# Heart Disease Prediction Using Hybrid Technique

**VenkateswaraRao Cheekati[1] , Dr .D.Natarajasivan[2], Dr. S.Indraneel[3]**
[1]Research Scholar, Department .of CSE Annamalai University,T.N, chvraograce@gmail.com
[2]Assistant.Prof., Department .of CS,Annamalai Univeristy , natarajasivan@gmail.com
Professor,Dept.of CSE ,.Ann's Engineering College, Chirala sreeram.indraneel@gmail.com

---

***Abstract***— One of the leading causes of death and disability around the world is coronary artery disease. A vast number of people around the world are afflicted by this terrible disease. This is especially true when one considers death rates and the enormous number of people who suffer from heart disease. This sickness cannot be diagnosed using conventional methods. In clinical data analysis, cardiovascular disease prediction is recognised as one of the most essential topics. The healthcare business generates enormous amounts of data. Using machine learning to develop a medical diagnosis system for predicting heart disease is more accurate than the traditional method. Machine learning (ML) has proven to be helpful in reducing the massive amounts of data generated by the healthcare industry's decision-making. Several studies provide only a brief glimpse into how machine learning can be used to predict cardiac disease. Using machine learning approaches, we offer a method for identifying relevant traits that can be used to improve the accuracy of cardiovascular disease prediction. Various combinations of features and recognised classification methods are used to build the prediction model. With the hybrid model for heart disease prediction, we're able to get a higher level of performance while maintaining a high degree of accuracy.

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

---

## INTRODUCTION

The terms "heart disease" and "cardiovascular disease" are frequently used interchangeably. Diseases of the cardiovascular system can cause heart attacks, angina (chest discomfort), and strokes because of restricted or clogged blood vessels. Cardiovascular diseases (CVDs) have been the leading cause of death in the world over the past several decades, not only in India but in the rest of the world as a whole. For this reason, a system that can reliably detect diseases like these in their early stages so that appropriate treatment can be administered is required. Many medical datasets have been analysed using machine learning algorithms and approaches to speed up the study of huge and complicated datasets.

Machine learning techniques have been used by a large number of researchers in recent years to aid the healthcare industry and medical professionals in the detection of heart-related disorders. There are various risk factors for heart disease, including diabetes, high blood pressure, and excessive cholesterol, as well as an irregular pulse rate. Data mining and neural networks have been used to find out the severity of heart disease in humans. The nature of heart disease necessitates cautious management because it is so complex. It's possible that failing to do so will have negative effects on the heart or result in early death. A person's risk of heart disease can be estimated by looking at their symptoms, such as their pulse rate, gender, and age.

The major goal of this research is to combine two models that provide improved accuracy in the performance of heart disease prediction. This study compares and evaluates the performance of several machine learning and deep learning models and methods. SVM, Naive Bayes, Decision Trees, Random Forest, and ensemble models are some of the most prominent supervised learning models. Deep learning and genetic algorithms were also incorporated as part of this effort. Combining different machine learning algorithms by utilising a weighted average, for example. Section II of the paper discusses relevant literature and previously published work. Section III describes the design process and modules; Section IV describes the data set; Section V

presents the results; and Section VI concludes with a discussion of the project's future goals. Sections III and IV describe the data set. Sections III and IV describe the results.

## II. LITERATURE

Stress and anxiety are common side effects of the hectic daily schedules we all undertake. This is accompanied by a dramatic increase in the number of people who are fat and addicted to cigarettes. Heart disease, cancer, and other debilitating illnesses can result. Prediction is a major problem with these disorders. The pulse rate and blood pressure vary from person to person. Blood pressure should be between 120/80 and 140/90 mmHg, which is clinically proven to be the optimal range. One of the most common causes of death worldwide is coronary artery disease. Men and women of all ages are at an increased risk of developing heart disease. There are several other factors that contribute to heart disease, such as gender, diabetes, and BMI. In this study, we have attempted to forecast and analyse heart disease by taking into account factors such as age, gender, blood pressure, heart rate, diabetes, and so on. It will be hard to predict heart disease because there are so many other things that play a role.

Machine learning techniques have been used by a large number of researchers in recent years to aid the healthcare industry and medical professionals in the detection of heart-related disorders. There are various risk factors for heart disease, including diabetes, high blood pressure, and excessive cholesterol, as well as an irregular pulse rate. The severity of cardiac disease in humans has been determined by using techniques such as data mining and neural networks. Due to the intricacy of the condition, proper care of cardiovascular disease requires extreme caution. If this is not done, there is a risk that it will have a harmful impact on the heart or lead to mortality at an earlier age.,

.

