PES UNIVERSITY

(Established under KarnatakaAct No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

Report on

‘Implementation of an IoT based Healthcare Kit’

Submitted by

Sinchitha H V (PES1UG19EC301)
Tanya Chanchalani (PES1UG19EC326)
Bhushan Kiran Munoli (PES1UGEC359)
Gaurav R (PES1201801939)

January - May 2022 & August - December 2022

Under the guidance of

Dr.Purushotham U
Associate Professor Department
of ECE PES University

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGG

PROGRAM B.Tech

CERTIFICATE

This is to certify that the Report entitled

‘Implementation of an IoT-based Healthcare Kit’

is a bonafide work carried out by

Sinchitha H V (PES1UG19EC301)

Tanya Chanchalani (PES1UG19EC326)

Bhushan Kiran Munoli (PES1UG19EC359)

Gaurav R (PES1201801939)

In partial fulfilment for the completion of 8th-semester course work in the Program of Study B.Tech in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period Jan–Dec 2022. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the 8th-semester academic requirements in respect of Capstone project work.

Signature with the date and Seal	*Signature with the date and Seal*	*Signature with the date and Seal*
Dr Purushotham U	*Dr Anuradha M*	*Dr. B. K. Keshavan*
Internal Guide	*Chairperson*	*Dean - Faculty of Engg. & Technology*

Name of the students:

Sinchitha H V (PES1UG19EC301)

Tanya Chanchalani (PES1UG19EC326)

Bhushan Kiran Munoli (PES1UGEC359)

Gaurav R (PES1201801939)

Name / Signature of Examiners:

Name Signature

DECLARATION

We, *Sinchitha H V, Tanya Chanchalani, Bhushan Kiran Munoli, and Gaurav R***,** hereby declare that the report entitled ***"Implementation of an IoT based Healthcare Kit***" is an original work done by us under the guidance of **Dr Purushotham U**, ***Associate Professor*****, ECE Department,** and is being submitted in partial fulfilment of the requirements for completion of 8th-semester course work in the Program of Study, B.Tech in Electronics and Communication Engineering.

PLACE: Bangalore
DATE: 15/12/2022

NAME AND SIGNATURE OF THE CANDIDATES

1. Sinchitha H V

2. Tanya Chanchalani

3. Bhushan Kiran Munoli

4. Gaurav R

ACKNOWLEDGEMENT

Firstly, we would like to sincerely thank our guide, Dr Purushotam U, Associate Professor, Department of Electronics and Communication, PES University, for providing guidance to help us think in the right direction and aiding us in the successful completion of our project work.

We extend our heartfelt thanks to **Dr Suryaprasad J**, Vice-Chancellor of PES University, for his support. We are also grateful to **Dr M. Anuradha**, Chairperson, Department of Electronics and Communication, PES University, for her support in completing this project.

Finally, we thank our panel members, our parents, our friends, and all the other people who were directly or indirectly involved for their cooperation and support in guiding and supporting us in the completion of this project.

Sinchitha H V (PES1UG19EC301)

Tanya Chanchalani (PES1UG19EC326)

Bhushan Kiran Munoli (PES1UGEC359)

Gaurav R (PES1201801939)

ABSTRACT

Cardiovascular diseases and Cardiac Arrhythmia are the most common cause of death worldwide over the last few decades in the developed as well as underdeveloped and developing countries. Early detection of these diseases and continuous supervision by clinicians can reduce the mortality rate.

However, it is not possible to monitor patients every day in all cases accurately, and consultation of a patient for 24 hours by a doctor is not available since it requires more patience, time, and expertise.

In this project, we have used ECG sensors, Arduino, and Raspberry Pi to obtain certain parameters pertaining to cardiovascular disease. We have developed machine learning models based on K-Nearest Neighbour, Logistic Regression, Support Vector Machine, and Random Forest for heart disease prediction through the various heart attributes of patients and to detect these impending heart diseases. The datasets that are used in this project are available publicly on the UCI website.

The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and, in turn, reduce complications, which can be a great milestone in the field of medicine.

Table of contents

1. Introduction **9**

1.1 Common Heart Disease **9**

1. Coronary Artery Disease 9
2. Cardiac Arrhythmia 10
3. Heart Attack 10
4. Heart Failure 10
5. Heart Valve Disease 11
6. Cardiomyopathy 11
7. Angina 12

1.2 Existing IoT Devices **13**

1. Holter Monitor 13
2. Zio XT patch 13
3. KardiaBand 13
4. BodyGuardian Heart 14
5. Implantable Cardio Defibrillator (ICD) 14
6. Pacemaker 14

2. Problem Statement **16**

2.1 Internet of Things: challenges and opportunities 16

2.2 The Goal 17

3. Literature Survey **18**

3.1 Deep Neural Network Based Real-Time Multi-Class Arrhythmia Classification in IoT-Cloud Platform 18

3.1.1 Introduction 18

3.1.2 Literature Review 18

3.1.3 Related work 19

a). MIT - BIH Arrhythmia Data-Set 19

b). Real-Time Signal Acquisition 20

c). Signal Processing 20

c.i). Denoising 20
c.ii) Peak detection 21
c.iii) Signal Segmentation and Normalisation 21
d). Deep Neural Network(DNN) 22
e). The cloud data storage and accessibility 23

3.1.4 Conclusion 23

3.2 IoT-Based Health Monitoring System by Using Raspberry Pi and ECG signal 24

3.2.1 Introduction 24
3.2.2 Related work 24
a) Arduino UNO 24
b).Raspberry pi-4 24
c). Microcontroller 24
d). Accelerometer sensor 24
e). ECG sensor(AD8232) 24
3.2.3 Conclusion 25

3.3 Early Detection of Cardiac Arrhythmia Disease using Machine Learning And IoT Technologies 25
3.3.1 Introduction 25
3.3.2 Related Work 25
3.3.2.1 ECG accretion 25
3.3.2.2 QRS detection algorithm 26
3.3.2.3 Extraction of features 26
3.3.3 Results 26

3.4 IoT Based Wearable Monitoring structure for detecting Abnormal Heart. 27
3.4.1 Introduction 27
3.4.2 Working and flow of the data 27
3.4.3 Signal collection and Processing 27
3.4.4 Storage and analysis 28
3.4.5 Warnings related to abnormalities. 28

3.4.6 Communication devices used for simulation 28
3.4.6.1 ZigbeeModule(Zonal Intercommunication Global-standard) 28
3.4.6.2 Bluetooth module 28
3.4.6.3 Wifi module: 29
3.4.6.4 Hydrogel Gel Electrodes and Sensors 29
3.4.7 Results 30

4 Implementation 31
4.1 Hardware Architecture 31
4.1.1 Components Used 31
1 .ECG sensor(AD8232) 31
2 .Arduino Uno 32
3. Raspberry pi-4 33
4.1.2 Protocol used for hardware communication 35
1. UART Protocol 35
4.1.3 Libraries used in hardware 37
1. Neurokit 2 37
2. Serial 38
3. Scipy 38
4.2 Construction of Machine Learning Models 38
4.2.1 Logistic Regression 39
4.2.2 K - Nearest Neighbour Classifier 41
4.2.3 Random Forest Classifier 43
4.2.4 Support Vector Classifier 45
4.3 Implementations details
4.3.1 Software Architecture 48
1. Arduino IDE 48
2.Python 48
3. Flask 49
4. Jupyter Notebook 50
5.HTML & CSS 50
6.Jinja 2 50
7. Gunicorn 50
8.MarkupSafe 50

4.4 Project Workflow 51
4.4.1 Hardware Explanation 51
4.4.1.1 Butterworth Filters 55
4.4.2 Software Explanation 71
4.4.3 Building GUI

5. Results **74**
5.1 Simulated output 74
5.1.1 ECG output waveform 74
5.1.2 Machine learning model output to predict Cardiovascular Disease(CVD) 75
5.1.3 Machine learning model output to predict Cardiac Arrhythmia 75
5.1.4 Accuracy Outputs of machine learning algorithms used to predict Cardiovascular Disease 76
5.1.5 Accuracy Outputs of machine learning algorithms used to predict Cardiovascular Disease 76
5.1.6 GUI Output of Cardiovascular Disease(CVD) and Cardiac Arrhythmia 77
5.1.7 Negative case GUI output of Cardiovascular Disease(CVD) 77
5.1.8 Positive case GUI output of Cardiovascular Disease(CVD) 78
5.1.9 Negative case GUI output of Cardiac Arrhythmia 78
5.1.10 Positive case GUI output of Cardiac Arrhythmia 79

6. Final Conclusion and Future Perspectives **80**
6.1 Conclusion 80
6.2 Future Scope 80
7. References **81**
8. Appendix **82**

List of Figures

Fig 1.1(a)	The four Heart Valves
Fig 1.1(b)	Coronary Artery disease
Fig 1.1(c)	Cardiac Arrhythmia
Fig 1.1(d)	Heart Failure
Fig 1.1(e)	Cardiomyopathy
Fig 1.2(a)	Zio XT Patch
Fig 1.2(b)	KardiaBand(Top Left -Top Right)
Fig 1.2(c)	Holter Monitor
Fig 1.2(d)	Body Guardian
Fig 1.2(e)	Pacemaker
Fig 1.2(f)	Implantable Cardiac Defibrillation(ICD)
Fig 3.1	Diagram with signal processing steps and proposed DNN
Fig 3.2	ECG sensor(AD8232) with three leads and three electrode Patches made up of Ag/Agcl
Fig 3.3	Band pass filter recording with the frequency set to 0.5Hz
Fig 3.4	An overview of the proposed DNN model
Fig 3.5	Graphical Representation of ReLU activation function
Fig 3.6	ECG accretion
Fig 3.7	Signal processing and feature extraction
Fig 3.8	Zig bee module
Fig 3.9	Bluetooth module
Fig 3.10	Wifi module
Fig 3.11	ECG patches with hydrogel electrodes
Fig 4.1	ECG Sensor(AD8232)
Fig 4.2	Arduino Uno R3 ATmega328

Fig 4.3	Raspberry Pi 4 model with 8GB RAM
Fig 4.4	UART device connections
Fig 4.5	UART data frame
Fig 4.6	Representation of simplex, half duplex and full duplex communication
Fig 4.7	Graph of Logistic Regression
Fig 4.8	Working of KNN
Fig 4.9	Working of Random Forest Algorithm
Fig 4.10(a)	Original Dataset
Fig 4.10(b)	Data with Decision Boundary
Fig 4.10(c)	Optimal Hyperplane
Fig 4.11(a)	Original Dataset
Fig 4.11(b)	Adding 3rd Dimension
Fig 4.11(c)	Obtaining Hyperplane
Fig 4.11(d)	Hyperplane mapped to 2D
Fig 4.12	Block Diagram of Project
Fig 4.13	Einthoven's Triangle Representation
Fig 4.14	Circuit Diagram
Fig 4.15 (a)	Raw Signal received from Sensor
Fig 4.15 (b)	Baseline wandering removed and Zoomed Signal
Fig 4.15 (c)	Powerline and Muscle noise removed and zoomed signal
Fig 4.16	Nth order Butterworth Formula
Fig 4.17 (a)	Butterworth Ideal Frequency Response
Fig 4.17 (b)	Frequency Response (ECG)
Fig 4.18	P, Q, R, S and T peaks marked in the completely Processed Signal
Fig 4.19	Slope Formula
Fig 4.20	Summary of Dataset

Fig 4.21	Null Values Heatmap
Fig 4.22	Samples in Each Class
Fig 4.23	Percentage of Positive and Negative samples
Fig 4.24	Distribution of Features
Fig 4.25	Correlation Matrix
Fig 4.26	Classification Report and Confusion Matrix for Logistic Regression
Fig 4.27	Classification Report and Confusion Matrix for KNN
Fig 4.28	Classification Report and Confusion Matrix for SVM
Fig 4.29	Classification Report and Confusion Matrix for Random Forest
Fig 4.30	Classification Report and Confusion Matrix for Decision Tree
Fig 4.31	Model Accuracy of SVM with Random Forest classifier
Fig 4.32	GUI to Detect Cardiovascular disease
Fig 4.33	Home Page of GUI to Detect Cardiac Arrhythmia
Fig 4.34	Page with Text Field and Results
Fig 5.1	Graph representing the plot of slopes extracted from ECG
Fig 5.2	Accuracy and classification report of Support Vector Classifier
Fig 5.3	Precision and Recall of SVM Kernel
Fig 5.4	Cardiovascular Disease Predictor for Negative case
Fig 5.5	Cardiovascular Disease Predictor for Positive cases
Fig 5.6	Detection of Arrhythmia for Negative case
Fig 5.7	Detection of Arrhythmia for Positive case

List of Tables

Table 4.1	Arduino Specification
Table 4.2	Raspberry Pi Specifications
Table 4.3	List of all libraries used in this project
Table 4.4	Implementation of ML Algorithms on UCI Dataset for CVD
Table 4.5	Implementation of ML Algorithms on UCI Dataset for Cardiac Arrhythmia
Table 5.1	Tabular Column of Cardiovascular Disease
Table 5.2	Tabular Column of Cardiac Arrhythmia

1. Introduction

The heart is a very prevalent organ in our body. It is a part of the cardiovascular system, which also consists of various arteries, veins, and capillaries. Absurd heartbeats and unusual blood flow from the heart are the major reasons for a person to develop heart disease. Approximately 17 million deaths occur annually due to heart abnormalities, as stated by the World Health Organization.

In this chapter, we give a fair introduction to certain types of heart diseases that exist and which are the most common among heart patients. The next subsection gives an introduction to the different types of heart diseases that exist.

1.1. Common Heart Diseases:

For the last 2-3 decades, the frequency of deaths resulting from heart diseases has gone up exponentially. In 2019, there were expected to be 25.6 lakh deaths due to cardiovascular disease in India. This was also the major cause of death in India.

One in four deaths that happen in India occurs due to one of the heart diseases. There are around seven kinds of heart diseases. The number of lives lost attributable to heart diseases in India has increased by 59% in the last two decades.

Cardiovascular Disease (CVD) is a term given to a bunch of diseases connected to the unusual condition of the heart and blood vessels involved. The common heart diseases that are seen among heart patients are briefed below.

1. Coronary Artery Disease:

This is a very familiar type of heart condition in India. This occurs due to blockage of coronary arteries. These are the blood vessels that deliver oxygen-rich blood to the heart for it to function.

CAD causes blockages in these arteries and makes them hard, which leads to a decreased supply of oxygenated blood to the heart. This causes a high-risk condition that may lead to severe chest pain or sometimes even a heart attack. This disease is usually seen in people who are aged around 60 having diabetes, high blood pressure and cholesterol.

2. Cardiac Arrhythmia:

This condition of the heart is due to abnormal heart rhythms, or in other terms, irregular heartbeat patterns. Based on these irregular patterns, there are two types of arrhythmia namely Tachycardia and Bradycardia. Tachycardia occurs due to a very fast heartbeat rate(usually above 100 bpm) whereas Bradycardia occurs due to a very slow heartbeat rate (usually less than 60 bpm). Interruptions of electrical signals within the heart are the core cause of these abnormal heartbeat patterns.

3. Heart Attack:

This is another type of heart disease that is responsible for a high amount of deaths. This condition is also known by another term Myocardial Infarction. The cause of this disease is the reduction of blood supply to a part of the heart muscle due to blockage in the coronary artery. Symptoms of heart attacks include chest pain, weakness, discomfort and shortness of breath. Certain complications following a heart attack are arrhythmias and heart failure. This is usually seen in people suffering from diabetes, high blood pressure, overweight, cholesterol and a high-fat diet.

4. Heart Failure:

This condition of the heart occurs when the heart is not functioning to its required standards. It doesn't pump blood as effectively as it should. This results in less supply of blood and oxygen to the heart muscles. Causes of this disease are blockage of the coronary artery, high blood pressure, cardiomyopathy and thyroid.

5. Heart Valve Disease:

To govern the blood flow from and to the heart, the heart consists of four valves. They are the Aortic valve, pulmonary valve, mitral valve and tricuspid valve. These ensure that blood flows in the correct direction at proper intervals. Problems associated with the valves are their improper opening and closure. This can build up excessive pressure in the heart chambers. Causes for the improper functioning of the valves are congenital heart disease, CAD, heart attack and rheumatic fever. Symptoms include chest pain, dizziness, fatigue, swelling and shortness of breath.

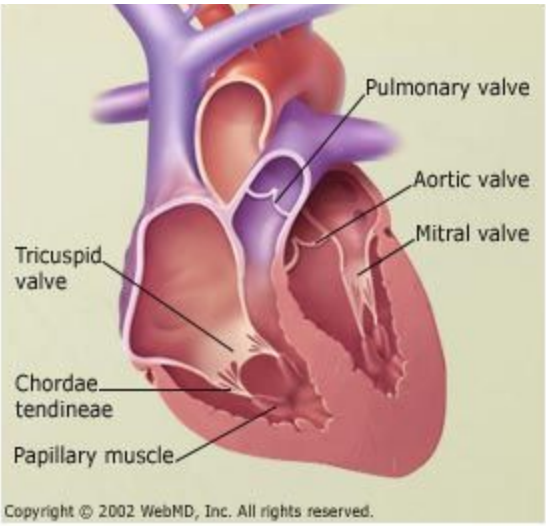


Fig 1.1 (a): The four Heart Valves

6. Cardiomyopathy:

Myocardium is termed the heart muscle. The improper functioning of the Myocardium leads to a heart condition called cardiomyopathy. The muscle may get hard and stretched making the heart pump blood less efficiently. Causes of this condition include alcohol consumption, genetic heart conditions and infections. One severe consequence of this condition is heart failure. Symptoms include breathlessness, obesity, fatigue, discomfort and swelling.

7. Angina:

Arteries getting narrowed leads to an inadequate supply of blood to the heart. This causes pain and discomfort in areas like the chest and abdomen. This pain is termed Angina. Symptoms of angina include extreme pain and stress. The pain is similar to tightness and squeezing of the chest. Angina levels can be stable or unstable. Stable angina is usually chest pain during physical activity. Sudden worsening of angina leads to unstable angina.

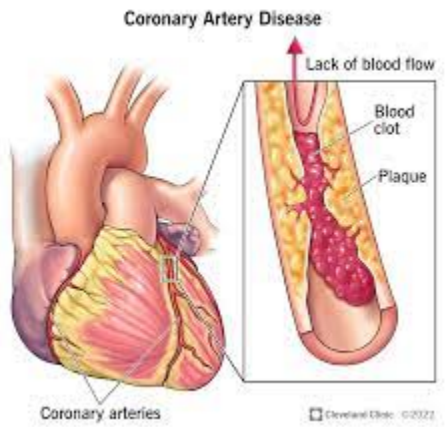


Fig 1.1 (b): Coronary Artery Disease

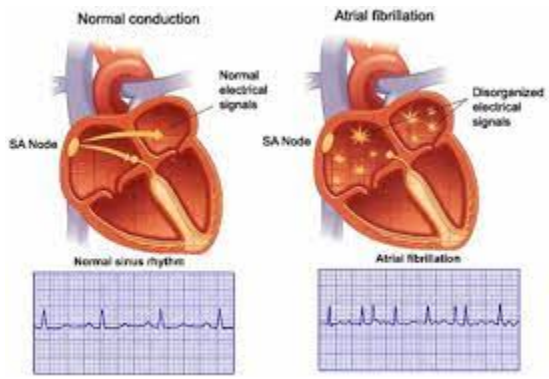


Fig 1.1 (c): Cardiac Arrhythmia

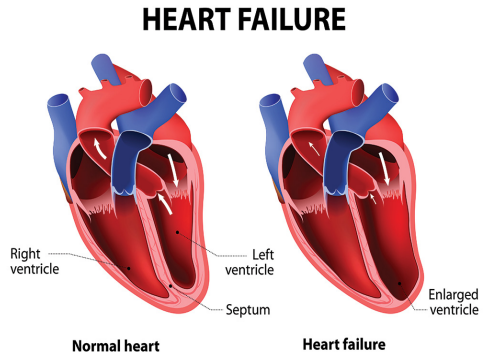


Fig 1.1 (d): Heart Failure

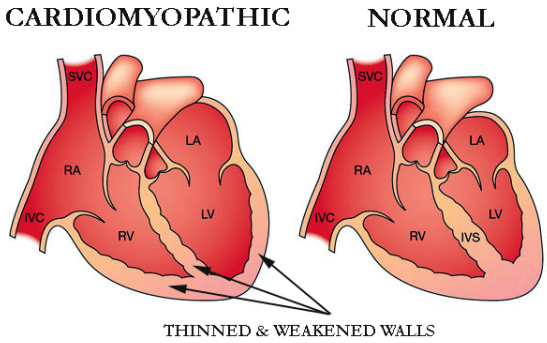


Fig 1.1 (e): Cardiomyopathy

1.2. Existing IoT Devices

There is a lot of research being done around the world on IoT devices that predict heart disease. Numerous attempts have been made to predict heart disease, and a few have been successful also due to the hardware used.

The devices listed below are some of the famous IoT devices that are being developed around the world.

1. **Holter Monitor**: It's a little portable device that records the heart rate. It is used to identify or determine the risk of irregular heartbeats (arrhythmias). It records a continuous ECG, usually for 24 to 48 hours. A Holter monitor test can be performed if a conventional electrocardiogram (ECG or EKG) does not provide sufficient detail on the cardiac status. Newer Holter monitors allow continuous monitoring for weeks. Holter monitors are usually considered bulky and uncomfortable, requiring skin preparation, patches and many wires. All this can lead to skin rashes, skin irritation and/or itching.

2. **Zio XT Patch**: The Zio XT wireless patch monitors EKG readings across the skin, enabling it to detect signs of arrhythmia or AFib. It sticks to the skin and is worn constantly by the patient, enabling quick identification of any heart issues. Because the patch may be worn continuously, including while sleeping, exercising, and taking a shower, it gives a considerably more thorough report of the patient's cardiac condition. The Zio XT monitor has humidity and temperature restrictions. If exposed, patients may notice a decline in adhesive effectiveness that causes the device to slide off or slip during the course of the patient's wear.

3. **KardiaBand**: The KardiaBand is the first 30-second EKG watch of its kind that is of a medical grade. This app, designed for Apple Watch, makes it easy to get better control over your cardiac health. Within 30 seconds, KardiaBand will evaluate your heart rate to identify atrial fibrillation. If you choose, you may also give your doctor a monthly report. This programme continuously scans your heartbeat for abnormalities and alerts you to get an EKG reading when it does. The Apple Watch and iPhone are compatible with the KardiaBand. However, several features are incompatible

with Android and need a premium subscription. Additionally, the Apple Watch's battery life is reduced by the SmartRhythm Monitoring feature, which also disables the Activity app and watches faces.

4. **BodyGuardian Heart**: BodyGuardian Heart is a wearable monitor that is wireless and not just for the elderly. Its medical-grade adhesive makes it easy to apply, and it will monitor your heart rate, as well as your activity level, respiration, and other vital signs. You may move around freely while wearing the device because it has no cables.

5. **Implantable Cardio Defibrillator (ICD)**: Seniors occasionally require more than just a wearable heart health device. The possibility of an implantable heart health gadget emerges. A senior may receive a pacemaker or an implanted cardioverter-defibrillator (ICD) for rhythm control. To track heart rate, an ICD is implanted in the heart tissue. It aids in re-establishing a healthy heart rhythm and guards against sudden cardiac death. Infection at the implant site is one of the potential hazards of getting an ICD installed. haemorrhage, bruising, or swelling. ICD leads can harm blood vessels.

6. **Pacemaker:** If a senior's heart beats too quickly, too slowly, or erratically, a pacemaker can be necessary. A pacemaker is a device that continuously checks your heart rate. It delivers electrical impulses to the heart to help it keep a healthy pace if it notices a rate that is either too sluggish or too fast. A battery-operated device called a pacemaker is implanted in the chest. Some pacemakers have sensors that keep an eye on vital signals like breathing rate and body movement. Additionally, they can convey information about heart health to the user or a doctor. The goal of pacemakers is to enhance your quality of life and prevent disruptions from heart issues. One of the sidearm's veins has a blood clot in it.

