datasets with DT, RF, LM are applied to find out the best classification method. Table5shows that results of existing and proposed methods.

The results show that RF and LM are the best. The RF error rate for dataset 4 is high (20.9%) compared to the other datasets. The LM method for the dataset is the best (9.1%) compared to DT and RF methods. We combine the RF method with LM and propose HRFLM method to improve the results. Table6show the results of the proposed method. Figure3shows the overall error rate of the dataset.

Figure4shows the overall classification error rate of the dataset.

A. BENCHMARKING OF THE PROPOSED MODEL

Benchmarking is needed to compare the performance of the existing models compared with the proposed model. This method is used to identify whether the proposed method is the best and improves accuracy or not. The accuracy is calculated with the number of feature selection and the model generated results. HRFLM has no restriction in selecting of features to use. All the features selected in this model accomplish the best results. Table7shows that comparison of various models with our proposed method. Figure5and Figure6shows the performance comparison of the various model with respect to proposed method respectively.

Table5depicts the details of features selected by various models from the UCI dataset for heart disease. The proposed method is used on all 13 attributes and classified, based on the error rate. This result clearly proves that all the features selected and ML techniques used, prove effective in accurately predicting heart disease of patients compared with known existing models.

VIII. CONCLUSION

Identifying the processing of raw healthcare data of heart information will help in the long term saving of human lives and early detection of abnormalities in heart condi- tions. Machine learning techniques were used in this work to process raw data and provide a new and novel discernment towards heart disease. Heart disease prediction is challenging and very important in the medical field. However, the mortality rate can be drastically controlled if the disease is detected at the early stages and preventative measures are adopted as soon as possible. Further extension of this study is highly desirable to direct the investigations to real-world datasets instead of just theoretical approaches and simulations. The proposed hybrid HRFLM approach is used combining the characteristics of Random Forest (RF) and Linear Method (LM). HRFLM proved to be quite accurate in the prediction of heart disease. The future course of this research can be performed with diverse mixtures of machine learning techniques to better prediction techniques. Furthermore, new feature-selection methods can be developed to get a broader perception of the significant features to increase the performance of heart disease prediction.

APPENDIX A

See Table 8.

APPENDIX B

See Table 9.

APPENDIX C

See Table 10.

TABLE 10. Hybrid model with HRFLM.

Data	Confusion matrix					Error	MODEL
1	98	0	0			0	RF
	5	6	0			45.5	
	0	0	0			0	
	0	0	0			0	
	0	0	0			0	
2	26	0	0			0	RF
	3	2	0			60	
	0	0	2			0	
	0	0	0			0	
	0	0	0			0	
3	3	0	0			0	LM
	0	6	0			0	
	0	1	3			25	
	0	1	0			20	
	0	0	0			0	
4	16	0	0			0	RF
	1	5	0			16.7	
	1	0	3			25	
	0	0	0			0	
	0	0	0			0	
5	2	3	0			60	RF
	0	7	0			0	
	0	0	1			0	
	0	2	0			50	
	0	0	0			0	
6	8	1	0			20	RF
	1	1	0			50	
	0	0	1			0	
	1	0	0			33.3	
	1	0	0			50	
7	0	0	0			0	RF
	0	4	3			42.9	
	0	0	15			0	
	0	0	3			60	
	0	0	0			0	
8	1	0	0			0	RF
	0	7	0			30	
	0	1	5			28.6	
	0	1	0	14		17.6	
	0	0	0	0		0	

ACKNOWLEDGMENT

The authors would like to thank IEEE Access journal and their respective Universities for their support.

REFERENCES

[1]A. S. Abdullah and R. R. Rajalaxmi, ''A data mining model for predicting the coronary heart disease using random forest classifier,'' in *Proc. Int. Conf. Recent Trends Comput. Methods, Commun. Controls*, Apr. 2012, pp. 22–25.

[2]A. H. Alkeshuosh, M. Z. Moghadam, I. Al Mansoori, and M. Abdar, ''Using PSO algorithm for producing best rules in diagnosis of heart disease,'' in *Proc. Int. Conf. Comput. Appl. (ICCA)*, Sep. 2017, pp. 306–311.

[3]N. Al-milli, ''Backpropogation neural network for prediction of heart disease,'' *J. Theor. Appl.Inf. Technol.*, vol. 56, no. 1, pp. 131–135, 2013.

