Automatic Classified Myocardial Infarction Detection Using Machine Learning and Forewarning System with Location of the Patient Using GSM Module

By

Soptorsi Paul Shrestha
17221001
Md. Hasnatul Amin
18121107
MD. Amir Faisal
17221008
Syed Md. Jawadul Alam
18121064

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering

Electrical and Electronic Engineering
BRAC University
September, 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Soptorsi Paul Shrestha	Md. Hasnatul Amin
Soptorsi Paul Shrestha 17221001	**Md. Hasnatul Amin** 18121107
MD. Amir Faisal	Syed Md. Jawadul Alam
MD. Amir Faisal 17221008	**Syed Md. Jawadul Alam** 18121064

Approval

The thesis/project titled "Automatic Classified Myocardial Infarction Detection Using Machine Learning and Forewarning System with Location of the Patient Using GSM Module" submitted by

1. Soptorsi Paul Shrestha (17221001)
2. Md. Hasnatul Amin (18121107)
3. MD. Amir Faisal (17221008)
4. Syed Md. Jawadul Alam (18121064)

of Summer 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 3rd October, 2021.

Examining Committee:

Supervisor:
(Member)

____

Dr. A. K. M. Abdul Malek Azad
Professor
Department of Electrical and Electronic Engineering BRAC University, Dhaka

Program Coordinator:
(Member)

____

Abu S.M. Mohsin, PhD
Assistant Professor
Department of Electrical and Electronic Engineering, BRAC University, Dhaka

Departmental Head:
(Chair)

____

Md. Mosaddequr Rahman, PhD
Professor and Chairperson
Department of Electrical and Electronic Engineering, BRAC University, Dhaka

Abstract

Myocardial Infarction (MI) is a crucially leading reasons of huge mortality and modality all over the world. The prior reasons for most of the demise are delayed diagnosis and disrupted medical attention. Our endeavor objectifies developing a portable system to detect MI instantly and incorporating a forewarning system using GSM and GPS module. The paper is focusing on the warning system delivering text messages containing classified MI information. Initially, the dataset has been preprocessed using Standard Scalar method and the preprocessed data has been trained and tested using K-Nearest Neighbor (KNN), Random Forest (RF), Support Vector Machine (SVM) and Naive Bayes (NB) to distinguish the MI affected ECG from normal ECG signal. The aim of this project is to avail immediate attention to a MI affected patient to ensure medical deliberation rapidly. Proper activation of the system will minimize the deadly effect of MI and hence reduce the mortality rate due to MI.

Keywords: Myocardial Infarction; GSM; GPS; Text Message; Standard Scalar Method; K-Nearest Neighbor; Random Forest; Support Vector Machine; Naive Bayes; ECG; Mortality Rate.

Dedication

This work is dedicated to our parents for all their constant support and motivation. We are grateful for their valuable effort on us. We also dedicate the work to our supervisor who has been leading us to pave the way to deliver a meaningful outcome.

Acknowledgement

We would like to express our gratitude to Dr. A K M Abdul Malek Azad, Professor, BRAC University, Department of Electrical and Electronic Engineering (EEE), for his assistance, guidance, and feedback for the completion of our thesis. We would also like to acknowledge the Control & Applications Research Centre (CARC) of BRAC University and research assistants associated with CARC for providing the continuous support needed to accomplish this project with their valuable inputs.

Table of Contents

List of Tables

List of Figures

List of Acronyms

MI	Myocardial Infarction
STEMI	ST-elevated Myocardial Infarction
NSTEMI	Non ST-elevated Myocardial Infarction
KNN	K-Nearest Neighbor
RF	Random Forest
NB	Naïve Bayes
SVM	Support Vector Machine
SMS	Short Message Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
SPST	Single Pole Single Throw
GPU	Graphics Processing Unit
IDE	Integrated Development Unit
HDMI	High Definition Multimedia Interface
SD	Secure Digital

Chapter 1

Introduction

Myocardial Infarction which is familiar to the world as the term heart attack causes enormous amount of demises throughout the world every year. In an enormous amount of cases this particular physical disruption leaves the patient on the way to death without even approaching any medical attention. To be precise, the failure of approaching medical attention often involves the delay in diagnosing Myocardial Infarction and hence causes further distress. Therefore, we have built the prototype in order to detect Myocardial Infarction and an alert system that will provide the instant location of the affected patient.

1.1 Motivation and Background Study

Death and acute morbidities caused by Myocardial Infarction are very much likely to occur during the primary hours of affection from the chest pain onset. Therefore, immediate diagnosis and rapid medical attention lessen the mortality and morbidity rate caused by Myocardial Infarction [1]. In most cases, the symptoms of Myocardial Infarction are pressure or tightness in the chest, breathing shortness, anxiety, sweating, vomiting, nausea, dizziness, increasing heartbeat, chest pain, pain in back, jaw and upper body areas or in some cases symptoms of basic flu [2]. The alarming point is there is a certain percentage of patients who may not show any symptoms at all and that may lead to an acute Myocardial Infarction as well. Apparently, the preliminary diagnosis of Myocardial Infarction involves a thorough Electrocardiogram (ECG) test defining a projection of the injured area in the heart and particularly displays the change in ECG signal. Since Myocardial Infarction involves the variation in ST segment of the ECG signal, the most efficient way to detect the intensity of cardiovascular situation is to execute an ECG test immediately after the projection of related symptoms. Generally, ECG signal tests can detect Myocardial Infarction with the basic efficiency from 95% to 97% and for extremely sensitive cases up to 30% [1]. The urgency of the ECG test and the necessity of immediate action have motivated us to implement a portable ECG diagnosis system and develop a forewarning method in order to minimize the deadly effect of Myocardial infarction.

This research is focused on analyzing Anterior and Inferior Myocardial Infarction and the detection procedure. Anterior MI is one of the common crucial heart diseases which ends up causing severe losses to humankind [3]. Multiple health issues are responsible for causing anterior MI which includes atherosclerotic plaque rupture as one of the main reasons. Eventually, a formation of thrombosis occurs which makes this heart situation even more severe and generally results in Anterior MI [4]. On the other hand, coronary artery occlusion causes Inferior MI because it affects the reducing perfusion in the inferior wall. As a result, this reducing perfusion may generate ischemia and finally infarction if it is not medically attended quickly [5]. Therefore, the time management while detecting myocardial infarction and immediate medical attention is extremely necessary to lessen the severity caused by Myocardial Infarction.