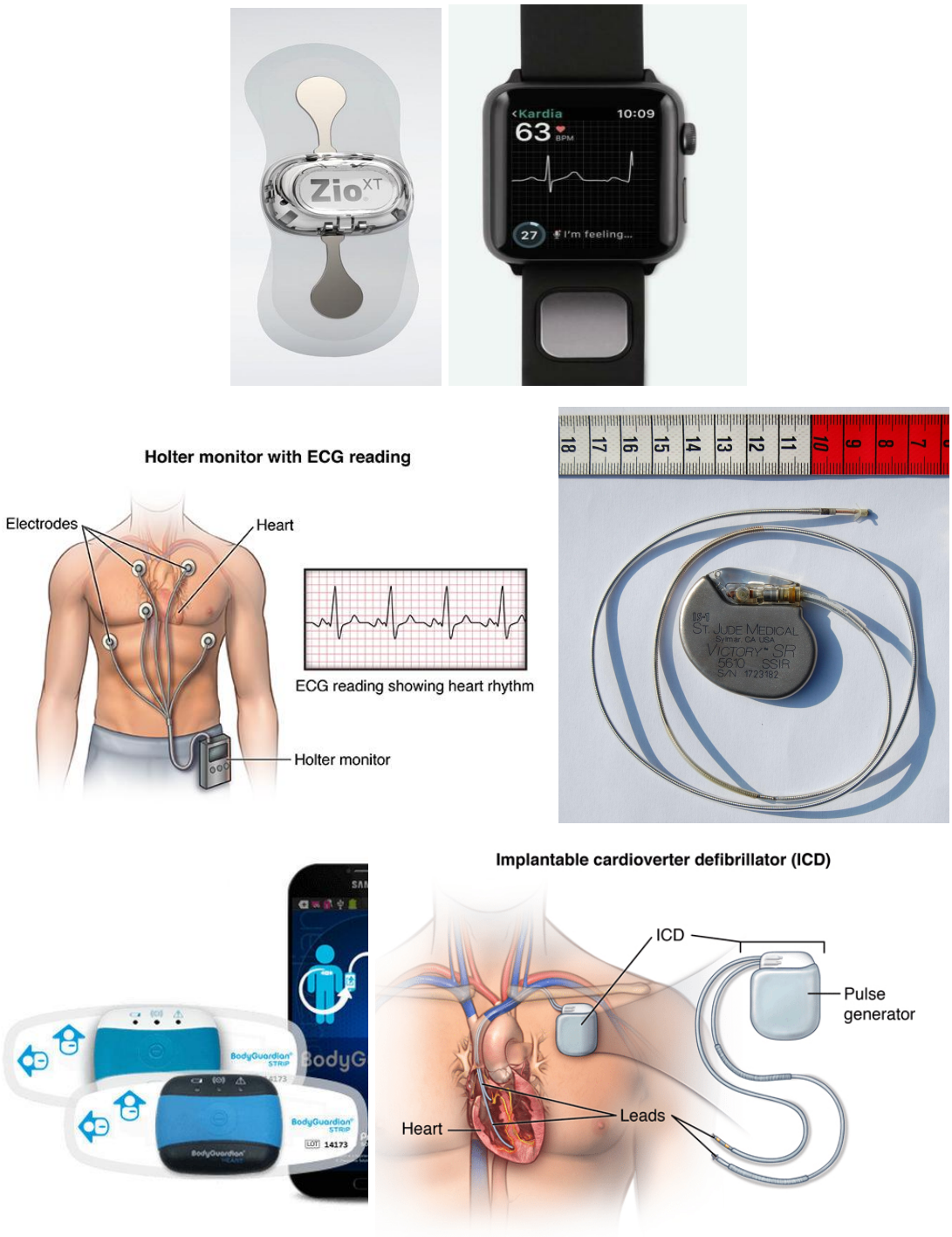


Figure 1.2(a-f) : Pictures of Zio XT Patch, KardiaBand(Top Left - Top Right), Holter Monitor, BodyGuardian , Pacemaker, Heart(Bottom left) and Implantable Cardiac Defibrillator(ICD)(Bottom Right)

2. Problem Statement

The main objective is to establish a reliable patient monitoring framework with the aim that patients, who are hospitalized or perform routine daily exercise, can screen and self-check their heart-related parameters safely. However, despite all the benefits, there are certain drawbacks that might reduce the overall system's effectiveness. This leads to more and more opportunities in the field for developing IoT devices in order to predict various heart diseases. Some of the related issues are described below.

2.1. Internet of Things: challenges and opportunities

The Internet of Things is pervasive around us and is playing a bigger role in our daily lives. IoT is a technology that incorporates a wide number of advanced systems, platforms, and products, built around intelligent systems, paradigms, smart objects, and sensors. A number of applications, including home automation, medical, agricultural, and industrial, have proved extremely valuable as a result of IoT. However, it still faces many challenges that require a great deal of attention. For IoT projects to be stable, we will need to overcome some of these issues:

1. Connectivity and maintenance costs can make it difficult for IoT sensors to collect and transfer data.
2. The efficiency of your devices and processing will mean little if the user experience isn't considered.
3. In terms of design, issues like limited computation power, limited energy, and limited memory must be resolved.

2.2. The Goal

Considering all the requirements, the goal of this project can be defined as,

"The idea here is to build and design a healthcare kit using IoT that measures ECG in order to predict the type of heart disease using machine learning. Additionally, provide an application interface for IoT devices for easy access and analysis"

Adaptive interfaces and reliable functionality can be a result of this as well. As a result, it should be both simple and accurate to make it easier to understand. The project aims to eliminate a number of existing disadvantages as well.

3. Literature Survey

This project summarises studies on IOT, implementation of machine learning algorithms and user interface. Integration of Internet Of Things(IOT) and Machine Learning applied to health promise to supply solutions to the challenges in healthcare systems by providing tools for lowering costs while increasing efficiency in diagnostics and treatment. The user interface provides a medium between the medical device interaction and the user.

[1] 3.1. Deep Neural Network Based Real-Time Multi-Class Arrhythmia Classification in IoT-Cloud Platform

Akhila Naz K A1*, Jeena R S2 and Niyas P

3.1.1. Introduction

Heart is the vital organ that is controlled by the central nervous system. Arrhythmia is the condition that leads to narrowing of heart arteries. Electrical conduction of the heart is processed in a specific manner . The electrocardiogram is obtained from the ECG sensor. It has three electrodes with patches attached. The patches are placed on the skin. This measures the electrical activity of the heart and ECG graph is plotted. Graph has several peaks and waveforms. Artificial Neural Networks classify different types of Arrhythmia. This algorithm is compared to biological neural networks. The Deep Neural Network is the subset of the Artificial Neural Network. It has multiple layers. The heart related information is accumulated on cloud storage. The client user and medical staff can access the required heart related information.

3.1.2. Literature Review

Arrhythmia results in inconsistent blood flow, causing various effects on organs like the brain and heart. In this paper using the Deep Neural Network(DNN) has multiple layers. It consists of three hidden layers and has maximum accuracy of 99.45%. Automatic heart beat classification is in progress.The data is divided into inter patient data and intra patient data. It classifies the data based on ECG recordings. Intra patient data differentiates into a number of training instances and testing.It

contains beat labels. Deep Neural Network is built on hierarchical patterns with seven hidden layers. MIT BIH dataset is used and features are extracted. The above proposed system in this paper consists of a Deep Neural Network with three hidden layers. This helps in classification of Arrhythmia into normal and abnormal cases with most efficient parameters. Cloud storage Medium stores variable ECG information and it is accessed by the client user. It is used for analysing the stored data through a web server. Web server operates with Master server with commanding and requesting operators. Master node controls the slave node. Slave node stores all the relative data of the client user. This approach stores a large amount of data.

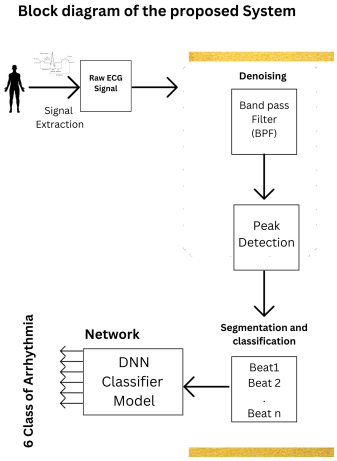


Fig 3.1: Diagram with signal processing steps and proposed DNN

3.1.3 Related work

This section gives an overview of the topics and methodology being used in the implementation of the above-proposed model.

a) MIT - BIH Arrhythmia Data-Set

The database is created and stores the ECG information and recording. This dataset consists of 47 subjects. It includes 25 men of ages 23-89 years and 27 women of the same age as men. The recording is obtained from two electrodes of the AD8232 sensor. This dataset is further split into training and testing instances.

b) Real-Time Signal Acquisition

It is the mechanism of collecting the raw data from the physical world. Sampling and converting are the key elements to obtain digit numeric values of the collected data and it is manipulated by the computer.

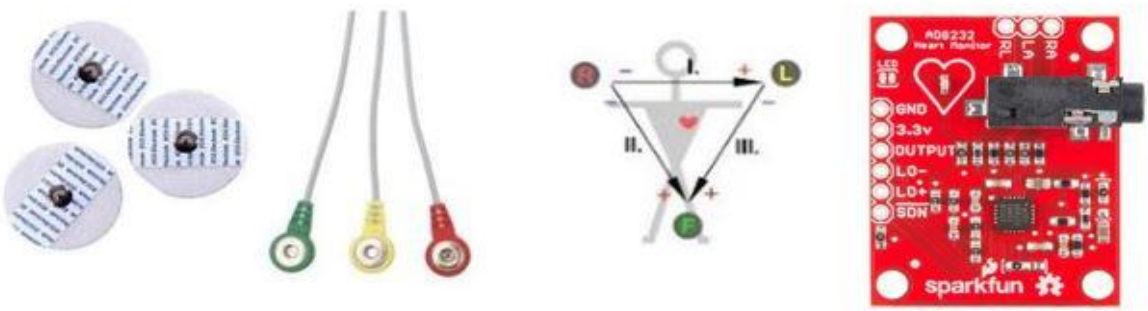


Fig 3.2: ECG sensor(AD8232) with three leads and three electrode patches made up of Ag/Agcl.

The surface electrodes are placed on specific locations of the body to measure the heart's electrical activity. It works on the exact principle of chemical equilibrium and detects the tension between muscle and body skin through electrolytic conduction. The three leads are arranged in anatomical relation. The third lead constitutes a positive electrode that is placed on the left leg. The negative electrode is placed on the left arm.The connection of three leads to the formation of an equilateral triangle. This figurative representation is called Einthoven's Triangle. A computer system records the information and displays the wave on a monitor.

c.) Signal Processing

c.i). Denoising

Patient movements and respiration results in interference. These interferences are captured in power line and baseline variations. After capturing, the errors are generated while detecting and analysing the ECG peaks . Noise is filtered denoising. Electrocardiogram signal is the weakest signal in the environment. An abundance of noise is generated due to muscle tensions and several other biological

parameters. Denoising is carried out by the band pass filter(BPF) with the frequency range set to [0.8Hz-48Hz].

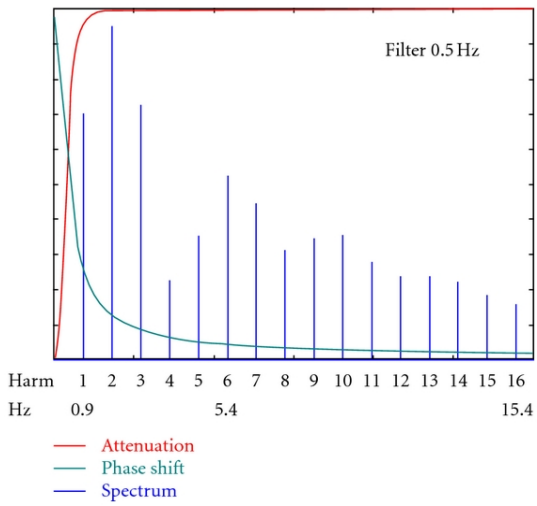


Fig 3.3: Bandpass filter recording with the frequency set to 0.5Hz

c.ii) Peak detection

It is important in detecting the various slopes of the ECG signal. The ECG signal is fragmented and differentiated into P-wave and T-wave.Peak numeric values are achieved by implementing the wfdb().This function directs and provides an easier medium to classify between the different types of waves. Maximum amplitude of every single R-peak of the QRS wave can be easily distinguished.It represents a single heartbeat. QRS detection is done using the Pan-Tompkins algorithm.

c.iii) Signal Segmentation and Normalisation

This segment explains about sampling of processed signals. After processing, the signal is segmented simultaneously. The signals are normalised. Normalisation is done to achieve easy mathematical computation.

d.) Deep Neural Network(DNN)

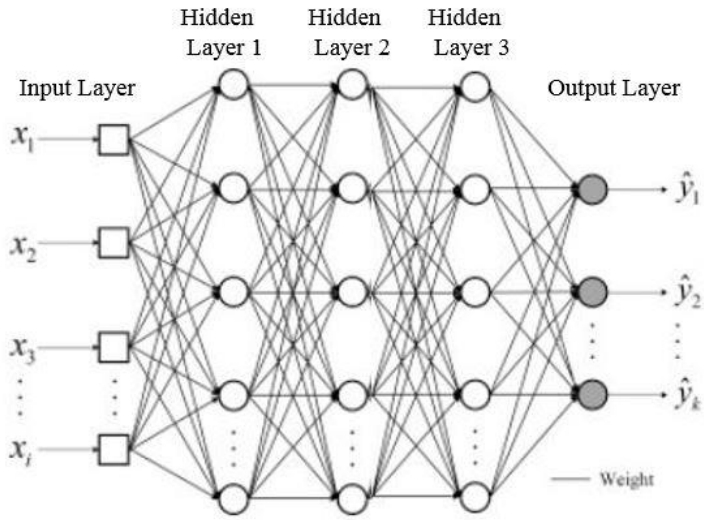


Fig 3.4 Hierarchical representation of Deep Neural Network

This model is built by installing Keras. It has a built-in function Tensorflow. It is built empirically. Classification performance is enhanced by building the model with three H1, H2 and H3 layers. Every layer has 300-500 neurons. It has an activation function. It aggregates the output based on maximum pooling and average pooling. There are no filters present.

Mathematical Equation of ReLU activation function.

$$y = \max(0, x) \qquad ..3.1.3.1$$

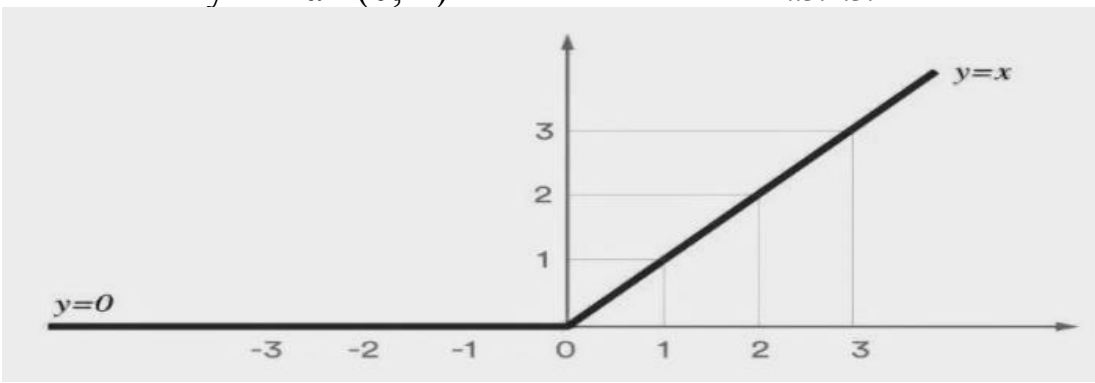


Fig 3.5 ReLU Representation

The task of predicting output more accurately with respect to actual output is termed as cost function(C). If classes are mutually exclusive, this function is used. It is used in Sparse Categorical Cross-Entropy

The mathematical equation,

$$f = - \{\log p(s \backslash in\ c)\} \qquad ..3.1.3.2$$

To minimise the cost function optimum weight has to be calculated. It also ensures the algorithm's accurate result. Different weights are assigned and multiple iterations are implemented. This technique helps to reduce cost function. The process is called gradient descent.

Mathematical equation, .

θ = parameter of weight

$\nabla J(\boldsymbol{\theta};x,y)$ = gradient of weight parameter **θ**

η = learning rate

$$\theta = \theta - \eta \nabla J(\theta; x, y) \qquad ..3.1.3.3$$

e). Cloud data storage and accessibility

The ECG data from different locations are organised and stored in a developed cloud platform.

3.1.4 Conclusion

Problems related to the heart are dangerous. It mostly affects elderly people without realising and knowing about it. The symptoms of heart disease are noticed when discomfort in the chest reaches the peak level. This paper aims at building an integrated model with an IoT sensor with machine learning algorithms. This integrated model helps in the detection of cardiovascular diseases early enough. It provides a medium between medical representatives and the client user. It collects the necessary heart-related information from the sensor embedded in IoT. The information is stored in a cloud computing platform through a web server. Client users and medical staff can access the information through the websites.

This innovation helps mankind by collecting data information through health status monitors which include total cholesterol and ECG.

[2] 3.2 IoT-Based Health Monitoring System by Using Raspberry Pi and ECG Signal

Megha Koshti[1], Prof. Dr Sanjay Ganorkar[2]

3.2.1 Introduction

The work done in the paper implements a healthcare system. It results in fast and noninvasive monitoring of the heart and the results are obtained from the lead I of electrocardiogram(ECG).

3.2.2 Related work

It aims at methodologies being used in the implementation IOT based monitoring system.

a) Arduino UNO

Microcontroller embedded on ATmega328 processor. 6 analogue and 14 digital pins. It is lost in cost and flexible. The memory used in the Arduino board is SRAM and the storage is FLASH and EEPROM.

b) Raspberry pi-4

It is faster in computing with memory 8GB. It has three USB ports. The operating frequency of the board is 700MHz by default.It has network connectivity

c) Microcontroller

It is a compressed micro computer. It controls the functions of the embedded systems. It contains components like memory, peripherals and processor.

d) Accelerometer sensor

It calculates acceleration of body in an instantaneous time frame.

e) ECG sensor(AD8232)

It is a sensor with three electrodes. They are connected with patches. The patches are made up of Ag/Agcl solutions.

3.2.3 Conclusion

The integration of IoT and machine learning domain helps in analysing and classifying patients with positive cases. As a result of this, the number of fatality rates can be reduced. The machine learning model helps in predicting the disease with the highest accuracy.

[3] 3.3 Early Detection of Cardiac Arrhythmia Disease using Machine Learning and IoT Technologies

Mr. Yathish D P , Dr. Dayanand Lal N

3.3.1. Introduction

As Arrhythmia is a lethal disease, early diagnosis of this can reduce the mortality rate. ECG readings are often used to diagnose cardiac problems. Many important features such as the location of chambers, heartbeats (BPM) and the presence of damage in heart muscles are detected in ECG. The usual method of detecting arrhythmia is by locating the peaks interval such as PR interval, QT interval, and QRS interval and comparing them for abnormalities. To extract these features the signal is processed to remove noise such as wandering baseline, muscle noise and power line interference. This study is a noninvasive procedure to display heartbeats over time. The important features are the time and frequency domain topologies.

3.3.2. Related work

This section gives an overview of the topics and methodologies used to build the model proposed above.

3.3.2.1. ECG accretion:

Board namely Arduino and sensor namely AD8232 are used. AD8232 is an ECG sensor which detects electrical activity. Arduino receives the data from the sensor via the COM port. A three-electrode ECG sensor is used to identify the heartbeat. The patches of the ECG are connected to the limbs to extract electrical impulses.

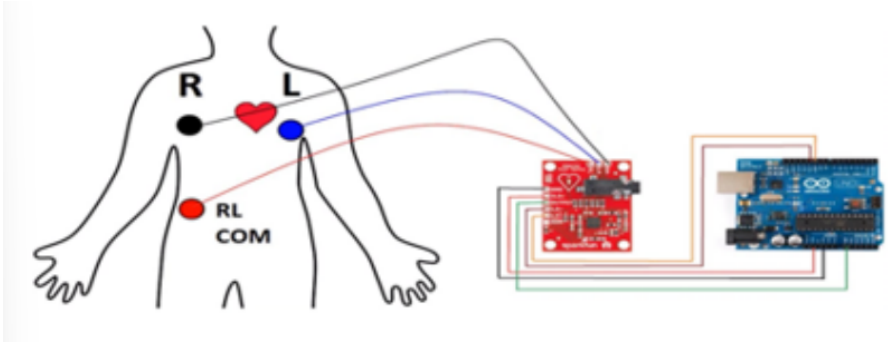


Fig 3.6 ECG accretion

3.3.2.2. QRS detection algorithm:

Filters namely BPF, HPF and LPF are used by the Pan Tompkins algorithm to reduce noise. The energy calculation is rendered with the help of a more extended ORS complex's moving window scale.

3.3.2.3. Extraction of features:

The statistical features are extracted namely mean, minimum, standard deviation, skewness, kurtosis and maximum. The proportion of the rr interval varies by a duration of 50ms.

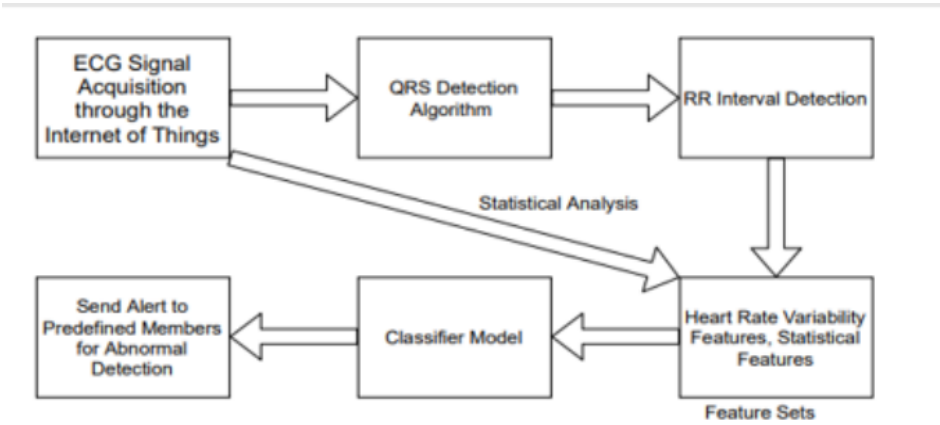


Fig 3.7 signal processing and feature extraction

3.3.3. Results:

The methods involved in the implementation are:

- Accretion of ECG: ECG AD8232 sensor is used to collect data and is sent to MATLAB for further processing.
- Extraction of Features: The statistical(mean, Kurtosis, maximum, skewness etc.) and dynamic characteristics(using Pan Tompkins algorithm) of the ecg are determined. With the R-R interval, the characteristics of the heart rate variability are determined.
- Classification: SVM classifier is used. The results of the classifier are compared and tested for different factors. Cardiac arrhythmia can be classified by the SVM classifier.

[4] 3.4 : IoT Based Wearable Monitoring structure for detecting Abnormal Heart

Padmavati Kora , A Rajani , M C Chinnaiah, K Swaraja, K meenakshi

3.4.1. Introduction

In this paper a portable ecg device is made to find abnormal heart conditions and to analyse the data for prior diagnosis. This device uses host processors, wireless sensors and temperature control.

3.4.2. Working and flow of the data:

The architecture is a full vertical system with all the advantages in accordance with the latest technology.

the approaches include devices such as:

- ECG electrodes.
- Bluetooth module.
- IOT module.

3.4.3. Signal collection and Processing:

The ecg is collected using a sensor and many noise signals generated during the collection and transmission of data which cause inconvenience are removed using digital filters.

3.4.4. Storage and analysis:

Cloud storage is a model of computer data storage having multiple servers and physical storage devices. This is owned, managed and hosted generally by a company. The data is stored in a digital format in the cloud. This can be further used for the diagnosis of diseases.

3.4.5. Warnings related to abnormalities.

As the device is connected to the cloud, this can capture real time data of a person's health. This will further inform the family and the doctor in charge if any abnormalities are detected.

3.4.6. Communication devices used for simulation:

3.4.6.1. Zig bee Module (Zonal Intercommunication Global-standard) :

A wireless network module under global management is called Zigbee. It consists of sensors, instrumentation, and control systems and is a low power network. Its advantages for residential use can be observed in low power sensors, HVAC control, industrial automation, and medical monitoring. The main networking standard based on 802.15.4 radios is ZigBee.

Fig 3.8 Zig bee module

3.4.6.2. Bluetooth module: The short-range 2.4G wireless communication module known as a Bluetooth module is the basic circuit set of the chip with an incorporated Bluetooth function. UHF radio waves are used by bluetooth to transfer data.

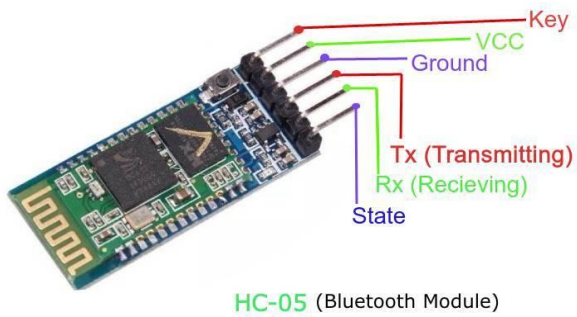


Fig 3.9 Bluetooth Module

3.4.6.3. Wifi module: To send and receive data over Wi-Fi, wifi modules or wifi microcontrollers are used. They can also receive commands via Wi-Fi. Wi-Fi modules are used for device communication. They are most commonly found in the domain of the Internet of Things. Wi-Fi modules are used to connect robotic and electronic projects to the internet. Wi-Fi modules enable the creation of IoT (Internet of Things) projects.

Fig 3.10 wifi module

3.4.6.4. Hydrogel Gel Electrodes and Sensors:

A conductive hydrogel is a type of polymer material with several uses due to qualities such as stiffness, can recover independently, conductivity, transparency, does not freeze, sensitivity, stretchability, heal on its own, and sensitivity to stain. Ag/AgCl electrodes are most frequently used in measuring ECG as these electrodes are passive electrodes. These electrodes are generally covered with a layer of hydrogel. While passive electrodes can assist produce high-quality ECG readings, they have some disadvantages.

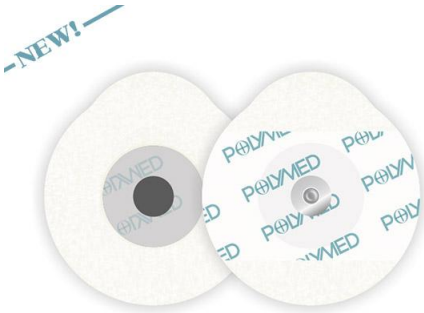


Fig 3.11 ECG patches with hydrogel electrodes

3.4.7 Results:

Weak ECG signals are accurately measured using ECG sensors and other circuit elements. The filtered ECG signal is amplified by an operational amplifier. ECG signals ranging from 0 to 3.3 volts are acquired using an ECG sensor AD8232 and a WIFI module using IEEE 802.11 b/g/ g/protocols. Simple and efficient access to IoT cloud and java is enabled by smartphones and machine learning techniques. Power is consistently supplied to every module in the ECG monitoring equipment.

