CARDIOVASCULAR DISEASE DETECTION USING ARTIFICIAL NEURAL NETWORKS

GROUP MEMBERS:-

Fanindra Nayak (20BIT0344)
Jatin Shah (20BIT0387)
Praanshull Verma (20BIT0229)

SITE
Fall semester - 2022-23
Course Code - ITE1015
Course Name : Soft Computing
Slot B1 + TB1

SUBMITTED TO:-

Dr. Balakrushna Tripathy

GROUP REPRESENTATIVE:-

fanindra.nayak2020@vitstudent.ac.in

Index

1 ABSTRACT

Cardiovascular illnesses are those that affect the heart. Heart disease is not a short-term illness like a fever or a cold. They can take years to diagnose and are difficult to detect and forecast based on symptoms alone. It is a significant source of morbidity and transience in modern culture. Diagnosis of cardiovascular disease using numerous medical tests is an essential but difficult undertaking that must be done correctly. If there are any flaws or mistakes in those projections, the patient's life may be jeopardized. As a result, a powerful tool for predicting heart disease at a cheaper cost has become a pressing requirement. The detection of such cardiovascular diseases, or heart diseases, may be accomplished through the use of common symptoms such as regular illness, or it may be predicted through the use of risk factors such as age, family history, diabetes, hypertension, high cholesterol, tobacco smoking, alcohol intake, obesity, or physical inactivity, and so on.

A small number of systems use these risk variables to predict cardiac disease. Heart disease patients share many of these apparent risk factors, which can be utilized to make an accurate diagnosis. A system based on such risk indicators would not only benefit medical professionals, but it would also alert individuals about the possibility of heart disease even before they enter a hospital. In this case, we will use ANN and binary classification on a dataset containing just risk variables for network prediction and training.

2 INTRODUCTION

In medical diagnosis, patients' information may include repetitive and connected symptoms and indicators, especially if the patients have more than one type of disease in the same category. The doctors might not be able to appropriately diagnose it. As a result, it is crucial to determine the important diagnostic aspects of a disease, which may aid physicians in diagnosing the condition early and properly. Because of their operators like selection, crossover, and mutation, genetic algorithms are often employed for improved solutions. Accurate and trustworthy decision making in cardiological prognosis can aid in the planning of appropriate surgery and therapy, as well as overall patient management throughout the disease's phases. The prediction of illnesses is not a simple process. To analyze and forecast the issue, we may require more than one soft computing, machine learning, or data mining approach.
Thus, the suggested problem is plainly connected to people's ignorance and, as a result, their disrespect for good medical care, particularly in the case of cardio-logical diseases. Thus, this approach intends to raise awareness among individuals by precisely predicting if they are at danger of getting a cardiac disease and, as a result, making them more proactive in choosing better lifestyle choices and attend frequent check-ups.

3 AIM

Our main aim in this review is to develop a heart disease prediction system , check its accuracy, verify if it is optimal using a genetic algorithm, and use Artificial Neural Network (ANN) to provide a very effective and accurate solution to one of the problems. The innovative part of the project is determining heart disease. Because it is a medical domain, it must be precise or else it may cause harm to a patient. As the population grows, so does the number of people with heart disease. Doctors face a huge load of work just for one patient, making it a tedious and time-consuming process.

4 Problem Statement

Due to the ever growing population on earth Cardiovascular illnesses are increasing day by day. Cardiovascular diseases are those that affect the heart. Heart disease is not a short-term illness like a fever or a cold. They can take years to diagnose and are difficult to detect and forecast based on symptoms alone. Here to develop a heart disease prediction system , check its accuracy, verify if it is optimal using a genetic algorithm.

5 Flow chart for Genetic Algorithm:

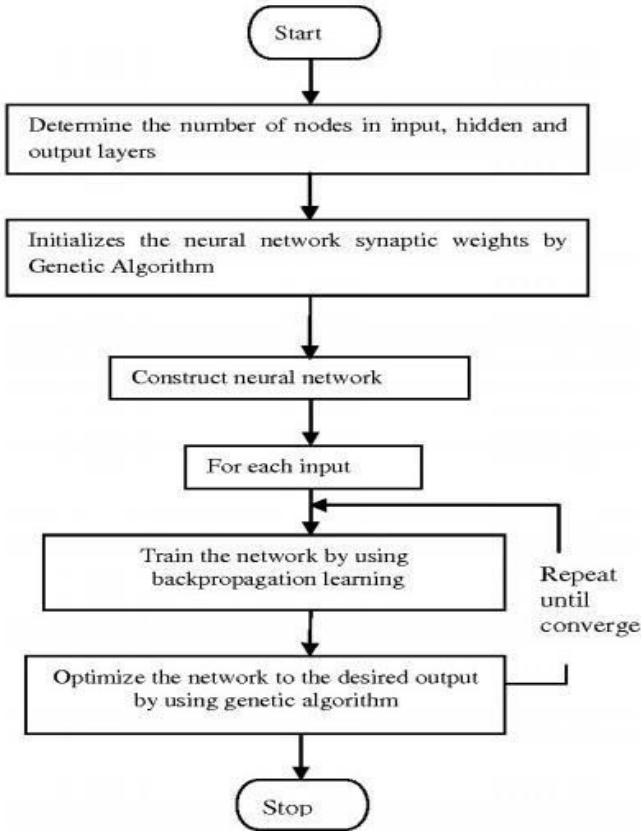


Figure 1 : Flow chart for Genetic Algorithm

6 Genetic algorithm general architecture:

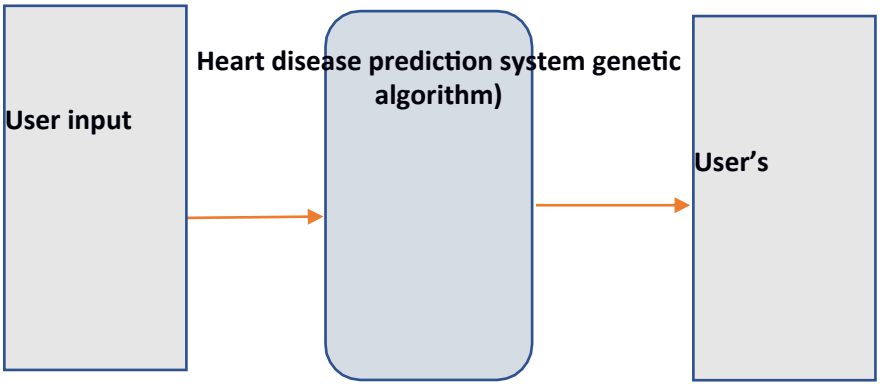


Figure 2 : Genetic algorithm general architecture:

7 Neural Network general architecture

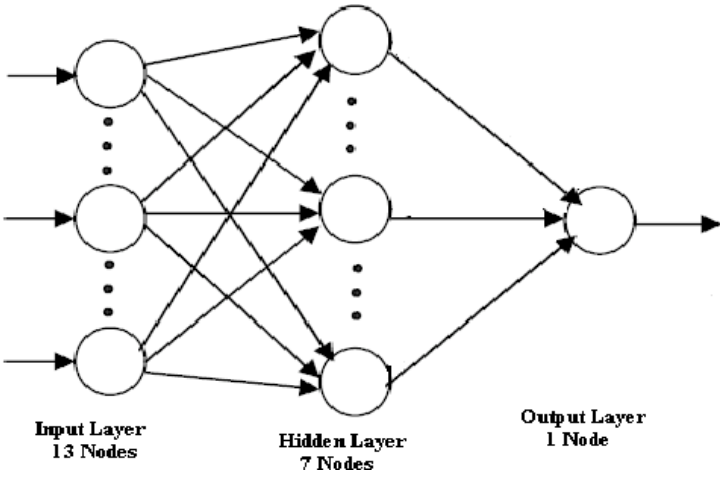


Figure 3 : Neural Network general architecture:

8 H/W and S/W REQUIREMENTS

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software, or application. In simple words, the SRS document is a manual of a project, provided it is prepared before you kick-start a project or application. This document is also known by the names SRS report and software document. A software document is primarily prepared for a project, software, or any kind of application. There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and nonfunctional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project, etc.