1.2 Literature Review

Multidimensional novel approaches have been applied to detect Myocardial Infarction. Among them some models are trained to implement on hospitalized patients and some are trained to be a portable system to diagnose MI remotely. A novel algorithm using Convolutional Neural network (CNN) algorithm has been proposed by utilizing multilead ECG in order to diagnose MI automatically. To run this model, raw ECG datasets have been used from PTB diagnostic database and to preprocess this raw multilead ECG signal beat segmentation and Fuzzy Information Granulation (FIG) have been applied. A real time micro embedded system has been utilized to execute the process from beat segmentation to diagnose MI using multi lead

CNN [6]. In [7], a diagnostic model was proposed to implement an automatic and ECG signal classification module. A combination of Recursive feature eliminator (REF), and K-Nearest Neighbors (KNN) and Artificial Neural Network (ANN) have been programmed to classify the ECG signal to operate the proposed diagnostic application model. The final results displayed accuracy level up to 80.60%, sensitivity level up to 86.58% and specificity level up to 64.71%. In another study [8], to distinguish and classify the ECG signal of a healthy person and MI affected person, Artificial Neural Network (ANN) and Support Vector Machine (SVM) have been applied. A projection of Pan Tompkins Algorithm has been done to preprocess the dataset collected from PTB diagnostic database. Back propagation with hidden layers and nodes are implemented to analyze the performance of the system. Another study [9] shows the application of harmonic phase distribution pattern on ECG signal data to determine MI. The changes in ECG waveform due to MI are projected in the phase distribution pattern of the Fourier Harmonics. The classification of ECG data for healthy and people affected by MI has been done by applying a threshold based classification rule and logistic regression and eventually shows an overall level of detection up to 95.6%, sensitivity of 96.5% and specificity of 92.7% for classifying all types of MI data from the PTB diagnostic ECG database. In [10], the detection and localization of MI has been done by the stacked Sparse Auto Encoder (SAE) and the decision tree which is bagged (TreeBagger) using the dataset from PTB diagnostic database. Here the optimization of the outcomes that the various decision trees provide are done more accurately using TreeBagger and the for MI detection, the accuracy level, the sensitivity level, and the specificity level up to 99.90%, 99.98%, and 99.52% respectively. Another approach is proposed in [11] to diagnose MI automatically using Flexible Analytic Wavelet Transform (FAWT) and it was applied to each ECG beat after beat segmentation. After operating the sample entropy into Random Forest (RF), J48 decision tree, back propagation neural network (BPNN), and least-square support vector machine (LSSVM), the

LSSVM shows the accuracy up to 99.31%. A proposition of latent topic Multiple Instance Learning to diagnose MI where the ECG signals are designed into a particular space where a number of topics define the spaces and hence support vector machine (SVM) has been applied on these ECG level topic vectors. This proposal of LTMIL achieves over 91% level of sensitivity and over 85% level of specificity in distinguishing MI affected ECGs from Healthy peoples' ECGs using the dataset from PTB diagnostic database [12]. In [13] another automated MI detection procedure has been approached applying Convolutional Neural network (CNN) on the ECG dataset from PTB diagnostic database. Apparently, using ECG beats with noise and without noise removal, this model has reached an average accuracy of 93.53% and 95.22% respectively. Another study [14] proposed a wearable heart attack detection and warning system by implementing a real-time ECG monitoring system where initially the ECG signal will be differentiated as normal or elevated. Then if the elevated heartbeat continues for thirty seconds, it will generate an audible alert system and along with the user, the system will notify the condition to a particular physician, designated hospital database, ambulance and next kin of the affected person as well. Detection of heart attack and monitoring system for heart rate applying IoT has been implemented in a proposal where the portable system including ECG sensor will alert heart attack or heart rate condition of the affected via android application [15]. In another research [16], an integrated IoT system including pulse sensor, Wi-Fi series communication device modules ESP8266 and microcontrollers have been used to detect heart attack by inspecting heart beat and an alert system to multiple smartphone contact numbers given to the system.

1.3 Problem Statement

Myocardial Infarction causes a significant number of deaths and multiple morbidities worldwide every year. It has been estimated that around 3 million people get affected with ST-elevated Myocardial Infarction (STEMI) and approximately 4 million people suffer from Non-ST-elevated Myocardial Infarction (NSTEMI) per year [17]. According to WHO, around 17.9 million people died due to Cardiovascular Diseases which represents 32% of global death. Among them, heart attack and stroke caused 85% of death. Moreover, various cardiovascular diseases occupy 38% of the 17 million premature deaths. Basically atherosclerotic plaque is one of the main predominant causes of Acute Myocardial Infarction and 70% of the lethal cardiac events occur due to atherosclerosis [18]. Insufficient blood supply to myocardium due to the blockage in blood vessels results in diminishing the balance of chemical reaction in cardiac cells. Eventually the imbalance of chemical reactions in cardiac cells give birth to various forms of ischemia, internal cardiac injuries, or necrosis.

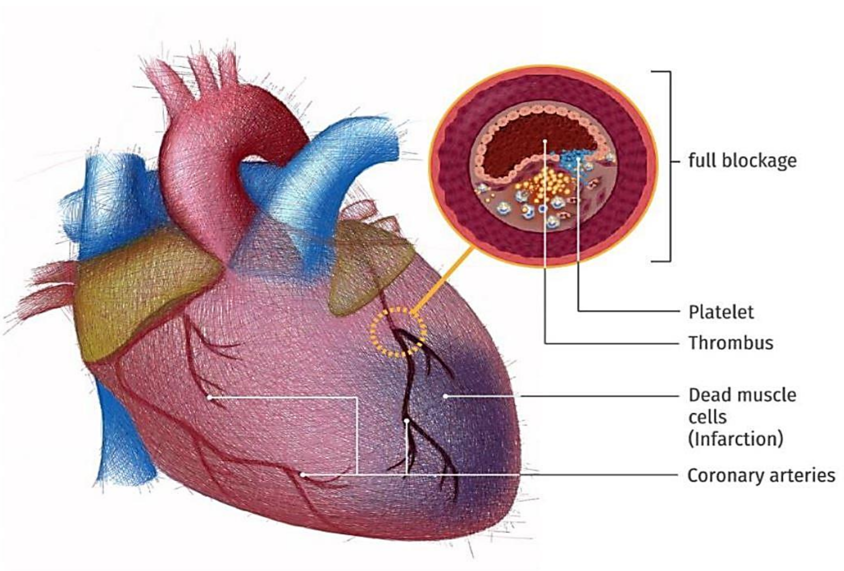


Fig 1: Illustration of Myocardial Infarction

Basically, the hindrance in blood flow through the artery due to plaques causes the deficiency of oxygen in the heart and drives obstacles in the chemical reactions. As Fig 1 [19] is showing the illustrated areas where the blood clots may take place due to the mentioned problems in the artery and ends up causing dead cells. These combined reasons are responsible for Myocardial Infarctions or heart attack [20]. The key to prevent Myocardial Infarction is to maintain a healthy balanced life and continuous deduction of harmful risk factors. There are nine chronic risk factors such as abnormal lipids, smoking, hypertension, diabetes, abdominal obesity, psychosocial factors, consumption of fruit and vegetables, regular alcohol consumption, and regular physical activity which are affiliated with causing Myocardial Infarction at a rate of more than 90% [21]. These modifiable risk factors and unhealthy behavioral lifestyle play significant roles behind the alarming mortality rate and morbidity due to cardiovascular diseases.

The initial approach to prevent these deadly cardiovascular diseases like Myocardial infarction is to sustain a healthy and balanced lifestyle both physically and psychologically. In the worst case scenario, if Myocardial Infarction takes place, immediate detection and initiate proper healthcare as soon as possible to avoid demise of the concerned patient.

1.4 Research Objectives

This research expedition involves immediate response to an MI affected patient. We aim to build up a portable prototype which will be able to detect several classes of MI depending on the variation of ECG signal especially in the ST segment. Eventually the further step is to develop a forewarning system using GSM and GPS modules in order to alarm any personnel related to the affected patient.