[4]C. A. Devi, S. P. Rajamhoana, K. Umamaheswari, R. Kiruba, K. Karunya, and R. Deepika, ''Analysis of neural networks based heart disease prediction system,'' in *Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI)*, Gdansk, Poland, Jul. 2018, pp. 233–239.

[5]P. K. Anooj, ''Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 24, no. 1, pp. 27–40, Jan. 2012. doi:10.1016/j.jksuci.2011.09.002.

[6]L. Baccour, ''Amended fused TOPSIS-VIKOR for classification (ATOVIC) applied to some UCI data sets,'' *Expert Syst. Appl.*, vol. 99, pp. 115–125, Jun. 2018. doi:10.1016/j.eswa.2018.01.025.

[7]C.-A. Cheng and H.-W. Chiu, ''An artificial neural network model for the evaluation of carotid artery stenting prognosis using a national-wide

database,'' in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 2566–2569.

[8]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)*, Dec. 2017, pp. 1011–1014.

[9]F. Dammak, L. Baccour, and A. M. Alimi, ''The impact of criterion weights techniques in TOPSIS method of multi-criteria decision making in crisp and intuitionistic fuzzy domains,'' in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, vol. 9, Aug. 2015, pp. 1–8.

[10]R. Das, I. Turkoglu, and A. Sengur, ''Effective diagnosis of heart disease through neural networks ensembles,'' *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7675–7680, May 2009. doi:10.1016/j.eswa.2008.09.013.

[11]M. Durairaj and V. Revathi, ''Prediction of heart disease using back propagation MLP algorithm,'' *Int. J. Sci. Technol. Res.*, vol. 4, no. 8, pp. 235–239, 2015.

[12]M. Gandhi and S. N. Singh, ''Predictions in heart disease using techniques of data mining,'' in *Proc. Int. Conf. Futuristic Trends Comput. Anal. Knowl. Manage. (ABLAZE)*, Feb. 2015, pp. 520–525.

[13]A. Gavhane, G. Kokkula, I. Pandya, and K. Devadkar, ''Prediction of heart disease using machine learning,'' in *Proc. 2nd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Mar. 2018, pp. 1275–1278.

[14]B. S. S. Rathnayakc and G. U. Ganegoda, ''Heart diseases prediction with data mining and neural network techniques,'' in *Proc. 3rd Int. Conf. Converg. Technol. (I2CT)*, Apr. 2018, pp. 1–6.

[15]N. K. S. Banu and S. Swamy, ''Prediction of heart disease at early stage using data mining and big data analytics: A survey,'' in *Proc. Int. Conf. Elect., Electron., Commun., Comput. Optim. Techn. (ICEECCOT)*, Dec. 2016, pp. 256–261.

[16]J. P. Kelwade and S. S. Salankar, ''Radial basis function neural network for prediction of cardiac arrhythmias based on heart rate time series,'' in *Proc. IEEE 1st Int. Conf. Control, Meas. Instrum. (CMI)*, Jan. 2016, pp. 454–458.

[17]V. Krishnaiah, G. Narsimha, and N. Subhash, ''Heart disease prediction system using data mining techniques and intelligent fuzzy approach: A review,'' *Int. J. Comput. Appl.*, vol. 136, no. 2, pp. 43–51, 2016.

[18]P. S. Kumar, D. Anand, V. U. Kumar, D. Bhattacharyya, and T.-H. Kim, ''A computational intelligence method for effective diagnosis of heart disease using genetic algorithm,'' *Int. J. Bio-Sci. Bio-Technol.*, vol. 8, no. 2, pp. 363–372, 2016.

[19]M. J. Liberatore and R. L. Nydick, ''The analytic hierarchy process in medical and health care decision making: A literature review,'' *Eur. J. Oper. Res.*, vol. 189, no. 1, pp. 194–207, 2008.

[20]T. Mahboob, R. Irfan, and B. Ghaffar, ''Evaluating ensemble prediction of coronary heart disease using receiver operating characteristics,'' in *Proc. Internet Technol. Appl. (ITA)*, Sep. 2017, pp. 110–115.

[21]J. Nahar, T. Imam, K. S. Tickle, and Y.-P. P. Chen, ''Computational intelligence for heart disease diagnosis: A medical knowledge driven approach,'' *Expert Syst. Appl.*, vol. 40, no. 1, pp. 96–104, 2013. doi:10.1016/j.eswa.2012.07.032.