HARDWARE REQUIREMENTS

Ram : 8 GB (Preferable 16 GB)
Storage Size : 5 GB (Recommended 15 GB)
Storage type : HDD (Preferable SSD)
Processor : 2 Cores (Preferably 8 Core)
CPU : Intel i5 7th gen (Preferably i7 9th gen or later)
Ubey Key : yubikey 5NFC
Connectivity : Wifi (Preferably Ethernet)

SOFTWARE REQUIREMENTS

Operating System: Windows 7 or later versions of windows
Presentation Tier: HTML, CSS
Business Tier : Python (with Flask web framework)
Os : Linux or Windows(Preferably Linux red hat)
Language Support : Python 3.10 or later
Drivers : Latest drivers for every hardware component is must
User Role : User preferably Administrator
Auto Update : Always enabled and latest updates should be installed.

9 LITERATURE REVIEW

[1] A Novel Approach for Prediction of Cardio Vascular Disease: An Improved Genetic Algorithm Approach Using Classifiers:

The proposed methodology is to find cardiovascular heart diseases and detect disease by using data mining methods and genetic algorithm. Proposed ensemble structure contains three modules such as pre-processing and data acquisition. Classifier training for disease discovering as well as prediction with 3 layered strategy. Improved Genetic algorithm (IGA) model generates greater accuracy level for all heart diseases compared to other classifiers .
.

[2] Adaptive genetic algorithm with fuzzy logic (AGAFL):

The objective of adaptive genetic algorithm with fuzzy logic (AGAFL) model isto predict heart disease which will help medical practitioners in diagnosing heart disease at early stages. The 9 model consists of the rough sets-based heart disease feature selectionmodule and the fuzzy rule-based classification module. The generated rules from fuzzy classifiers are optimized by applying the adaptive genetic algorithm. There are two algorithms proposed in this paper one is for feature selection from dataset and another onefor classification of heart patients. First algorithm is genetic algorithm in which, fitness function, selection, crossover, mutation, accepting are submodules. Second algorithm is pseudocode consist of fuzzification, fuzzy rule generation, defuzzification as sub- modules.

[3] Heart Disease Prediction System using Genetic Algorithm:

In this Heart Disease Prediction System using Genetic Algorithm paper simple genetic algorithm is used without any other combination of other soft computing techniques. The data source we have used for experimental testing are commonly used and considered as a de facto standard for heart disease prediction reliability ranking. The results obtained show that Genetic Algorithm Technique are not only competitive with other evolutionary techniques, but also with industry standard algorithms, and can be successfully applied to heart disease prediction.

[4] Hybrid Architecture of Heart Disease Prediction System using Genetic Neural Network

The main objective of Hybrid Architecture of Heart Disease Prediction System using Genetic Neural Network is to develop a prototype of heart disease forecasting system using data mining and neural network concepts. It has the combination of genetic 10 algorithm and neural networks. The proposed system architecture has major components as follows: patient database, pre processing, tokenization, training the model, test the model, design fitness function, application of genetic algorithm, results collection and prediction of heart disease. Genetic algorithm is used to optimize the weights.

[5] Coronary Heart Disease prediction using genetic algorithm-based decision tree:

The aim of Coronary Heart Disease prediction using genetic algorithm-based decision tree is to support the doctors in taking decision to classify healthy and coronary heart disease (CHD) patients using popular modified decision tree by using genetic algorithm. Performance analysis of the proposed method is compared against datamining approach, probability rule base classification; Five machine-learning algorithms include K-Nearest Neighbour (KNN), artificial neural network, support vector machine (SVM), to classify dataset data processing is done alongside feature selection- maximal redundancy maximal relevancy(mRMR) feature, decision tree, and modified decision tree using genetic algorithm. Analysis was performed with reference to accuracy, precision, classification error, execution, and recall. Results show that the decision tree using genetic approach predicts the CHD patient more accurately than other existing algorithms.

[6] Prediction of heart disease using Artificial Neural Network:

In this Prediction of heart disease using Artificial Neural Network paper, they have compared the results of applying all the neural network techniques. The accuracy is calculated and visualized such as ANN gives 94.7% but with Principle Component 11 Analysis (PCA) accuracy rate improve to 97.7%. For the dataset selection they have used back propagation technique and to optimize the weights to reduce errors in the network. The collection of data sets, its pre processing, building patterning matching classifier model with ANN. After training them we will ensemble them to get the better decision. Principle component analysis (PCA) is used for transforming the big dimension to lower one. For this Correlation Matrix Eigenvalues and Eigen Vectors is calculated based on which new feature is extracted with more relevancy. The results are shown in the form of confusion matrix which can compare the various models in neural networks and by that confusion matrix we can review accuracy, sensitivity, precision and F1 score.

[7] Study and Analysis of Prediction Model for Heart Disease:

An Optimization Approach using Genetic Algorithm: The main idea behind writing this paper is to study diverse prediction models for the heart disease and selecting important heart disease feature using genetic algorithm. The optimized prediction models using genetic algorithm performance is better than traditional prediction models. The performances of the different prediction models retested with different heart disease data sets and validated with real-time data sets. The methodologies used in this paper are Random Forest, Support Vector Machine, Decision Tree, Naive Bayes Classifier, Genetic Algorithm, K-Cross Validation. Observation shown that in most of the cases Naive bayes classifier performance is having more accuracy compared with other three classification methods with respect to all the dataset. The main advantages of this paper are the integration genetic algorithm with prediction models improves the performances of the models.

[8] Identification of heart disease using fuzzy neural genetic algorithm with data mining techniques:

The Fuzzy validation of parameters of anomalous information smoothly decide the anomalies in the extensive size database. The presentation of fuzzy set idea effectively manages the vulnerability issues. The participation work in fuzzy sets achieves the qualities inside the threshold limit .The restorative information gathered from various instruments is indeterminate in nature. The information vulnerability causes the learning flaws to be specific, ambiguous and incoherent. Harsh set hypothesis is utilized to characterize the information.

[9] Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm:

13 One of the factors affecting the performance of artificial neural network is the initial weights utilized in the network structure. In this regard, the proposed model sought to ameliorate the performance of neural network through enhancing the primary weights used in it. In this study, the initial weights of neural network were identified via genetic algorithm.

[10] Design of Heart Disease Diagnosis System using Fuzzy Logic:

Proposed diagnosis system shows the fuzzy membership functions, which have been used in the implementation part, fuzzy expert system designing (ranges for all input attributes) and fuzzy rule base. The most important application of fuzzy expert system is an uncertainty problem. First step in fuzzy expert system designing is determination of input and output variables. The system consists of 10 input fields and 1 output field. The output field calculates the presence of heart disease in the patient . Fuzzy rule base is one of the most important part of fuzzy interface system and quality of the result is depends upon this rule base. Fuzzy rule base is the collection of single attribute or combination of attribute with AND/OR operators [1]. In the implementation part, 86 rules are used with the proper combination. In the rule base, the entire variable with the proper output fuzzy set value is used. For example, if value of BP, OP, CP is low then result is also “Healthy” but in case if values of BP, SCHOL, CP is high then disease condition is very high.

[11] Analysis of heart disease prediction system using artificial neural networks:

Until now, all the research methodology has been focused on clinical parameters ,but they can use additional attributes in consideration too in this research paper . They used Artificial neural network to predict heart disease.

[12] Detection of heart disease applying fuzzy logics and its comparison with neural networks:

The process starts with data processing. The second step includes the reduction of number of attributes for the processing in the fuzzy method . After the successful development of fuzzy rule model the early stage prediction will be made lastly, the result will be analysed and will be ended for development phase . Under this process the input data sets will be fuzzified, which will be pass to inference engine. Proper algorithm and decision will be made in accordance with the predefined

fuzzy rule base. Then these data set will be send for defuzzification and finally we will be obtaining our output. Fuzzification means fuzzifying all inputs and outputs .It means the degree of determining of which inputs and outputs belong to its corresponding fuzzy set.

[13] Enhanced decision tree algorithm using genetic algorithm for heart disease prediction:

In this paper the methodologies used are basic decision tree algorithm, genetic algorithm, enhanced decision tree algorithm based on genetic algorithm, fitness function for the rule, crossover and mutation operations for the rule. They proposed the enhanced decision tree based on genetic algorithm which takes advantage of genetic algorithm optimization ability.

[14] Cardiovascular disease prediction using genetic algorithm and neuro-fuzzy system:

In this paper, the heart disease prediction is done by using genetic algorithm and fuzzy system. The methodologies performed here are genetic algorithm, NFS(neuro fuzzy system), back propagation algorithm and feature selection. In this paper they applied NFS to the dataset which is nothing but the risk factors, for prediction and training of network will be done using back propagation algorithm and weight optimization will be done by genetic algorithm.