A person's risk of heart disease can be estimated by looking at their symptoms, such as their pulse rate, gender, and age. Such methods are employed. In this study, the primary goal is to increase the accuracy of heart disease diagnosis. This study compares and evaluates the performance of several machine learning and deep learning models and methods. SVM, Naive Bayes, Decision Trees, Random Forest, and ensemble models are some of the most prominent supervised learning models. We are employing a hybrid approach to improve the accuracy of heart disease prediction by combining two different models. On top of that, we're putting in place genetic algorithms and deep learning.

### 2.1 Existing System

There are several approaches for predicting heart illness, including decision trees, logistic regression, and Naive Bayes, but they are not able to accurately predict heart disease at an early stage. All of the currently available models failed to provide the most accurate forecasts, which led to the failure of various strategies. Every existing model has issues like the Naive Bayes algorithm can't be used for large datasets and Naive Bayes considers that features are independent and fails at tasks to find relationships between features, in decision trees because of pruning, it leads to a reduced size of the tree, leading to poor accuracy, KNN is a non-parametric method that has no idea about underlying data; etc. Furthermore, categorization is erroneous in certain existing models, resulting in errors as a result of the model's poor fit. We are combining the methods to come up with a prediction model that uses both different and related strategies.

### 2.2 Related Works

**2.2.1** A hidden naive Bayes classifier-based method for the early detection of heart disease

In terms of attribute dependencies, HNB provides a more accurate classification than naive Bayes when applied to the dataset and tool in question. HNB is a Bayesian classifier that avoids intractable complexity and takes into account the influence of all features. There are parents for each feature in hidden bayes, which include the influences of other features.

Prediction of heart disease with a hidden nave BayesS is a Bayes algorithm.

Step-1: Data set on cardiovascular illness as an input

Step-2 Classification of whether or not a person is suffering from heart disease.

Step – 3 A preprocessing filter discretization and an inter-quartile range are used in Step 2. (IQR)

Partition the data sets into training and test sets in step three.

Step 4: HNB trains the heart disease data set.

Step 5: HNB is entrusted with the analysis of the test data.

Measure the HNB's accuracy in step six.

Here's the HNB algorithm in a nutshell:

One or more database sets are used as input.

Classifier output: naive Bayes

For each value of c in class C, proceed to step 1.

Step-2: Data from Database D is used to calculate P (C).

Step – 3 Next, we'll take a look at Ai and Aj.

In the fourth step, we'll take the data set D and use computingSt to find the function P.

Step- 5 Conditional mutual information MI = IP (Ai; Aj|C) and weights Wij from D are combined.

**Drawback:** HNB is a structure-extension-based algorithm and needs more training time.

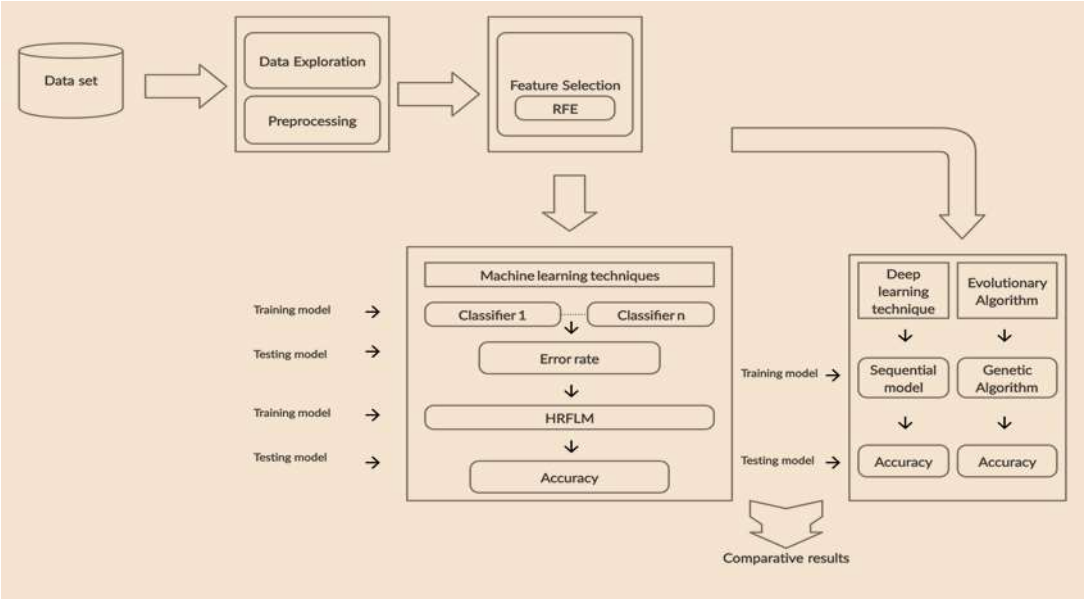
### 2.2.2 Human heart disease prediction system using data mining techniques