The method initiates with the training process of the model using a Data frame which includes raw ECG signals. Then the model will get trained through multiple machine learning algorithms. After the training procedure the model will perform necessary calculations and classifier models to detect the test data. The model will be able to detect particular types of MI.

The further process is to generate a serial communication via coding which will implicate the forewarning system. The forewarning system is incorporated with the application of GSM and GPS module. The detected result will be sent through text message to some personnel related to the MI affected patient with the current location.

1.5 Thesis Organization

The rest of the dissertation is organized as follows –

Chapter 2: *Infrastructure Coordination*

This chapter evolves the development of the overall project and the necessary software and hardware components, their features and specifications. Initially, a system block diagram is projected for brief understanding and consecutive description will establish an overview of the research. Gradually, machine learning algorithms that have been implemented for the detection procedure are also mentioned with proper reasoning and model architecture. The software and hardware requirements have been described elaborately according to the utilization procedure. The incorporation of hardware and software parts have been demonstrated, observed and analyzed thoroughly. Each equipment has been explained with visualization explicitly.

Chapter 3: *Data Processing*

The entire dataset processing is explained in this section thoroughly. At first the description of the chosen dataset has been provided with adequate reasoning for choosing this particular one from such a certain database which will be applicable for the prototype and accomplish the objective of this project. Then the preprocessing procedure has been analyzed elaborately with necessary algorithms and programming applications. Finally, the selected features that are going to be executed in the classifier models and machine learning algorithm has been discussed vastly.

Chapter 4: *Software Implementation*

This chapter signifies the detection of Myocardial Infarction with classification. The application of machine learning algorithms and implementation of several classifier models has been described adequately. Basically, the detection has been done using K-Nearest Neighbor (KNN), Random Forest (RF), Support Vector Machine (SVM) and Naive Bayes (NB) theorem and each class has shown a different level of accuracy and sensitivity. The ratio of test data and train data defines a crucial factor since it fluctuates both the level of accuracy and sensitivity. Thus the entire detection procedure has been explained at large scale in this section.

Chapter 5: *Hardware Implementation*

The forewarning system is the incorporation of a GPS and GSM module which has been described elaborately in this chapter. The incorporation has been done in the Arduino platform and the system sends a text message with classified MI information and the current location of the affected person to any relative. Basically the serial communication method has been described with required clarity and documentation.

This chapter provides an overall descriptive procedure of interfacing the detection method with Raspberry Pi. Since the forewarning system requires necessary information regarding the MI affected person and the location, this method will be executed through raspberry pi. Basically raspberry pi is working as a bridge between detection system and Arduino which implicates the GPS and GSM module. In this section, we have shown the experimentation methods after commissioning the software and hardware parts. Multiple trial and error loops were executed to reach the final outcome and every step is analyzed with corresponding methods and diagrams.

Chapter 6: *Result Analysis*

This chapter is the overall reflection of the outcome of this entire project. The result has been analyzed by comparing the accuracy of each approach of machine learning algorithm and the finalization of the effective method to detect MI and further execution of the forewarning system with the proper classification of MI and location of the deceased person with maximum reliability.

Chapter 7: *Sustainability and Durability Evaluation*

The chapter creates a projection on the environmental, economic, social impacts since these are very significant points for a project which is directly related to human health. Moreover, the clinically-alimented ethical issues have been discussed briefly as well as the risk managements while demonstrating this project.

Chapter 8: *Conclusion and Future Work*

Finally, in this portion of the report we have reflected our future thoughts on this project. We are implying to build an embedded system which is going to be more comfortable to carry and eventually will be more effective. Moreover, we intend to introduce some developments in detection procedure and preprocessing of dataset which will increase the efficiency of the system eventually.

However, the individual sections that have been briefly presented, will be discussed thoroughly in the following chapters. The chapters overall initiated to generate an overall idea about what is going on with the project. So, from the initial state of developing the project, every details of the hardware and software programs needs to be checked individually. The individual features and specifications have been discussed with particular application. Then the execution of the hardware and software parts have been shown with appropriate reasoning, evidence and applicable properties. Finally, a glimpse of our future intention with this research work has been given shortly.

Chapter 2

Infrastructural Coordination

The Infrastructure of the prototype is coordinated with a software and hardware system. Basically, the detection of Myocardial Infarction is executed by the software module and the later part which involves the forewarning system with location via text message is operated by the hardware module. Finally, the software and hardware module is integrated to generate the entire prototype.

2.1 Proposed Model

Here is a flow diagram of our proposed model which indicates the entire procedure of the prototype. The prototype is basically a combination of hardware and software setup which runs through simultaneously to project the final output.

Initially, the ECG datasets from ECG datasets are given as inputs to train the model so that the system can detect and decide any test data if it categorizes under Inferior MI, Anterior MI or Healthy ECG signal. After successfully detecting the type of MI or Healthy, the system initiates the forewarning procedure via Arduino Uno and GSRM module. Which means a text message with location will be sent to a certain contact number related to the patient making aware of his or her current condition.

Here is a block diagram given below to visualize the brief idea of the prototype where the steps are explicitly shown. The whole procedure evolves around these steps and executes according to the inherent program by the incorporation of hardware and software.