[15] Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks:

This Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks proposes a genetic algorithm (GA) based trained recurrent fuzzy neural networks (RFNN) to diagnosis of heart diseases. The results showed that 97.78% accuracy was obtained from testing set.

[16] A Computational Intelligence Method for Effective Diagnosis of Heart Disease using Genetic Algorithm:

In this paper a novel method for the diagnosis of heart disease has been proposed using Genetic Algorithms. In this approach an effective association rules are inferred using Genetic Algorithm approach which uses tournament selection, crossover, mutation and new proposed fitness function.

[17] A Comparative Study on Heart Disease Prediction Methods of ANFIS and GAFL:

In this paper for the prediction of heart disease, they used methods like ANFIS(Adaptive Neuro Fuzzy Inference System) and GAFL(Genetic Algorithm Fuzzy Logic model). This paper intended to discuss and compare, diagnosis and prediction system for heart disease based on soft computing technique that is ANFIS and GAFL model called Genetic Algorithm Fuzzy Logic model for effective heart disease prediction.

10 ARCHITECTURE OF THE SYSTEM DIAGRAM OF THE PROJECT

11 Flow Chart

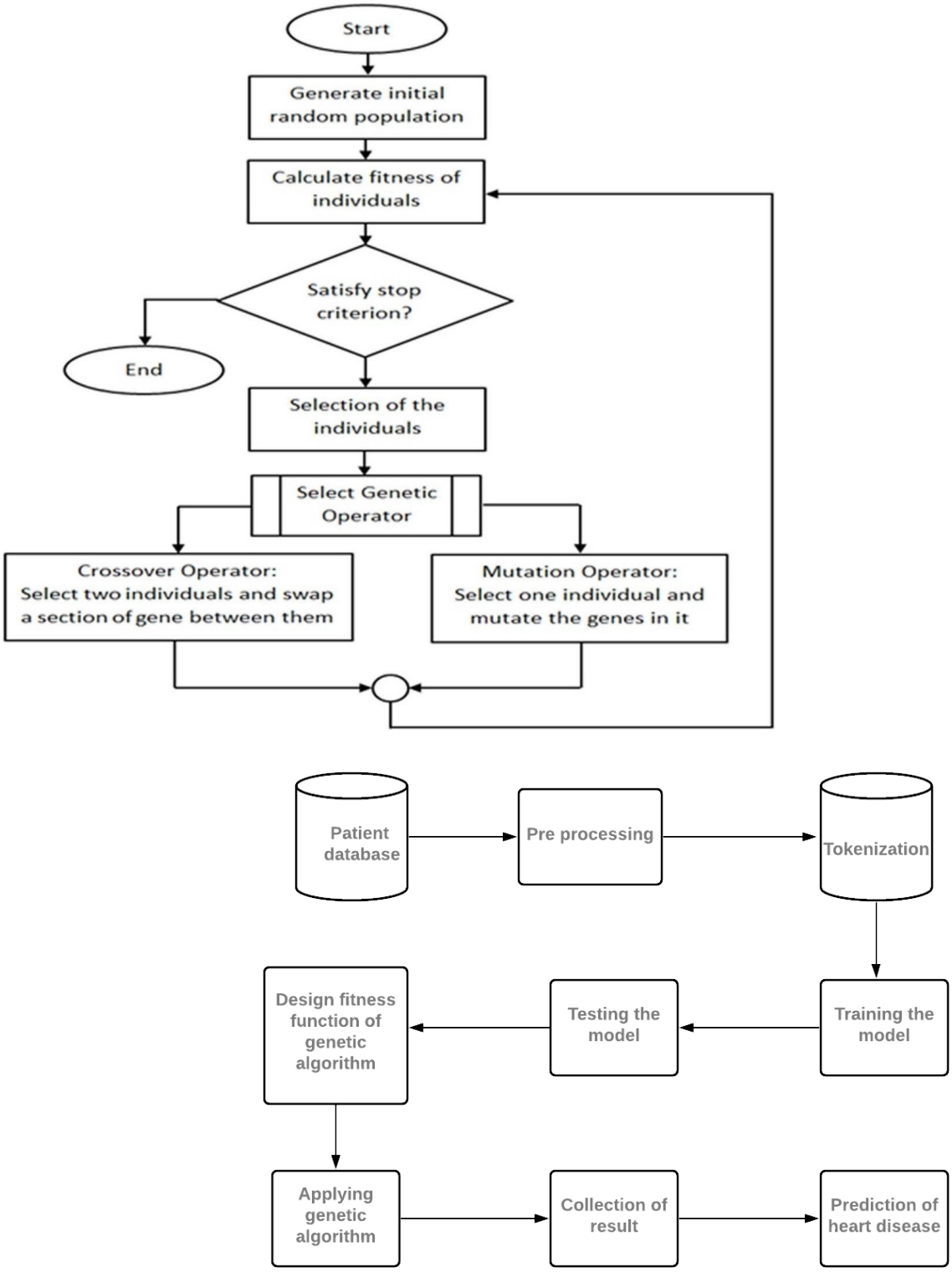


Figure 4 : Flow Chart for processing :

12 Language to be used

The code of genetic algorithms and artificial neural networks is in the Python programming language. We have used the Kagale notebook platform for writing and executing the code. Installing the latest Kagale ebook on an updated Windows 10 will help us import new libraries. Kagale is a project and community whose goal is to "develop open- source software, open-standards, and services for interactive computing across dozens of programming languages".

We will use Python as a language to develop this model. We will go with Kagel notebook for easier development and availability of the project

13 Algorithms used and processes followed in the model.

A.Neural Networks

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

How do neural networks work?

Think of each individual node as its own linear regression model, composed of input data, weights, a bias (or threshold), and an output. The formula would look something like this:

Mathematical formula used to determine summation

$\Sigma wixi + bias = w1x1 + w2x2 + w3x3 + bias$

Mathematical formula used to determine the output

$output = f(x) = 1 \text{ if } \Sigma w1x1 + b >= 0; 0 \text{ if } \Sigma w1x1 + b < 0$

Once an input layer is determined, weights are assigned. These weights help determine the importance of any given variable, with larger ones contributing more significantly to the output compared to other inputs. All inputs are then multiplied by their respective weights and then summed. Afterward, the output is passed through an activation function, which determines the output. If that output exceeds a given threshold, it “fires” (or activates) the node, passing data to the next layer in the network. This results in the output of one node becoming in the input of the next node. This process of passing data from one layer to the next layer defines this neural network as a feedforward network.

Most deep neural networks are feedforward, meaning they flow in one direction only, from input to output. However, you can also train your model through backpropagation; that is, move in the opposite direction from output to input. Backpropagation allows us to calculate and attribute the error associated with each neuron, allowing us to adjust and fit the parameters of the model(s) appropriately.

ii.Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems.It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

