Heart Disease Prediction Using Hybrid Technique

Vanitha Guda , Shalini K, Shivani C
[#]*Computer Science Engineering, Chaitanya Bharathi Institute of Technology,(A)*
[1]*gvanitha_cse@cbit.ac.in*
shivanichennoju@gmail.com
kothurushalini99@gmail.com

***Abstract*— Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. It is a deadly disease that a large population of people around the world suffers with. When considering death rates and the large number of people who suffer from heart disease, it is revealed how important early diagnosis of heart disease is. Traditional way of diagnosis is not sufficient for such an illness. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis. The amount of data in the healthcare industry is huge. Developing a medical diagnosis system based on machine learning for prediction of heart disease provides more accurate diagnosis than the traditional way. Machine learning (ML) has been shown to be effective in assisting i.e., making decisions and predictions from the large quantity of data produced by the healthcare industry. Various studies give only a glimpse into predicting heart disease with ML techniques. In this paper, we propose a 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 high accuracy through the prediction model for heart disease with the hybrid technique.**

***Keywords*—Machine Learning, Classification, Prediction, Accuracy**

I. INTRODUCTION

The term "heart disease" is often used interchangeably with the term "cardiovascular disease." Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Heart related diseases or Cardiovascular Diseases (CVDs) are the main reason for a huge number of deaths in the world over the last few decades and has emerged as the most life-threatening disease, not only in India but in the whole world. So, there is a need for a reliable, accurate and feasible system to diagnose such diseases in time for proper treatment. Machine Learning algorithms and techniques have been applied to various medical datasets to automate the analysis of large and complex data.

Many researchers, in recent times, have been using several machine learning techniques to help the healthcare industry and the professionals in the diagnosis of heart related diseases. 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 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. Heart disease is predicted based on symptoms namely, pulse rate, sex, age, and many others.

The main objective of the work is to improve the performance accuracy of heart disease prediction by combining two models which gives better accuracy. This paper presents comparative results of various models based on machine learning and deep learning algorithms and techniques and analyses their performance. Models based on supervised learning algorithms such as SVM, Naive Bayes, Decision Trees, Random Forest and ensemble models are found very popular among the researchers. Along with that we implemented deep learning and genetic algorithms. The idea of hybrid technique i.e, combination of machine learning techniques using weighted average. Paper presents literature and existing works in section –II. Design work flow and module description in section –III and data set description presented in section –IV, results in section-V and conclusion and future work scope in section VI.

II. LITERATURE

In our day to day life, people are undergoing a routine and busy schedule which leads to stress and anxiety. In addition to this, the percentage of people who are obese and addicted to cigarette goes up drastically. This leads to diseases like heart disease, cancer, etc. The challenge behind these diseases is its prediction. Each person has different values of pulse rate and blood pressure. But medically proven, the pulse rate must be 60 to 100beats per minute and the blood pressure must be in the range of 120/80 to 140/90.

Heart disease is one of the major causes of death in the world. The number of people affected by heart disease increases irrespective of age in both men and women. The other factors like gender, diabetes, BMI also contribute reasons for the heart disease. In this paper, we have tried prediction and analysis of heart disease by considering the parameters like age, gender, blood pressure, heart rate, diabetes and so on. Since many other factors are involved in heart disease, so the prediction of this will be a challenging task.

Many researchers, in recent times, have been using several machine learning techniques to help the healthcare industry and the professionals in the diagnosis of heart related diseases. 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 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.

Heart disease is predicted based on symptoms namely, pulse rate, sex, age, and many others. So ML techniques are used. The main objective of this research is to improve the performance accuracy of heart disease prediction. This paper presents comparative results of various models based on machine learning and deep learning algorithms and techniques and analyses their performance. Models based on supervised learning algorithms such as SVM, Naive Bayes, Decision Trees, Random Forest and ensemble models are found very popular among the researchers. We are using a hybrid method to combine 2 models to improve accuracy of heart disease prediction. Along with that we are also implementing deep learning and genetic algorithms.