In this particular research, we will be utilising three distinct data mining strategies, namely the Decision Tree, the Naive Bayes, and the KNN. As a novel approach to the prediction of cardiac disease, KNN is currently the focus of our investigation, alongside two additional methodologies. The k-nearest neighbours method is a non-parametric approach to pattern recognition that can be used for classification and regression analysis. In both scenarios, the input is made up of the k training instances that are located in the feature space that are the closest together. Whether k-NN was used for classification or regression, the result of pattern recognition depends on the method that was used.

### 2.2.3 Prediction of heart disease using neural network [3]

- The Cleveland dataset, which can be found in the UCI repository, was used here. MATLAB R2015a was used in the development of the proposed system for predicting cardiac disease. This system makes use of a multilayer perceptron neural network. The artificial neural network that was built consists of three layers: an input layer, a hidden layer, and an output layer.
- The input layer was planned to have a total of 13 neurons in it. It was decided that the number of attributes in the data set would correspond directly to the number of neurons.
- It was planned for the hidden layer to include three neuronal components. It was decided that this would serve as the starting point for the count. The number was altered so that it increased by one at a time until it reached the number of neurons in the input layer. This was accomplished by comparing the performance of each individual neuron and then choosing the one that performed the best. This strategy is based on one of the recommended best practises for machine learning, which states that the average number of neurons found in an input layer and an output layer should determine the total number of neurons found in a hidden layer.
- The output layer was planned to have a total of two neuronal components. The NN that was created is a classifier that is going to operate in Machine Mode, and this means that it will produce a class label (for example, "Disease Presence" or "Disease Absence"). The decision to use two neurons is based on the idea that there is one node for each class label in the model's output layer.
- .

## III. DESIGN OF THE WORKFLOW




3.1 Block Diagram of the Work flow

Figure 3.1. The block diagram shown in Figure 3.1 provides a high-level overview of the flow of algorithms used in the project. Pre-processing, feature extraction, classification, and comparing outcomes are the phases included in this procedure. At long last, the results of each of the models' accuracy assessments are shown.

### 3.2 Module Description

3.2.1 NUMPY

NumPy is an abbreviation that can mean either "Numerical Python" or "Numeric Python." It is a component of Python that is available as open source and enables users to perform rapid mathematical computations on arrays and matrices. NumPy, along with other machine learning modules such as Scikit-learn, Pandas,

Matplotlib, TensorFlow, and so on, are what are required to finish off the Python Machine Learning Ecosystem. This is due to the fact that arrays and matrices are fundamental components of the machine learning ecosystem.

High-level mathematical functions and scientific calculations require key array-oriented computing functionalities. These functionalities are built for multidimensional arrays of data.

### 3.2.2 PANDAS

pandas Panda is a package for the Python programming language that provides data structures that are quick, versatile, and expressive. Its purpose is to simplify and streamline the process of dealing with structured (tabular, multidimensional, potentially heterogeneous) and time series data. Its goal is to become the most fundamental and high-level building block for undertaking data analysis in Python that is applicable to the current world. In addition to this, its overarching objective is to become the most effective and adaptable data analysis and manipulation tool that is freely available in any programming language.

### 3.2.3 MATPLOTLIB

Matplotlib is a fantastic visualisation tool in Python that can be used to plot arrays in two dimensions. Matplotlib is a data visualisation library that is built on NumPy arrays and is designed to operate with the larger SciPy stack. It is compatible with multiple platforms. One of the most significant advantages of visualisation is that it gives us visual access to massive amounts of data in the form of images that are simple to understand. It is a complete library for Python programmers to use in the production of interactive, animated, and static visualisations. Matplotlib has many different kinds of graphs, such as line, bar, scatter, and histogram plots.