4 Implementation

This section speaks about the design of the IoT Healthcare Kit, building Machine Learning models to predict the presence of the disease and the implementation of graphical user interfaces, one for testing Cardiovascular Disease and one for testing Cardiac Arrhythmia.

4.1 Hardware Architecture

4.1.1 Components Used

1. **[7]ECG sensor(AD8232)** : The ECG AD8232 sensor is used to determine the electrical movement of the human heart.This activity may be tracked similarly to an electrocardiogram, with the output being an analogue readout.ECGs may be highly loud, thus the AD8232 chip can be used to minimise the noise.The ECG sensor operates similarly to an operational amplifier in order to obtain a clear signal from the intervals.The AD8232 has a rapid restoration feature that shortens the otherwise lengthy settling tails of the high-pass filters.The AD8232 automatically adapts to a higher filter cutoff after a rapid signal shift that rails the amplifier (such as a leads-off situation).

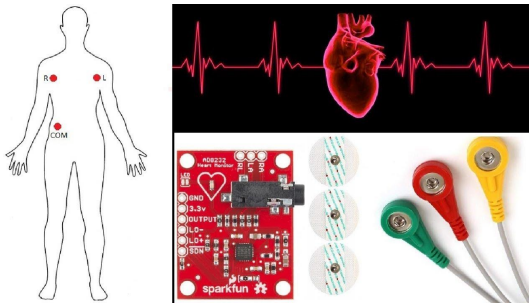


Fig 4.1 ECG Sensor (AD8232)

Features and Specifications:

- Operation of single supply ranges from 2V to 3.5V
- The front end is integrated fully with only lead ECG
- The virtual ground can be generated through integrated reference RFI filter is used internally
- The current supply is low like 170 µA and the output is rail to rail

- Shutdown pin
- CMRR is 80 dB
- Incorporated RLD amplifier (right leg drive Electrode configurations are 2 or 3
- The operational amplifier is uncommitted . It accepts half cell potential up to ±300 mV
- Three-pole adaptable LPF with adaptable gain
- The signal gain is high using DC blocking capacity
- Filter settling can be improved by quick restore
- Two-pole adaptable HPF
- 4 mm × 4 mm and 20-lead LFCSP package

2. [8]**Arduino Uno**: It consists of several equipped inbuilt devices . The board has 6 pins which take analog inputs and 14 pins which produce digital outputs. It also consists of other circuits and shields that are embedded in the board. It has a USB connector, power supply jack and ICSP(In-Circuit-Serial-Programming) header. Arduino IDE is used to program this board and can be used on online as well as offline platforms. The PWMs pins are Pulse Width Modulation capable pins. Its input voltage ranges from 7V to 20V. It can be powered using a USB as well as an external power supply. It is low-cost, flexible and easy-to-use programmable as a result of these features its widely used board in various projects.

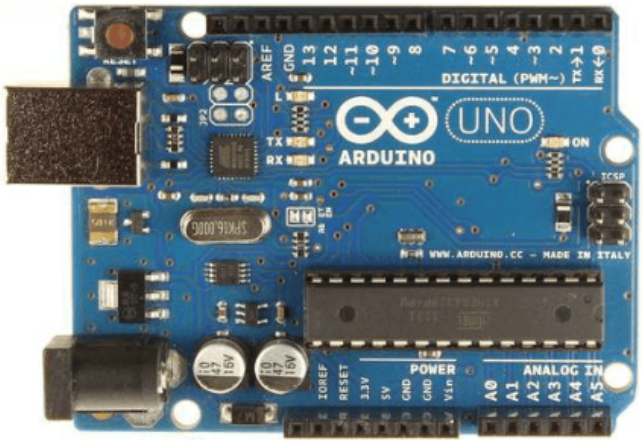


Fig 4.2 Arduino Uno R3 ATmega328

Specifications of Arduino UNO

Microcontroller	Atmega328P
Operating Voltage	5V
Input voltage(recommended)	7-12
Inout Voltage(limit)	6-20V
Digital I/O Pins	14(of which 6 provides PWM output)
PWM Digital I/O Pins	6
Analog Input pins	6
DC Current per I/O pin	20mA
DC Current for 3.3V pin	50mA
Flash Memory	32 KB(ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB(ATmega328P)
EEPROM	1 KB(Atmega328P)
Clock Speed	16 MHz
LED_BULTIN	13

Table 4.1 Arduino Specifications

3. **[9] Raspberry pi -4 :**

The latest variant of Rpi is Raspberry pi-4 Model B. It has 8GB RAM. It provides a dual display with a 4K output feature. The pi-4 board can run two monitors at once in 4K. It has upgraded USB port capacity. It has 3 USB ports, which can transmit data ten times faster.

This supplies slightly higher peak currents to the board. Type-C Power Supply is embedded on the board, it has 5 Volt 15 Watts. It has a number of advantages like faster processing, wireless, enhanced ethernet connectivity, software and OS.

Fig 4.3 Raspberry Pi 4 model with 8GB RAM

Specification of Raspberry pi-4 Model B.

SoC	Broadcom BCM2835(CPU+GPU+DSP+SDRAM+USB port)
CPU	ARM1176jzf-S a 700 MHz(ARM 11 family)
GPU	Broadcom video Core IV, OpenGL ES 2.0, -2y VC-1, 1080p30 H.264/MPEG-4 AVC
Memory(SDRAM)	512 MB(share with the GPU)
Ports USB 2.0	2(throw integrated USB Hub)
Video inputs	Connector[MIPI] CSI
Video outputs	RCA(PAL and NTSC), HDMI(rev 1.3 and 1.4), Interface DSI to LCD panel
Audio outputs	3.5mm Connector, HDMI
Integrated Memory	SD/MMC/SDIO slot
Network connectivity	10/100 Ethernet(RJ-45)
Energy	700 mA,(3.5)
Power Supply	5V throw Micro USB or GPIO header
Size	85.60mm X 53.98
Supporting Operating system	Debian, Fedora, Arch Linux, Slackware Linux, RISC OS

Table 4.2 Raspberry Pi specifications

4.1.2 Protocol used for hardware communication:

1. UART: (Universal Asynchronous Receiver Transmitter)

UART is a serial communication protocol and it requires only two wires for communication. It has a particular data format and a transmission speed which are configurable. The word serial means that one bit is transmitted at a time. The transmission is achieved by sending all the bits sequentially over a single wire. This reduces channel size and overall communication cost in comparison to parallel communication. In this protocol a clock is not used to synchronise or validate data transmitted.Tx and Rx pins are used for data transmission. The TX pin of device 1 and RX pin of device 2 are connected and vice versa. The ground of device 1 is connected to the ground of device 2.

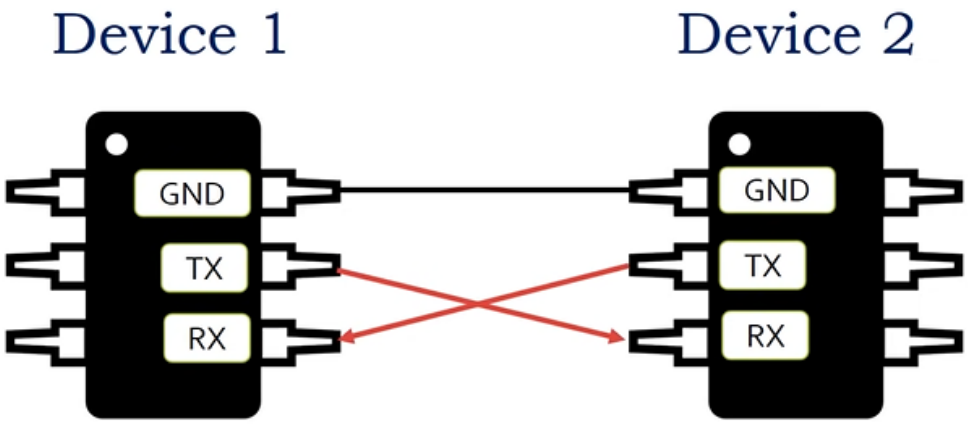


Fig 4.4 UART device connections

The data synchronisation is achieved with the help of

- Baud rate.
- Start and Stop bit.

Speed of data transfer is represented by Baud rate(units are bps). Some of the baud rate values include 2400, 4800, 9600 bps etc.

Data frame of UART communication protocol:

1) The idle, no data state, is a high voltage signal. The line is held at high voltage to show that the line and transmitter are not damaged.
2) A frame in this protocol is made of start bit, data bits, parity bit (optional) and one or more stop bits. Start bit indicated the start of a packet and a new signal. The next 5 to 9 bits, based on the configuration, represents the data that the packet is carrying. The parity bit, if configured, will be placed after all the data bits. The next 1 or 2 bits are stop bits and are always in the high state and represent the end of the packet.
3) The errors, if any while transmitting, are checked using a parity bit.

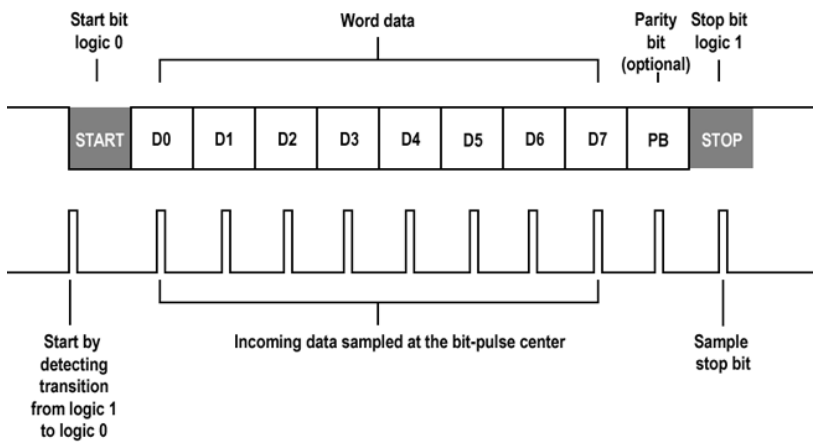


Fig 4.5 UART data frame

Based on the application the UART protocol can be configured in 3 different ways.

1) Simplex: This is a one way communication with no provision for the receiver to send the information back to the transmitter.
2) Half Duplex: It is a bidirectional communication mechanism where the data can be sent by both the transmitter and receiver but cannot be sent at the same time.
3) Full Duplex: It is a bidirectional communication mechanism where the data can be sent by both the transmitter and receiver and can be sent at the same time.

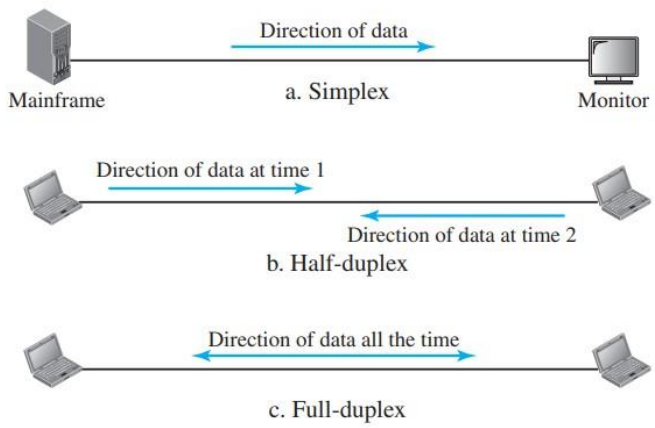


Fig 4.6 Representation of simplex, half duplex and full duplex communication

The advantages of UART include:

1) It has fewer errors because error detection is performed with the help of a parity bit.
2) It does not require more wires as data transmission is performed with the help of 2 wires only. The clock signal is also not required as the baud rate is defined.
3) Cost required to set up a serial communication system like UART is much less than the cost required to set up a parallel communication system.

The disadvantages of UART include:

1) Maximum data frame size is 9 bits.
2) Multiple masters or multiple slaves are not supported by this protocol
3) The data transmission speed is restricted. This becomes a bottleneck for applications which require a higher data transmission rate.

4.1.3. Libraries used in hardware:

1) **Neurokit2:** is a user-friendly package providing easy access to advanced biosignal processing routines. Researchers and clinicians can analyse physiological data with minimal lines of code. The community of this library includes programmers of all levels of expertise. The signals that can be processed using this library are:

- ECG - Electrocardiogram signal - Test used to check electrical activities of the heart.
- EDA - Electrodermal Activity - Test used to check traffic of electric signals in nerves.

- EEG - Electroencephalography -Test used to monitor and check electrical activities of various states of the brain.
- EMG- Electromyography - Test used to check the neuromuscular activities in a patient.
- EOG - Electrooculography - Test to check the corneo-retinal standing potential.
- HRV - Heart rate variability - It is indicative of autonomic regulation of the heart rate.
- PPG - Photoplethysmography - It is an optical measurement of arterial volume, just using a single photodiode.

2) **Serial:** This module is responsible for accessing the serial port. It includes Python backends for Windows, OSX, Linux, BSD (potentially any POSIX-compliant system), and IronPython. The "serial" module automatically selects the suitable backend. This module, using python properties, provides access to port settings. It supports the transmission of data of different byte sizes, parity, stop bits and flow control with CTS/RTS and Xoff/Xon. Provides file-like API with read and write. The files in this module are 100% python code. We use this module to set up communication between Rpi and Arduino. For efficient usage, this package requires Python 2.7 or Python 3.4 or newer, and Windows 7 or newer.

3) **Scipy:** It runs on NumPy. The scientific Python library is abbreviated as Scipy. It adds new utility functions for optimization, statistics, and signal processing. SciPy, like NumPy, is open source, so we may freely use it. It enables users to alter and view data using a variety of high-level Python commands. SciPy is based on the NumPy Python extension. It significantly increases the capabilities of an interactive Python session by offering the user with high-level commands and classes for manipulating and displaying data.

4.2 Construction of Machine Learning Models

Models developed using techniques and algorithms of machine learning are the tools used to effectively solve problems involving high complexity. After obtaining the data and preprocessing it, we need to split the data into training and testing datasets. The next stage involves building a machine learning model using certain algorithms. The final stage involves testing the model on a

random test sample and obtaining its performance parameters. The subsequent sections give a brief rundown of different algorithms that are implemented in building the machine learning model.

4.2.1 Logistic Regression

The probability of a target variable is predicted using the supervised learning classification algorithm known as logistic regression. Since the dependent variable's nature is dichotomous, there are only two viable classes. The dependent variable, to put it simply, is binary in nature, with data represented as either 1 (success/yes) or 0 (failure/no).

A logistic regression model makes mathematical predictions about P (Y=1) as a function of X. One of the simplest machine learning methods, it may be applied to a number of categorization issues, including spam detection, diabetes prediction, and other issues. An "S"-shaped logistic function, commonly referred to as a logistic sigmoid function, is fitted in logistic regression and predicts two maximum values (0 or 1).

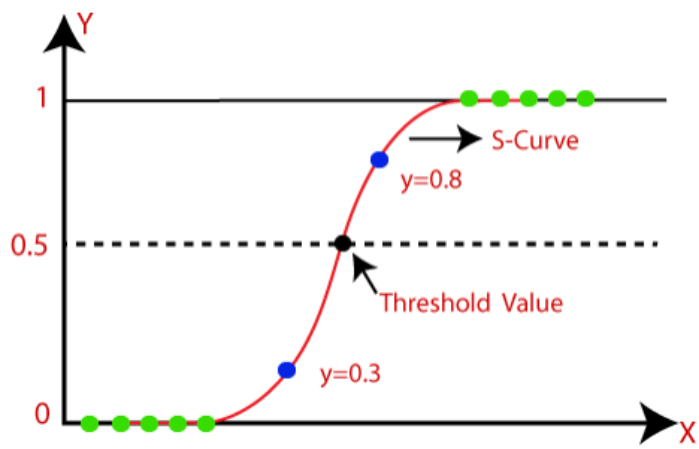


Fig 4.7: Plot of Logistic Function used

The general mathematical equation for this algorithm is given below,

$$y = \frac{1}{e^{-(a + b1x1 + b2x2 + b3x3 + \ldots)}} \qquad \ldots\ldots (4.1)$$

where, y - dependent or response variable

x - independent or predictor variable

a and b are coefficients

The **ADVANTAGES** of Logistic Regression can be seen below,

- One of the simplest ML algorithms is logistic regression, which in some situations offers excellent training efficiency. With this technique, training a model uses significantly less processing resources and takes much less time.
- The anticipated parameters provide information about each feature's significance. Additionally given is the association's direction. The relationship between the features is thus determined by this algorithm.
- Unlike decision trees or support vector machines, this approach enables models to be quickly modified to incorporate new data. Stochastic gradient descent can be used to update the model.
- Along with classification results, logistic regression produces probabilities that are accurately calibrated. Compared to models that merely provide the final categorization as results, this has a benefit.
- This algorithm is less prone to overfitting in datasets with lower dimensions
- When the dataset contains features that can be separated linearly, this approach performs quite well.
- Using a softmax classifier, this approach can be expanded to multi-class classification. Multinomial Logistic Regression is the term used for this.
- The logistic regression model's resultant weights are quite easy to understand.

The **LIMITATIONS** of Logistic Regression can be seen below,

- Large datasets and sufficient training samples for each of the criteria that need to be satisfied are prerequisites for using logistic regression.

- Little to no multicollinearity between the independent variables is needed. Information repetition may result in incorrect parameter training when attempting to minimise the cost function.
- Due to its linear decision surface, logistic regression cannot be used to tackle non-linear situations. Real-world situations rarely involve linearly separable data.
- When there are anomalies or outliers in the data, logistic regression produces inaccurate findings.

4.2.2 K Nearest Neighbour Classifier

K Nearest Neighbour classifier is one of the most popular ML algorithms based on supervised learning technique. The KNN algorithm is designed to do pattern recognition tasks. It makes the assumption that the new test sample and the existing samples are similar, and it places the new sample in the class that best matches the existing classes. The KNN algorithm saves all the available data and categorises a new sample based on its resemblance to nearby examples.

Both classification and regression issues can be solved with the KNN technique. Instead of regression issues, classification issues are highly preferable. Since it makes no assumptions about the underlying data, KNN is a non-parametric method. Because it does not instantly learn from the training set, it is also known as a slow learner algorithm. A K-nearest neighbour classifier seeks to predict the label for a new sample by locating the class with the closest neighbours. By using Euclidean distance, the closest class to the sample that needs to be classified is found.

Optimal value of K can be found using cross validation. Deciding the value of K is the most integral part of the algorithm as this highly impacts the performance of the model. High values of K lead to overfitting of the model. If the value of K is very high than the recommended value, the computational time will increase drastically.

The **WORKING** of the KNN algorithm is briefed below,

- Select K number of neighbours

- For each sample, the euclidean distance of K number of neighbours is considered
- From the euclidean distances calculated, consider the K nearest neighbours with respect to the sample considered
- Out of these K neighbours, compute the amount of the data points in every category
- Allot the test sample to that category which has the highest number of neighbours

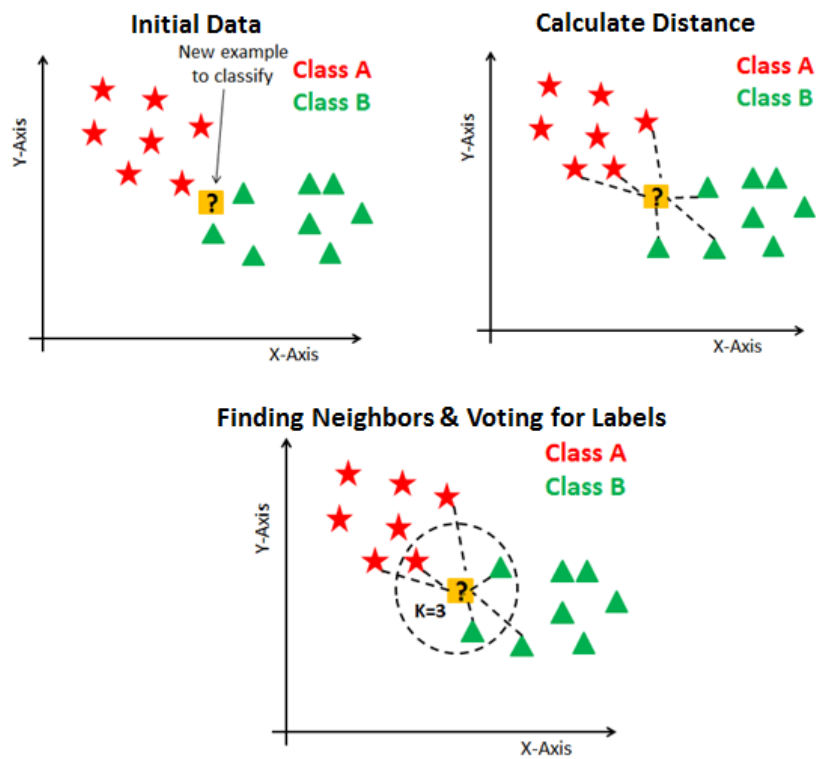


Fig 4.8 : Working of KNN

The **ADVANTAGES** of K-Nearest Neighbour algorithm is briefed below,

- It is straightforward and resistant to noisy training data.
- In the case of unusually huge training data, the algorithm performs well.
- It is an algorithm with several uses. It can therefore be applied to classification, regression, and search issues.

The **LIMITATIONS** of K-Nearest Neighbour algorithm is briefed below,

- Determining the value of K is to be done every time and this might be complex sometimes.

- The high computation cost is caused by the need to determine the separation between each data point for each training sample.
- Training time of the model significantly shoots up with a rise in the number of predictors or independent variables thus making the model slow.

4.2.3 Random Forest Classifier

A popular supervised machine learning approach for classification and regression issues is the Random Forest classifier. It can be thought of as numerous decision trees applied to different dataset subsets. Random Forest takes prediction from each tree and takes their majority vote for classification and average for regression. This is how the final output is predicted.

Random Forest is based on ensemble learning. Ensemble uses a method called bagging which generates a unique training subset from each sample training set. The prediction relies on the concept of majority voting. Another method used by Ensemble is boosting. Increasing the quantity of trees in the random forest algorithm prevents the overfitting problem and results in higher accuracy. It can handle datasets that have continuous and categorical variables. Random Forest algorithm is comparatively slower than decision trees as Random Forests consist of a set of decision trees. The prediction computed from each of the decision trees is intended to have extremely small correlations. Some important features of this algorithm are stability, diversity, and parallelization. Random forest algorithms are immune to the curse of dimensionality due to the reduction in the feature space considered.

The **WORKING** of the Random Forest algorithm is briefed below,

- Select K samples present in the training dataset
- Individual decision trees are constructed for each training sample considered. The selected data points are termed as subsets
- Choose the number N. This is the amount of decision trees that are to be built
- Each decision tree will generate an output

- Next step is voting which involves averaging the decision tree
- The highest voted prediction result is considered as the ultimate output in case of the classification problem. Similarly, averaging is done in case of a regression problem.

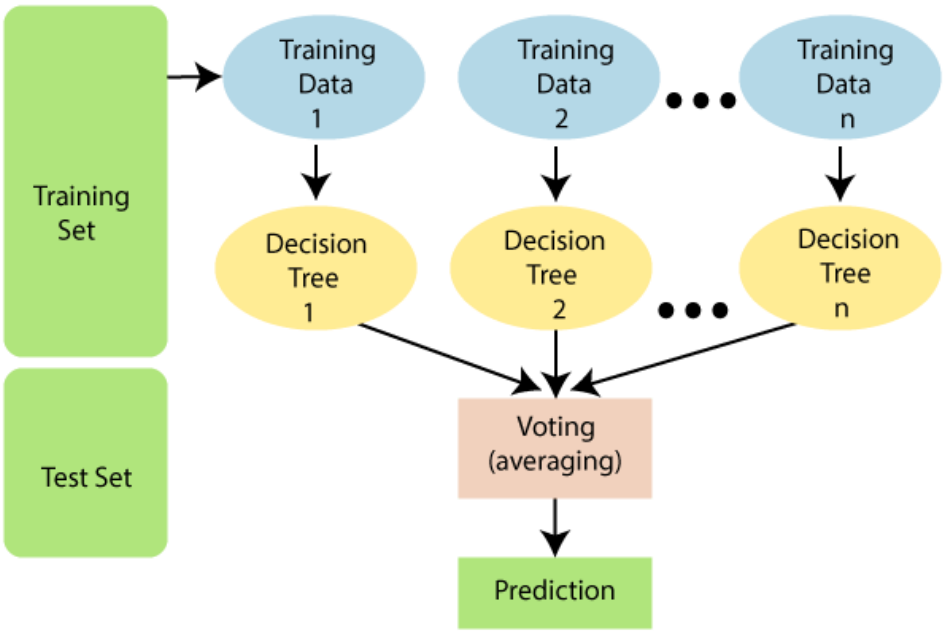


Fig 4.9 : Working of Random Forest Algorithm

Hyperparameters like n_estimators, max_features and mini_sample_leaf increase the prediction capability of the algorithm whereas hyperparameters like n_jobs, random_state and oob_score increase the speed of the algorithm.