2.1 Existing System

Although many methods like Decision Trees, Logistic regression and Naive Bayes are used for predicting heart disease, they did not provide the early prediction of heart disease with more accuracy. Almost all the existing models lacked in getting the most accurate predictions resulting in failure of techniques. Every existing model has issues like Naive Bayes algorithm can't be used for large datasets and Naive Bayes considers that features are independent and fails at tasks to find relationship between features, in decision trees because of pruning it leads to reduced size of tree leading to poor accuracy, KNN is a non-parametric method which has no idea about underlying data etc. And also in some existing models classification is inaccurate leading to errors because the model did not fit well. So we are trying to combine the techniques to give the comparative predictive results in order to produce a prediction model using not only distinct techniques but also by relating two or more techniques with comparative results.

2.2 Related Works

2.2.1 Heart disease prediction system based on hidden naive Bayes Classifier [1]

Dataset used clevel and and Tool used weka 6.4 to apply HNB is a more accurate classification compared to naïve Bayes, with respect to attribute dependencies. HNB is equivalent to a Bayesian classifier which avoids the intractable complexity and takes the influence from all features into account. In hidden naïve Bayes, a parent is created for each feature, which integrates the influences from other features.

Algorithm: Heart disease prediction using hidden naïve Bayes
Input: Heart disease data set
Output: Classification whether a person is having heart Disease or not
Step 1: Heart data set is loaded
Step 2: Apply pre-processing filter discretization and inter quartile range (IQR)
Step 3: Partition the data sets into training and test set
Step 4: Heart disease data set is trained by HNB
Step 5: The test data set is given to HNB for testing
Step 6: Measure the accuracy of the HNB

Algorithm for hidden naïve Bayes (HNB) is as follows
Input: A set of data base
Output: Hidden naïve Bayes classifier
Step 1: For each value of c of class C
Step 2: Calculate probabilities P(C) from Database D
Step 3: For attributes Ai and Aj
Step 4: Compute P (ai|aj, c) from D
Step 5: Compute conditional mutual information MI=IP(Ai; Aj| C) and weights W_{ij} from D
Drawback: HNB is a structure-extension-based algorithm and needs more training time.

2.2.2 Human heart disease prediction system using data mining techniques

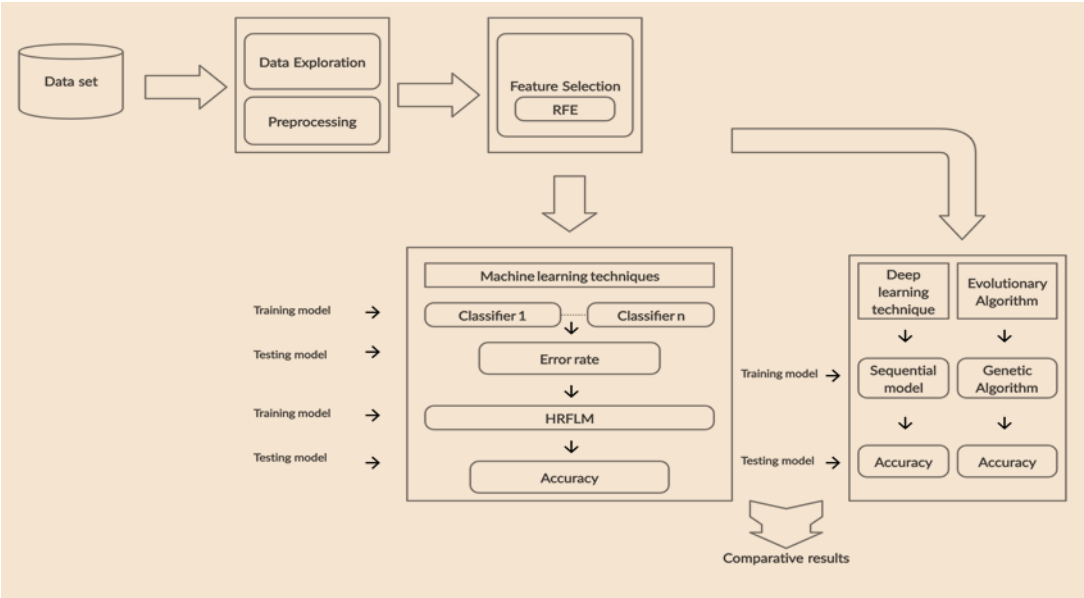
In this paper we are using three different data mining techniques namely – Decision Tree, Naïve Bayes and KNN. We are using KNN as a new method for heart disease prediction and we are comparing it with other two techniques. In recognition of patterns, the k-NN is a method which is non-parametric and used for classification and regression. In both of the cases, the input consists of the k closest training examples in the feature space. The output of pattern recognition depends on whether k-NN is used for classification or regression:

2.2.3 Prediction of heart disease using neural network [3]

Dataset used here is Cleveland from UCI repository. The proposed heart disease prediction system which uses a multilayer perceptron neural network was developed in MATLAB R2015a. The designed ANN has three layers: namely an input layer, a hidden layer and an output layer.

- Input Layer was designed to contain 13 neurons. Number of neurons was decided to be equal to the number of attributes in the data set.
- Hidden Layer was designed to contain 3 neurons. This number was decided as a start-up point. The number was changed increasing one by one until it reached the number of neurons of the input layer by comparing performance of them and then selecting the best one. This approach is based on one of machine learning best practices that the number of neurons of a hidden layer should be the mean of the number of the neurons of input and output layers.
- Output Layer was designed to contain 2 neurons. The designed NN is a classifier going running in Machine Mode which means returning a class label (e.g., "Disease Presence"/"Disease Absence"). Deciding 2 neurons is based on the idea that the output layer has one node per class label in the model.

III. Design of the Workflow



3.1 Block Diagram of the Work flow

Figure 3.1 is a block diagram that gives a brief flow of algorithms in the project. The steps include preprocessing, feature extraction, classification and comparative results. Finally output of each accuracies of models are displayed.

3.2 Module Description

3.2.1 NUMPY

NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation.

3.2.2 PANDAS

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.

3.2.3 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. It is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

3.2.4 SCIKIT

Scikit-learn is a free software machine learning library for the Python programming language. It is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Some popular groups of models provided by scikit-learn include:

- Clustering: for grouping unlabeled data such as KMeans.
- Cross Validation: for estimating the performance of supervised models on unseen data.

Ensemble methods: for combining the predictions of multiple supervised models.
Feature extraction: for defining attributes in image and text data.
Feature selection: for identifying meaningful attributes from which to create supervised models.
Parameter Tuning: for getting the most out of supervised models.
Manifold Learning: For summarizing and depicting complex multi-dimensional data.
Supervised Models: a vast array not limited to generalize linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

3.2.5 TENSORFLOW

Tensor Flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It provides multiple APIs (Application Programming Interfaces). Tensor Flow is basically a software library for numerical computation using data flow graphs where nodes in the graph represent mathematical operations, edges in the graph represent the multidimensional data arrays (called tensors) communicated between them. (Tensor is the central unit of data in TensorFlow).

3.3 Algorithms Applied

3.3.1 Logistic Regression

Logistic regression is a predictive analysis algorithm and based on the concept of probability. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

Types of Logistic Regression

1. Binary Logistic Regression: The categorical response has only two 2 possible outcomes. Example: Spam or Not.
2. Multinomial Logistic Regression : Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan).
3. Ordinal Logistic Regression : Three or more categories with ordering. Example: Movie rating from 1 to 5.

3.3.2 Naive Bayes

Naive Bayes classifier is a straightforward and powerful algorithm for the classification task. Even if we are working on a data set with millions of records with some attributes, it is suggested to try Naive Bayes approach. Naive Bayes classifier gives great results when we use it for textual data analysis. Such as Natural Language Processing. To understand the naive bayes classifier we need to understand the Bayes theorem. Bayes theorem named after Rev.