### 3.2.4 SCIKIT

Scikit-learn is a machine learning package that may be used with the Python programming language. It is available for free. It is a machine learning Python module built on top of SciPy that is given under the 3-Clause BSD licence. It includes a variety of methods for classification, regression, and clustering, including support vector machines, random forests, and gradient boosting, and it is designed to interoperate with a variety of other systems.

NumPy and SciPy are Python libraries for numerical and scientific computing, respectively.

One of the most well-known types of models scikit-learn offers is clustering, which is used to group unlabeled data and includes KMeans.

• Cross Validation: This technique is used to estimate how well supervised models perform on new data.With ensemble methods, you can combine the results of different kinds of supervised models.

The process of establishing properties in picture and text data using "feature extraction."

The process of choosing which important attributes will be used to build supervised models is called "feature selection."

Getting the most out of supervised models requires a process known as parameter tuning.

For simplifying and displaying complex, multi-dimensional data, manifold learning is useful.

Supervised models include a wide variety of different approaches, some of which include but are not limited to generalised 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 that may be used for dataflow and differentiable programming in a variety of contexts and applications. In addition to its use in symbolic math, the library is also put to use in machine learning applications like neural networks. It offers a number of different APIs (Application Programming Interfaces). Tensor Flow is essentially a software library that is used for numerical computation through the use of data flow graphs. The nodes in the graph represent mathematical operations, and the edges in the graph represent the multidimensional data arrays that are communicated between them. Tensors are referred to as "tensors." (In TensorFlow, the tensor serves as the primary organisational unit for data.)

## 3.3 Algorithms Applied

### 3.3.1 Logistic Regression

When it comes to the classification task, the Naive Bayes classifier is an algorithm that is both simple and effective. It is recommended that we give the Naive Bayes strategy a shot, even if we are working with a dataset that contains millions of records with some properties. When applied to the study of textual data, the Naive Bayes classifier produces very satisfying results. Processing of natural languages is one example. It is necessary for us to have an understanding of the Bayes theorem in order to comprehend the naive Bayes classifier. The Bayes theorem, after its namesake, Rev.

The following equation can be used to get the conditional probability: P(H|E) = P(E|H)*P(H)/P(E), where P(H) denotes the likelihood that hypothesis H is correct.This is what statisticians refer to as the prior probability. The probability of the evidence is denoted by the symbol P(E) (regardless of the hypothesis). The probability that the evidence indicates that the hypothesis is correct is denoted by the symbol P(E|H). Given that the evidence exists, the probability of the hypothesis is denoted by the symbol P(H|E). The Bayes principles are implemented into this naïve Bayes learning model through the use of independent characteristics. Every instance of the data type D is put in the category that has the highest likelihood of coming up again.

Different Applications of the Naive Bayes Algorithm

When the values of an attribute aren't discrete, the Gaussian Naive Bayes method assumes that the values for each class follow the Normal Distribution, which is also called the Gaussian Distribution. When working with data that has a multinomial distribution, it is recommended to utilise the Multinomial Naive Bayes method. It is considered to be a classic algorithm of the highest quality. which is applied to the classification of text (classification). The occurrence of a word in a document is referred to as a "event" in the process of text classification.

### 3.3.2 Naive Bayes

When it comes to the classification task, the Naive Bayes classifier is an algorithm that is both simple and effective. It is recommended that we give the Naive Bayes strategy a shot, even if we are working with a dataset that contains millions of records with some properties. When applied to the study of textual data, the Naive Bayes classifier produces very satisfying results. Processing of natural languages is one example. It is

necessary for us to have an understanding of the Bayes theorem in order to comprehend the naive Bayes classifier. The Bayes theorem, after its namesake, Rev.

The following equation can be used to get the conditional probability: P(H|E) = P(E|H)*P(H)/P(E), where P(H) denotes the likelihood that hypothesis H is correct.This is what statisticians refer to as the prior probability. The probability of the evidence is denoted by the symbol P (E) (regardless of the hypothesis). The probability that the evidence indicates that the hypothesis is correct is denoted by the symbol P (E|H). Given that the evidence exists, the probability of the hypothesis is denoted by the symbol P (H|E). The Bayes principles are implemented into this nave Bayes learning model through the use of independent characteristics. Every instance of the data type D is put in the category that has the highest likelihood of coming up again.