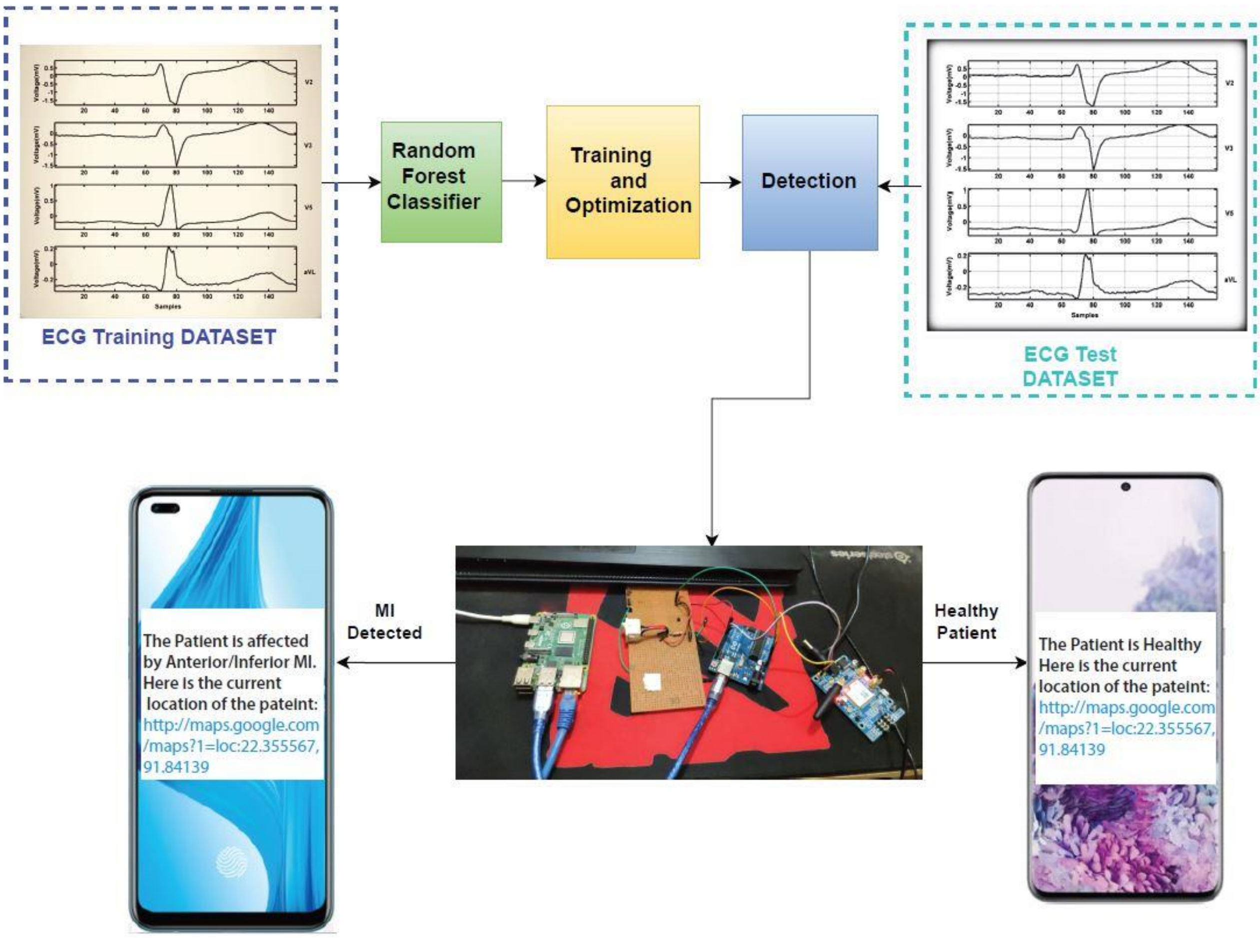


Fig 2.1: Block Diagram of the Infrastructure of the System

In brief, Fig 2.1 is the theoretical demonstration of our project which implicates the gist idea of the whole execution of the prototype. Eventually, all the steps are explained in the later chapters.

2.1.1 Model Description

The following block diagram represents the entire infrastructure of the prototype that we have generated briefly. Therefore, the blocks are accustomed to each other following certain flow that denotes the brief steps of the working procedure of the prototype.

The whole procedure revolves around an entire cycle to detect each test data. The system initializes with the input of training dataset which is going to train the model and creates a coding environment which is able to distinct between MI patient and normal or healthy patient. Basically, the model is trained by Random Forest Classifier since it has shown the best result while detecting among other machine learning algorithm. So, we have trained the system using Random Forest Classifier and the next step is to run the detection procedure. For the detection procedure we have to insert a test data manually and the trained system will be able to detect the inserted ECG data whether it falls into which category of MI or in healthy category.

The next step incorporates hardware setting and that is driven by a raspberry Pi. The code of training and testing procedure is then uploaded into the raspberry Pi and the detected signal then passes through an Arduino Uno which drives the next step. The next step is a forewarning system which acquires a signal from GSRM module and hence a message including location of the affected patient is received by a particular contact number related to the patient.

2.1.2 Machine Learning Algorithm

To execute the detection of Myocardial Infarction, we have applied multiple machine learning algorithms in order to decide which algorithm is appropriate for our dataset and hence assembling with the hardware system as well. Therefore, we have approached Random Forest Classifier, K-Nearest Neighbor method, Support Vector Machine and Naive Bayes Theorem and the methods are explained briefly in the following.

2.1.2.1 Random Forest

Random Forest is a machine learning algorithm based on supervised method and is extracted from algorithms of decision tree. Basically this method is used for problems based on regression and classification. It is based upon learning of ensembling which is a method that incorporates many classifiers to simplify compound problems and hence provide solutions. A random forest algorithm includes many decision trees and these decision trees provide certain predictions which finally drive the outcome of the algorithm. The forest obligated to the random forest algorithm is trained by the aggregating of bagging or bootstrap. Here, bagging is a meta-algorithm which is essentially ensemble which increases the accuracy of a program. Random forest eliminates the drawbacks of an algorithm of decision tree by reducing the overfitting problem of a dataset and maximizes the precision level as well. The features of Random Forest Algorithm are given below:

Random Forest provides more accuracy than the algorithm from the implementation of decision tree. Moreover, it generates an efficient methodological approach to incorporate the data or information which are missing. This algorithm produces adequate prediction without any requirement of hyper-parameter tuning. It accommodates the capability of solving the issue of overfitting in decision trees. In this algorithm, a subset of features is selected randomly in every random forest at the node's splitting point [22].

Now the classification Method of Random Forest algorithm is explained briefly in the following:

Random forests classification employs an ensemble methodology to acquire the result. Initially, the training data is fed categorically to various decision trees shown in Fig 2.2[22] allocated for training. This training dataset includes the features of observations and features that will be selected randomly while splitting the nodes takes place.

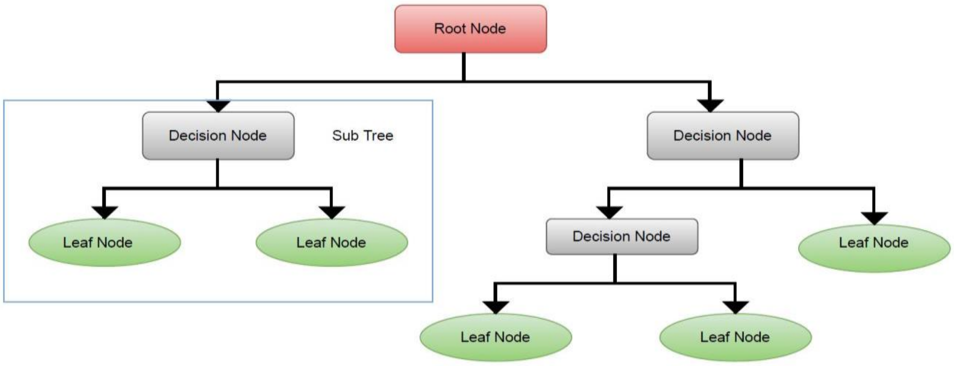


Fig 2.2: Integrated Flowchart of Decision Tree

A random forest system depends on the activities of multiple verities of decision trees. Each decision tree is implemented upon decision nodes, leaf nodes, and a root node. Each tree produces leaf nodes and the leaf nodes accumulated from the final outcome produced by the