The **ADVANTAGES** of Random Forest algorithm is briefed below,

- This algorithm avoids the problem of overfitting as the ultimate result is calculated on the basis of majority voting or averaging.
- Ability to handle datasets that are large and have high dimensionality
- It can maintain its performance capability even when there are null and missing values in the dataset.

- The property of parallelization is exhibited by Random Forest as each decision tree is independent of all other decision trees.
- It is highly stable and maintains diversity.
- Splitting of data samples into training and the testing dataset is not needed as 30% of the samples are not seen by the decision tree made out of bootstrap.

The **LIMITATIONS** of Random Forest algorithm is briefed below,

- Implementation of this algorithm leads to high complexity. Here, prediction of output cannot be done by just following the path of a single tree as in case of decision trees.
- Bearing to its complexity, the training time of this algorithm is very high compared to other algorithms. This is because each decision tree which is a part of the algorithm has to generate a different output for the supplied input data.

4.2.4 Support Vector Classifier

A popular supervised machine learning approach for classification and regression issues is the vector classifier. They operate more quickly and perform better than other neural network models. Each data point in the procedure is plotted as a point in n-dimensional space. The value of a specific coordinate is what each feature's associated value is. This algorithm's primary goal is to establish the optimum decision boundary (termed as hyperplane). In order to assign the proper label to a new test sample when tested after training the model, this hyperplane is utilised to divide the n-dimensional space into classes.

Support vectors are used to find the hyperplane. SVMs come in both linear and nonlinear varieties. When a dataset can be divided into two classes by drawing a single straight line, the linear SVM classifier is utilised. When a dataset cannot be classified using a straight line and is instead non-linearly separable, a non-linear SVM classifier is utilised. Kernel function refers to the mathematical operation utilised for the transformation. Different types of kernel are linear, polynomial, sigmoid and radial basis function.

The **WORKING** of Linear Support Vector classifier is briefed below,

- The algorithm takes all the data samples and outputs the decision boundary that separates the tags.
- Everything on one side of the decision boundary belongs to one particular class whereas the samples on the opposite side of the decision boundary belong to the other class
- The best decision boundary is given the term hyperplane.
- The final aim of this algorithm is to maximise the distance between the support vectors and the hyperplane. The resultant is an optimal hyperplane which best classifies data items into classes

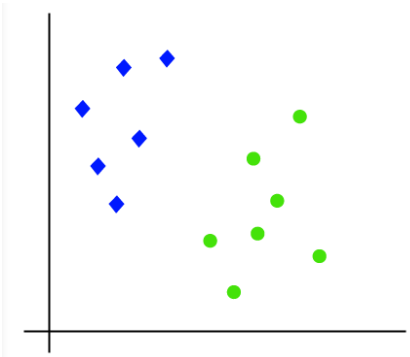


Fig 4.10 (a) : Original Dataset

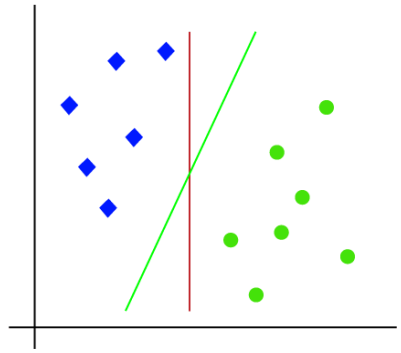


Fig 4.10 (b) : Data with Decision Boundary

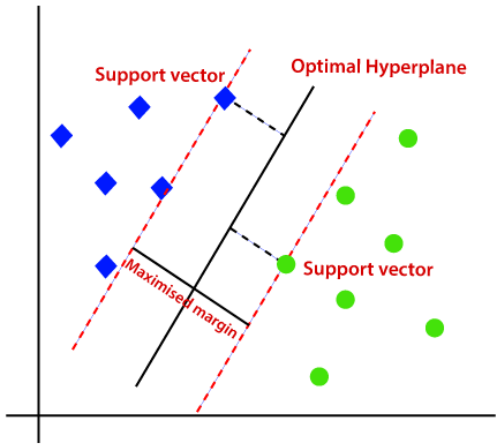


Fig 4.10 (c) : Optimal Hyperplane

The **WORKING** of Non - Linear Support Vector classifier is briefed below,

- In nonlinear cases, we cannot have a linear decision boundary. We start off by adding a third dimension
- If x and y are the two dimensions that are present, the third dimension z is calculated as $z = x^2 + y^2$. This will provide the third dimensional space
- A hyperplane that is parallel to the X axis at a certain z value is constructed by the algorithm.
- After this, the mapping is done similar to how it was done in two dimensional case

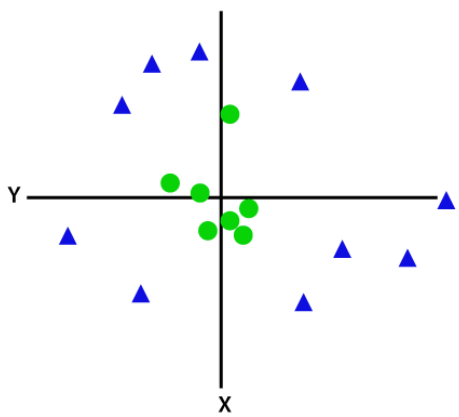


Fig 4.11 (a) : Original Dataset

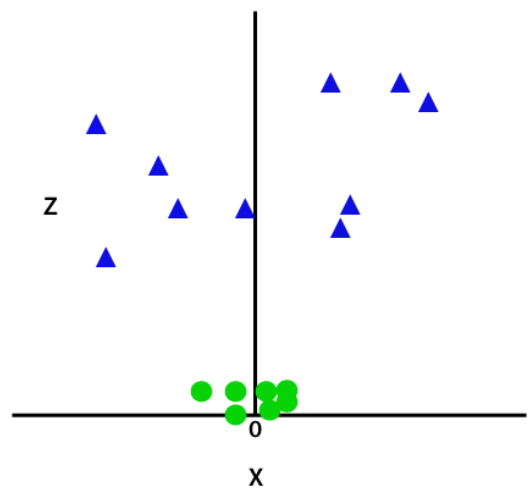


Fig 4.11 (b) : Adding 3rd Dimension

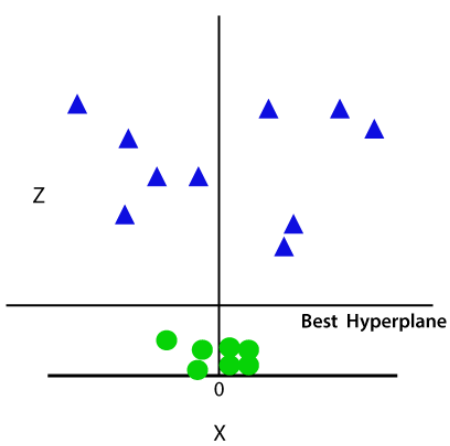


Fig 4.11 (c) : Obtaining Hyperplane

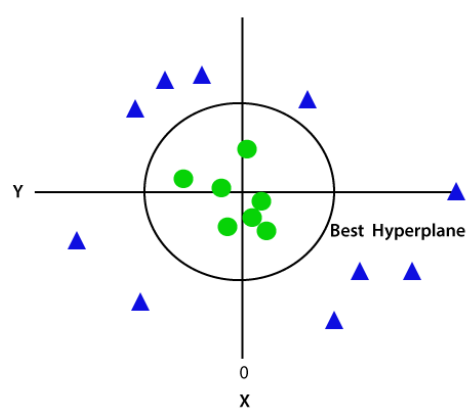


Fig 4.11 (d) : Hyperplane mapped to 2D

The **ADVANTAGES** of Support Vector classifier is briefed below,

- When there is a distinct and understandable margin between classes, this approach performs well.
- More efficient in environments with large spatial dimensions and where the number of dimensions exceeds the number of specimens
- Both classification and regression issues can benefit from the use of this approach. Additionally, it does well with image data.
- Efficient usage of memory

The **LIMITATIONS** of Support Vector classifier is briefed below,

- Not suitable for very large datasets
- Performance of the algorithm drastically reduces when there is more noise in the dataset.
- The algorithm might tend to underperform if the amount of features or properties is greater compared to the amount of training data samples.
- In presence of overlapping classes, the algorithm will underperform
- Choosing an optimal kernel function is a difficult task.

4.3 Implementation details

4.3.1 Softwares Architecture

1. **Arduino IDE**: Arduino IDE, or Arduino code (IDE), is associated with integrated surroundings that has a text editor, a message space and a console. It conjointly offers buttons for common functions and a menu system. This can also be ASCII text file software, that is employed to put in writing and transfer code to Arduino boards. The program eliminates the requirement for makefiles and command-line interfaces
2. **Python 3.8**: This is an advanced programming language that consumes 11% less memory than Python 3.7 making it more efficient and faster.

Some of the libraries used for making the machine learning models in this project are listed below:

Matplotlib	To produce publication-quality plots
SkLearn	Contains tools for machine learning and statistical modelling
NumPy	To conduct mathematical computations on arrays
Pandas	Provides high-performance data structures and tools for data analysis
TKinter	A standard Python interface to the Tk GUI toolkit
Operator	Exports a group of effective functions that match the intrinsic operators
Seaborn	Used to visualise data
Collections	Containers used for storing data, include data structures, such as lists, tuples, arrays, dictionaries, etc.
Math	Provides standard mathematical constants and functions
TensorFlow	Is an Open source library for numerical computations
Keras	Is Used for implementing neural networks

Table 4.3: List of all the libraries used in this project

3. **Flask**: It is described as a lightweight python microframework which is minimally dependent on external libraries. It is an API of python that allows us to build various types of web applications. It provides useful tools, libraries and techniques to build a web app in python language. It can also take the form of a web page, a blog, or a wiki, or it could also be used to make applications for commercial use. Henceforth, making it an easy-to-use python backend development tool.

4. **Jupyter Notebook:** This is a web interface that enables you to create and share documents with realtime code, computations, graphical representations etc. In this project, it is utilised to construct the Python programs.
5. **HTML & CSS:** These are two of the core technologies used to produce Web pages. The former language is used for structuring the page, while the latter language enables the creation of layouts for web pages for a broad range of screens or monitors.
6. **Jinja2:** Jinja2 is used by Flask to enable HTML templating. This is a Python programming language web template engine. It enables the clean addition of loops and variables into the Html document. Jinja is comparable to the Django template engine, except it supports Python-like expressions and verifies that templates are assessed in a sandbox.
7. **Gunicorn:** This is an HTTP server written entirely in Python for WSGI applications. It enables the simultaneous execution of any Python programme by executing many Python processes within a single dyno. It strikes the ideal mix between performance, versatility, and setup ease.
8. **MarkupSafe:** MarkupSafe introduces a text object for wanting to escape characters in HTML. Characters that are of special importance are replaced so that they appear authentic. This prevents injection attacks, allowing untrusted user input to be shown on a website securely.

4.4 Project Workflow

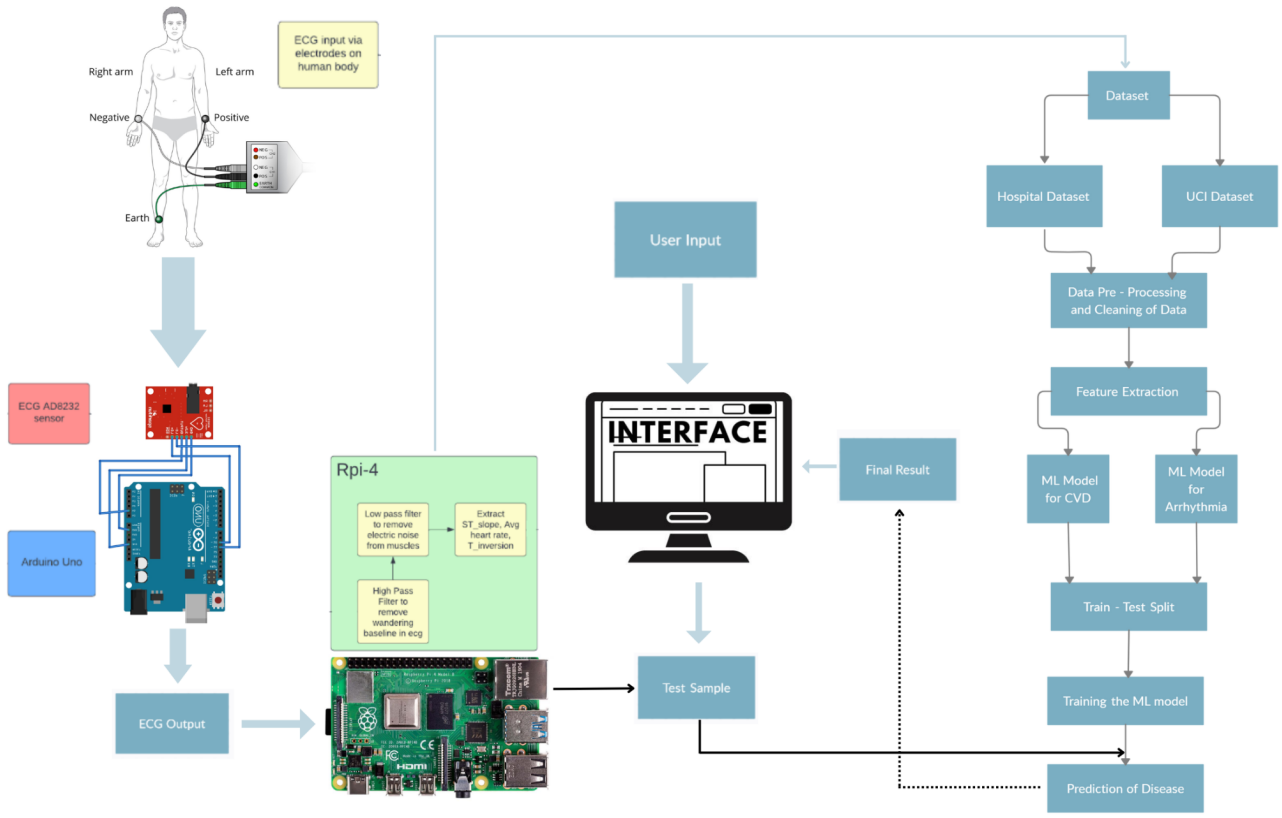


Fig 4.12: Block Diagram of the Project

4.4.1 Hardware Explanation

The ecg patches and ecg sensor AD8232 helps to collect the data of the electrical activities in the heart. Either the limb lead II or the chest leads V5 are used to extract the ECG signals.Chest leads are preferred as they produce less noise caused by the muscles . The ECG sensor AD8232 is a single lead sensor. It has 3 electrodes which are arranged in a specific triangular shape namely Einthoven's Triangle. This is an illustrative method of construction of three limb leads in an electrocardiographic triangle. The parts involved in the triangle are two shoulders and the pelvis. The middle of the triangle is the place where the heart is located. Different combinations of the leads give us different ECG graphs.

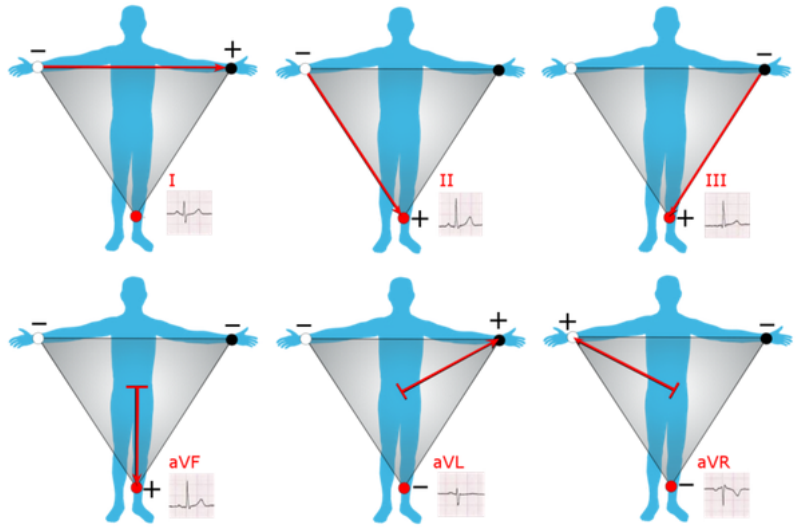


Fig 4.13 : Einthoven's triangle representation.

Einthoven's triangle can be useful in identifying erroneous lead placement. Incorrect lead placement can result in recording errors, which can lead to misdiagnosis. Lead I changes polarity when the arm electrodes are inverted, causing leads II and III to switch.The polarity of the Lead II is changed if the right arm and right leg electrodes are interchanged, causing Lead I to become Lead III and vice versa. Reversing the right leg and left arm changes the polarity of lead III and switches Leads I and Lead II.

Lead I = LA - RA .

Lead II = LL - RA.

Lead III = LL - LA.

The LO- Pin of sensor AD8232 and the Pin 11 of the arduino are connected. Similarly, the LO+ Pin and Pin 10 are connected. If both the pins are high, the arduino reads the input data from its analog PIN A0 which is connected to the output pin of the sensor. The power to the sensor is supplied by a 3.3V arduino pin. The ground connections are made. The voltage readings obtained from the ECG machine are in millivolts. Lead II is considered to detect CVD. This is further transferred to the Rpi-4 board with the help of UART serial communication protocol.

In the Rpi board 5000 data points are collected. The average time to collect this data is 30 seconds. All these data are stored in a .csv file. The .csv file is used to process the collected ECG signal. The signal is processed to remove noises caused due to breathing of a patient which results in baseline

wandering of the signal, the power line interference and also the muscle noise which manifests as high frequency noise in the ECG.

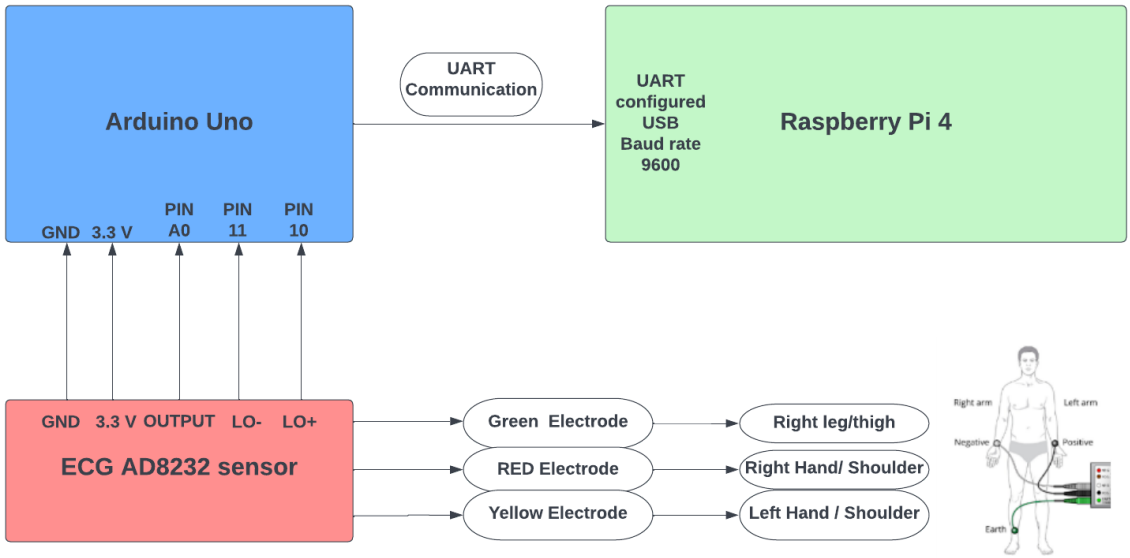


Fig 4.14 : Circuit Diagram

The wandering baseline frequency is about 0.5 hz to 2hz. A high pass butterworth filter is used. Its cutoff frequency is 2 hz and is used to remove any noise signals present below 2 hz frequency.The power line interference is caused because of the power provided to the ecg sensor by the Arduino and the muscle noise is caused by the electric activities in the muscles. Both these noises range from 60Hz to 100Hz and above. Hence a low pass butterworth is used. Its cutoff frequency is 50 hz and is used to remove any noise signals which are present above 50 hz frequency. The sampling frequency(Fs) used in both the cases is 160 Hz. Cutoff frequency for HPF is(FcH)=2 Hz. So Wn for HPF = 0.01. Cutoff frequency for LPF is(FcL)=2 Hz. So Wn for LPF = 0.31.

$$\omega_n = \frac{f_c}{f_s} \qquad \ldots (4.4.1.1)$$

where, ω_n = Critical Frequency ; f_c = Cutoff frequency ; f_s = Sampling frequency

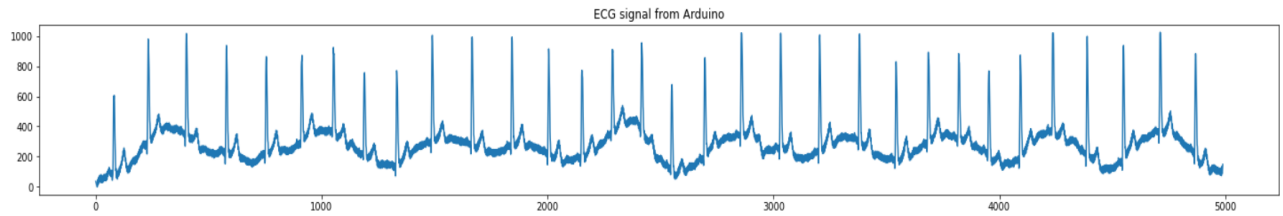


Fig 4.15 (a) : Raw Signal received from Sensor

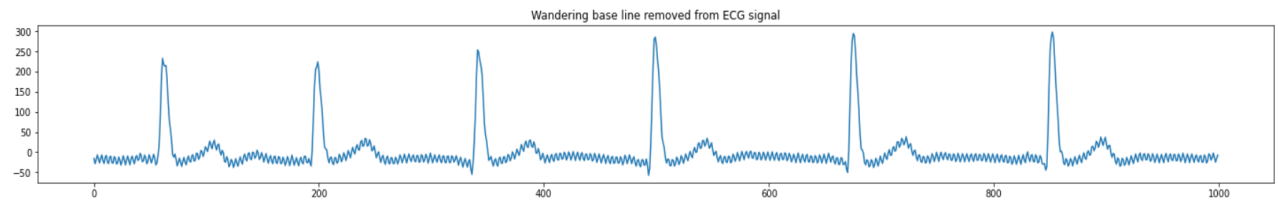


Fig 4.15 (b) : Baseline wandering removed and Zoomed Signal

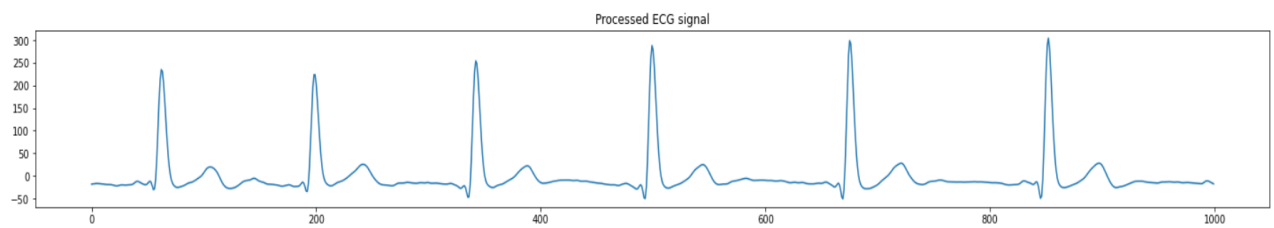


Fig 4.15 (c) : Powerline and Muscle noise removed and Zoomed Signal

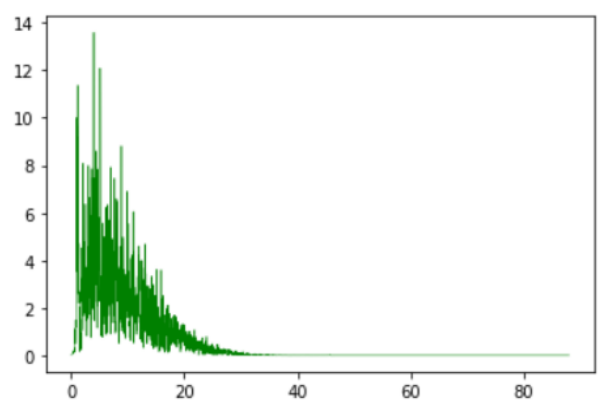


Fig 4.15(d) : Frequency plot of filtered ECG.

Fig 4.15(d) is a zoomed out graph which represents the frequency plot of the filtered ECG. The y axis represents the count of the frequencies and the x axis represents the frequency in Hz. The graph shows that the frequencies below 2 Hz and above 50Hz are removed efficiently.

4.4.1.1 Butterworth Filter:

It is an analogue filter that produces the best output response with no ripple in the pass or stop bands. Because of this the width of the transition band increases and the filter response is flat to maximum extent. The form or width of the roll-off is often known as the "transition band" in applications that utilise filters to alter the frequency spectrum of a signal, such as communications or control systems. The transition band of the first order filter might be lengthy or wide, compelling us to use higher order filters. These filters are usually referred to as "high-order" or "nth-order" filters.

$$H_{(j\omega)} = \frac{1}{\sqrt{1+\varepsilon^2\left(\frac{\omega}{\omega_p}\right)^{2n}}}$$

Fig 4.16: Nth order butterworth formula

This filter is also known as a maximally flat magnitude filter because it can produce a frequency response that is completely flat, theoretically. The frequencies above the cut-off point band roll down to zero at a rate of 20dB per decade or 6dB per octave in the stop band as the quality factor is only 0.707 .The fundamental drawback of the Butterworth filter is that it moves from pass band to stop band by obtaining pass band flatness at the price of large transition bands.The below illustrated figure shows the ideal frequency response and and also the standard estimate of low pass butterworth filter with various order.