Different Applications of the Naive Bayes Algorithm

When the values of an attribute aren't discrete, the Gaussian Naive Bayes method assumes that the values for each class follow the Normal Distribution, which is also called the Gaussian Distribution.

When working with data that has a multinomial distribution, it is recommended to utilise the Multinomial Naive Bayes method. It is considered to be a classic algorithm of the highest quality. which is applied to the classification of text (classification). The occurrence of a word in a document is referred to as an "event" in the process of text classification.

**Bernoulli Naive Bayes:** Bernoulli The method of Naive Bayes is applied to the data when the data is distributed in accordance with multivariate Bernoulli distributions. That is to say, there may be more than one feature, but we are treating each one as though it were a binary-valued (Bernoulli, boolean) variable. Therefore, the features have to be rated on a binary scale.

### 3.3.3 Decision Trees

Decision Trees (DTs) are a form of non-parametric supervised learning that can be used for classification and regression. The sine curve is approximated using a series of if-then-else decision rules that are learned by decision trees as they process data. When there are more levels in the tree, the decision-making rules become more intricate, and the model becomes more accurate.

The construction of classification or regression models using a tree-like structure is known as a decision tree. It takes a data set and divides it into subsets that are progressively smaller while simultaneously developing an associated decision tree in an iterative fashion. The completed task produces a tree that has decision nodes as well as leaf nodes. A decision node can have two or more branches coming off of it. Each leaf node in the tree reflects a choice or classification. The root node is the decision node at the very top of a tree, and it corresponds to the predictor with the highest accuracy. Data of either a categorical or numerical kind can be processed using decision trees.

### 3.3.4 Support Vector Machine

Support The Vector Machine is a linear model that can be applied to both classification and regression issues. It is effective for a wide variety of real-world challenges and can tackle both linear and nonlinear problems. The concept behind it is straightforward: The method makes a line or a hyperplane that separates the data into the right groups.

In accordance with the method, it selects the points from both classes that are situated in the proximity of the line or hyperplane. The names given to these points are "support vectors." Now it determines how far apart the line and the support vectors are by computing the distance between them. This particular distance is referred to as the margin. The objective of the model is to achieve the highest possible margin. The ideal hyperplane is

the one in which the margin is the highest possible. Therefore, the decision boundary that the support vector machine creates will attempt to strike such a balance in order to ensure that there is as much of a gap as possible between the two groups. In the event that the data cannot be separated linearly, it will make the data linearly separable by transforming the data into a higher dimension. The process of selecting a suitable kernel function might be challenging.

.**3.3.5 Random Forest**

A random forest is an example of an algorithm for supervised classification. It contributes a lot of trees to the formation of the forest. In general, there is a correlation between the number of trees in a forest and the robustness of the forest. In the same manner, the random forest classifier produces high-accuracy results when the number of trees in the forest is increased. This ensemble classifier creates a number of decision trees and uses them together in order to achieve the best possible outcome. Random Forest pseudocode:

1. Select "k" features at random from the whole set of "m" features, where "k" is less than "m."

2. Using the optimal split point, compute the "d" node among the "k" characteristics.

3. Using the method that provides the best split, divide the node into daughter nodes.

4. Continue to repeat steps 1 to 3 until you have reached the desired number of nodes (l).

5. Construct the forest by performing steps 1 to 4 "n" times in order to produce "n" times that number of trees.

**3.3.6 Gradient Boosting Gradient Boosting** Another method for carrying out supervised machine learning tasks such as classification and regression is known as an ensemble learner. This indicates that it will produce a complete model by building upon a set of models used individually. These individual models have a limited capacity for prediction and are prone to overfitting, but integrating a large number of such limited models into an ensemble will lead to an overall result that is significantly better. The gradient-boosting algorithm is comprised of the following three components:

• A function of loss that needs to be optimised.

• As a learner, you are unable to make accurate forecasts.

• An additive model with less capable students to reduce the loss function

**3.3.7 K-nearest neighbours [12]**

The KNN algorithm makes use of '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. This further means that the KNN algorithm makes use of 'feature similarity' to predict the values of new data points. It is a non-parametric algorithm that is also a lazy one. With the aid of the following stages, we are able to comprehend how it works. First, we need to determine the value of K, which is the closest together data points. Any integer may serve as K. Perform the following operations on each point of the test data:

Calculate the distance between each row of the test data and each row of the training data using either the Euclidean, Manhattan, or Hamming distance algorithm. Most of the time, the Euclidean method is used to figure out how far away something is now, arrange them in ascending order based on the distance value you were given. After that, it will select the first K rows from the array that has been sorted. Now, it will decide

what category the test point belongs to based on the category that appears the most frequently among these rows.