[22]J. Nahar, T. Imam, K. S. Tickle, and Y.-P. P. Chen, ''Association rule mining to detect factors which contribute to heart disease in males and females,'' *Expert Syst. Appl.*, vol. 40, no. 4, pp. 1086–1093, 2013. doi:10.1016/j.eswa.2012.08.028.

[23]S. N. Rao, P. Shenoy M, M. Gopalakrishnan, and A. Kiran B, ''Applicability of the Cleveland clinic scoring system for the risk prediction of acute kidney injury after cardiac surgery in a South Asian cohort,'' *Indian Heart J.*, vol. 70, no. 4, pp. 533–537, 2018. doi:10.1016/j.ihj.2017.11.022.

[24]T. Karayılan and Ö. Kılıç, ''Prediction of heart disease using neural network,'' in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Antalya, Turkey, Oct. 2017, pp. 719–723.

[25]J. Thomas and R. T. Princy, ''Human heart disease prediction system using data mining techniques,'' in *Proc. Int. Conf. Circuit, PowerComput. Technol. (ICCPCT)*, Mar. 2016, pp. 1–5.

[26]C. Raju, ''Mining techniques,'' in *Proc. Conf. Emerg. Devices Smart Syst. (ICEDSS)*, Mar. 2016, pp. 253–255.

[27]D. K. Ravish, K. J. Shanthi, N. R. Shenoy, and S. Nisargh, ''Heart function monitoring, prediction and prevention of heart attacks: Using artificial neural networks,'' in *Proc. Int. Conf. Contemp. Comput. Inform. (IC3I)*, Nov. 2014, pp. 1–6.

[28]F. Sabahi, ''Bimodal fuzzy analytic hierarchy process (BFAHP) for coronary heart disease risk assessment,'' *J. Biomed. Informat.*, vol. 83, pp. 204–216, Jul. 2018. doi:10.1016/j.jbi.2018.03.016.

[29]M. S. Amin, Y. K. Chiam, K. D. Varathan, ''Identification of significant features and data mining techniques in predicting heart disease,'' *Telem- atics Inform.*, vol. 36, pp. 82–93, Mar. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0736585318308876

[30]S. M. S. Shah, S. Batool, I. Khan, M. U. Ashraf, S. H. Abbas, and S. A. Hussain, ''Feature extraction through parallel probabilistic principal component analysis for heart disease diagnosis,'' *Phys. A, Stat. Mech. Appl.*, vol. 482, pp. 796–807, 2017. doi:10.1016/j.physa.2017.04.113.

[31]Y. E. Shao, C.-D. Hou, and C.-C. Chiu, ''Hybrid intelligent modeling schemes for heart disease classification,'' *Appl. Soft Comput. J.*, vol. 14, pp. 47–52, Jan. 2014. doi:10.1016/j.asoc.2013.09.020.

[32]J. S. Sonawane and D. R. Patil, ''Prediction of heart disease using multilayer perceptron neural network,'' in *Proc. Int. Conf. Inf. Commun. Embedded Syst.*, Feb. 2014, pp. 1–6.

[33]C. Sowmiya and P. Sumitra, ''Analytical study of heart disease diagnosis using classification techniques,'' in *Proc. IEEE Int. Conf. Intell. Techn. Control, Optim. Signal Process. (INCOS)*, Mar. 2017, pp. 1–5.

[34]B. Tarle and S. Jena, ''An artificial neural network based pattern classification algorithm for diagnosis of heart disease,'' in *Proc. Int. Conf. Comput., Commun., Control Automat. (ICCUBEA)*, Aug. 2017, pp. 1–4.

[35]V. P. Tran and A. A. Al-Jumaily, ''Non-contact Doppler radar based prediction of nocturnal body orientations using deep neural network for chronic heart failure patients,'' in *Proc. Int. Conf. Elect. Comput. Technol. Appl. (ICECTA)*, Nov. 2017, pp. 1–5.

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

[37]T. Vivekanandan and N. C. S. N. Iyengar, ''Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease,'' *Comput. Biol. Med.*, vol. 90, pp. 125–136, Nov. 2017.

[38]S. Radhimeenakshi, ''Classification and prediction of heart disease risk using data mining techniques of support vector machine and artificial neural network,'' in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, New Delhi, India, Mar. 2016, pp. 3107–3111.