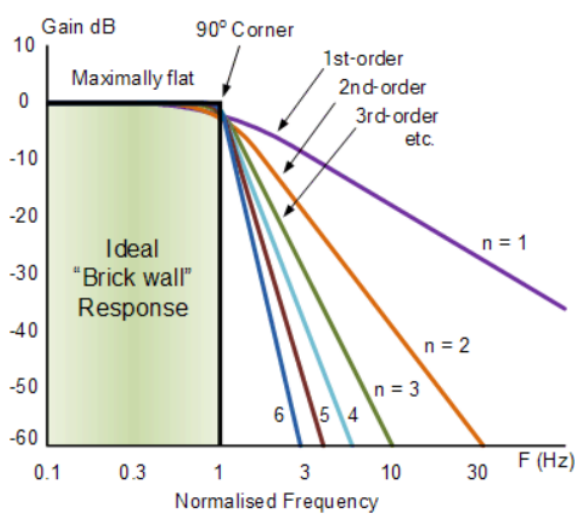


Fig 4.17(a): Butterworth Ideal frequency Response

The implementation of butterworth filter is mainly by 2 methods: matched z-transform and bilinear transform. These two approaches can be used to critique an analogue filter design. When we consider the Butterworth filter, which contains all-pole filters, we can say that the approaches impulse variance and matched z-transform are similar.

n	Normalised Denominator Polynomials in Factored Form
1	$(1+s)$
2	$(1+1.414s+s^2)$
3	$(1+s)(1+s+s^2)$
4	$(1+0.765s+s^2)(1+1.848s+s^2)$
5	$(1+s)(1+0.618s+s^2)(1+1.618s+s^2)$
6	$(1+0.518s+s^2)(1+1.414s+s^2)(1+1.932s+s^2)$
7	$(1+s)(1+0.445s+s^2)(1+1.247s+s^2)(1+1.802s+s^2)$
8	$(1+0.390s+s^2)(1+1.111s+s^2)(1+1.663s+s^2)(1+1.962s+s^2)$
9	$(1+s)(1+0.347s+s^2)(1+s+s^2)(1+1.532s+s^2)(1+1.879s+s^2)$
10	$(1+0.313s+s^2)(1+0.908s+s^2)(1+1.414s+s^2)(1+1.782s+s^2)(1+1.975s+s^2)$

Fig 4.17(b) : Examples of normalised second order low pass polynomials.

The advantages of the Butterworth filter include

- Smooth curve.
- There are no ripples in the pass band.
- There are no ripples in the stop band.

The disadvantages of Butter worth filter include:

- Very slow roll off speed.
- The order required is usually high.
- This filter involves a lot of time delay.

The signal, after this process, is a filtered ECG. The frequency response of the final signal is plotted. the frequencies less than 1Hz and more than 100Hz are removed. The ECG signal is plotted for visualisation. The Y coordinates of the graph represent the amplitude of the signal or the strength of the signal in millivolts and the X coordinates represent the count of the signal data. The X coordinates of the P, Q, R, S and T are found with the help of functions present in the library named Neurokit2. The coordinates are returned in the form of an array. These are used further to find three important outputs namely:

- ST slope.
- Max heart rate at the time of sample collection.
- T inversion.

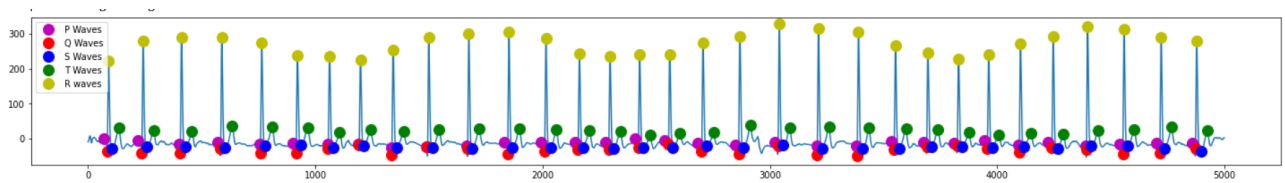


Fig 4.18 : P, Q, R, S and T peaks marked in the completely processed signal.

ST slope, as the name goes, is the slope of the ST segment. To find the slope of ST segment 2 methods are used. The first method includes using the slope formula.

$$m = \frac{\text{rise}}{\text{run}} = \frac{y_2 - y_1}{x_2 - x_1}$$

Fig 4.19 : Slope formula

The x and the y coordinates of S peak are x1 and y1 respectively. The x and y coordinates of the Tpeak are x2 and y2 respectively. An array of slopes are obtained. The mean of this slope is found to give the approximate slope of the ST segment.

The second method includes differentiation. As the ST segment is not always a straight line, the array segment is differentiated and the slope is obtained. Similar to the previous method a list of slopes are obtained. The mean of this is considered for further evaluation. The acceptable value of the slope is between 0 to 1. Slope more or less than this can indicate abnormalities.

To find the heart rate, the number of R peaks is counted. This is then multiplied by 2. The reason is, the ECG signal is collected for a time duration of 30 seconds and the total number of R peaks is extrapolated to 60 seconds by multiplying it by 2.

To find the T inversion, the amplitude of T peaks is extracted and checked if any peak is less than -0.1mV. If the value is less than -0.1mV, it indicates abnormalities and is detected as an inverted T peak and the value 1 is returned. If T inversion is not detected 0 is returned. The max heart rate in general is calculated using the formula:

Max heart rate = 220 - age … (4.4.1.1)

The target heart rate is 75% of max heart rate. But there is no scientific backing to his formula. The presence of the number 220 was never explained correctly. Recently scientists have conducted tests which shows that the formula :

HRmax = 205.8-0.685(age) … (4.4.1.2)

can be used in some cases. Nevertheless, this formula also is erroneous and cannot be agreed upon. Hence in this project, both these formulas are not used to find the max heart rate of a person. The avg heart rate, the t inversion and the ST slope values are later stored in the .csv files which are further added to the interface which runs a ML model to predict the probability of disease.

4.4.2 Software Explanation

There are three parameters that are obtained from the hardware. Maximum heart the rate at the time of collection, ST peak and T inversion are the three parameters obtained which would be sent as inputs to the interface for CVD along with other important parameters.

The UCI Dataset for CVD prediction consists of multivariate data where attributes are categorical, integer and real. The considered dataset contains 303 instances and 75 attributes. Among these, a subset of fourteen attributes is mostly included in published experiments. The important parameters considered in our project from the UCI dataset for the prediction of CVD are:

- Age : Age in years
- Gender : 0-female , 1-male
- Chest pain type (cp) : 1-typical angina, 2-atypical angina, 3-non anginal pain, 4-asymptomatic
- Resting blood pressure (trestbps) : Measured in mm of Hg
- Cholesterol (chol) : Serum cholesterol in mg/dl
- Resting ECG results (restecg) : 0-normal, 1-ST-T wave abnormality, 2-probable or definite left ventricular hypertrophy
- Exercise induced angina (exang) : 0-absent, 1-present
- Maximum heart rate (thalach)
- Fasting Blood Sugar (fbs) : 0-if fbs<120 mg/dl, 1-if fbs>120 mg/dl
- Oldpeak : ST depression induced by exercise relative to rest
- Slope : the slope of the peak exercise ST segment 0-upsloping, 1-flat, 2-downsloping
- Number of major vessels involved (ca) : 0-3
- Thalassemia (thal) : 0-normal, 1-fixed defect, 2-reversible defect
- Condition : 0-no disease, 1-disease

The UCI Dataset for the prediction of Cardiac Arrhythmia consists of multivariate data where attributes are categorical, integer and real. The dataset has 452 instances and 279 attributes. Among these, 206 attributes are valued linearly and the remaining are nominal. This dataset also consists of missing values. Hence, important parameters considered in our project from the UCI dataset for Cardiac Arrhythmia are :

- Age
- Gender
- Height: Height in centimetres
- Weight: Weight in kilograms
- QRS duration: Average of QRS duration in msec
- P-R interval: Average duration between onset of P and Q waves in msec
- Q-T interval: Average duration between onset of Q and offset of T waves in msec
- T interval: Average duration of T wave in msec
- P interval: Average duration of P wave in msec
- Vector angles in degrees on front plane of of QRS
- T

- P
- QRST
- J
- Heart rate: Number of heartbeats per minute (linear)

Further, two separate machine learning models were developed and trained using the UCI datasets repository for CVD and Cardiac Arrhythmia. The data required is cumulated, cleaned and processed. This is done with the aim of extracting features according to the parameters necessary for the two heart diseases selected. We decided to implement multiple ML algorithms in order to increase efficiency and get better accuracy.

Some algorithms used for CVD are Logistic Regression, KNN, Support vector machine(SVM), Random Forest and Decision Tree Classifier.

The reason for implementing logistic regression classifier is that we had a classification problem where a test sample had to be categorised into two classes. Along with classification results, logistic regression produces probabilities that are accurately calibrated. Logistic regression is particularly effective and less prone to overfitting in situations where characteristics in the UCI dataset can be linearly separated. Hence, a logistic regression classifier is used.

The reason for implementing the K-Nearest Neighbour classifier is that it is an adaptable and flexible algorithm and can be used for problems dealing with classification. In our problem, we have to categorise a new test sample into two classes. This algorithm is simple and robust to noisy data. In cases of exceptionally large datasets, the KNN algorithm is highly effective. Hence KNN classifier is implemented.

The reason for implementing the SVM classifier is that it works well on classification problems. It works when the dataset's two classes are clearly separated by a large margin. This makes the usage of SVM a good decision. The classifier is efficient in memory usage and has higher speed and improved performance in comparison to other neural network models. Hence Support Vector classifier is implemented.

A decision tree classifier was employed because it may use a variety of feature subsets and decision rules at different stages of classification. These divide complicated data into more manageable chunks, these have better interpretability and work faster with categorical/numerical values. These classifiers also need less computation power, therefore minimising the implementation time required to compute the final results. Hence, to further enhance the results Random Forest Classifier was applied as it offers more randomness to the model while growing trees. It is also observed that these also tend to provide the highest accuracy in certain cases.

Upon exploring the dataset, we found the following results :

- 'heart' is the variable that holds the dataset. .info() function gives the summary of the dataset. We can see that there are 297 rows and 14 columns as a part of the dataset. It also talks about all the attributes present and their datatypes. The number of non null values can be seen for each attribute. Heart_df is the copy of the original dataset.

```
print(heart.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 297 entries, 0 to 296
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        297 non-null    int64
 1   sex        297 non-null    int64
 2   cp         297 non-null    int64
 3   trestbps   297 non-null    int64
 4   chol       297 non-null    int64
 5   fbs        297 non-null    int64
 6   restecg    297 non-null    int64
 7   thalach    297 non-null    int64
 8   exang      297 non-null    int64
 9   oldpeak    297 non-null    float64
 10  slope      297 non-null    int64
 11  ca         297 non-null    int64
 12  thal       297 non-null    int64
 13  condition  297 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 32.6 KB
None
```

Fig 4.20 : Summary of Dataset

- .isnull().values.any() returns the number of null values that are present across the dataset. A null value heatmap is displayed which depicts that there are no null values present.

```
heart_df.isnull().values.any()
```

```
False
```

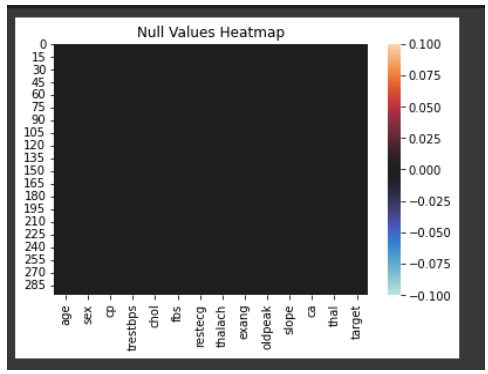


Fig 4.21: Null Values Heatmap

- From the value_counts() we get to know that out of the total instances present, 160 belong to class Disease and 137 belong to class No Disease

```
heart_df['target'].value_counts()
```

```
0    160
1    137
Name: target, dtype: int64
```

Fig 4.22 : Samples in Each Class

- We can observe from the pie graph that 54% of the people have heart disease and 46% do not. We can also conclude that the dataset considered for this work is a balanced dataset.

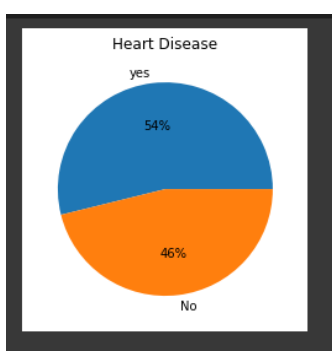


Fig 4.23 : Percentage of Positive and Negative samples

- Distribution graphs for each of the features are plotted using matplotlib to analyse how each feature impacts the final prediction result. From the below distribution graphs, we can observe that most people are aged between 50 and 65. Most people are male. Common heart rate is between 150-180 and blood pressure levels are in the range of 120 to 145. Cholesterol levels are seen to be in the range of 200-300 for most people.

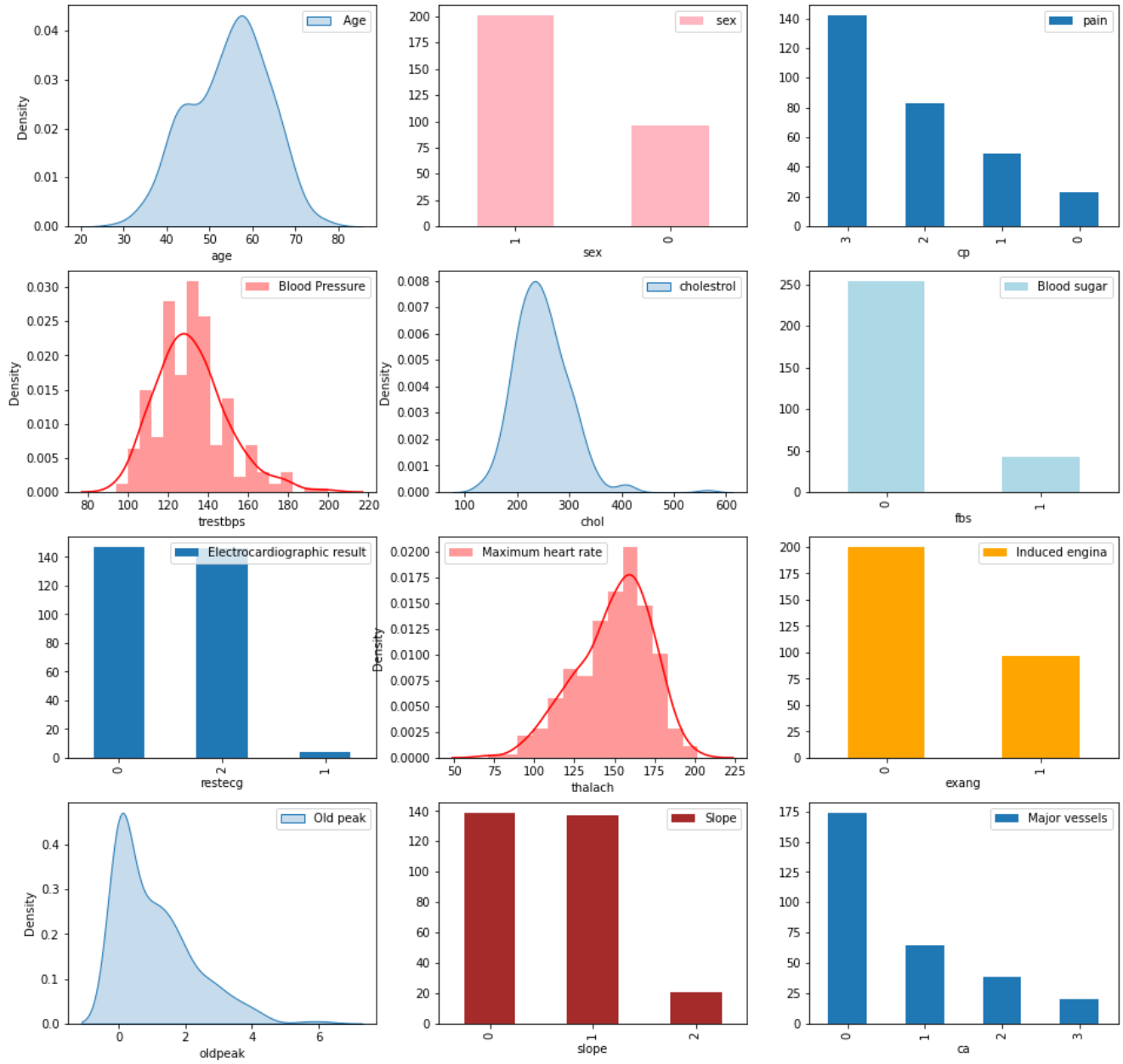


Fig 4.24: Distribution of Features

We can make the following inferences when we analyse the effect of individual features on the classification:

- Age: People aged between 40-75 are at high risk of having CVD
- Chest Pain Type: It is observed that a high percentage of people with heart disease have asymptomatic chest pain type
- Blood Pressure: It is seen that for the range 110-140 of blood pressure, the risk of heart disease shot up
- Cholesterol: People with serum cholesterol levels between 200-300 are at high risk of having CVD
- Resting ECG results: People with probable or left ventricular hypertrophy have the risk of heart disease and people with normal ECG do not have the disease
- Maximum Heart Rate: People with heart rate above 150 are at high risk of developing heart disease.
- Old Peak: If ST depression is around 1.5, people have CVD. If it is around 0.6, there are minimal chances of CVD
- Exercise-Induced Angina: Most people who have exercise-induced angina are likely to develop CVD
- Slope: People with flat ST segment have CVD and people with upsloping ST segment have no CVD
- Number of Vessels: People who do not have major vessels coloured by fluoroscopy have no disease
- Thalassemia: It was seen that people with reversible defect of thalassemia have a high risk of heart disease.

We plot the correlation map with the following command,

```
plt.figure(figsize=(15, 15))
plt.title('Correlation Matrix', size=20)
sns.heatmap(heart_df.corr(), annot = True, cmap = "Greens")
plt.show()
```

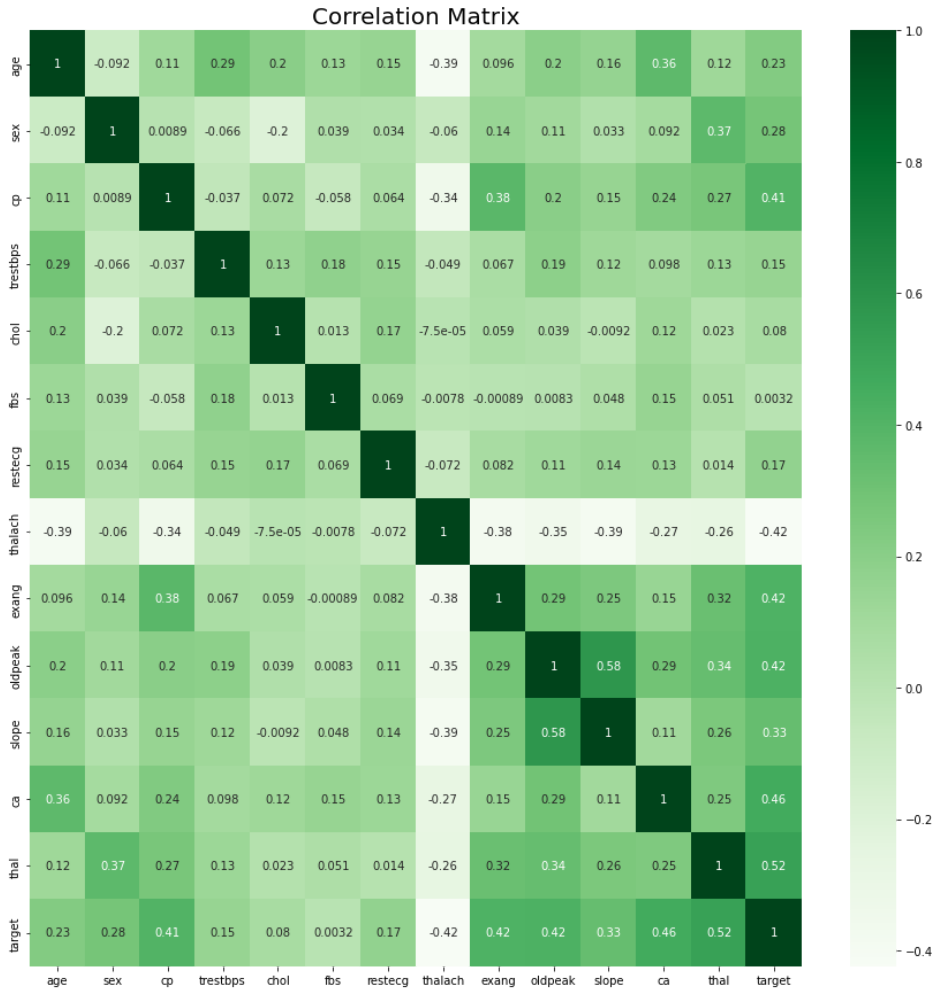


Fig 4.25: Correlation Matrix

From the above plot, we can make the following conclusions:

- There is a relation between the number of major vessels and age
- There is a relation between old peak and slope of the ST segment
- Exercise induced angina and the type of chest pain are related.
- There is the relation between maximum heart rate and age
- There is an inverse relationship between maximum heart rate and final prediction output
- Chest pain type, exang, oldpeak, slope, number of major vessels and thalassemia are directly correlated with the prediction of heart disease.

The type of dataset used in building the model is multivariate. The dataset is trained and tested with a train-test split ratio of 75:25(75% training dataset and 25% testing dataset) which makes the

dimensions 222 x 75 records. The instances and attributes size is 297 x 14. The number of attributes considered is 14, amongst the entire number of attributes which is 75. The amount of positive cases present in the dataset is 137 and that of negative cases is 160. The results were of two values : 0 and 1. If the result obtained is 0, it indicates no disease and if the result obtained is 1, it indicates the disease is present.

Source	UCI
Disease Considered	CVD (Cardiovascular Disease)
Type of dataset	Multivariate
Attribute Characteristics	Categorical, Integer, Real
Size (Rows x Cols)	297x14
Train:Test split	75:25 (222 & 75 records)
Number of Attributes Considered	14
Name of Attributes	age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, num
Total Positive Cases	137
Total Negative Cases	160
Number of Missing Values	0
Output Values	0 - no disease ; 1 - disease
Logistic Regression Accuracy	76.0 %
KNN Accuracy	74.67 %
Support Vector Classifier Accuracy	78.67 %
Random Forest Classifier Accuracy	77.33 %
Decision Tree Classifier Accuracy	70.67 %

Table 4.4: Implementation of ML Algorithms on UCI Dataset for CVD

On implementing several ML algorithms on the UCI dataset for CVD we obtain the following results:

```
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.79      0.77        38
           1       0.77      0.73      0.75        37

    accuracy                           0.76        75
   macro avg       0.76      0.76      0.76        75
weighted avg       0.76      0.76      0.76        75

Accuracy: 76.0%
```

```
cm = confusion_matrix(y_test, y_pred_LR)
cm
```

```
array([[30,  8],
       [10, 27]])
```

Fig 4.26: Performance Analysis of Logistic Regression

```
Classification Report
              precision    recall  f1-score   support

           0       0.74      0.76      0.75        38
           1       0.75      0.73      0.74        37

    accuracy                           0.75        75
   macro avg       0.75      0.75      0.75        75
weighted avg       0.75      0.75      0.75        75

Accuracy: 74.67%
```

```
cm = confusion_matrix(y_test, y_pred_knn)
cm
```

```
array([[29,  9],
       [10, 27]])
```

Fig 4.27: Performance Analysis of KNN

```
Classification Report
              precision    recall  f1-score   support

           0       0.79      0.79      0.79        38
           1       0.78      0.78      0.78        37

    accuracy                           0.79        75
   macro avg       0.79      0.79      0.79        75
weighted avg       0.79      0.79      0.79        75

Accuracy: 78.67%
```

```
cm = confusion_matrix(y_test, y_pred_SVC)
cm
```

```
array([[30,  8],
       [ 8, 29]])
```

Fig 4.28 : Performance Analysis of SVM

```
Classification Report
              precision    recall  f1-score   support

           0       0.76      0.82      0.78        38
           1       0.79      0.73      0.76        37

    accuracy                           0.77        75
   macro avg       0.78      0.77      0.77        75
weighted avg       0.77      0.77      0.77        75

Accuracy: 77.33%
```

```
cm = confusion_matrix(y_test, y_pred_RF)
cm
```

```
array([[31,  7],
       [10, 27]])
```

Fig 4.29: Performance Analysis of Random Forest

```
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.63      0.69        38
           1       0.67      0.78      0.72        37

    accuracy                           0.71        75
   macro avg       0.71      0.71      0.71        75
weighted avg       0.71      0.71      0.71        75

Accuracy: 70.67%
```

```
cm = confusion_matrix(y_test, y_pred_DT)
cm
```

```
array([[24, 14],
       [ 8, 29]])
```

Fig 4.30: Performance Analysis of Decision Tree

As for cardiac Arrhythmia, SVM with Random Forest Classifier is used with linear, RBF, and polynomial kernels. Initially, Principal Component Analysis(PCA) is applied for feature selection. Following that, SVM (Support Vector Method) is used to effectively predict data with very high dimensions, which is then combined with a Random Forest classifier to improve accuracy, generalisation, and comprehensibility because it is profoundly preferable for multiclass problems and (speedier than non-linear) SVMs. This model constantly samples with replacements on a subset of the training dataset. In the next step, a decision tree is used to fit the model. Each tree created consists of multiple yes-no questions based on different attributes. The tree does not consider all of the attributes, confirming that separate decision trees are not connected. As a result, the classifier is less likely to overfit. The Gini index is used to calculate impurity. To give a window to modify data, several kernels are employed to enhance SVM classifier output. In general, kernel functions enable a non-linear decision surface to be converted into a linear equation in higher dimensional spaces by transforming the training dataset. A linear kernel is used when data can be separated using a single line, in other words, when the data is linearly separable, the polynomial kernel is similar but the boundary is of some defined but arbitrary order. And since the Radial Basis Kernel only maintains the support vectors during training and not the complete dataset, it solves the space complexity issues when trying to find a non-linear classifier or regression line.