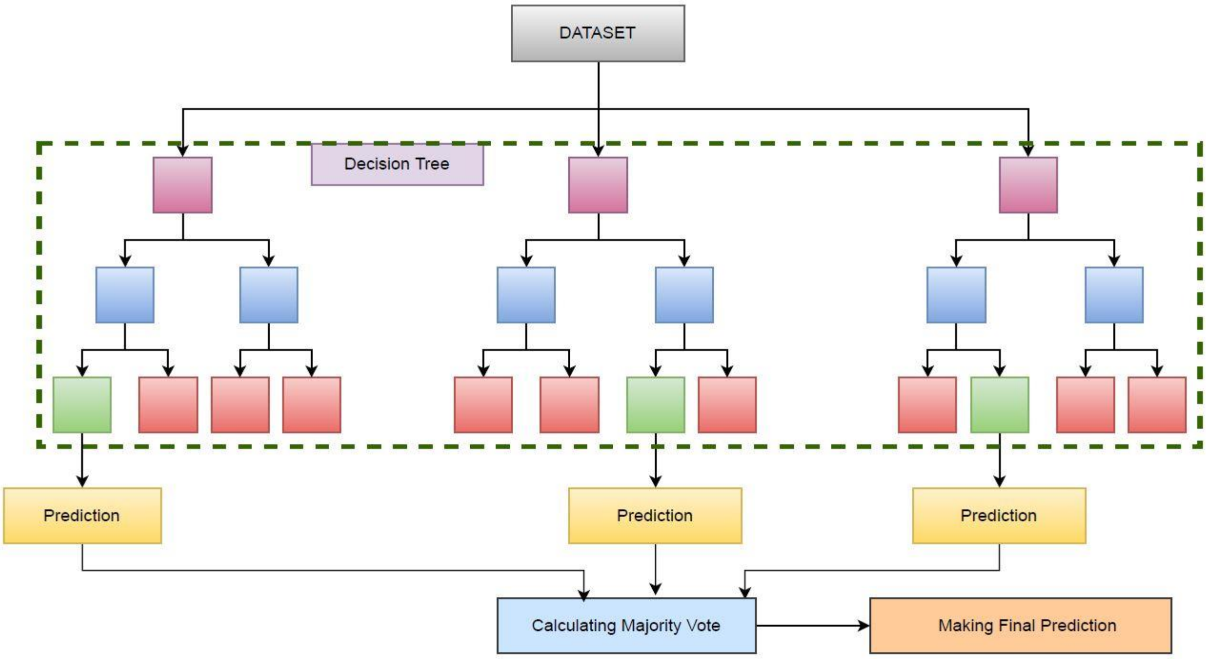


Fig 2.3: A Simple Random Forest Classifier

particular decision tree. The final output is projected by the maximized voting system which is visible in Fig 2.3 [22]. In this case, the selected outcome by the maximized nodes of the decision trees becomes the final consequent of the random forest system. The above visualization shows a simple random forest classifier [22].

2.1.2.2 KNN

K-Nearest Neighbor [KNN] is a supervised learning technique and one of the easiest approaches that assumes the similarity between the given categories with the new datasets or cases and finally takes the decision to assign the new case under the most similar given category which is represented in Fig 2.4[23]. Though this method is used for classification purposes mostly, it is also used for regression. Since KNN does not generate any assumption of prediction depending upon such data which are underlying and works as a non-pragmatic program [23].

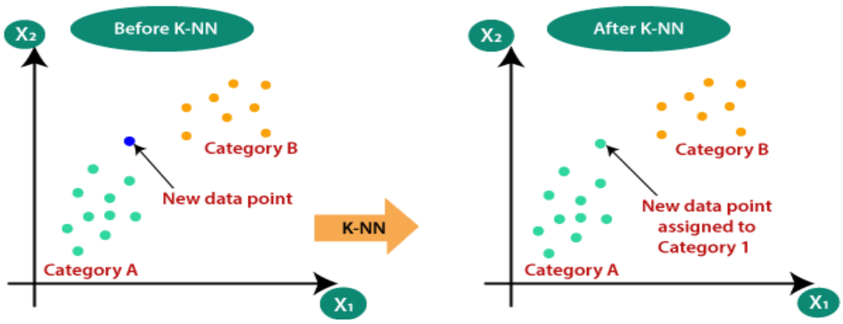


Fig 2.4: Architecture of KNN Algorithm

KNN algorithm works in an explicit procedural method which involves some steps that initializes with selecting the neighboring number. Then the calculation of the distance in Euclidean method for the number of neighbors ‘K’ has to be done. According to the calculated value of the Euclidean distance, the k nearest neighbor has to be selected. The next step requires counting of data points in each category from the k nearest neighbors. Finally, the data points are assigned to the category where the number of the neighbor is accountable as maximally similar.

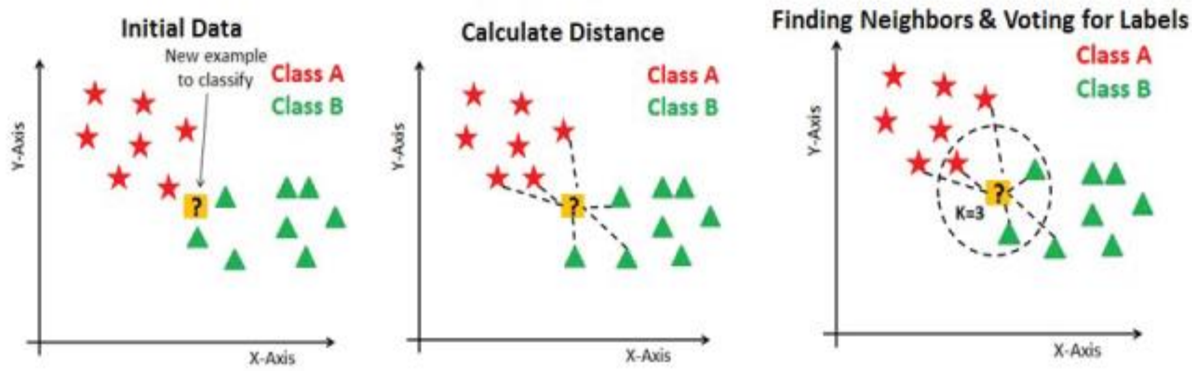


Fig 2.5: Steps of KNN Algorithm

Therefore, for visualization purposes we can represent the steps of KNN method by the following. First of all, the 'K' in KNN signifies the number of nearest neighbors and that is the crucial component in making assumptions [24]. The application of KNN can be briefly shown in three basic steps which are explicitly shown in Fig 2.5[24].

1. Calculation of Euclidean Distance
2. Selection of Closest Neighbors according to the Distance
3. Selected Neighbors and hence Voting for Labels [24].

2.1.2.3 Naive Bayes

The Naive Bayes method basically originates from Bayes Theorem which grounds from conditional probability. The conditional probability works as the following equation 1 [25]:

$$P(S|L) = \frac{P(L|S) * P(S)}{P(L)} \qquad \text{.....................(1)}$$

This method is basically a cluster of supervised learning algorithms which makes a "naive" assumption where the values of class variables are given so that each set of features accumulates a conditional independence. For instance, if the class variable is given as y, dependent feature vector is to the Naive Bayes theorem formulates the following equation 2 [26]:

$$P(y|x_1, \dots\dots.., x_n) = \frac{P(y)P(x_1, \dots\dots, x_n|y)}{P(x_1, \dots\dots, x_n)} \dots\dots\dots\dots\dots\dots\dots\dots (2)$$

The benefits of Naïve Bayes theorem are significantly mentioned for the rapid prediction in classifying test data sets and in the cases of multiple prediction of classes. In comparison with numerical variables with categorical input variables, Naive Bayes works efficiently with categorical values. Moreover, Naive Bayes provides the freedom of prediction even with a smaller amount of training data and works more effectively than other algorithms. Though freedom of prediction is a crucial benefit of this theorem, it is often difficult to manage the availability of independent predictors and that is why Naive Bayes is often called a bad estimator [25].

The four most significant effective sides of Naive Bayes algorithm are prediction of Real time, Prediction of Multi Class, Classification of text or Sam filtration or Analysis of sentiment and Recommendation structures [25]. To be specific, this algorithm has particularly programmed well in cases of classification of several documents and spam filtering [26].