The formula for calculating the conditional probability: P(H|E)=P(E|H)*P(H)/P(E), where, P(H) is the probability of hypothesis H being true. This is known as the prior probability. P(E) is the probability of the evidence(regardless of the hypothesis). P(E|H) is the probability of the evidence given that hypothesis is true. P(H|E) is the probability of the hypothesis given that the evidence is there. This naive bayes learning model applies Bayes rules through independent features. Every instance of data D is allotted to the class of highest subsequent probability.

Types of Naive Bayes Algorithm

Gaussian Naive Bayes : When attribute values are continuous, an assumption is made that the values associated with each class are distributed according to Gaussian i.e., Normal Distribution.

Multi Nomial Naive Bayes : Multi Nominal Naive Bayes is preferred to use on data that is multinomial distributed. It is one of the standard classic algorithms. Which is used in text categorization (classification). Each event in text classification represents the occurrence of a word in a document.

Bernoulli Naive Bayes: Bernoulli Naive Bayes is used on the data that is distributed according to multivariate Bernoulli distributions .i.e., multiple features can be there, but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. So, it requires features to be binary valued.

3.3.3 Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules.The deeper the tree, the more complex the decision rules and the fitter the model.

Decision tree build classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

3.3.4 Support Vector Machine

Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of it is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

According to the algorithm it finds the points closest to the line or hyperplane from both the classes. These points are called support vectors. Now, it computes the distance between the line and the support vectors. This distance is called the margin. The model goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane. Thus the support vector machine tries to make a decision boundary in such a way that the separation between the two classes is as wide as possible. If the data is not linearly separable, it makes data linearly separable by converting data into higher dimension. Selecting the appropriate kernel function can be tricky.

3.3.5 Random Forest

Random forest is a supervised classification algorithm. It creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. This ensemble classifier builds several decision trees and incorporates them to get the best result. Random Forest pseudocode:

1. Randomly select “k” features from total “m” features where k << m.
2. Among the “k” features, calculate the node “d” using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until “l” number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for “n” times to create “n” number of trees.

3.3.6 Gradient Boosting Gradient Boosting (an ensemble learner) is another technique for performing supervised machine learning tasks, like classification and regression. This means it will create a final model based on a collection of individual models. The predictive power of these individual models is weak and prone to overfitting but combining many such weak models in an ensemble will lead to an overall much improved result. Gradient boosting Algorithm involves three elements:

- A loss function to be optimized.
- Weak learner to make predictions.
- An additive model to add weak learners to minimize the loss function.

3.3.7 K-nearest neighbours [12]
KNN algorithm uses 'feature similarity' to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. It is a non parametric and lazy algorithm. We can understand its working with the help of following steps − we need to choose the value of K i.e. the nearest data points. K can be any integer. For each point in the test data do the following −

Calculate the distance between test data and each row of training data with the help of any of the methods namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
Now, based on the distance value, sort them in ascending order.Next, it will choose the top K rows from the sorted array.Now, it will assign a class to the test point based on the most frequent class of these rows.

Voting Classifier [7]
A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output. It simply aggregates the findings of each classifier passed into the Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each of them, it creates a single model which trains by these models and predicts output based on their combined majority of voting for each output class. Voting Classifier supports two types of voting.

3.3.8 Deep Learning [9]: Deep learning is an increasingly popular subset of machine learning. Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. Then the model spits out a prediction. The weights are adjusted to find patterns in order to make better predictions. The user does not need to specify what patterns to look for — the neural network learns on its own. We use a sequential model of deep learning. Sequential is the easiest way to build a model in Keras. It allows you to build a model layer by layer. Each layer has weights that correspond to the layer that follows it. We use the 'add()' function to add layers to our model. We will add two layers and an output layer.