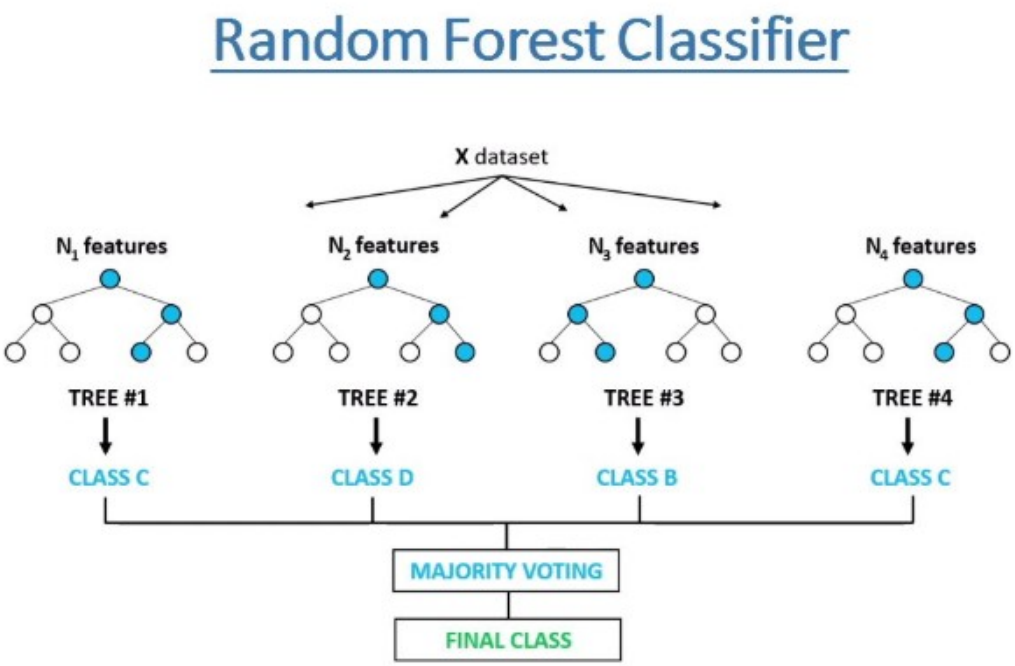
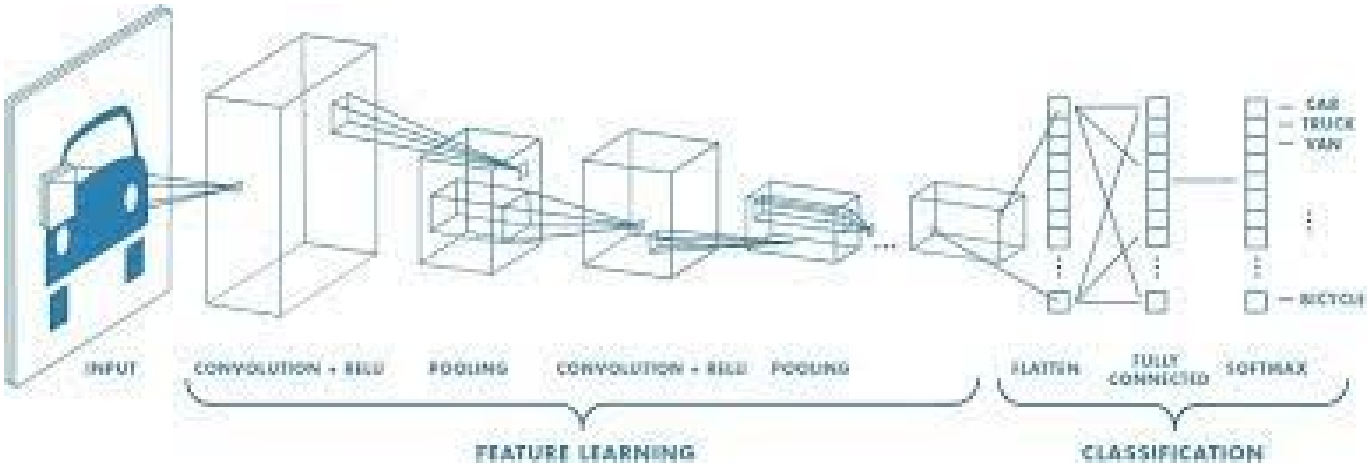


Figure 5 : Random Forest Classifier diagram

C. CNN

CNNs are used for image classification and recognition because of its high accuracy.It works by getting an image, designating it some weightage based on the different objects of the image, and then distinguishing them from each other.The main advantage of CNN is that it automatically detects the important features without any human supervision.



14 Process of heart diseases predictions

Patient database

Generally in hospitals certain patient databases are maintained for future use. We are using that data as our dataset from Kaggle website. This data is the main source of our project and with the help of database we will perform all the other operations in the flow chart.

Pre-processing

We have two types of attributes in the database; primary attributes(more important) and secondary attributes(less important). Through pre#processing we will refine the data by separating more important attributes from less one.

Tokenization

It is the process of turning sensitive data into non-sensitive data called "tokens" that can be used in a database or internal system without bringing it into scope. Tokenization can be used to secure sensitive data by replacing the original data with an unrelated value of the same length and format.

We will replace the fuzzy values of the data as crisp values and change the data into bit strings so that the data can be easily used in genetic algorithm.

Training the model

Training of the model is done by artificial neural network in which we will perform updation of weights with the help of old weights present in database. Then by using threshold value and activation function according to the data obtained we will compare and provide the output and updated weights as results.

Testing the model

Testing the gained results provide the accuracy of the model. We are performing testing through genetic.

GENETIC ALGORITHM

With the help of genetic algorithms, they enhanced or modified the decision tree, using genetic algorithms in decision tree functionality. The model consists of the rough sets-based heart disease feature selection module and the fuzzy rule-based classification module. The generated rules from fuzzy classifiers are optimized by applying the adaptive genetic algorithm. First, important features which effect heart disease are selected by rough set theory. The second step predicts the heart disease using the hybrid AGAFL classifier. The adaptive genetic algorithm (AGA) is an improved version of the genetic algorithm, in which, adaptive mutations are employed for achieving desired optimizing results.

A genetic algorithm uses mutations on each parent chromosome, resulting in random gene interchanging. The suggested adaptive mutation method bases the rate of mutation estimation on the fitness of the chromosome. The pace of mutation determines the performance of mutation. The fitness function is employed in genetic algorithms to develop optimized rules; the higher the fitness value, the better the rule, and vice versa. The performance of optimized prediction models based on genetic algorithms outperforms that of standard prediction models. It is a well-known method for resolving optimization challenges. It is also used to improve network characteristics like as learning rate, network momentum, and the number of MFs (Membership Functions) for each input. The Genetic Algorithm might be a faster and more accurate replacement for the present time-consuming procedures We can also identify the correct conclusion even if the dataset has less attributes. It is a practical and efficient method for both optimization and machine learning applications.

The genetic algorithm works well in noisy situations because it is stochastic and resilient. The genetic algorithm will assist to lower the error rate. The genetic algorithm employs

the notion of association rule mining to infer the best and most effective rules from the provided input data set. For the prediction of cardiac illness, we may employ decision trees, naïve bayes classifiers, support vendor machines, and other techniques in this genetic algorithm. These classification algorithms were chosen because they are often used in research and have the potential to provide accurate findings for heart disease prediction.

15 Input and output of the project

16 Dataset Information

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. To now, the Cleveland database is the only one that has been used by ML researchers. The "target" field refers to the patient's presence of cardiac disease. It has an integer value ranging from 0 (no presence) to 4. Experiments with the Cleveland database have mostly focused on attempting to identify presence (values 1,2,3,4) from absence (values 0). (Value 0). The patients' names and social security numbers were recently deleted from the database and replaced with fictitious values. One file, holding the Cleveland database, has been "processed." This directory contains all four unprocessed files.

This dataset might be useful for us to perform training and testing of the algorithm that we are going to implement . As the central objective of the project is heart disease detection , we definitely need a heart dataset . So, we considered this dataset .

17 Dataset Attributes

Only	14	Attributes	Full description
1	#3	(age)	Patient Age
2	#4	(sex)	Male/Female
3	#9	(cp)	Chest pain type
4	#10	(trestbps)	Resting blood pressure (in mm Hg on admission to the hospital)
5	#12	(chol)	Serum cholestoral (mg/dl)
6	#16	(fbs)	Fasting blood sugar
7	#19	(restecg)	Resting ECG results
8	#32	(thalach)	Maximum heart rate achieved
9	#38	(exang)	Exercise induced angina
10	#40	(oldpeak)	ST depression included by exercise relative to rest
11	#41	(slope)	Slope of the peak exercise ST segment
12	#44	(ca)	Number of major vessels (0-3) colored by fluoroscopy
13	#51	(thal)	3 = normal, 6 = fixed defect, 7 = reversible defect.
14	#58	(num)	Angiographic disease status (Diagnosis of heart disease)

Figure 6 : Dataset

Link

https://www.kaggle.com/rishidamarla/heart-disease-prediction

18 Implementation of the project Using ANN

In [1]:

```
# Importing teh nessacery module
import sys
import pandas as pd
import numpy as np
import sklearn
import matplotlib
import keras
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import seaborn as sns
cleveland = pd.read_csv('../input/heart-disease-prediction/Heart_Disease_Prediction.csv')
cleveland.loc[0:]
```

Out[1]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	Absence
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	Absence
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	Absence
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	Absence
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3	Presence

270 rows × 14 columns

In [2]:

```
# remove missing data (indicated with a "?")

data = cleveland[~cleveland.isin(['?'])]
data.loc[0:]
```

Out[2]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	Absence
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	Absence
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	Absence
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	Absence
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3	Presence

270 rows × 14 columns

In [3]:

```
# drop rows with NaN or undefined values from DataFrame

data = data.dropna(axis=0)
data.loc[0:]
```

Out[3]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	Absence
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	Absence
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	Absence
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	Absence
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3	Presence