2.1.2.4 SVM

In the applications which involve regression, classification, or outlier’s detection, one of the most effective machine learning approaches is Support Vector Machines [SVM]. SVM algorithm works in such a way that it aims to classify certain data points by creating a best fitting decision boundary that is able to categorize the data points efficiently. The best decision boundary is merely called hyper planes which are generated by the support vector machine. Here some absolute points are selected by the SVM algorithm which works as vectors while creating the hyperplanes [27].

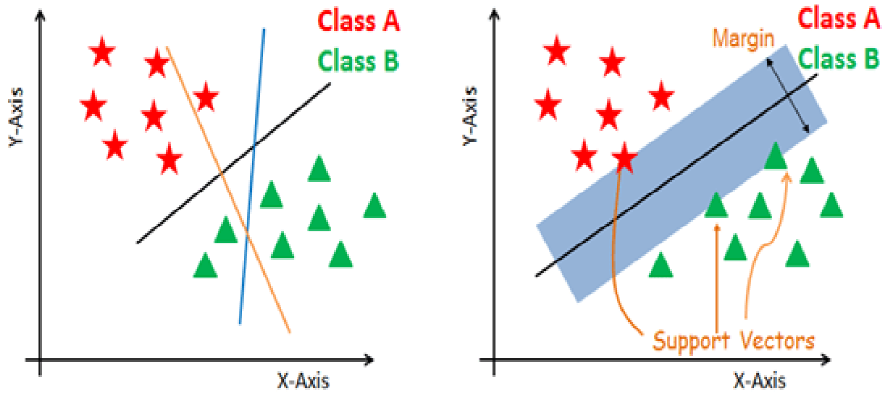


Fig 2.6: Architecture of SVM Algorithm and Hyperplanes

The best fitted hyperplanes classify the data points ensuring maximum margin distance so that the categories are explicitly represented as Fig 2.6 [27].

SVM provides some significant benefits which involve the following. In the programs where the units of dimensions are greater than the ones of samples and thus SVM performs very effectively. So, this benefit obliges that SVM is effectively applicable for the spaces or programs which are high in dimensions. SVM also provides the efficiency of memory as it exercises the model with subsets which involve some training points required while generating the decision function. These decision functions are eventually known as support vectors. Moreover, the decision functions require multiple types of Kernel functions. In SVM, the kernels can be particularly customized according to the necessity of the program [27].

Beside the benefits there are some inconveniences of support vector machines as well and by taking precautions may minimize these adverse effects which are represented in the following.

In the cases where the units of features are extremely greater than the counts of samples, while selecting kernel function it is favorable to abstain from over-fitting. Another significant point should be maintained and that is regularization. Furthermore, the Support vector machine does not generate any estimation of probability and so that needs to be estimated by an expensive five-fold cross validation method [28].

2.2 Software Requirement

To perform the detection of Myocardial Infarction, some software implementations are required and that involves certain programming software. Since our program is written in Python, we have used Google Colab to execute software operations.

2.2.1 Google Colab

Google Colaboratory or mostly known as the short form “Colab” is a well-known platform to implement and execute any arbitrary python code through browser and vastly used in machine learning and data analysis. Basically Colab is accommodates the functions of Jupyter Notebook which provides unconditional availability to computational purposes installing GPUs. This is a product from Google research which is availing its coding platform for multidimensional utilization without any particular subscription. Working with a huge dataset with multiple decision making algorithms was a difficult task to do yet it was quite smoothly executed using Google Colab. In google collab sharing is much easier as multiple person can access and edit a single file without having it saved on that person computer. Despite some limitations, Google Colab is a useful platform to operate project based research works [29].

2.2.2 Spyder IDE

In our project we have used Spyder Integrated Development Environment (IDE). This IDE consists of advanced editing, debugging, interactive execution and analysis tools [30]. However, as we are using python for the data analysis, this IDE is written in python. The version we have used for our project is Spyder 4. We have installed this software in our raspberry pi 4 where Spyder IDE will execute machine learning codes and will send an alphabet as a signal in the Arduino UNO R3 microprocessor. We have used Google Colab for running and testing the test data but the issue with Google Colab is as this is a hardware project we have to use compatible software to execute our hardware. Therefore, we have used Spyder IDE in our hardware as it is compatible to run and send signals in our arduino.

2.2.3 Arduino IDE

We have used an available program spaced named Arduino IDE software. Arduino IDE, downloaded from [31] is used for writing and uploading code in an Arduino Board. This software is implemented based on C and C++. Programs can be executed in the editor and hence easily uploaded to the microprocessor of our Arduino UNO R3. The GPS and GSM system which is set up with our Arduino needs to have correct serial communication and instruction from the Arduino for it to function properly. The version we have used for our project is 1.8.16 which is released on 6th September 2021[32].

2.3 Hardware Requirement

Accomplishing the forewarning system requires hardware implementation and certain electronic devices have been used to execute the project properly. The prototype that we have built consists of a microcontroller unit, GPS, GSM and power sources for all of the components. After carrying out several experiments we could know the limitations and the drawbacks of the components and finally came up with the following components.

2.3.1 Arduino Uno R3

The Arduino Uno as shown in Fig 2.7 is basically a microcontroller device which is structured as the basis of ATmega328. It accommodates digital input/output ports counted as 20 and out of them 6 pins have been utilized as PWM (pulse width modulation) outputs and 6 used as analog port, one resonator of 16 MHz, a USB port, a power input, a built in system programming (ICSP) header, an in built LED and a reset button. Besides, both input output Pin of 20 Ma and 3.3V Pin is 50 mA are dedicated for the DC current and. It can be simply connected to the computer with a cable or powered up by AC-to-DC adapter or 9v maximum battery. The Operating Voltage of the Arduino itself is 5V but the recommended input voltage from the user ranges from 7V to 12V. Moreover, the boot loader utilizes a flash memory of 32 KB and 0.5 KB memory is used by It also has SRAM of 2 KB with EEPROM of 1 KB and the CLK speed of the microcontroller is 16MHz [33].

Programming of the Arduino Uno R3 which is visible in Fig 2.7 can be implemented by using Arduino IDE software with any of the versions. The microcontroller on the PCB board that forms the whole Arduino is available and a boot loader pre-burns that allows to upload codes without using any hardware programmer externally.