**Voting Classifier [7]**

A Voting Classifier is a type of machine learning model that is trained on an ensemble of many models and then makes a prediction about an output (class) based on the models' highest likelihood of selecting that class as the output. It merely takes the findings from each classifier that were fed into the Vote Classifier, compiles them, and then determines the output class based on the voting result that received the highest majority. Instead of creating multiple independent dedicated models and determining the accuracy of each of them, the idea is to create a single model that is trained by these models and predicts output based on their combined majority of voting for each output class. This eliminates the need to create separate models from scratch. The Voting Classifier allows users to choose between two distinct voting methods.

**3.3.8 Deep Learning [9]:** The subfield of machine learning known as deep learning is becoming an increasingly popular option. Neural networks are used to construct the models that are used in deep learning. Inputs are received by a neural network, and these inputs are subsequently processed in the hidden layers using weights that are modified during the training process. After that, the model will generate a prediction for you. In order to improve the accuracy of the predictions, the weights are changed in order to search for trends. The user does not need to tell the neural network what patterns to look for because the network will learn on its own. The sequential model of deep learning is the one that we utilise. Building a model in Keras using the sequential method is the most straightforward approach. It enables you to construct a model in stages, layer by layer. Weights are assigned to each layer so that they are consistent with the layer that comes after it. In order to add layers to our model, we make use of the 'add' function. We will add a total of three layers: one input, one middle, and one output.

**3.3.9 Genetic Algorithm [11]**: The natural selection process in biology served as the inspiration for the optimization method known as the genetic algorithm. It makes use of concepts such as mutation, crossover, and selection as its foundation. The genetic algorithm is a type of classical evolutionary algorithm that is based on randomness. When we talk about randomness, we are referring to the process of applying random alterations to previously found solutions in order to come up with new ones as part of the genetic algorithm's search for a solution. The theory of evolution proposed by Charles Darwin serves as the foundation for genetic algorithms. It is a process that is slow and gradual, and it operates by making small adjustments to the process that are sluggish and subtle. The genetic algorithm also makes small changes to its answers so that it can get to the best state as quickly as possible.

.

**3.3.10 Proposed model - Hybrid Random Forest with Linear Model (HRFLM) Technique**

Rather than using either the Random Forest method or the Linear Method, the hybrid HRFLM strategy that was described is what is being employed (Logistic Regression). The HRFLM model was shown to be quite accurate in its ability to predict cardiac disease. In order to successfully carry out the hybrid approach, there are three phases.

1. determining the probability of each model's output. In order to put this into action, we are going to be utilising the pred proba function, which provides the probabilities for the target in the form of an array. The ratio of the total number of categories in the target variable to the total number of probabilities in each row is 1.

2. Using the log loss function, determine the optimal weight for combining the two models so that the overall classification error rate is minimized. The degree to which your prediction is different from the true label is measured by a statistic called the log loss function. This shows how uncertain your prediction is.

3. Using the weight that was optimised in the step before this one, combine the two models with the assistance of a weighted average, and then proceed to make the forecast. When compared to the other methods that are now available, the results of the hybrid classification method have demonstrated a greater level of accuracy and performance in the prediction of cardiovascular disease.

## IV. DATASETS

### 4.4.1 Cleveland dataset

The data on heart illness is taken from the machine learning repository at the University of California, Irvine. There are four different database options (i.e. Cleveland, Hungary, Switzerland, and the VA Long Beach). A total of 303 records can be found in the collection. Even though there are 76 attributes included in the Cleveland dataset, the data set that is given in the repository only provides information for a subset of 14 of those attributes. The Cleveland Clinic Foundation is where the information for the Cleveland dataset was obtained. There are thirteen different factors that go into the process of predicting heart disease. But in the end, only one of these factors determines whether a patient has heart disease or not.