270 rows × 14 columns

In [4]:

```
# print the shape and data type of the dataframe

print(data.shape)
print(data.dtypes)
```

```
(270, 14)
Age                         int64
Sex                         int64
Chest pain type             int64
BP                          int64
Cholesterol                 int64
FBS over 120                int64
EKG results                 int64
Max HR                      int64
Exercise angina             int64
ST depression             float64
Slope of ST                 int64
Number of vessels fluro     int64
Thallium                    int64
Heart Disease              object
dtype: object
```

In [5]:

```
# print data characteristics, usings pandas built-in describe() function

data.describe()
```

Out[5]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Nu ve
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.00000	270.000000	27
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630	1.05000	1.585185	0.6
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.470952	1.14521	0.614390	0.9
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.00000	1.000000	0.0
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000	0.00000	1.000000	0.0
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000	0.80000	2.000000	0.0
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000	1.60000	2.000000	1.0
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.20000	3.000000	3.0

In [6]:

```
# initiating a taget coloumn

data['target'] = 0

df = pd.DataFrame(data,columns=[
 'Age', 'Sex',
 'Chest pain type',
 'BP',
 'Cholesterol',
 'FBS over 120',
 'EKG results','Max HR',  'Exercise angina',  'ST depression',  'Slope of ST','Number of vessels fluro',  'Thallium','Heart D
isease','target'])

df.loc[df['Heart Disease']=='Presence','target']='1'
df.loc[df['Heart Disease']=='Absence','target']='0'

print(df)
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	\
0	70	1	4	130	322	0	2	
1	67	0	3	115	564	0	2	
2	57	1	2	124	261	0	0	
3	64	1	4	128	263	0	0	
4	74	0	2	120	269	0	2	
..	...	...	...	...	...	...	...	
265	52	1	3	172	199	1	0	
266	44	1	2	120	263	0	0	
267	56	0	2	140	294	0	2	
268	57	1	4	140	192	0	0	
269	67	1	4	160	286	0	2	

	Max HR	Exercise angina	ST depression	Slope of ST	\
0	109	0	2.4	2	
1	160	0	1.6	2	
2	141	0	0.3	1	
3	105	1	0.2	2	
4	121	1	0.2	1	
..	...	...	...	...	
265	162	0	0.5	1	
266	173	0	0.0	1	
267	153	0	1.3	2	

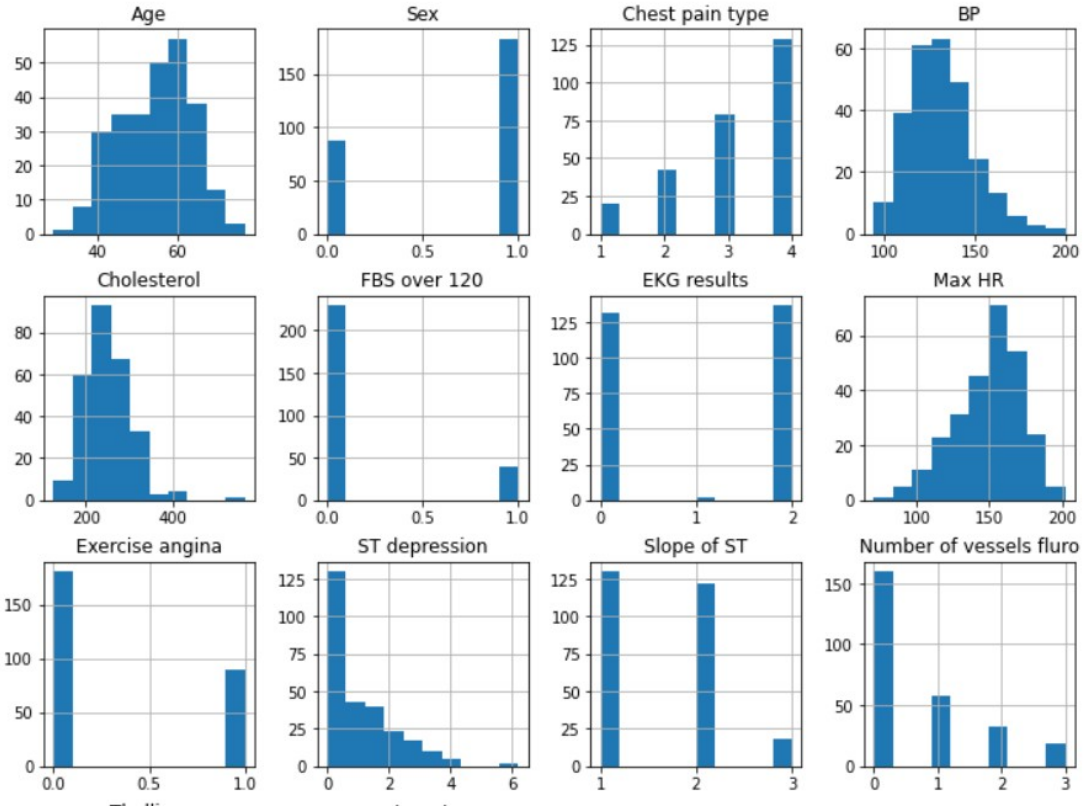
	Number of vessels fluro	Thallium	Heart Disease	target
0	3	3	Presence	1
1	0	7	Absence	0
2	0	7	Presence	1
3	1	7	Absence	0
4	1	3	Absence	0
..	...	...	...	...
265	0	7	Absence	0
266	0	7	Absence	0
267	0	3	Absence	0
268	0	6	Absence	0
269	3	3	Presence	1

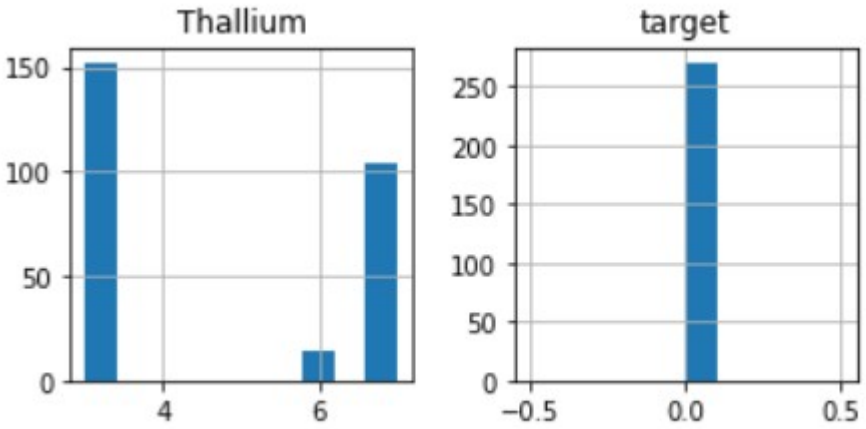
[270 rows x 15 columns]

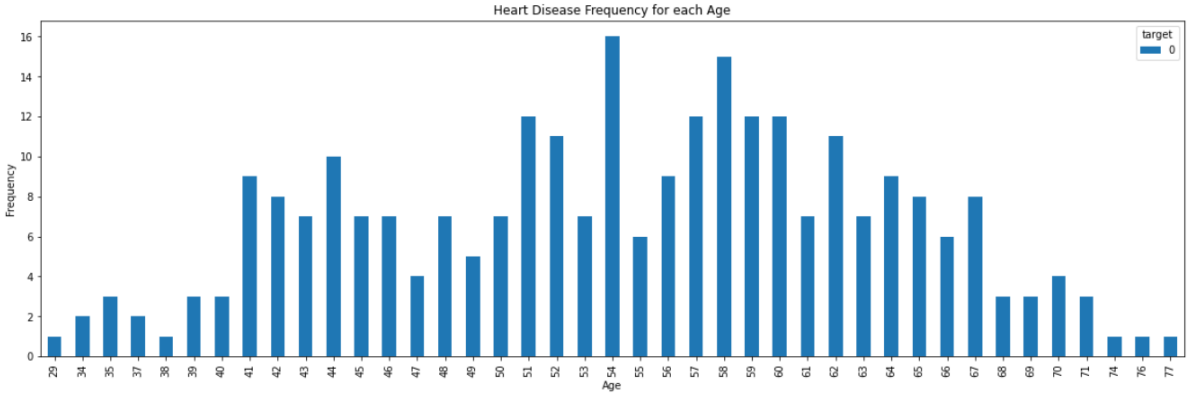
In [7]:

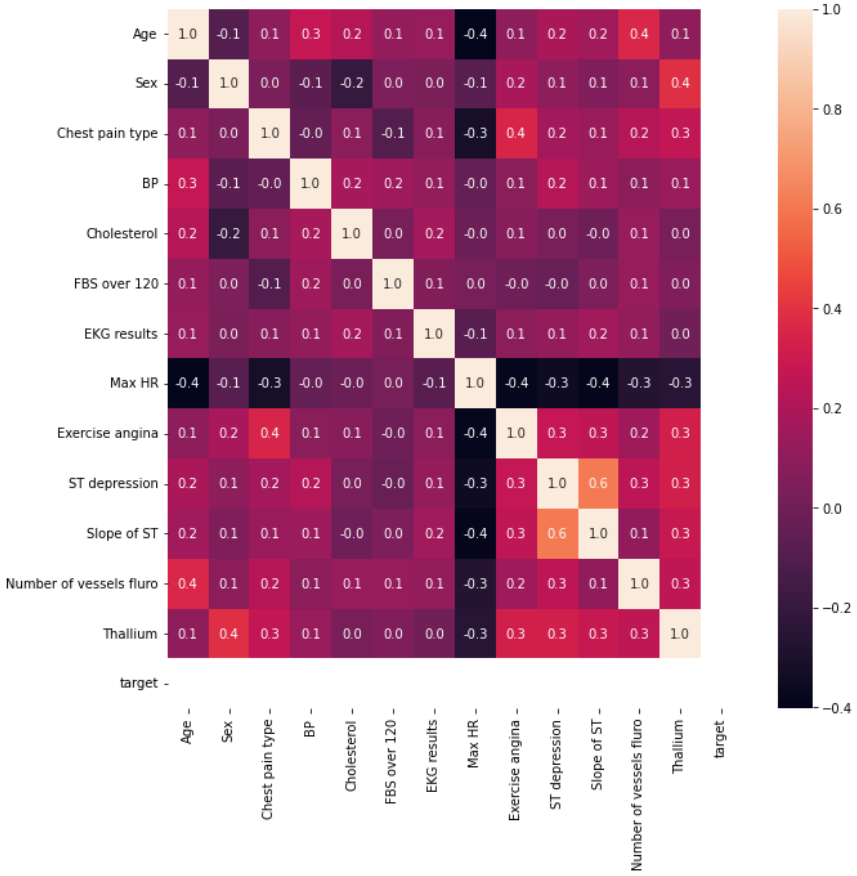
```
# plot histograms for each variable