3.3.9 Genetic Algorithm [11]: Genetic algorithm is an optimization method inspired by the biological process of natural selection. It is based on the terms such as mutation, crossover and selection. The genetic algorithm is a random-based classical evolutionary algorithm. By random it means that in order to find a solution using the genetic algorithm, random changes applied to the current solutions to generate new ones. Genetic Algorithm is based on Darwin's theory of evolution. It is a slow gradual process that works by making changes to the making slight and slow changes. Also, genetic algorithm, makes slight changes to their solutions slowly until getting the best solution.

3.3.10 Proposed model - Hybrid Random Forest with Linear Model (HRFLM) Technique
The proposed hybrid HRFLM approach is used combining the characteristics of Random Forest and Linear Method (Logistic Regression). HRFLM proved to be quite accurate in the prediction of heart disease. There are three steps in performing the hybrid technique

1. Finding out the output probabilities of each model. To implement this we are using the pred_proba function which gives the probabilities for the target in array form. The number of probabilities for each row is equal to the number of categories in the target variable.

2. With the help of log loss function, finding the optimized weight that perfectly combines the two models which has low classification error rate. Log Loss function is a metric which takes into account the uncertainty of your prediction based on how much it varies from the actual label.

3. Using the optimized weight from the above step combining the two models with the help of weighted average and then performing the prediction. The results of the hybrid classification method have proved a higher degree of accuracy and performance in prediction of heart disease compared to the other existing methods.

IV. DATASETS

4.4.1 Cleveland dataset

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 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. There are 13 attributes that feature in the prediction of heart disease, where only one attribute serves as the output or the predicted attribute to the presence of heart disease in a patient.

4.4.2 Framingham dataset

The Framingham Heart Study dataset is dedicated to identifying common factors or characteristics that contribute to cardiovascular disease (CVD). In 1948, an original cohort of 5,209 men and women between 30 and 62 years old were recruited from Framingham, MA. An Offspring Cohort began in 1971, an Omni Cohort in 1994, a Third Generation Cohort in 2002, a New Offspring Spouse Cohort in 2004 and a Second Generation Omni Cohort in 2003. Core research in the dataset focuses on cardiovascular and cerebrovascular diseases. The data include biological specimens, molecular genetic data, phenotype data, samples, participant vascular functioning of the National Heart, Lung and Blood Institute and Boston University.

V. RESULTS AND DISCUSSION

Table 5.1 Performance metrics of ML classifiers

	Algorithm	Accuracy	Classification Error Rate	Specificity	Sensitivity	Precision
0	Logistic Regression	85.71	14.29	82.61	88.89	83.33
1	Random Forest	83.52	16.48	85.00	82.35	87.50
6	GB	81.32	18.68	78.26	84.44	79.17
2	Naive Bayes	80.22	19.78	79.07	81.25	81.25
3	Decision Tree	73.63	26.37	73.17	74.00	77.08
7	Adaboost	73.63	26.37	68.63	80.00	66.67
5	KNN	68.13	31.87	65.91	70.21	68.75
4	SVM	64.84	35.16	68.97	62.90	81.25

Table 5.1 shows the machine learning classifiers sorted by classification error rate .Here we got Logistic regression and Random Forest with least classification error rate so we combined these two models using a hybrid method.