### 4.4.2 Framingham dataset

The data collected as part of the Framingham Heart Study is aimed at determining the factors or qualities that, on the whole, are associated with an increased risk of cardiovascular disease (CVD). In 1948, residents of Framingham, Massachusetts, ranging in age from 30 to 62 years old, were selected to participate as members of the first cohort. A New Offspring Spouse Cohort did not begin until 2014; a Third Generation Cohort in 2012; an Offspring Cohort in 1971, an Omni Cohort in 1994, a Third Generation Cohort in 2002, and a Second Generation Omni Cohort in 2013. The study's primary emphasis is placed on cardiovascular and cerebrovascular disorders as its primary research topic. The National Heart, Lung, and Blood Institute and Boston University gave the research team biological samples, molecular genetic data, phenotypic data, samples, and information about how the participants' blood vessels worked.

## V. RESULTS AND DISCUSSION

Table 5.1 Metrics of performance for machine learning 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 It displays the machine learning classifiers arranged according to their classification error rate. After finding that logistic regression and random forest produced the results with the lowest rate of classification error, we decided to merge these two models in order to create a hybrid technique.

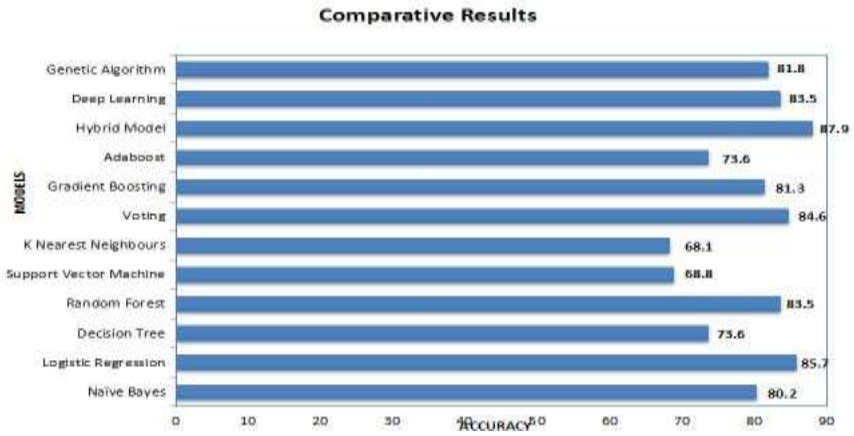



**Figure 5.1 Bar plot depicting accuracy of all models**

Figure 5.1 is a bar plot that shows how accurate different machine learning models are. These models include Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, KNN, Voting Classifier, Gradient Boosting, Adaboost, HRLFM, and Deep Learning Model, among others.

**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 The dataset displays the accuracy as well as the error rate of each model that was used on the dataset, with the results being arranged in ascending order of the error rate that each model produced. We were able to obtain HRLFM with the maximum possible precision here. When compared to the other methods that are now available, the results of the hybrid classification method have demonstrated a greater level of accuracy and performance in the prediction of cardiovascular disease.

## VI CONCLUSION AND FUTURE SCOPE

A hidden naive Bayes classifier-based method for the early detection of heart disease In terms of attribute dependencies, HNB provides a more accurate classification than naive Bayes when applied to the dataset and tool in question. HNB is a Bayesian classifier that avoids intractable complexity and takes into account the influence of all features. In hidden bayes, there are parents for each feature, and these parents take into account the effects of other features. Prediction of heart disease with a hidden nave BayesS is a Bayes algorithm. The early diagnosis of irregularities in heart problems as well as the saving of human lives could both be aided by the localization of the processing of raw healthcare data pertaining to heart information. This would be a long-term benefit of locating this processing. Throughout the course of this research, machine learning strategies were utilised to process raw data in order to give an original and cutting-edge interpretation of heart disease. These methods were also utilised by the researchers that conducted the study. Predicting cardiac disease is a challenging and extremely important undertaking, hence the medical profession places a large amount of focus on doing so. On the other hand, if the disease is detected when it is still in its early stages and preventative measures are put into place as soon as it is practicable to do so,

it is feasible to significantly reduce the fatality rate. After making use of all the different machine methods, we discovered that logistic regression always comes out on top in terms of accuracy. As a result, we merged logistic regression with random forest because random forest is a model that is known for its high level of stability. When the hybrid method was used to combine these two models, the accuracy with which heart disease could be predicted increased by a lot.

**FUTURE SCOPE:** This research must be continued and expanded in order to direct the investigations toward real-world raw data. Also, the performance of predicting heart disease can be improved by coming up with new ways to choose features, which can give a more complete picture of the important ones.

.

## 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