data.hist(figsize = (12, 12))
plt.show()
pd.crosstab(data.Age,data.target).plot(kind="bar",figsize=(20,6))
plt.title('Heart Disease Frequency for each Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
plt.figure(figsize=(10,10))
sns.heatmap(data.corr(),annot=True,fmt='.1f')
plt.show()
```




Thallium
150
100
50
0
4
6
target
250
200
150
100
50
0
−0.5
0.0
0.5


Heart Disease Frequency for each Age
target
0
Frequency
Age


Age 1.0 -0.1 0.1 0.3 0.2 0.1 0.1 -0.4 0.1 0.2 0.2 0.4 0.1
Sex -0.1 1.0 0.0 -0.1 -0.2 0.0 0.0 -0.1 0.2 0.1 0.1 0.1 0.4
Chest pain type 0.1 0.0 1.0 -0.0 0.1 -0.1 0.1 -0.3 0.4 0.2 0.1 0.2 0.3
BP 0.3 -0.1 -0.0 1.0 0.2 0.2 0.1 -0.0 0.1 0.2 0.1 0.1 0.1
Cholesterol 0.2 -0.2 0.1 0.2 1.0 0.0 0.2 -0.0 0.1 0.0 -0.0 0.1 0.0
FBS over 120 0.1 0.0 -0.1 0.2 0.0 1.0 0.1 0.0 -0.0 -0.0 0.0 0.1 0.0
EKG results 0.1 0.0 0.1 0.1 0.2 0.1 1.0 -0.1 0.1 0.1 0.2 0.1 0.0
Max HR -0.4 -0.1 -0.3 -0.0 -0.0 0.0 -0.1 1.0 -0.4 -0.3 -0.4 -0.3 -0.3
Exercise angina 0.1 0.2 0.4 0.1 0.1 -0.0 0.1 -0.4 1.0 0.3 0.3 0.2 0.3
ST depression 0.2 0.1 0.2 0.2 0.0 -0.0 0.1 -0.3 0.3 1.0 0.6 0.3 0.3
Slope of ST 0.2 0.1 0.1 0.1 -0.0 0.0 0.2 -0.4 0.3 0.6 1.0 0.1 0.3
Number of vessels fluro 0.4 0.1 0.2 0.1 0.1 0.1 0.1 -0.3 0.2 0.3 0.1 1.0 0.3
Thallium 0.1 0.4 0.3 0.1 0.0 0.0 0.0 -0.3 0.3 0.3 0.3 0.3 1.0
target
Age
Sex
Chest pain type
BP
Cholesterol
FBS over 120
EKG results
Max HR
Exercise angina
ST depression
Slope of ST
Number of vessels fluro
Thallium
target
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

In [8]:
```
print(data)
```

```
     Age  Sex  Chest pain type   BP  Cholesterol  FBS over 120  EKG results  \
0     70    1                4  130          322             0            2
1     67    0                3  115          564             0            2
2     57    1                2  124          261             0            0
3     64    1                4  128          263             0            0
4     74    0                2  120          269             0            2
..   ...  ...              ...  ...          ...           ...          ...
265   52    1                3  172          199             1            0
266   44    1                2  120          263             0            0
267   56    0                2  140          294             0            2
268   57    1                4  140          192             0            0
269   67    1                4  160          286             0            2

     Max HR  Exercise angina  ST depression  Slope of ST  \
0       109                0            2.4            2
1       160                0            1.6            2
2       141                0            0.3            1
3       105                1            0.2            2
4       121                1            0.2            1
..      ...              ...            ...          ...
265     162                0            0.5            1
266     173                0            0.0            1
267     153                0            1.3            2
268     148                0            0.4            2
269     108                1            1.5            2

     Number of vessels fluro  Thallium Heart Disease  target
0                          3         3      Presence       0
1                          0         7       Absence       0
2                          0         7      Presence       0
3                          1         7       Absence       0
4                          1         3       Absence       0
..                       ...       ...           ...     ...
265                        0         7       Absence       0
266                        0         7       Absence       0
267                        0         3       Absence       0
268                        0         6       Absence       0
269                        3         3      Presence       0

[270 rows x 15 columns]
```

In [9]:
```
# Dropping the heart dieases coloumn

df= df.drop(['Heart Disease'], axis=1)
```

In [10]:
```
print(df)
X = np.array(df.drop(['target'], 1))
y = np.array(df['target'])

# create X and Y datasets for training

from sklearn import model_selection
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, stratify=y,random_state=42, test_size = 0.2)

from keras.utils.np_utils import to_categorical

Y_train = to_categorical(y_train, num_classes=None)
Y_test = to_categorical(y_test, num_classes=None)

print (Y_train.shape)
print (Y_train[:10])

X_train[0]
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	\
0	70	1	4	130	322	0	2	
1	67	0	3	115	564	0	2	
2	57	1	2	124	261	0	0	
3	64	1	4	128	263	0	0	
4	74	0	2	120	269	0	2	
..	...	...	...	...	...	...	...	
265	52	1	3	172	199	1	0	
266	44	1	2	120	263	0	0	
267	56	0	2	140	294	0	2	
268	57	1	4	140	192	0	0	
269	67	1	4	160	286	0	2	

	Max HR	Exercise angina	ST depression	Slope of ST	\
0	109	0	2.4	2	
1	160	0	1.6	2	
2	141	0	0.3	1	
3	105	1	0.2	2	
4	121	1	0.2	1	
..	...	...	...	...	
265	162	0	0.5	1	
266	173	0	0.0	1	
267	153	0	1.3	2	
268	148	0	0.4	2	
269	108	1	1.5	2	

	Number of vessels fluro	Thallium	target
0	3	3	1
1	0	7	0
2	0	7	1
3	1	7	0
4	1	3	0
..	...	...	...
265	0	7	0
266	0	7	0
267	0	3	0
268	0	6	0
269	3	3	1

[270 rows x 14 columns]

```
(216, 2)
[[1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [1. 0.]]
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:2: FutureWarning: In a future version of pandas all arguments
of DataFrame.drop except for the argument 'labels' will be keyword-only
```

Out[10]:

```
array([ 42.,   0.,   3., 120., 209.,   0.,   0., 173.,   0.,   0.,   2.,
         0.,   3.])
```