Data preprocessing is first done by deleting columns with more than forty per cent of missing values, the missing values are later imputed by substituting the attribute median, and lastly, the attributes are normalised. In attributes with a higher deviation, it is recommended to use the median instead of the mean of missing values. For feature selection using PCA, according to Kaiser's rule components with eigenvalues, greater than or equal to 1 are selected.

Similar to the previous model, for cardiac arrhythmia, the dataset is of type multivariate, and the train-test split ratio considered for this model is 70:30(70% training dataset and 30% testing dataset) which makes the dimensions 316 x 136 records. Instances and attributes size is 452 x 280. The number of attributes considered is 15, amongst the total number of attributes which is 279 in which 206 are linear and 73 are nominal with categorical class attributes. And, the total number of positive cases found from the dataset is 207, and that of the negative cases found is 245. Since there were a lot of missing values found in the dataset which accumulated up to 376 values. The final results are obtained specifying 16 classes based on the value of attributes: class 1 identifies normal, and class 2 to 15 identifies classes indicating various types of arrhythmia.

Disease Considered	Cardiac Arrhythmia
Source	UCI
Type of Dataset	Multivariate
Attribute Characteristics	Categorical, Integer, Real
Size (Instances x Attributes)	452 x 280
Train: Test split	70:30 (316 & 136 records)
Total Number of Attributes	279
Number of Attributes considered	15
Total Positive Cases	207
Total Negative Cases	245
Output Values	0 - Class 1, Normal 1 - Class 2-15; various types of Arrhythmia
Name of Attributes	Age, Sex, Height, Weight, QRS duration, P-R interval, Q-T interval, T interval, P interval, Vector angles in degrees on the front plane of QRS, T, P, QRST and J, Heart rate
Number of Missing Values	376
SVM w/ RF (Linear, RBF, Polynomial)	73%, 66%, 68%
Linear SVM Accuracy	74%

Table 4.5: Implementation of ML Algorithms on UCI Dataset for Cardiac Arrhythmia

On implementing SVM using Random Forest classifier on the UCI dataset for Arrhythmia we obtain the following results:

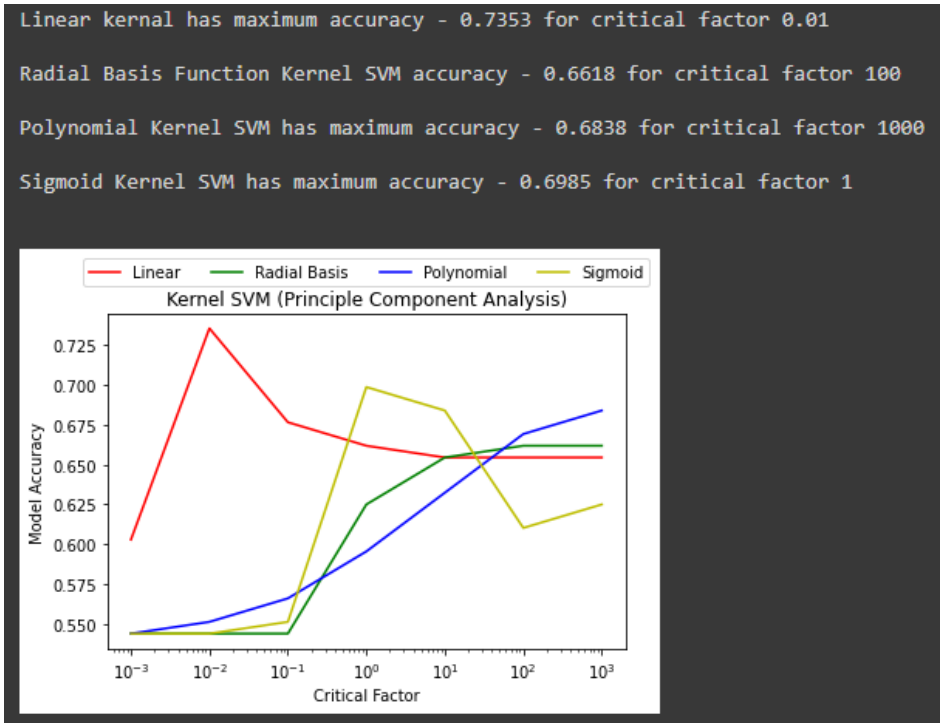


Fig 4.31: Model Accuracy of SVM with Random Forest classifier

After training the models, in order to test them, data is manually entered into the interface according to the parameters specified for each disease. The inputs are processed in the ML models connected to the two interfaces. And the final outputs are displayed through the GUI built for both diseases.

4.4.3 Building GUI

The interface for CVD is built mainly using Flask which is a microframework written in Python, along with HTML to provide page structure, and CSS to provide page layout. A few of the other required modules are libraries mentioned in the software requirements which have to be installed into the IDE. Then the machine learning model that is connected is run in order to obtain the .pkl(pickle) file of the required algorithm, indicating that the model has finished being trained and is

ready for testing. The command to start the server has to be executed for the web page to run on the web browser with all input fields. At last, the user provides the necessary details into the input fields, and the final results are calculated using the machine learning model by clicking the predict buttons and displayed for the user on the GUI.

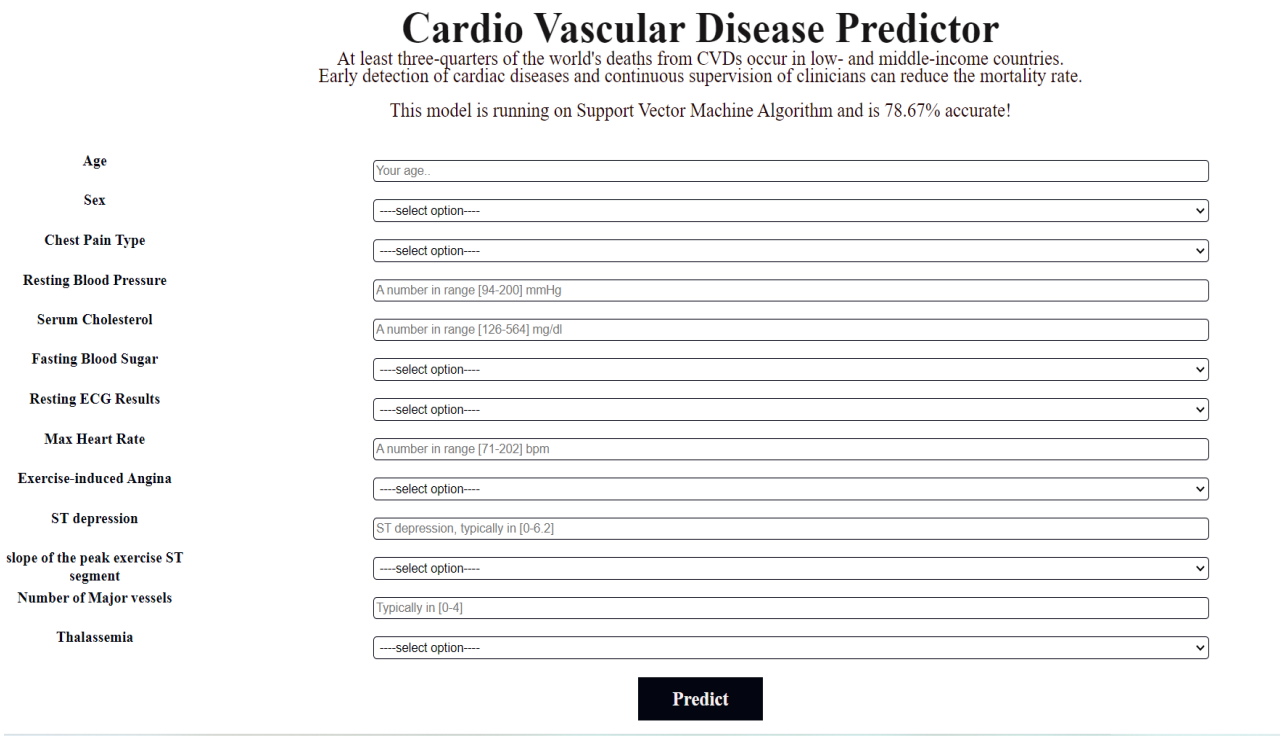


Fig 4.32: GUI to Detect Cardiovascular disease

The interface for the prediction of Cardiac Arrhythmia is built using a Python module Tkinter. GUI application development is made quick, simple, and dependable with Tkinter. Custom-made buttons are incorporated into the Tkinter module to facilitate the functioning of the GUI. The home page, which is the initial page, has a title and a button to go to another page. Then the user enters data from the UCI dataset in a text box on the second page for the prediction of Cardiac Arrhythmia. The input data is tested with SVM, Decision Tree and KNN which output the class of Arrhythmia. The final part is displaying the most probable class of arrhythmia from the obtained results.

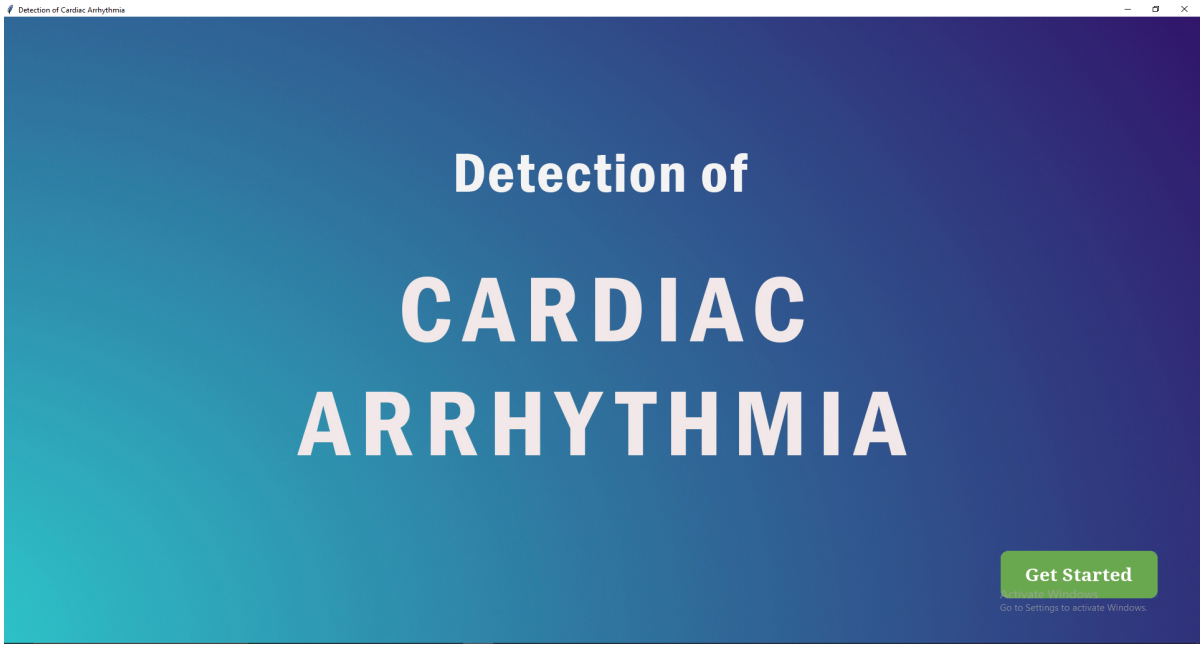


Fig 4.33: Home Page of GUI to Detect Cardiac Arrhythmia

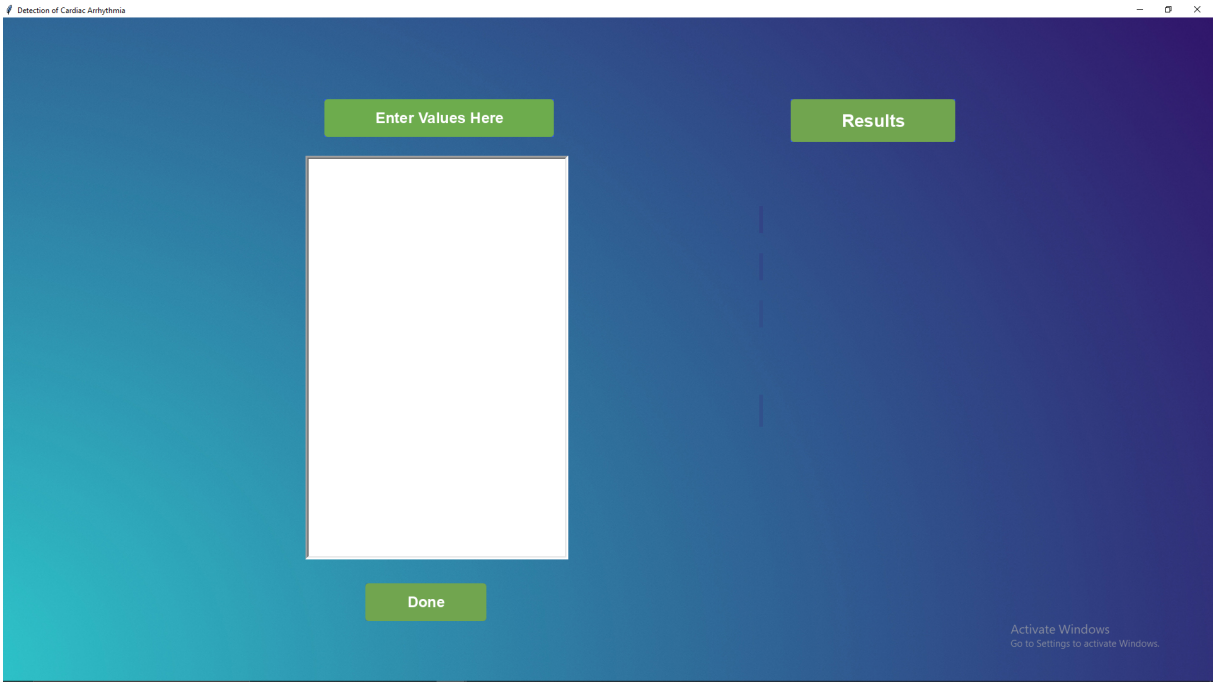


Fig 4.34 : Page with Text Field and Results

5. Results

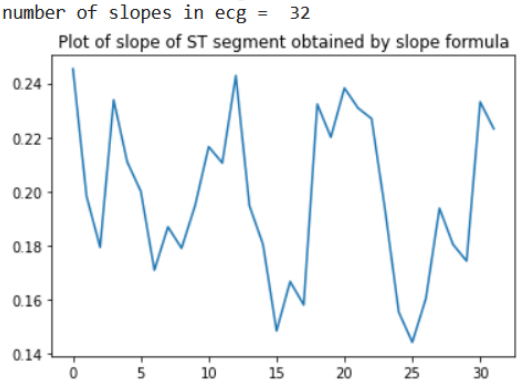
The proposed intelligent healthcare kit is being deployed and tested. The machine learning model performed up to the expectation and achieved the highest accuracy. The main operation that was performed is from an ECG sensor, it measures electrical activity and records the electrical signals in the form of graphs. The main discussion of the above project is to Classify the two types of heart disease(CVD and Cardiac Arrhythmia). The secondary operation is to detect the risk of disease by creating an application interface. The output of the same is executed on the Graphical User Interface(GUI) platform.

5.1 Simulated outputs

5.1.1 ECG output waveform

The ECG sensor(AD8232) is connected to an Arduino Uno board and simulated on Arduino IDE. The blue signal indicates the slope, it is plotted with respect to count. The slopes are extracted from the modified ECG signals that are being processed.

```
Slopes obtained using formula =  [0.2452477  0.19830768 0.17944603 0.23381315 0.21091395 0.20007977
 0.17100005 0.18696656 0.17907661 0.19486838 0.21654041 0.21057328
 0.24281082 0.19480789 0.18050972 0.14858213 0.16680103 0.15818027
 0.23225104 0.22005703 0.2381933  0.23083638 0.22688075 0.19304215
 0.15553912 0.14437222 0.16054441 0.19383145 0.18048518 0.17438015
 0.23307587 0.22317471]
number of slopes in ecg =  32
```




```
mean slope using formula =  0.19766216230330091
mean slope using differentiation =  0.20175104504029476
```

```
number_of_ecg_r_peaks    32
THE SAMPLE WAS COLLECTED FOR 30 SECONDS. TO FIND THE HEART RATE NUMBER OF R PEAKSx2 =  64
No T wave inversion detected
Enter Patient ID : 123123
Enter age : 21
average_heart_rate =  64
```

Fig 5.1 Graph representing the plot of slopes extracted from ECG.

5.1.2 Machine learning model output to predict Cardiovascular Disease (CVD)

Logistic Regression	76.0 %
K - Nearest Neighbour	74.67 %
Support Vector Classifier	78.67 %
Random Forest Classifier	77.33 %
Decision Tree Classifier	70.67 %

Table 5.1: Tabular Column of Cardiovascular Disease

The output value is obtained in binary format in which 0 indicates no disease and 1 indicates the person is diagnosed with heart problems. The KNN algorithm, SVM, Random Forest Classifier and Decision Tree Classifier are machine algorithms used to classify and predict the above heart disease. Support Vector Classifier shows the highest accuracy with 78.67%.

5.1.3 Machine learning model output to predict Cardiac Arrhythmia

SVM w/ RF (Linear, RBF, Polynomial)	73%, 66%, 68%
Linear SVM Accuracy	74%

Table 5.2 Tabular Column of Cardiac Arrhythmia

The output value is obtained; 0 indicates class 1, Normal, person is not diagnosed with Arrhythmia and the person is categorised as normal. 1 indicates class 2-15, a person is diagnosed with various kinds of arrhythmia. Support Vector Machine implemented with RBF, Linear, and polynomial kernel is used to classify different types of Arrhythmia and predict the same. Linear SVM shows the highest accuracy with 74%. Principle Component Analysis(PCA) is initially done to extract features. SVM uses various kernels. Results are later enhanced with Random Forest Classifier.

5.1.5 Accuracy Outputs of ML algorithms used to predict CVD

```
Support Vector Classifier

[ ] SVC_model= SVC()
    SVC_model.fit(x_train_scaler, y_train)
    y_pred_SVC= SVC_model.predict(x_test_scaler)
    SVC_model.score(x_test_scaler,y_test)

    0.7866666666666666

    print('Classification Report\n', classification_report(y_test, y_pred_SVC))
    print('Accuracy: {}%\n'.format(round((accuracy_score(y_test, y_pred_SVC)*100),2)))

    Classification Report
                  precision    recall  f1-score   support

               0       0.79      0.79      0.79        38
               1       0.78      0.78      0.78        37

        accuracy                           0.79        75
       macro avg       0.79      0.79      0.79        75
    weighted avg       0.79      0.79      0.79        75

    Accuracy: 78.67%

[ ] cm = confusion_matrix(y_test, y_pred_SVC)
    cm

    array([[30,  8],
           [ 8, 29]])
```

Fig 5.2 Accuracy and classification report of Support Vector Classifier

5.1.5 Accuracy Outputs of machine learning algorithms used to predict Cardiac Arrhythmia

```
Accuracy for SVM - Linear Kernal -  0.7353

              precision    recall  f1-score   support

           1       0.70      0.96      0.81        74
           2       0.75      0.69      0.72        13
           3       1.00      1.00      1.00         4
           4       0.60      0.75      0.67         4
           5       1.00      0.25      0.40         4
           6       0.00      0.00      0.00         8
           7       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         1
           9       1.00      0.67      0.80         3
          10       1.00      0.67      0.80        15
          14       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          16       0.00      0.00      0.00         7

    accuracy                           0.74       136
   macro avg       0.47      0.38      0.40       136
weighted avg       0.66      0.74      0.68       136

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classifi
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classifi
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classifi
  _warn_prf(average, modifier, msg_start, len(result))
```

Fig 5.3 Precision and Recall of SVM Linear Kernel

5.1.6 GUI Output of Cardiovascular Disease(CVD) and Cardiac Arrhythmia

A separate independent user interface is built (using HTML, CSS, Flask framework with python) for each disease where the user provides necessary input. The GUI shows the number of normal or abnormal outputs depending on the inputs given by the user

5.1.7 Negative case GUI output of Cardiovascular Disease(CVD)

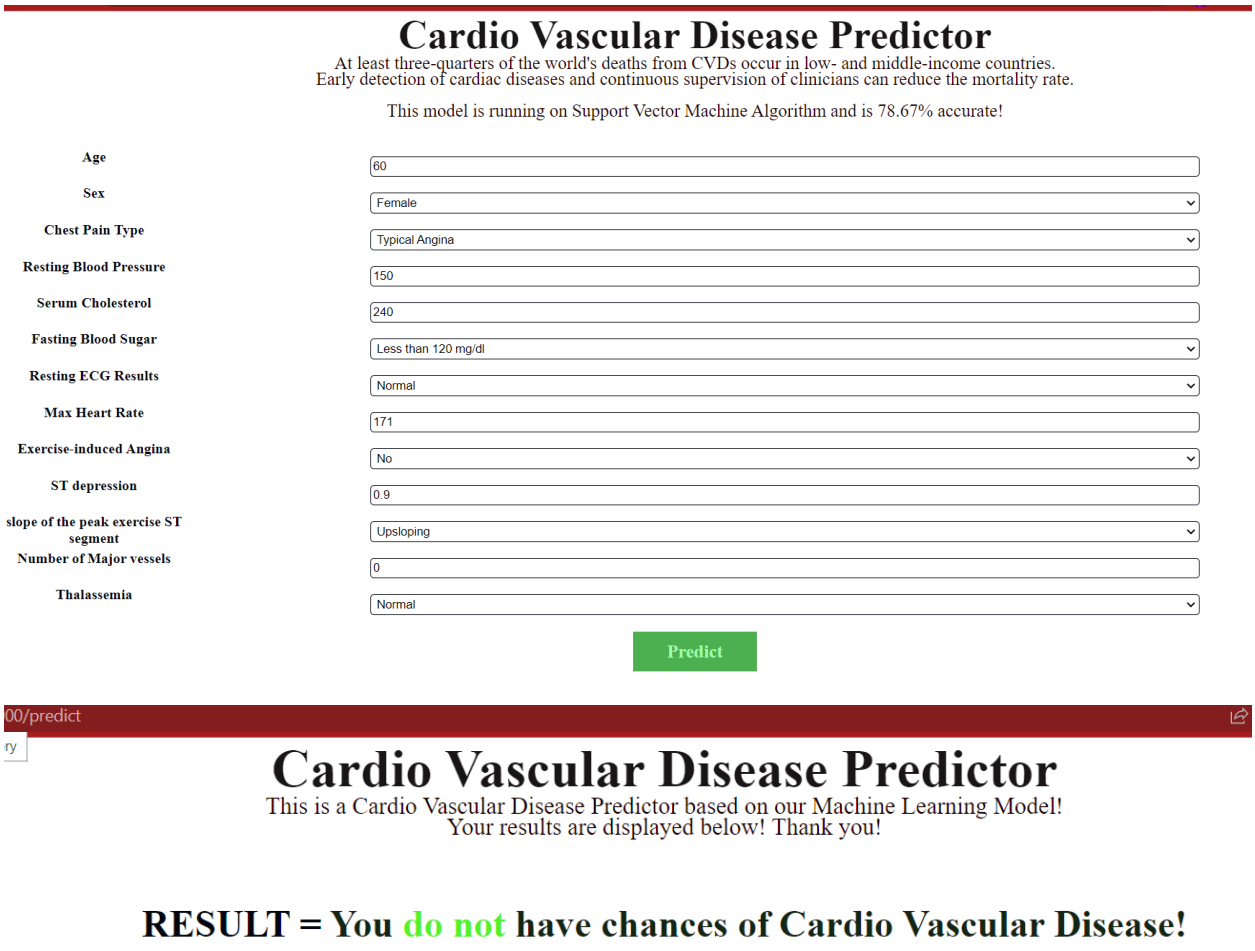


Fig 5.4 : Cardiovascular Disease Predictor for Negative case

5.1.8 Positive case GUI output of Cardiovascular Disease(CVD)

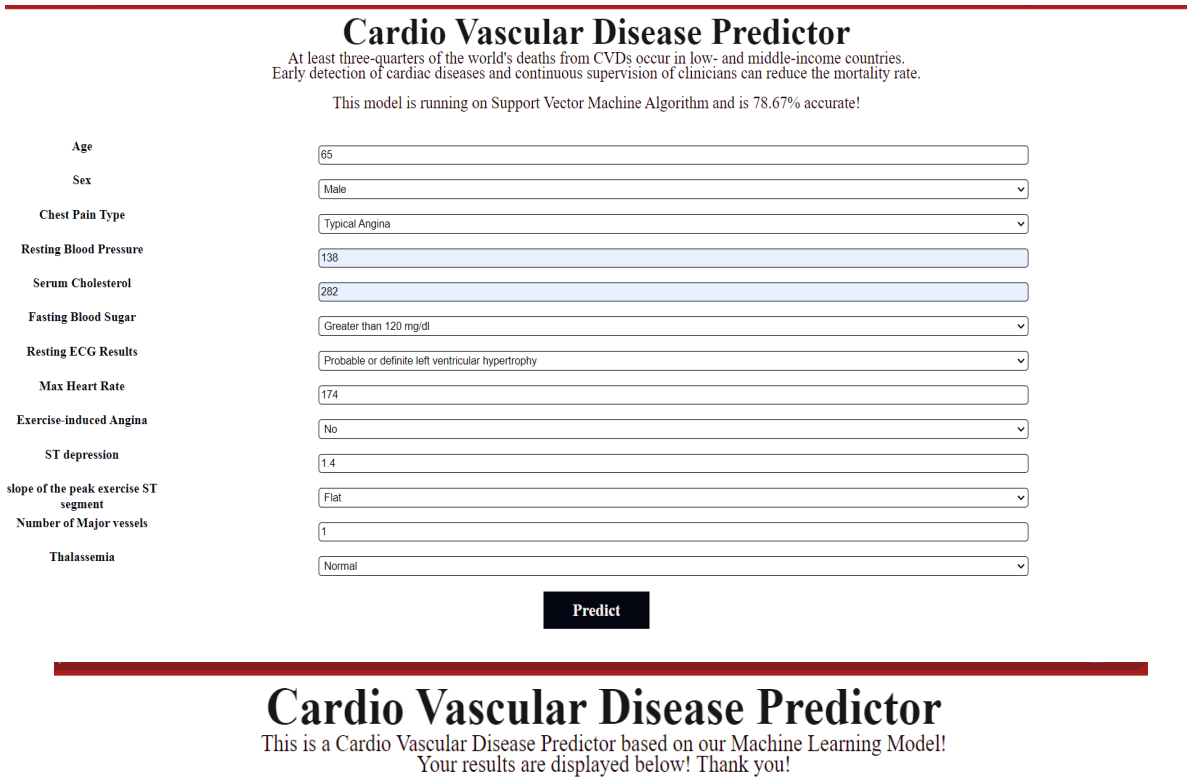
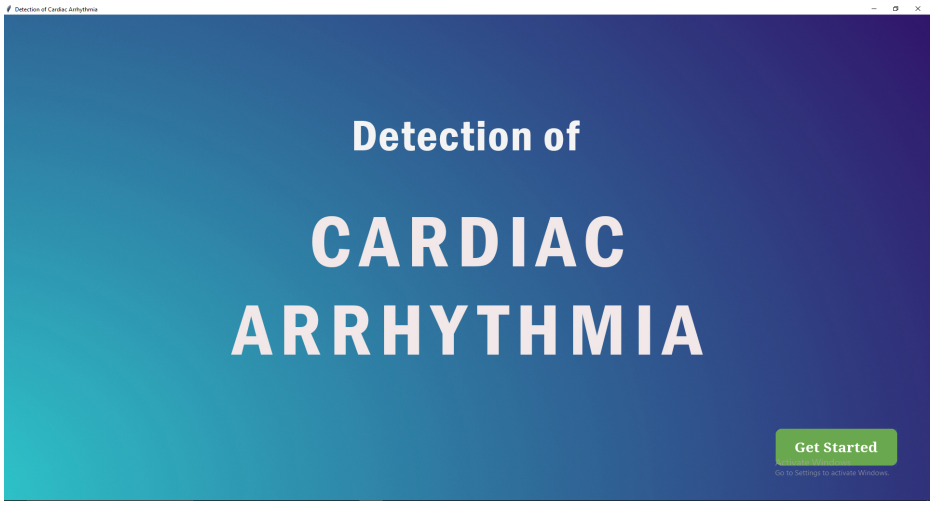