[39]R. Wagh and S. S. Paygude, ''CDSS for heart disease prediction using risk factors,'' *Int. J. Innov. Res. Comput.*, vol. 4, no. 6, pp. 12082–12089, Jun. 2016.

[40]O. W. Samuel, G. M. Asogbon, A. K. Sangaiah, P. Fang, and G. Li, ''An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction,'' *Expert Syst. Appl.*, vol. 68, pp. 163–172, Feb. 2017.

[41]S. Zaman and R. Toufiq, ''Codon based back propagation neural network approach to classify hypertension gene sequences,'' in *Proc. Int. Conf. Elect., Comput. Commun. Eng. (ECCE)*, Feb. 2017, pp. 443–446.

[42]W. Zhang and J. Han, ''Towards heart sound classification without segmentation using convolutional neural network,'' in *Proc. Comput. Cardiol. (CinC)*, vol. 44, Sep. 2017, pp. 1–4.

[43]Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, ''ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis,'' in *Proc. Symp. Appl. Comput.*, Apr. 2017, pp. 506–509.

[44]J. Wu, S. Luo, S. Wang, and H. Wang, ''NLES: A novel lifetime extension scheme for safety-critical cyber-physical systems using SDN and NFV,'' *IEEE Internet Things J.*, no. 6, no. 2, pp. 2463–2475, Apr. 2019.

[45]J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, ''Big data analysis-based secure cluster management for optimized control plane in software-defined networks, *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 27–38, Mar. 2018.

[46]J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, ''FCSS: Fog computing based content-aware filtering for security services in information centric social networks,'' *IEEE Trans. Emerg. Topics Comput.*, to be published. doi:10.1109/TETC.2017.2747158.

[47]G. Li, J. Wu, J. Li, K. Wang, and T. Ye, ''Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of things,'' *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4702–4711, Oct. 2018.

[48]J. Wu, K. Ota, M. Dong, and C. Li, ''A hierarchical security framework for defending against sophisticated attacks on wireless sensor networks in smart cities,'' *IEEE Access*, vol. 4, pp. 416–424, 2016.

[49]H. Li, K. Ota, and M. Dong, ''Learning IoT in edge: Deep learning for the Internet of Things with edge computing,'' *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.

SENTHILKUMAR MOHAN received the M.S. degree in software engineering, in 2007, the M.Tech. degree in IT networking, in 2013, and the Ph.D. degree from Vellore Institute of Technology (VIT), in 2017. He was a Project Associate with IIT Madras, from 2009 to 2010. He is currently an Assistant Professor (senior) with the School of Information Technology and Engineering, VIT, Vellore. He has ten years of experi- ence in teaching and research. His current research interests include big data and machine/deep learning.

CHANDRASEGAR THIRUMALAI received the Bachelor of Engineering degree in computer science and engineering from Dr. Pauls Engineer- ing College affiliated to Anna University, India, and the Master of Technology degree in com- puter science and engineering from Pondicherry Central University. He is currently pursuing the Ph.D. degree with VIT University, India. His area of specialization includes linear cryptanaly- sis, public key cryptosystems, fuzzy expert systems, automata, and networking. He has published several international journals and conferences.

GAUTAM SRIVASTAVA received the B.Sc. degree from Briar Cliff University, USA, in 2004, the M.Sc. and Ph.D. degrees from the University of Victoria, Victoria, BC, Canada, in 2006 and 2012, respectively. He then taught for three years at the Department of Computer Science, University of Victoria, where he was regarded as one of the top undergraduate professors in the computer science course instruction. From there in 2014, he joined a tenure-track position at Brandon University, Brandon, MB, Canada, where he is currently active in various professional and scholarly activities. He was promoted to the rank Associate Professor, in 2018. He, as he is popularly known, is active in research in the field of data mining and big data. In his eighth-year academic career, he has published a total of 50 papers in high-impact conferences in many countries and in high-status journals (SCI, SCIE) and has also delivered invited guest lectures on big data, cloud computing, the Internet of Things, and cryptography at many Taiwanese and Czech universities. He received the Best Oral Presenter Award in FSDM 2017, which was held at National Dong Hwa University (NDHU), Shoufeng, Taiwan, in 2017. He is an Editor of several international scientific research journals. He currently has active research projects with other academics in Taiwan, Singapore, Canada, Czech Republic, Poland, and USA. He is constantly looking for collaboration opportunities with foreign professors and students.