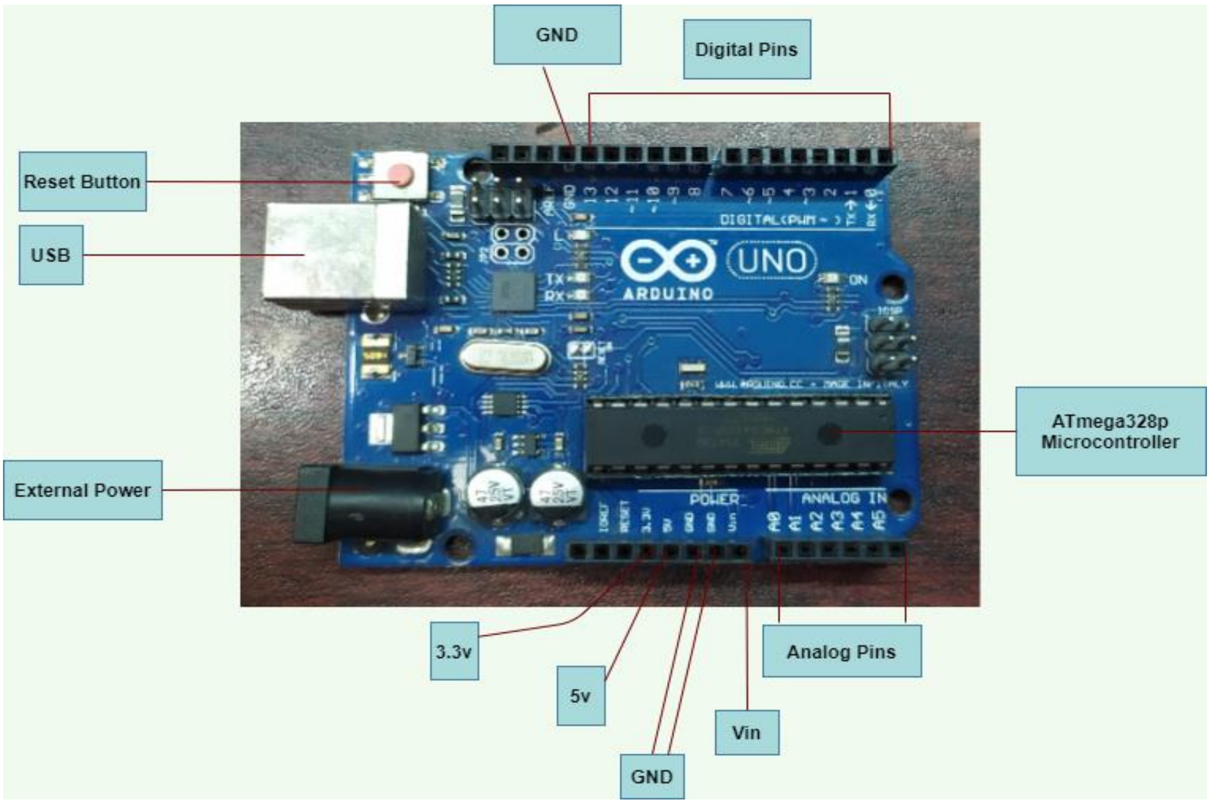


Fig 2.7: An Illustration of Arduino Uno R3

Besides the communication of this microcontroller can be done using an STK500. Protocol. We also can upload the program in the Arduino Uno by avoiding the boot loader using the header like the In-Circuit Serial Programming [33].

2.3.2 Sim808 GPRS/GSM Module

It is a combined module having the function of both GPS and the GSM. This device is an inherently commissioned system of Quad-Band GSM/GPRS module which incorporates GPS technology and hence executes satellite navigator implementation. The entire design of SMT potentially shows uprising efficiency in saving both time and costs for users to implement applications based on GPS requirement. The industry-standard interface and GPS function avails the multiple assets which are allowed to get tracked effortlessly at any particular location and immediate time with signal coverage. The prime features of the module are:

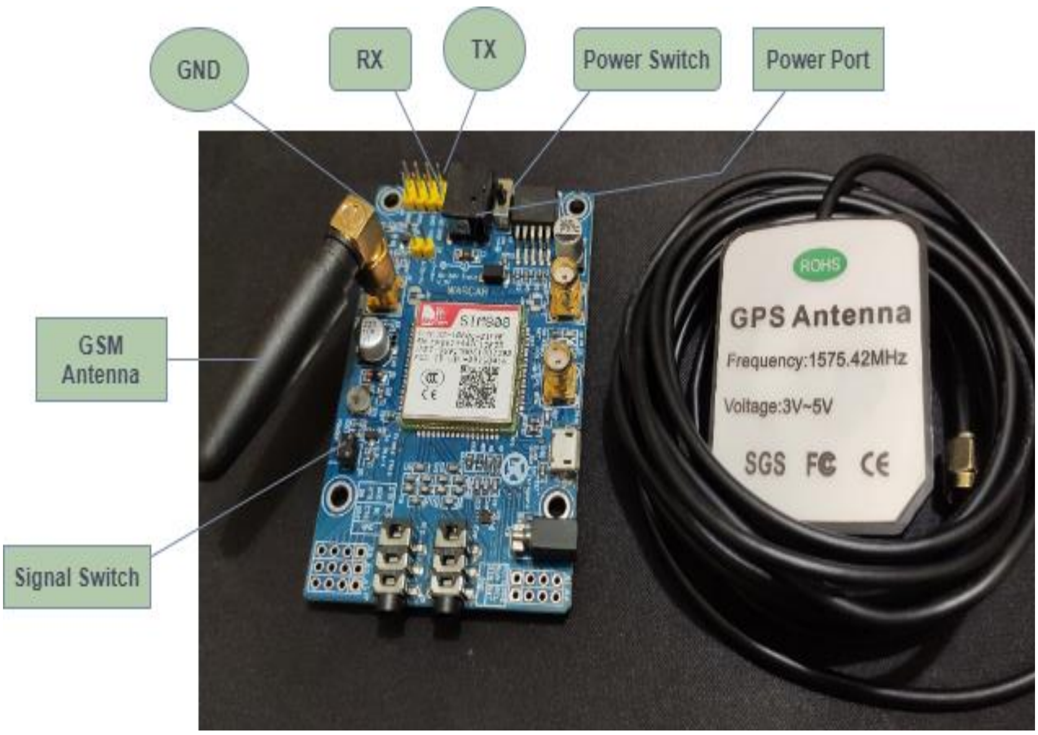


Fig 2.8: Integrated Sim808 GPRS Module

Have Quad-band of 850/900/1800/1900MHz, GPRS multi-slot class 12/10, GPRS mobile station, it is Compliant to GSM phase 2/2+, has Bluetooth: compliant with 3.0+EDR, Dimensions: 24*24*2.6mm, weight is approximately 3.3g, can be controlled via AT commands (3GPP TS 27.007, 27.005 and SIMCOM enhanced AT Commands), have a supply voltage range 3.4 ~ 4.4V, Low power consumption, operating temperature is -40℃ ~85℃. Some of the specifications are GPS Receiver channels: 22 tracking / 66 acquisition [34]. Here Fig 2.8 is an integrated GPRS System.