Fig 5.5 : Cardiovascular Disease Predictor for positive cases

5.1.9 Negative case GUI output of Cardiac Arrhythmia



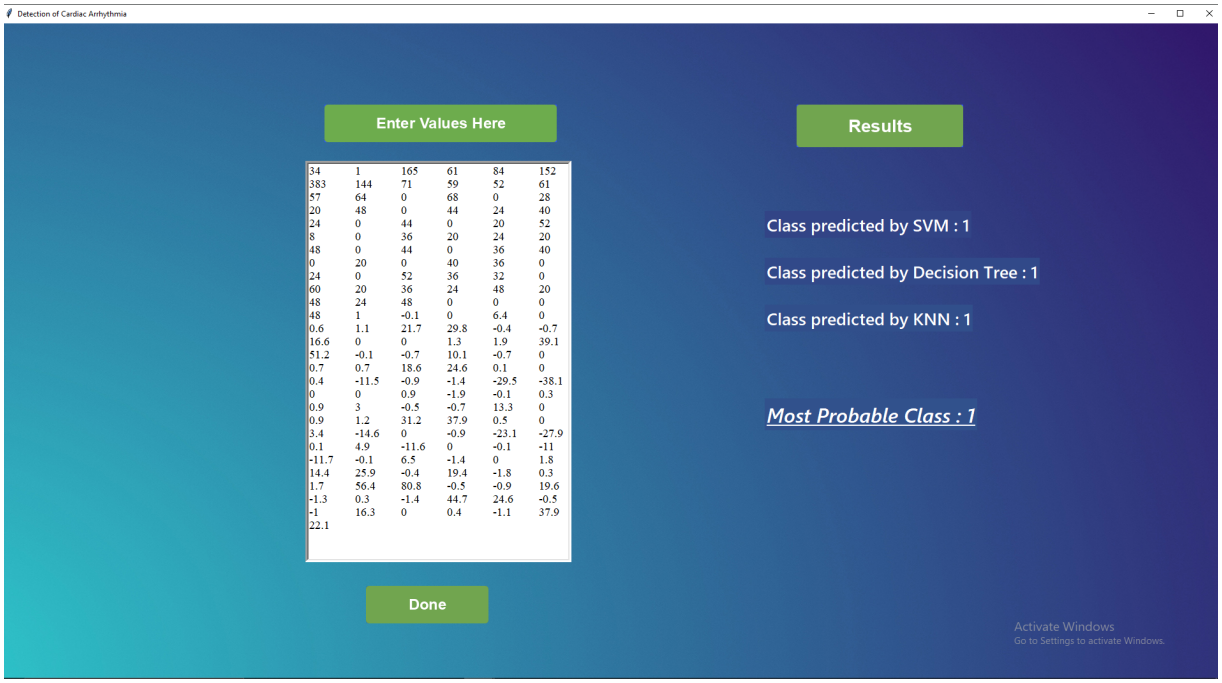


Fig 5.6 : Detection of Arrhythmia for negative case

5.1.10 Positive case GUI output of Cardiac Arrhythmia

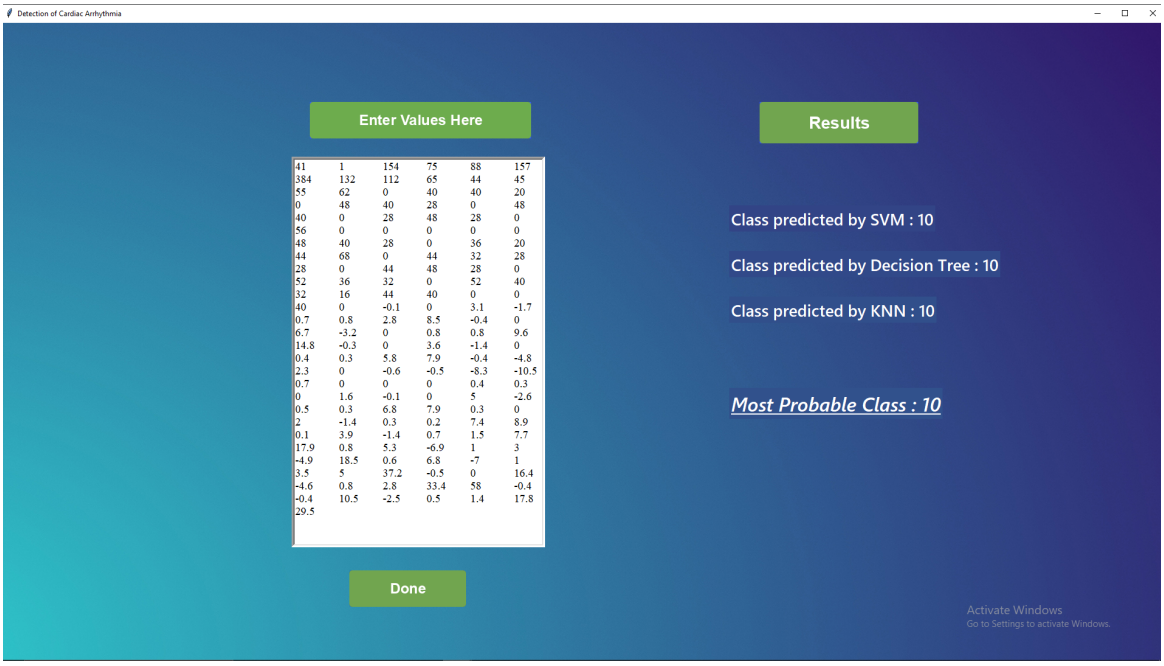


Fig 5.7 Detection of Arrhythmia for positive case

6. Final Conclusion and Future Perspectives

6.1 Conclusion

In this project we have implemented a machine-learning algorithm to classify two types of heart disease:- Cardiovascular disease(CVD) and Cardiac Arrhythmia.

Data acquisition for heart parameters is being implemented using the hardware, signal processing is used to detect peaks in ECG signals/ waveforms which are fed as inputs to the interface along with other important parameters. The machine learning model processes these inputs received to send back the results to be displayed as the output on the GUI made for it.

Using UCI datasets we have trained and tested the accuracy of the algorithms used in the machine-learning models. With an accuracy of 78.67% for predicting CVD and 74% for predicting Cardiac arrhythmia, the SVM algorithm was found to deliver the best results for both CVD and arrhythmia among many other algorithms.

6.2 Future Scope

Several countries are experiencing an increase in death rates due to various heart diseases. Therefore, prevention of disease before it can increase will reduce death rates. And, to get the best accuracy in the proposed system, we have incorporated several of the most recent machine learning algorithms. However, in addition to the approach used in this project, it can still be enhanced in a number of ways, such as by incorporating various deep learning techniques, augmenting the data set, using alternative methods for attribute selection, and using it to forecast a number of other diseases.

Especially in rural areas, healthcare monitoring and forecasting frameworks have proven indispensable in saving lives. By implementing the Internet of Things, we can improve patient care, administrative efficiency, and operational effectiveness.

7. References

[1] Akhila Naz K A, Jeena R S, and Niyas P, **"Deep Neural Network based Real Time Multi-class Arrhythmia classification in IoT cloud platform"**,2021, DOI: https://doi.org/10.21203/rs.3.rs-603991/v1

[2] Padmavathi Kora, A Rajini, M C Chinnaiah & K Meenakshi, **"IoT Based Wearable Monitoring Structure for Detecting Abnormal Heart"**, 2021 International Conference on Sustainable Energy and Future Electric Transportation (SEFET), 21-23 January 2021
DOI: 10.1109/SeFet48154.2021.9375787

[3] Y. D P and D. L. N, **"Early Detection of Cardiac Arrhythmia Disease using Machine Learning and IoT Technologies,"** 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) IEEE, 2021, pp. 1658-1661, DOI:10.1109/ICOSEC51865.2021.9591884.

[4] Shadman Nashif, Md. Rakib Raihan, Md. Rasedul Islam, Mohammad Hasan Imam**, "Heart Disease Detection by Using Machine Learning Algorithms and a Real-TimeCardiovascular Health Monitoring System"**, Department of Electrical & Electronic Engineering, American International University-Bangladesh (AIUB), Dhaka, Bangladesh, Vol.6 No.4, November 2018
DOI:10.4236/wjet.2018.64057

[5] Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H.,Schölzel, C., & Chen, S. A. (2021). **NeuroKit2: A Python toolbox for neurophysiological signal processing. Behaviour Research Methods**, 53(4), 1689–1696. https://doi.org/10.3758/s13428-020-01516-y

[6] Rahul Kher (2019) Signal Processing Techniques for Removing Noise from ECG Signals. J Biomed Eng 1: 1-9.

[7]ECG AD823 sensor datasheet:
https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf

[8] Arduino datasheet: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[9] Raspberry Pi datasheet: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

Capstone_Report_Final_S13 (1) (1).pdf

ORIGINALITY REPORT

9%	3%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Teaching and Learning with Technology Student Paper	1%
2	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	1%
3	Submitted to Swinburne University of Technology Student Paper	<1%
4	Submitted to Gitam University Student Paper	<1%
5	www.elprocus.com Internet Source	<1%
6	www.researchgate.net Internet Source	<1%
7	Submitted to SRM University Student Paper	<1%
8	Yathish D P, Dayanand Lal N. "Early Detection of Cardiac Arrhythmia Disease using Machine Learning and IoT Technologies", 2021 2nd	<1%

Implementation of IOT-Based Healthcare Kit

Sinchitha H V, Tanya Chanchalani, Bhushan Kiran Munoli, Gaurav R

Project Guide: Dr Purushotham U

MOTIVATION

The prime objective is to build up a dependable patient-checking framework with the goal that the patients, who are either hospitalized or executing their ordinary day-to-day life exercises can screen and self-examine their heart-related health parameters safely.

THEORY

- The IOT-based healthcare kit is embedded with ECG(AD8232) sensor.
- Two separate ML models are developed and trained using UCI dataset for CVD and Arrthymia.
- The input given to ML model is processed and connected to the two interfaces.

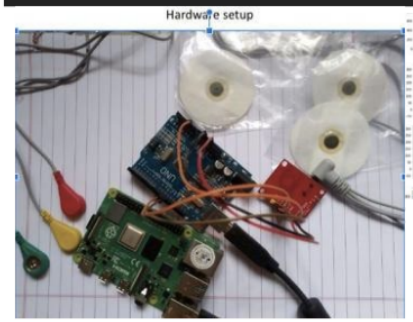
ANALYSIS

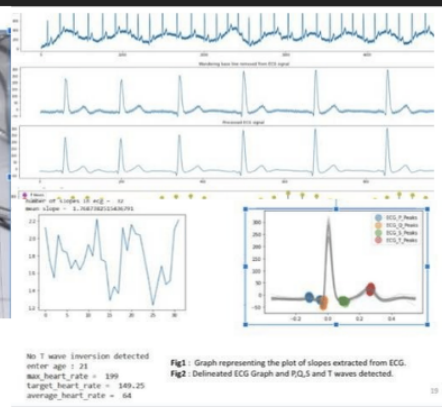
- The electrical activity of the heart is charted as an ECG graph.
- The SVM algorithm predicts the model with accuracies of 73% for Arrhythmia and 78.67% for CVD.
- The GUI shows the number of positive cases based on the input.

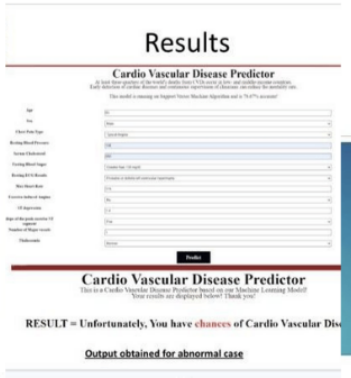
IMPLEMENTATION

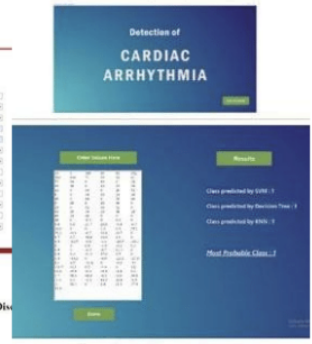
- The ECG electrode is placed on the body and enables recording of electrical currents.ECG AD8232 is used to calculate the electrical activity of the heart.
- The analog output of the sensor is fed into the Arduino Uno, It is connected to Rpi using UART.
- ML model is designed for two separate diseases(CVD and Arrhythmia)
- SVM and Logistic Regression to predict CVD.
- SVM and Random Forest Classifier to predict Cardiac Arrhythmia
- Two separate interfaces are built using framework and the output is executed on GUI.

RESULTS AND CONCLUSION









Conclusion:

- In this project we have implemented machine learning algorithm to classify two types of heart disease:- Cardiovascular disease(CVD) and Cardiac Arrhythmia.
- It was observed that SVM algorithm gave the best results for both CVD and Arrhythmia among other algorithms even when tested on the hospital dataset, and hence proved to be the best for prediction of both diseases.
- A separate independent user interface is built (using HTML, CSS, Flask framework with python) for each disease where the user provides necessary input.
- And, the obtained data is tested using the machine learning models and final results are displayed on the GUI.
- The result of CVD include 0(CVD absent) and 1 (CVD present) and Arrythmia include 0 (No arrythmia) 1-15 (different types of arrythmia) and 16 (unknown values)

Implementation of IoT-Based Healthcare Kit

Sinchitha H V
Electronics and Communication Engineering
PES University
Bengaluru, India
sinchitha18@gmail.com

Tanya Chanchalani
Electronics and Communication Engineering
PES University
Bengaluru, India
tanyachanchalani@gmail.com

Bhushan Kiran Munoli
Electronics and Communication Engineering
PES University
Bengaluru, India
bhushanmunoli01@gmail.com

Gaurav R
Electronics and Communication Engineering
PES University
Bengaluru, India
gaurav.gaurav.r53@gmail.com

Dr Purushotham U
Project Guide
PES University
Bengaluru, India
purushothamu@pes.edu

***Abstract*—Cardiovascular diseases and Cardiac Arrhythmia are the most common causes of death worldwide over the last few decades in various countries across the world. However, it is not possible to monitor patients every day in all cases accurately and consultation with a patient for 24 hours by a doctor is not available as it needs more patience, time and expertise.**

In this project we have used ECG sensors, Arduino and Raspberry Pi to obtain certain parameters on cardiovascular disease. We have developed machine learning models based on K-Nearest Neighbour, Logistic Regression, Support Vector Machine and Random Forest for heart disease prediction through the various heart attributes of patients and to detect these impending heart diseases. The datasets that are used in this project are available publicly on the UCI website.

The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications which can be a great milestone in the field of medicine

***Keywords*—CardiovascularDiseases, Cardiac Arrhythmia, ECG sensors, Arduino Uno, Raspberry Pi, K-Nearest Neighbour, Logistic Regression, Support Vector Machine, Random Forest Classifier, Decision Trees, Data augmentation.**

I. Introduction

The heart is a very prevalent organ in our body. It is a part of the cardiovascular system that also consists of blood vessels like arteries, veins and capillaries. A load of heart diseases is rapidly increasing all around the world. It is highlighted as a silent killer which leads to many deaths across the globe. The early diagnosis [15] of such diseases plays a very vital role and reduces complications. If not diagnosed in the early stages, the condition of the person will worsen and might eventually die due to reluctance. The proposed solution is to develop two machine learning models, one for each of the two diseases-cardiovascular disease and cardiac arrhythmia. Along with the development of these, three hardware components are used to obtain ECG-related data from the patient. These components are ECG AD8232 sensor, Arduino and Raspberry Pi-4 board. The models aim to predict the presence of the disease. Several algorithms [26] like Logistic Regression, K-Nearest Neighbour, Support Vector Machine, Random Forest and Decision Tree would be implemented to train the model. All of them would be tested against a test sample and their performances would be noted.

We are trying to achieve the best results for the classification and prediction of the two types of heart diseases considered by comparing various Machine Learning (ML) models based on accuracy and time and applying the one which best suits the requirement. The final part of our work involves building a graphical user interface (GUI), one for the prediction of each disease. For the prediction of CVD, ECG-related attribute values are recorded from the hardware components and inputted to the GUI. Other important features are input directly into the GUI which is built using HTML, CSS and Flask modules on Python. The GUI for the prediction of cardiac arrhythmia is built using Tkinter on Python and all attribute values are directly inputted on the GUI.

II. Literature Review

We reviewed some research papers as part of a literature survey. In Real-Time Multi-Class Arrhythmia classification in IoT-cloud platform[1] showed Data Acquisition is the mechanism of collecting raw data from the physical world. AD8232 sensor is used for the above process. Signal processing is applied to denoise and refactor. Classification performance is enhanced by building a machine learning model with Deep Neural Network(DNN) with three hidden layers. This machine learning algorithm is a subset of the Artificial Neural Network(ANN)[12], [10]. It is compared to biological neural networks.The exhibited accuracy of the model is 99.45%. DNN is built empirically with seven layers. It classifies arrhythmia into normal and abnormal classes with efficient parameters. IoT-Based Health Monitoring System by using Raspberry Pi and ECG signal[2], [17] developed a monitoring system that tracks the heart pulse of the patient. It results in fast and noninvasive monitoring of the heart.

Early Detection of Cardiac Arrhythmia Disease using Machine Learning and IoT Technologies [3] implemented a hardware model for early diagnosis of cardiac problems. Many important features such as the location of chambers,

heartbeats(BPM) and the presence of damage in heart muscles are detected in electrocardiograms. The first operation is ECG accretion, three electrodes with the patches are placed in specific locations on the body. The patches of the ECG are connected to the limbs to extract electrical impulses. A QRS detection algorithm is applied to detect Q-R-S peaks by using filters namely BPF, HPF and LPF. These filters reduce noise in the signal.SVM classifier is used. The results of the classifier are compared and tested for different factors. Cardiac Arrhythmia is classified by the classifier. IoT Based on a Wearable Monitoring structure for detecting Abnormal Heart [4] This paper explains a portable ECG device is made to find abnormal heart conditions and to analyze the data prior to diagnosis. The architecture is a full vertical system with all the advantages following the latest technology. The approaches include [29] devices such as ECG electrodes, Bluetooth modules and IoT modules. This device is connected to the cloud. This approach informs the family and doctor in charge of any abnormalities that are detected.

A python toolbox for neurophysiological signal processing, Behaviour research methods [5]. This paper explains the implementation of Neurokit2. It is a user-friendly package and provides access to advanced biosignal processing. It includes libraries of all levels of expertise. Processing of signals like EEG, HRV, and PPG is done by using the above libraries. Signal Processing Techniques for removing Noise from ECG signals[6]. It focuses on removing noises from the signal. An Analog filter(Butterworth filter) is considered to do the above filtering task. This type of filter produces sequences with no ripple count.

III. METHODOLOGY USED

A. *Mainly used data and dataset*

For our analysis, we used a dataset from the UCI(University of Irvine) to classify and predict two types of heart diseases[9]. The arrhythmia UCI dataset differentiates between the presence and absence of cardiac arrhythmia and is classified into 16 groups. The cardiovascular UCI dataset consists of 14 important attributes. Both UCI datasets are multivariate. There are no missing values in the Cardiovascular UCI dataset but the dataset for Cardiac Arrhythmia has 376 missing values. The features from the datasets are extracted and used in machine learning models to train and test them[34].

B. *Hardware Implementation*

1. Boards used:

Raspberry Pi 4 Model B, ECG AD8232 and Arduino Uno.

2. ECG(Electrocardiogram) Signal Collection:

The ECG electrodes having hydrogel are placed in the form of Einthoven's triangle[31]. The red electrode is [19] on the right arm, the yellow electrode on the left arm and the green electrode on the right leg. Lead II is extracted from the ECG sensor. A change in the arrangement of the electrodes can lead to a different lead ECG signal according to Einthoven's triangle[31].

3. Role of Arduino:

As the sensor is attached to the Arduino, the signal collected is passed through the inbuilt filters in the sensors and a final single lead ECG is passed to the Arduino[11]. The Arduino helps to collect accurate signals even if the signal is in millivolts.

4. Communication between Arduino and Rpi:

With the help of the UART(Universal Asynchronous Receiver-Transmitter) protocol, the data collected by the Arduino is sent to the Rpi. Arduino uses UART for serial communication. A USB port of Rpi[23] is configured to receive UART packets. The baud rate is set. In this project, the baud rate is set to a default of 9600 bps. The communication is carried on by running a program on Rpi, to receive the signal from the configured port.

5. Role of Rpi:

Rpi is used for signal processing and ECG parameter detection[32].

6. Noise in the ECG signal:

Usually when ECG readings are taken there are a few basic noise signals which are inevitable. Some of the noise includes baseline wandering caused by the breathing of a patient at a frequency of 0.5Hz to 2Hz. The power line interference with a frequency of 50Hz to 60Hz. Noise is caused by the electric activities in the muscles which are usually above 100Hz. Noise caused by the movement of the electrodes during feature extraction is of a frequency range of 1Hz to 10 Hz[7]. These are the basic noise present in ECG

7. Removal of noise in the ECG signal:

The baseline wandering is removed by using a high pass Butterworth filter of cutoff frequency 2Hz[6]. The power line interference is removed by using a low pass Butterworth filter of cutoff frequency 50Hz. In this project, the noise produced due to electrode movement was not removed as extra care was taken to not produce noise of this kind.

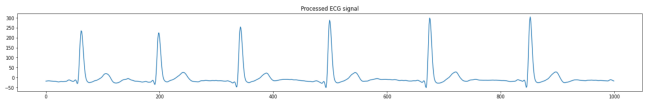


Fig. 1. ECG filtered signal

The y axis represents the magnitude of the ECG signal and the x axis represents the count of the signal.