Training the data

In [11]:

```
# Importing different modules

from keras.models import Sequential
from keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from keras.layers import Dropout
from keras import regularizers
```

In [12]:

```
# define a function to build the keras model

def create_model():
 # create model
 model = Sequential()
 model.add(Dense(16, input_dim=13, kernel_initializer='normal',
kernel_regularizer=regularizers.l2(0.001), activation='relu'))
 model.add(Dropout(0.25))
 model.add(Dense(8, kernel_initializer='normal',
kernel_regularizer=regularizers.l2(0.001), activation='relu'))
 model.add(Dropout(0.25))
 model.add(Dense(2, activation='softmax'))
 # compile model
 adam = Adam(learning_rate=0.001)
 model.compile(loss='categorical_crossentropy', optimizer='rmsprop',
metrics=['accuracy'])
 return model
model = create_model()
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 16)                224
_________________________________________________________________
dropout (Dropout)            (None, 16)                0
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 136
_________________________________________________________________
dropout_1 (Dropout)          (None, 8)                 0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 18
=================================================================
Total params: 378
Trainable params: 378
Non-trainable params: 0
_________________________________________________________________
```

In [13]:

```
history=model.fit(X_train, Y_train, epochs=50, batch_size=10,validation_data=(X_test,Y_test))
```

```
2022-10-04 07:30:47.521127: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization
Passes are enabled (registered 2)


Epoch 1/50
22/22 [==============================] - 1s 16ms/step - loss: 0.8782 - accuracy: 0.5093 - val_loss: 0.6666 - val_accuracy:
0.5741
Epoch 2/50
22/22 [==============================] - 0s 4ms/step - loss: 0.7144 - accuracy: 0.5046 - val_loss: 0.6702 - val_accuracy:
0.6296
Epoch 3/50
22/22 [==============================] - 0s 4ms/step - loss: 0.7022 - accuracy: 0.5880 - val_loss: 0.6590 - val_accuracy:
0.5556
Epoch 4/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6775 - accuracy: 0.5833 - val_loss: 0.6530 - val_accuracy:
0.6296
Epoch 5/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6720 - accuracy: 0.5602 - val_loss: 0.6536 - val_accuracy:
0.5556
Epoch 6/50
```

```
------
Epoch 6/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6719 - accuracy: 0.5694 - val_loss: 0.6208 - val_accuracy:
0.6852
Epoch 7/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6374 - accuracy: 0.6435 - val_loss: 0.6144 - val_accuracy:
0.6481
Epoch 8/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6250 - accuracy: 0.6204 - val_loss: 0.5859 - val_accuracy:
0.6852
Epoch 9/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6557 - accuracy: 0.6435 - val_loss: 0.6391 - val_accuracy:
0.6111
Epoch 10/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6444 - accuracy: 0.6250 - val_loss: 0.6017 - val_accuracy:
0.7037
Epoch 11/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6331 - accuracy: 0.6435 - val_loss: 0.6044 - val_accuracy:
0.6852
Epoch 12/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6080 - accuracy: 0.6806 - val_loss: 0.5553 - val_accuracy:
0.7037
Epoch 13/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5929 - accuracy: 0.6481 - val_loss: 0.5492 - val_accuracy:
0.7778
Epoch 14/50
22/22 [==============================] - 0s 4ms/step - loss: 0.6127 - accuracy: 0.6713 - val_loss: 0.5337 - val_accuracy:
0.7222
Epoch 15/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5705 - accuracy: 0.6991 - val_loss: 0.5189 - val_accuracy:
0.7778
Epoch 16/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5477 - accuracy: 0.7222 - val_loss: 0.5186 - val_accuracy:
0.7593
Epoch 17/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5973 - accuracy: 0.6574 - val_loss: 0.5238 - val_accuracy:
0.7593
```

```
Epoch 18/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5304 - accuracy: 0.7315 - val_loss: 0.4928 - val_accuracy:
0.7222
Epoch 19/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5804 - accuracy: 0.7222 - val_loss: 0.5191 - val_accuracy:
0.7963
Epoch 20/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5679 - accuracy: 0.6806 - val_loss: 0.5142 - val_accuracy:
0.7407
Epoch 21/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5658 - accuracy: 0.7037 - val_loss: 0.4850 - val_accuracy:
0.7963
Epoch 22/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5886 - accuracy: 0.7361 - val_loss: 0.5009 - val_accuracy:
0.7963
Epoch 23/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5706 - accuracy: 0.6898 - val_loss: 0.5296 - val_accuracy:
0.7407
Epoch 24/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5968 - accuracy: 0.7269 - val_loss: 0.4739 - val_accuracy:
0.7963
Epoch 25/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5454 - accuracy: 0.7222 - val_loss: 0.4745 - val_accuracy:
0.7778
Epoch 26/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5486 - accuracy: 0.7222 - val_loss: 0.4776 - val_accuracy:
0.7778
Epoch 27/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5269 - accuracy: 0.7685 - val_loss: 0.5729 - val_accuracy:
0.6852
Epoch 28/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5534 - accuracy: 0.7407 - val_loss: 0.4598 - val_accuracy:
0.8333
Epoch 29/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5339 - accuracy: 0.7315 - val_loss: 0.5501 - val_accuracy:
0.6852
Epoch 30/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5693 - accuracy: 0.7222 - val_loss: 0.4312 - val_accuracy:
0.7963
```

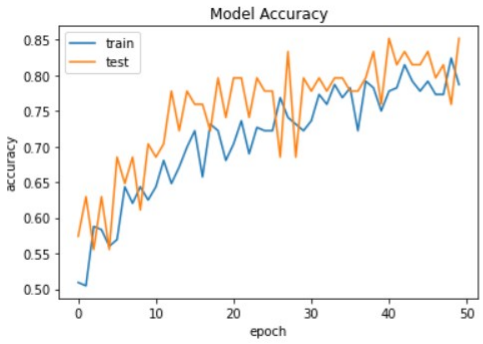
```
Epoch 31/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5450 - accuracy: 0.7361 - val_loss: 0.4499 - val_accuracy:
0.7778
Epoch 32/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5107 - accuracy: 0.7731 - val_loss: 0.4679 - val_accuracy:
0.7963
Epoch 33/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5353 - accuracy: 0.7593 - val_loss: 0.4376 - val_accuracy:
0.7778
Epoch 34/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5304 - accuracy: 0.7870 - val_loss: 0.4465 - val_accuracy:
0.7963
Epoch 35/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5170 - accuracy: 0.7685 - val_loss: 0.4772 - val_accuracy:
0.7963
Epoch 36/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5192 - accuracy: 0.7824 - val_loss: 0.4467 - val_accuracy:
0.7778
Epoch 37/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5281 - accuracy: 0.7222 - val_loss: 0.4784 - val_accuracy:
0.7778
Epoch 38/50
22/22 [==============================] - 0s 3ms/step - loss: 0.4767 - accuracy: 0.7917 - val_loss: 0.4282 - val_accuracy:
0.7963
Epoch 39/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5083 - accuracy: 0.7824 - val_loss: 0.4336 - val_accuracy:
0.8333
Epoch 40/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5061 - accuracy: 0.7500 - val_loss: 0.4842 - val_accuracy:
0.7593
Epoch 41/50
22/22 [==============================] - 0s 3ms/step - loss: 0.4917 - accuracy: 0.7778 - val_loss: 0.4035 - val_accuracy:
0.8519
Epoch 42/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5015 - accuracy: 0.7824 - val_loss: 0.4257 - val_accuracy:
0.8148
Epoch 43/50
22/22 [==============================] - 0s 3ms/step - loss: 0.4802 - accuracy: 0.8148 - val_loss: 0.3951 - val_accuracy:
0.8333
```

```
Epoch 44/50
22/22 [==============================] - 0s 3ms/step - loss: 0.4658 - accuracy: 0.7917 - val_loss: 0.4066 - val_accuracy:
0.8148
Epoch 45/50
22/22 [==============================] - 0s 3ms/step - loss: 0.4917 - accuracy: 0.7778 - val_loss: 0.4119 - val_accuracy:
0.8148
Epoch 46/50
22/22 [==============================] - 0s 3ms/step - loss: 0.4825 - accuracy: 0.7917 - val_loss: 0.3972 - val_accuracy:
0.8333
Epoch 47/50
22/22 [==============================] - 0s 3ms/step - loss: 0.5163 - accuracy: 0.7731 - val_loss: 0.4215 - val_accuracy:
0.7963
Epoch 48/50
22/22 [==============================] - 0s 4ms/step - loss: 0.5050 - accuracy: 0.7731 - val_loss: 0.4381 - val_accuracy:
0.8148
Epoch 49/50
22/22 [==============================] - 0s 8ms/step - loss: 0.4861 - accuracy: 0.8241 - val_loss: 0.4912 - val_accuracy:
0.7593
Epoch 50/50
22/22 [==============================] - 0s 6ms/step - loss: 0.4468 - accuracy: 0.7870 - val_loss: 0.3859 - val_accuracy:
0.8519
```