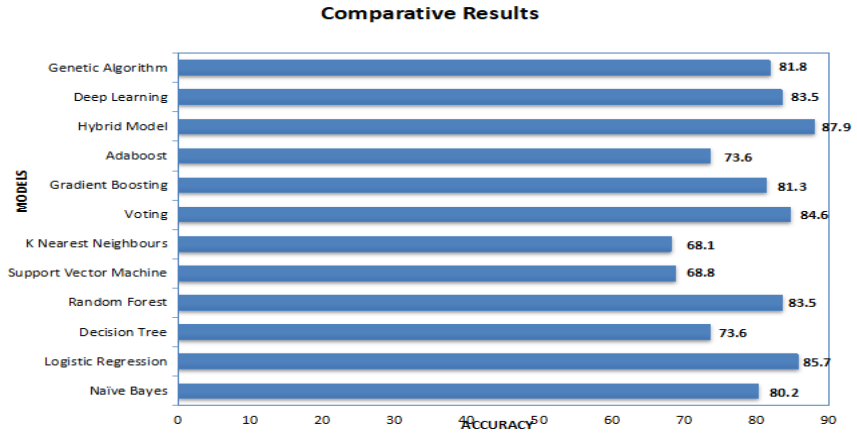


Figure 5.1 Bar plot depicting accuracy of all models

Figure 5.1 is a bar plot showing the accuracy of Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, KNN, Voting Classifier, Gradient Boosting, Adaboost , HRLFM, Deep Learning Model and

Table 5.2 Accuracies measures of ML Classifier

	Algorithm	Accuracy	Classification Error Rate
9	Hybrid	87.91	12.09
1	Logistic Regression	85.71	14.29
6	Voting	84.62	15.38
3	Random Forest	83.52	16.48
0	Deep Learning	83.52	16.48
1	Genetic algorithm	81.83	18.17
7	Gradient Boosting	81.32	18.68
0	Naive Bayes	80.22	19.78
2	Decision Tree	73.63	26.37
8	Adaboost	73.63	26.37
5	KNN	68.13	31.87
4	SVM	64.84	35.16

Table 5.2 shows the accuracy and error rate of all models applied on the dataset sorted in ascending order of classification error rate. Here we got HRLFM with highest accuracy. The results of the hybrid classification method have proved a higher degree of accuracy and performance in prediction of heart disease compared to the other existing methods.

VI CONCLUSION AND FUTURE SCOPE

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 conditions. 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. After applying all the machine techniques we found out that logistic regression always stands out on top in terms of accuracy and so we combined logistic regression with random forest because random forest is a highly robust model. Because of the combination of those two models with the help of the hybrid method it resulted in the increase in accuracy in predicting heart disease.

FUTURE SCOPE: Further extension of this study is highly desirable to direct the investigations to real-world raw data. 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.

REFERENCES

[1] A.Jabbar,ShirinaSamreen,"Heartdiseaseprediction system based on hidden naïve bayes classifier " , https://ieeexplore.ieee.org/document/8053261

[2] Abhishek Rairikar, Vedant Kulkarni, Vikas Sabale, Harshavardhan Kale, Anuradha Lamgunde " Heart disease prediction using data mining techniques", 2017 International Conference on Intelligent Computing and Control (I2C2), IEEE, Coimbatore, India, March 2017

[3] TulayKarayilan, OzkanKilic ,"Prediction of heart disease using neuralnetwork",https://www.researchgate.net/publication/320829299_Prediction_of_heartdisease_using_neural_network

[4] A.T.Sayad, P. P. Halkarnikar, "Diagnosis of heart disease using neural network approach", M. Tech, Fourth Semester, Department of Technology, Shivaji University, Kolhapur, India , Volume- 2, Issue-3, July-2014.

[5] "Feature Selection using RFE ", https://scikit-learn.org/stable/modules/generate/ sklearn.feature_selection.RFE.html

[6] Avinash Golande, Pavan Kumar T, " Heart Disease Prediction Using Effective Machine Learning Techniques ", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019

[7] "Ensemble Voting Classifier", https://scikit-learn.org/stable/modules/generated/ sklearn.ensemble.VotingClassifier.html

[8] "Performance Metrics", https://www.geeksforgeeks.org/confusion-matrix- machine-learning/Eijaz Allibhai, " Deep learning model using Keras "

[9] https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37

[10] Manas Narkar," Heart disease prediction using keras,deep learning " , https://medium.com/@manasnarkar/heart-disease-prediction-using-keras-deep-learning-960a1b7b98ee

[11] "Genetic Algorithm in machine learning", https://dkopczyk.quantee.co.uk/genetic - algorithm/

[12] "KNN",lhttps://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.html