8. Peak detection:

The P, Q, R, S and T peaks are detected using the python library neurokit2. This returns an array of x coordinates of the peaks. This can be further used to detect different parameters

9. ST slope detection:

2 methods to detect the ST slope are implemented. The first is by using the slope formula. Here the x and y coordinates of S and T peaks are determined and the slope formula was used. This method is very inaccurate as the ST segment is usually not a straight line. The second method is by using differentiation. The array of x and y coordinates of the complete ST considered, first order difference of both the x and y are calculated, as the lists are NumPy arrays, they were divided and the mean of the array was considered as the slope of the ST slope. Our ECG signal has 32 ST segments. This resulted in finding 32 separate slopes. To narrow it down to only one value, the mean of all the slopes was considered. The normal range of the ST slope is 0 to 0.5. Slight variations are entertained. Further, [28] efficiency can be increased by detecting the J point in ECG and considering the ST slope from the J point to the T point.

10. Max heart rate during sample collection:

The count of the total number of R peaks in the sample is obtained. The sample is collected for 30 seconds. So to find BPM the count of R peaks is doubled. The R peaks are considered that represent the major contraction of the heart muscles to pump blood throughout the body.

11. T inversion:

The baseline in the ECG signal [18] can have a variation of 0.1mV. If the amplitude of the T peak is less than -0.1mV then there is a possible T inversion in the ECG signal. T inversion is equal to 1, and is detected as positive, or else it is 0.

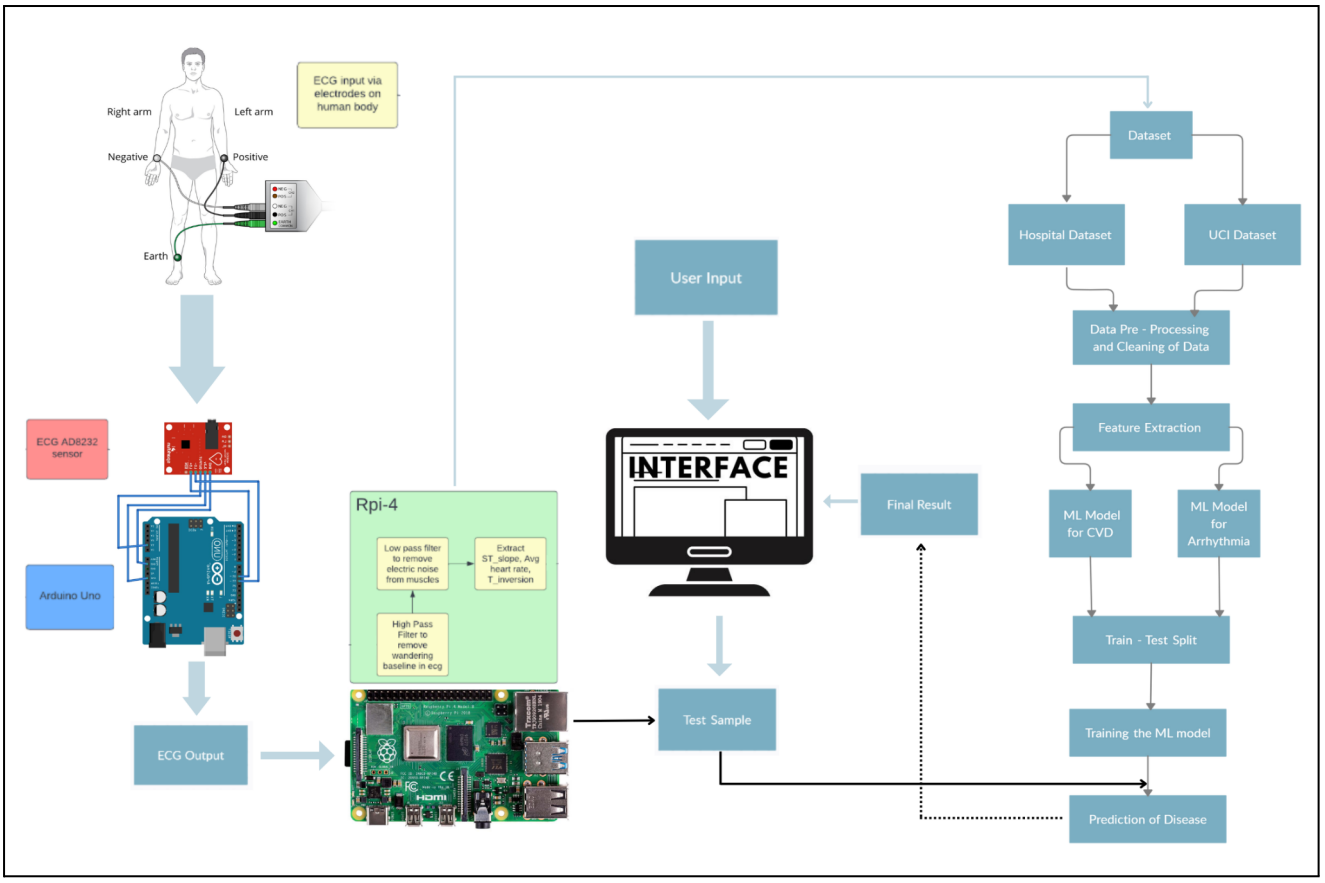


Fig. 2. Block diagram of project

C. *Feature Extraction from ECG*

There are mainly 3 parameters extracted from ECG. The slope of ST segment, heart rate and presence or absence of T inversion. The ST segment is the connecting segment between S and T peaks in ECG. The slope of this line is obtained in 2 ways. One is using the slope formula. This method is not efficient as the segment is not a straight line. To reduce the error, the 1st discrete difference of coordinate values of the ST segment are considered.

To find the T inversion, the magnitude of the T peaks are checked if they are less than -0.1mV. If yes 1 is returned, else 0 is returned. To find the heart rate, the number of R peaks are calculated and doubled as the signal is collected for a time duration of 30 seconds. Later these 3 parameters are stored in the .csv file with the age and patient ID.

These three parameters obtained from the hardware are Maximum heart rate at the time of collection, ST peak and T inversion which would be sent as inputs to the interface for CVD(Cardiovascular Disease) along with other important parameters.

D. *Implementation of Machine Learning Models*

1. *Feature Selection*:

The UCI Dataset for CVD prediction consists of multivariate data where attributes are categorical, integer and real. The considered dataset contains 303 instances and 75 attributes[22][24]. The dataset is trained and tested with a train-test split ratio of 75:25(75% training dataset and 25% testing dataset) which makes the dimensions 222 x 75 records. The instances and attributes size is 297 x 14. Hence, the number of attributes considered is 14, amongst the entire number of attributes which is 75. The amount of positive cases present in the dataset is 137 and that of negative cases found in the dataset is 160. The results are displayed as two values : 0 and 1. If the result obtained is 0, it indicates no disease and if the result obtained is 1, it indicates the disease is present. Among these, a subset of fourteen attributes is mostly included in published experiments for CVD, which are: age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, num or the predicted attribute.

The UCI Dataset for the prediction of Cardiac Arrhythmia consists of multivariate data where attributes are categorical, integer and real. The dataset has 452 instances and 279 attributes out of which 206 attributes are valued linearly and the remaining are nominal. Hence, the total number of positive cases observed was 207, and for negative cases was 245. The train test split of this dataset was selected to be 70:30 dividing it into 316 and 136 records. Therefore the output includes 16 classes based on the value of attributes. Class 1 indicates Normal cases or no presence of this disease, and classes 2 to 15 of them indicate various types of Arrhythmia. Lastly, class 16 indicates the unknowns due to a lot of missing values in the dataset. Hence, the fifteen attributes there were selected in order to help predict this disease were: Age, Sex, Height, QRS duration, P-R interval, T interval, P interval, Vector angles in degrees on the front plane of QRS, T, P, QRST and J, Heart rate.

Further, two separate machine learning models were developed and trained using the UCI datasets repository for CVD and Cardiac Arrhythmia. The data required is cumulated, cleaned and processed. This is done with the aim of extracting features according to the parameters necessary for the two heart diseases selected. In this approach, multiple ML algorithms are implemented in order to increase efficiency and get better accuracy.

2. *Algorithms:*

Some algorithms used for the prediction of

CVD are Logistic Regression, K-Nearest Neighbour, Support vector machine, Random Forest and Decision Tree Classifier[21][30]. The intention of using logistic regression is that it works better on classification problems and along with the classification results, it produces probabilities that are accurately calibrated. KNN was used as it is simple and robust to noisy data. It is highly effective in cases of exceptionally large datasets. The SVM classifier was used as the UCI dataset's two classes are clearly separated by a large margin. This makes the usage of SVM a good decision. Added advantages of SVM are that it has efficient memory usage, higher speed and improved performance compared to other neural network models. A decision tree classifier [14] was employed as it uses a variety of feature subsets and decision rules at different stages of classification. Data is divided into chunks which have better interpretability and work faster with categorical/numerical values thus reducing the computation power and minimizing the implementation time required to compute the final results[33]. To further enhance the results Random Forest Classifier was applied to have more randomness to the model while growing trees.

Two algorithms were used for the prediction of Cardiac Arrhythmia - SVM with linear, RBF(radial basis function), polynomial kernels and Linear SVM[8]. For feature selection, Principal Component Analysis is applied. This is followed by SVM is used to predict the data with very high dimensions which are later combined with a Random Forest Classifier to enhance the accuracy, generalization and comprehensibility[20]. The next step involves decision trees. The tree does not consider all of the attributes, confirming that separate decision trees are not connected. This makes the classifier less prone to overfitting. Kernels are employed to enhance the SVM classifier output[35]. The kernel functions enable a non-linear decision surface to be converted into a linear equation in higher dimensional spaces by transforming the training dataset. A linear kernel is used when data is linearly separable. A polynomial kernel is used when data is separated by a boundary that is of some defined arbitrary order. A radial basis kernel is used to solve the space complexity issues when trying to find a non-linear classifier or regression line. Data preprocessing is done first by deleting columns that have more than 40% missing values. These missing values are later imputed by substituting the attribute median and then the attributes are normalized. For attributes having higher deviation, it is recommended to use the median instead of the mean of the missing values. For feature selection using PCA(Principal Component Analysis), according to Kaiser's rule components with eigenvalues, greater than or equal to 1 are selected.

After training the models, in order to test them, data is manually entered into the interface according to the parameters specified for each disease. The inputs are processed in the ML models connected to the two interfaces. And the final outputs are displayed through the GUI built for both diseases.

3. *Building Graphical User Interface:*

The interface for CVD is built mainly using Flask which is a microframework written in Python, along with HTML to provide page structure, and CSS to provide page layout. A few of the other required modules are Jinja2 as it enables the clean addition of loops and variables into the Html document, other than that Gunicorn and MarkupSafe have to be installed. Then the machine learning model that is connected is run in order to obtain the .pkl(pickle) file of the required algorithm, indicating that the model has finished being trained and is ready for testing. The command to start the server has to be executed for the web page to run on the web browser with all input fields[25]. At last, the user provides the necessary details into the input fields, and the final results are calculated using the machine learning model by clicking the predict button, output is displayed for the user on the second page.

The interface for the prediction of Cardiac Arrhythmia is built using a Python module Tkinter. GUI(Graphical user interface) application development is made quick, simple, and dependable with Tkinter. Custom-made buttons are incorporated into the Tkinter module to facilitate the functioning of the GUI. The home page, which is the initial page, has a title and a button to go to another page. Then the user enters data from the UCI dataset in a text box on the second page for the prediction of Cardiac Arrhythmia. The input data is tested with SVM, Decision Tree and KNN which output the class of Arrhythmia. The final part is displaying the most probable class of arrhythmia from the obtained results.

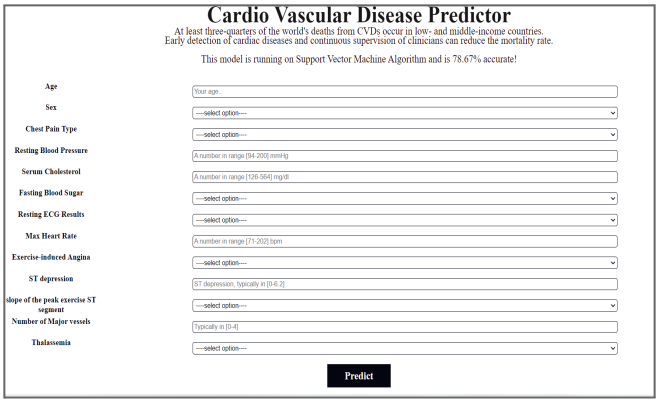


Fig. 2. GUI to detect cardiovascular disease

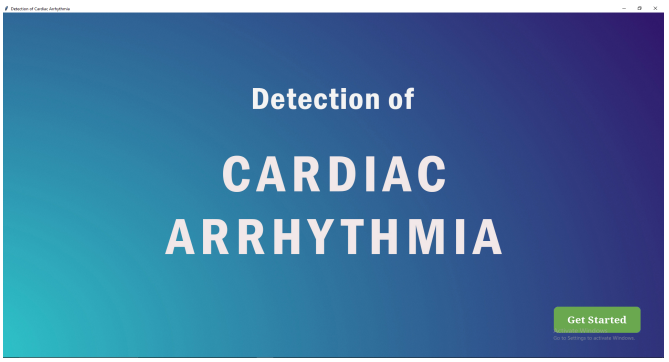


Fig. 3. Home page of GUI to detect cardiac arrhythmia

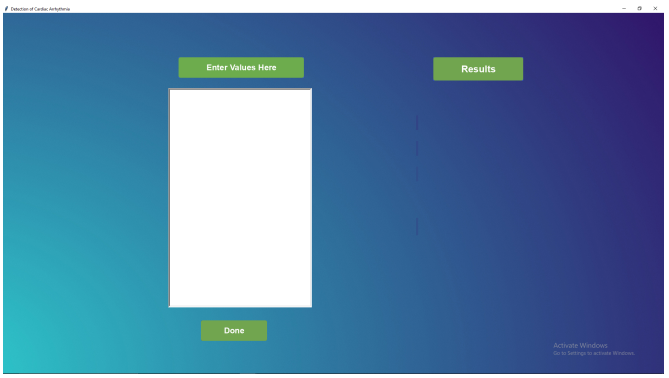


Fig. 4. Second page with text field and results

IV. RESULTS

The proposed intelligent healthcare kit is being deployed and tested. The machine learning model performed up to the expectation and achieved the highest accuracy. The main operation that was performed is from an ECG sensor[13], it measures electrical activity and records the electrical signals in the form of graphs. The main discussion of the above project is to Classify the two types of heart disease(CVD and Cardiac Arrhythmia). The secondary operation [27] is to detect the risk of disease by creating an application interface. The output of the same is executed on the GUI platform.

The output value is obtained in binary format in which 0 indicates no disease and 1 indicates the person has a chance of having CVD. After testing the model with multiple machine learning algorithms it was noted that the Support Vector Classifier showed the highest accuracy of 78.67%.

For cardiac Arrhythmia the output included 16 classes based on the value of attributes. Initially, PCA was applied for feature extraction, followed by SVM with various kernel functions, and to further enhance the results random forest classifier was applied, but it was observed that SVM with Linear Kernel and regularization parameter of 0.01 was 74% accurate amongst the other kernels used for predicting Cardiac Arrhythmia].

TABLE I. RESULTS OF CVD PREDICTION

Sl.No.	Model	Accuracy
1	Random Forest Classifier	77.33%
2	Decision Tree	70.67%
3	Linear SVM	78.67%
4	Logistic Regression	76.0%
5	K-nearest Neighbors	74.67%

TABLE II. RESULTS OF CARDIAC ARRHYTHMIA PREDICTION

Sl.No.	Model	Accuracy
1	SVM with Random Forest (Linear, RBF, Polynomial Kernel)	73%, 66%, 68%
2	Linear SVM	74%

V. CONCLUSION AND FUTURE WORK

In this project we have implemented machine learning algorithms to classify CVD and Cardiac Arrhythmia. Data acquisition for heart parameters is being implemented using the hardware, signal processing is used to detect peaks in ECG signals/ waveforms which are fed as inputs to the interface along with other important parameters. The machine learning model processes these inputs received to send back the results to be displayed as the output on the GUI made for it.

Using UCI datasets available for CVD and Cardiac Arrhythmia, we have trained and tested the accuracy of the algorithms used in the machine-learning models. It was observed that the SVM algorithm gave the best results for both CVD and Arrhythmia among other algorithms with an accuracy of 78.67% and 74% respectively.

Several countries are experiencing an increase in death rates due to various heart diseases. Therefore, prevention of disease before it can increase will reduce death rates. And, to get the best accuracy in the proposed system, we have incorporated several of the most recent machine learning algorithms. However, in addition to the approach used in this project, it can still be enhanced in several ways, such as by incorporating various deep learning techniques, augmenting the data set, using alternative methods for attribute selection, and using it to forecast several other diseases. Especially in [16] rural areas, healthcare monitoring and forecasting frameworks have proven indispensable in saving lives. By implementing the Internet of Things, we can improve patient care, administrative efficiency, and operational effectiveness.

ACKNOWLEDGMENT

We would like to express our gratitude to Dr Purushotham U, Department of Electronics and Communication Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this Capstone Project. We take this opportunity to thank Dr Anuradha M, Chairperson, Department of Electronics and Communication Engineering, PES University, for all the knowledge and support we have received from the department.

We are deeply grateful to Dr M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro-Chancellor, PES University, Dr Suryaprasad J, Vice-Chancellor, PES University for providing various opportunities and enlightenment to every step of the way.

Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends.

REFERENCES

[1] K. Abdulnazar, A. Naz, R S Jeena and P. Niyas, “Deep Neural Network Based Real Time Multi-Class Arrhythmia Classification in IoT-Cloud Platform,” research square, v1, 2021, in press.

[2] PP. Kora, A. Rajani, M. C. Chinnaiah, K. Swaraja and K. Meenakshi, "IoT Based Wearable Monitoring structure for detecting Abnormal Heart," International Conference on Sustainable Energy and Future Electric Transportation (SEFET), 2021, pp, in press.

[3] Y. D P and D. L. N, "Early Detection of Cardiac Arrhythmia Disease using Machine Learning and IoT Technologies," 2nd International Conference on Smart Electronics and Communication (ICOSEC) IEEE, 2021, pp. 1658-1661, in press.

[4] S. Nashif, R. Raihan, M. R. Islam, M. H. Imam, “Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System,” World Journal of Engineering and Technology, 06. 854-873, 2018, in press.

[5] D. Makowski, T. Pham, Z. J. Lau, J. C. Brammer, F. Lespinasse., H. Pham, C. Schölzel, and S. H. Annabel Chen, “NeuroKit2: A Python toolbox for neurophysiological signal processing,” Behaviour Research Methods, 53(4), 1689–1696, 2021, in press.

[6] R. Kher, “Signal Processing Techniques for Removing Noise from ECG Signals,” J Biomed Eng 1: 1-9, 2019, in press.

[7] Q. He, A. Man and A. Elchouemi, "Heart disease monitoring and predicting by using machine learning based on IoT technology," 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), 2020, pp. 1-10, in press.

[8] Sahoo, Sudarsan & Borthakur, Priyanuj & Baruah, Niharika & Chutia, Bhaskar, “IoT and Machine Learning Based Health Monitoring and Heart Attack Prediction System,” Journal of Physics: Conference Series. 1950. 012056, 2021, in press.

[9] V. Sharma, S. Yadav and M. Gupta, "Heart Disease Prediction using Machine Learning Techniques," 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 177-181, in press.

[10] M. Gudadhe, K. Wankhade and S. Dongre, "Decision support system for heart disease based on support vector machine and Artificial Neural Network," International Conference on Computer and Communication Technology (ICCCT), 2010, pp. 741-745, in press.

[11] N. L. Fitriyani, M. Syafrudin, G. Alfian and J. Rhee, "HDPM: An Effective Heart Disease Prediction Model for a Clinical Decision Support System," in IEEE Access, vol. 8, pp. 133034-133050, 2020, in press.

[12] H. Yan, J. Zheng, Y. Jiang, C. Peng and Q. Li, "Development of a decision support system for heart disease diagnosis using multilayer perceptron," IEEE International Symposium on Circuits and Systems (ISCAS), 2003, pp. V-V, in press.

[13] Bui Ngoc, Thang & Duc Tri, Phan & Nguyen, Thanh Phuoc & Hoang, Giang & Choi, Jaeyeop & Bui, Quoc & Vo, Hung & Oh, Junghwan, “Real-Time Filtering and ECG Signal Processing Based on Dual-Core Digital Signal Controller System,” IEEE Sensors Journal, 2020, pp. 1-1, in press.

[14] S. K. J. and G. S., "Prediction of Heart Disease Using Machine Learning Algorithms," 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019, pp. 1-5, in press.

[15] M. A. Jabbar, B. L. Deekshatulu and P. Chandra, "Alternating decision trees for early diagnosis of heart disease," International Conference on Circuits, Communication, Control and Computing, 2014, pp. 322-328, in press.

[16] S. Harjai and S. K. Khatri, "An Intelligent Clinical Decision Support System Based on Artificial Neural Network for Early Diagnosis of Cardiovascular Diseases in Rural Areas," 2019 Amity International Conference on Artificial Intelligence (AICAI), 2019, pp. 729-736, in press.

[17] G Sowmya, “IOT Based Health Monitoring System,” International Journal for Research in Applied Science and Engineering Technology, 2021, 9. 1176-1178, in press.

[18] A. Walinjkar and J. Woods, "ECG classification and prognostic approach towards personalized healthcare," International Conference On Social Media, Wearable And Web Analytics (Social Media), 2017, pp. 1-8, in press.

[19] R. Ö. Doğan and T. Kayıkçıoğlu, "Remote patient monitoring and Electronic Health Record system based on web services," 24th Signal Processing and Communication Application Conference (SIU), 2016, pp. 1785-1788, in press.

[20] J. Singh, A. Kamra and H. Singh, "Prediction of heart diseases using associative classification," 5th International Conference on Wireless Networks and Embedded Systems (WECON), 2016, pp. 1-7, in press.

[21] S. Dhar, K. Roy, T. Dey, P. Datta and A. Biswas, "A Hybrid Machine Learning Approach for Prediction of Heart Diseases," 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018, pp. 1-6, in press.

[22] S. Islam, N. Jahan and M. E. Khatun, "Cardiovascular Disease Forecast using Machine Learning Paradigms," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 487-490, in press.

[23] AA. Nancy, D. Ravindran, PMDR. Vincent, K. Srinivasan, DG. Reina, “IoT-Cloud-Based Smart Healthcare Monitoring System for Heart Disease Prediction via Deep Learning,” Electronics 2022, 11, 2292, in press.

[24] BU. Rindhe, R. Patil, N. Ahire, S. Gagare, M. Darade, “Heart Disease Prediction Using Machine Learning,” International Journal of Advanced Research in Science, Communication and Technology, 267-276, 2021, in press.

[25] M. Pal, S. Parija, G. Panda, K. Dhama, RK. Mohapatra, “Risk prediction of cardiovascular disease using machine learning classifiers,” Open Med (Wars), 2022, 17(1):1100-1113. PMID: 35799599; PMCID: PMC9206502, in press.

[26] M. J. Gaikwad, P. S. Asole and L. S. Bitla, "Effective Study of Machine Learning Algorithms for Heart Disease Prediction," 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), 2022, pp. 1-6, in press.

[27] L. Jayasree and D. Usha, “A Comparison Analysis of Machine Learning Algorithms on Cardiovascular Disease Prediction”, International Journal on Future Revolution in Computer Science and Communication Engineering(IJFRCSCE), vol. 8, no. 3, pp. 14–22, Sep. 2022, in press.

[28] J. H. Lim, Y. W. Hau, H. T. Yew and S. B. Dass, "ASIC Layout Design-Space Exploration of Pan-and-Tompkins Pre-Processing Algorithm for High Efficiency Electrocardiogram Monitor," IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), 2020, pp. 1-6, in press.

[29] S. K. Jain and B. Bhaumik, "An Energy Efficient ECG Signal Processor Detecting Cardiovascular Diseases on Smartphone," in IEEE Transactions on Biomedical Circuits and Systems, vol. 11, no. 2, pp. 314-323, 2017, in press.

[30] KVV. Reddy, I. Elamvazuthi, AA. Aziz, S. Paramasivam, HN. Chua and S. Pranavanand, "Prediction of Heart Disease Risk Using Machine Learning with Correlation-based Feature Selection and Optimization Techniques," 7th International Conference on Signal Processing and Communication (ICSC), 2021, pp. 228-233, in press.

[31] GD. Gargiulo, P. Bifulco, M. Cesarelli, A. Mcewan, H. Moeinzadeh, AJC. O'Loughlin et al, “On the Einthoven Triangle: A Critical Analysis of the Single Rotating Dipole Hypothesis,” Sensors. 18. 2353, in press.

[32] S. K. K. L, N. K. G and M. J. A, "Coronary Artery Disease Prediction using Data Mining Techniques," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 693-697, in press.

[33] R. Katarya and P. Srinivas, "Predicting Heart Disease at Early Stages using Machine Learning: A Survey," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 302-305, in press.

[34] SAA. Shah, AH. Saleh, M. Ebrahimian and R. Kashef, "Early Detection of Heart Disease Using Advances of Machine Learning for Large-Scale Patient Datasets," IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2022, pp. 274-280, in press.

[35] M. Nahiduzzaman, MJ. Nayeem, MT. Ahmed and MSU. Zaman, "Prediction of Heart Disease Using Multi-Layer Perceptron Neural Network and Support Vector Machine," 2019 4th International Conference on Electrical Information and Communication Technology (EICT), 2019, pp. 1-6, in press.