In [14]:

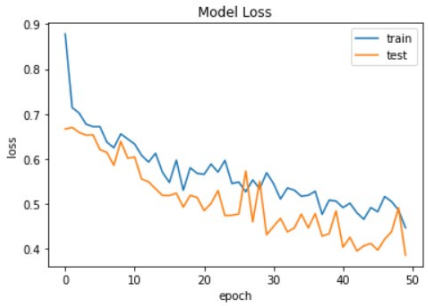
```
# Plotting the Model accuracy

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'])
plt.show()
```



In [15]:

```
# Plotting the Model Losss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'])
plt.show()
```



20 Output

Genral Classification result

In [16]:
```
# generate classification report using predictions for ANN

from sklearn.metrics import classification_report, accuracy_score
categorical_pred = np.argmax(model.predict(X_test), axis=1)
print('Results for Neural network Model')

y_test_int = y_test.astype('int')

print(accuracy_score(y_test_int,categorical_pred))
print(classification_report(y_test_int, categorical_pred))
```

```
Results for Neural network Model
0.8518518518518519
              precision    recall  f1-score   support

           0       0.87      0.87      0.87        30
           1       0.83      0.83      0.83        24

    accuracy                           0.85        54
   macro avg       0.85      0.85      0.85        54
weighted avg       0.85      0.85      0.85        54
```

Finished with ANN

Random Forest

[25]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
```

[26]:
```
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
```

[26… RandomForestClassifier()

[23]:
```
pred = rf.predict(X_test)
```

[27]:
```
cm = confusion_matrix(y_test,pred)
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.86      0.83      0.85        30
           1       0.80      0.83      0.82        24

    accuracy                           0.83        54
   macro avg       0.83      0.83      0.83        54
weighted avg       0.83      0.83      0.83        54
```

21 Result Analysis

After we see the accuracy score from few of the algorithm we can conclude that it will be useful to use the genetic algorithm for predicting the cardiovascular heart disease as the accuracy of the algorithm is quite high compare to simple ANN and the other algorithm we see this by the accuracy score of Genetic Algorithm that is 0.87 and the accuracy score of 0.85 as it being in medical fied even a little more accuracy matters towards the test analysis of the reports. Hence we can say that Genetic Algorithm is useful

22 Conclusion

In conclusion, cardiovascular disease prediction using artificial neural networks is a powerful tool for predicting heart disease at a cheaper cost. The detection of such cardiovascular diseases, or heart diseases, may be accomplished through the use of common symptoms such as regular illness, or it may be predicted through the use of risk factors such as age, family history, diabetes, hypertension, high cholesterol, tobacco smoking, alcohol intake, obesity, or physical inactivity, and also they are able to take into account a variety of risk factors and make accurate predictions. This tool is essential for diagnosing heart disease at an early stage, when it is most treatable. Hence predicting the disease before it becomes a huge problem. Using AI will also help the doctors and medical people as it will be easy and accurate for them to analyze the lab reports and make an accurate prediction of the diseases.

23 Future Studies

For future studies it was found that the dataset should be normalized; otherwise, the training model gets overfitted sometimes and the accuracy achieved is not sufficient when a model is evaluated for real-world data problems which can vary drastically to the dataset on which the model was trained. It was also found out that the statistical analysis is also important when a dataset is analyzed. The difficulty which came here is that the sample size of the dataset is not large. If a large dataset is present, the results can increase very much in deep learning and ML as well. The algorithm applied by us in ANN architecture increased the accuracy. The dataset size can be increased and then deep learning with various other optimizations can be used and more promising results can be achieved. Machine learning and various other optimization techniques can also be used so that the evaluation results can again be increased. More different ways of normalizing the data can be used and the results can be compared. And more ways could be found where we could integrate heart-disease-trained ML and DL models with certain multimedia for the ease of patients and doctors.

24 Base Paper / Reference Paper

Reference paper

[1] Reddi, Sivaranjani & srinivasa naresh, Vankamamidi & Murthy, Nistala V.E.S.. (2019). Coronary Heart Disease prediction using genetic algorithm based decision tree. 10.1515/9783110621105-004.

[2] Gadekallu, Thippa & Reddy, Praveen & Lakshman, Kuruva & Rajput, Dharmendra & Kaluri, Rajesh & Srivastava, Gautam. (2020). Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. Evolutionary Intelligence. 13. 10.1007/s12065-019-00327-1.

[3] Bano, Shaikh. (2019). Heart Disease Prediction System using Genetic Algorithm. International Journal for Research in Applied Science and Engineering Technology. 7. 2178-2182. 10.22214/ijraset.2019.6366.

[4] Kumkum Chaudhary, Radhika Naidu, Rhea Rai, Narendra Gawai. (2019). Hybrid Architecture of Heart Disease Prediction System using Genetic Neural Network.
V6/i5/IRJET-V6I5857

[5] Uyar, Kaan & Ilhan, Ahmet. (2017). Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks. Procedia Computer Science. 120. 588-593. 10.1016/j.procs.2017.11.283.

[6] Awan, Shahid & Riaz, Muhammad & Khan, Abdul. (2018). PREDICTION OF HEART DISEASE USING ARTIFICIAL NEURAL NETWORK. 13. 102-112.

[7] Prasadgouda B Patil, Dr. P Mallikarjun Shastry, Dr Ashokumar P S. (2020). A Novel Approach for Prediction of Cardio Vascular Disease: An Improved Genetic Algorithm Approach Using Classifiers. International Journal of Advanced Science and Technology, 29(7s), 4493 – 4504.

[8] Srikanth Meda1 , Raveendra Babu Bhogapathi 2 1 Research scholar, Acharya Nagarjuna University, Guntur & Associate Professor in the Department of Computer Science and Engineering at R.V.R. & J.C. College of Engineering, Guntur 522019, India 2 Professor, Department of Computer Science and Engineering, R.V.R & J.C College of Engineering, Guntur 552019, India

[9] Akruti Dave, Prof. Gayatri Pandi , Master of Engineering Student, Head of Department Department of Computer Engineering ,L. J Institute of Engineering &Technology (Gujarat Technological University), Ahmedabad, India

[10] Kasbe, Tanmay & Pippal, Ravi. (2017). Design of heart disease diagnosis system using fuzzy logic. 3183-3187. 10.1109/ICECDS.2017.8390044.

[11] Zeinab Arabasadi , Roohallah Alizadehsani , Mohamad Roshanzamir , Hossein Moosaei , Ali Asghar Yarifard , Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm , Computer Methods and Programs in Bio medicine , Volume 141,2017,Pages 19-26,ISSN 0169-2607.

[12] Zabeen, Ashiya & Utsav, Ankur & Lal, Kanhaiya. (2018). Detection of Heart Disease Applying Fuzzy Logics and Its Comparison with Neural Networks. 461-467. 10.1109/RTEICT42901.2018.9012315.

[13] Kumar, S., & Sahoo, G. (2018). Enhanced decision tree algorithm using genetic algorithm for heart disease prediction. International Journal of Bioinformatics Research and Applications, 14(1-2), 49-69.

[14] Nikam, S., Shukla, P., & Shah, M. (2017). Cardiovascular disease prediction using genetic algorithm and neuro-fuzzy system.

[15] Kumar, P. S., Anand, D., Kumar, V. U., Bhattacharyya, D., & Kim, T. H. (2016). A computational intelligence method for effective diagnosis of heart disease using genetic algorithm. International Journal of Bio-Science and Bio-Technology, 8(2), 363-372.

[16] Salem, T. (2018). Study and analysis of prediction model for heart disease: an optimization approach using genetic algorithm. International Journal of Pure and Applied Mathematics, 119(16), 5323-5336.

[17] Santhanam, T., & Ephzibah, E. P. (2015). Heart disease prediction using hybrid genetic fuzzy model. Indian Journal of Science and Technology, 8(9), 797.

BASE PAPER

https://link.springer.com/article/10.1007/s12553-020-00508-4