2.3.3 Raspberry Pi 4

Raspberry Pi 4 is basically a mini computer with the capabilities of a desktop computer. This comes in the size of a credit card which can be easily plugged into a monitor or TV and can be controlled by a standard keyboard and mouse. Presence of GPIO pins has enabled its ability to

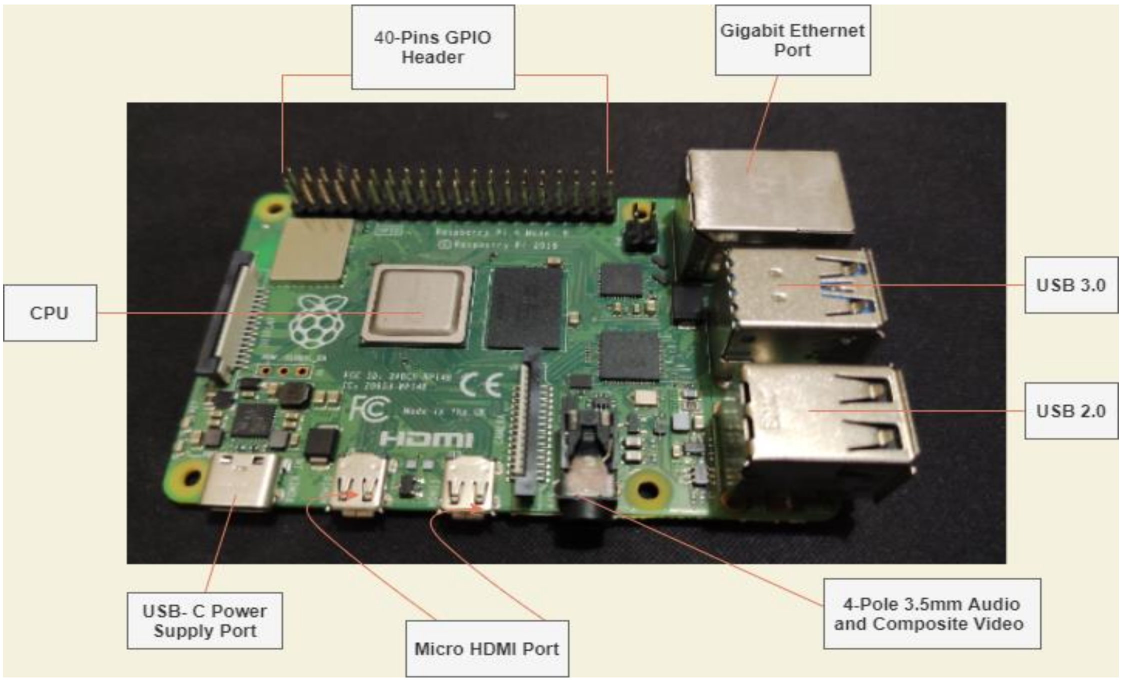


Fig 2.9: An Illustration of Raspberry Pi 4

support multiple sensors and thus for project-based research purposes it has become useful and the parts are specifically visualized in Fig 2.9. In case of digital maker projects, it has been applied as a wide array and the programming is done with Scratch and Python mostly. Raspberry Pi 4 has high processing speed which makes its usage even more applicable.

Our work requires an interfacing between decision command and forewarning system and that has been easily executed by Raspberry Pi due to its versatility.

Despite a few drawbacks related to the storage system, it is very handy for project based research works. Our project required Raspberry pi 4 since the specifications of this type integrate the necessity and activity of our system.

2.3.3.1 Specification of Raspberry Pi 4

This specification of Raspberry Pi 4 is defined by multiple significant points and they are given below:

This is a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decodes at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability. (via a separate PoE HAT add-on). It has a processor of Broadcom BCM2711and quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz with a memory of 2GB LPDDR4. It also contains Micro SD card slot for installing operating system and unit storage. Besides it has a connectivity a 2.4 GHz and 5.0 GHz, 11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports with GPIO: Standard 40-pin GPIO header (fully backwards-compatible with previous boards). For Video & sound, it has 2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port along with multimedia of H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics. Finally, the module Operates at a temperature of 0–50ºC [35].

2.3.4 Power Adapter of 5V-2A for Sim808 Module

Power Adapter- The power adapter rated 5v-2A is used basically to power up the Sim808 GSM/GPRS module. Power rating above this may burn out the module

Fig 2.10: An Illustration of Power Adapter

So, Fig 2.10 visualizes the kind of power adapter that we have used in our project.

2.3.5 32GB SD Card

A 32 GB SD card has been used for uploading the datasets and programming codes to execute the prototype smoothly.

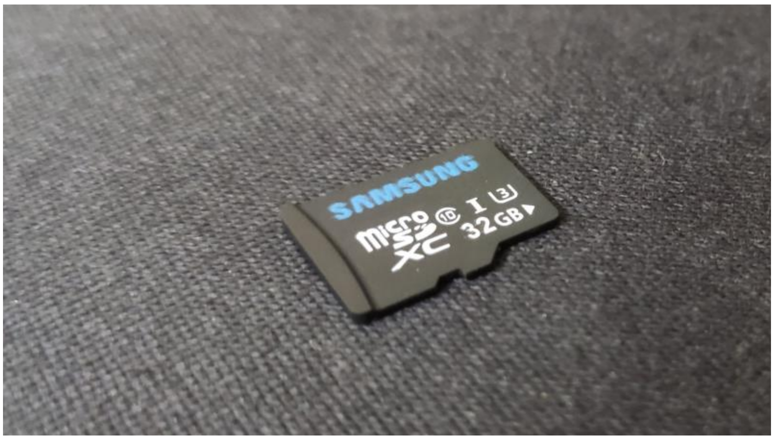


Fig 2.11: An Illustration of 32 GB SD Card

Fig 2.11 is the visual representation of the 32 GB SD card that we have used in our project.

2.3.6 Cables

a. HDMI cable: HDMI (High-Definition Multimedia Interface) cable is utilized for transmitting digital audio or video from sources like Computer as in Fig 2.12(a) [36].

a. HDMI Cable

b. Ethernet Cable

c.Type-C Cable

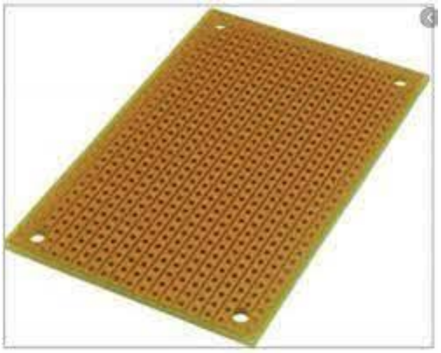
Fig 2.12: Illustrations of Cables a. HDMI Cable, b. Ethernet Cable, c. Type-C Cable

b. Ethernet cable: Ethernet cable is used to provide internet connection by connecting the devices into local network presented in Fig 2.12(b) [37].

c. Type-C cable: The Type-C charger in our project is used to power up the Raspberry Pi visualized in Fig 2.12(c) [38].

2.3.7 Vero Board, 16k Resistor, SPST Switch and Jumper Wires

In our project, the Vero board such as Fig 2.13(a) [39] is used as a connecting medium for Arduino, Switch, 16k resistor Fig 2.13(b) [40] for the testing purpose. In our project we have used a switch such as Fig 2.13(c) [41] to on/off or pause the prototype whenever necessary.



a. Vero board

c. SPST Switch

b. 16k Resistor

d. Jumper Wires

Fig 2.13: Illustrations a. Vero board, b. 16k Resistor, c. SPST Switch, d. Jumper Wires

Along with this switch, we used a 16k resistor to prevent short circuit. Finally, the jumper wires Fig 2.13(d) [42] are used to make all the connection of the components.

Therefore, these are the components that we have used while developing the prototype. All of them are easily available and applicable. Each component has been tested individually to ensure smooth activity after incorporation. After the development of the entire project, they have executed properly.

Chapter 3

Data Processing

Data processing requires some crucial steps including data collection, data preprocessing, feature selection and these steps are necessary to prepare the datasets for the algorithmic or any programming approach.

3.1 Dataset Description

In our project, we have utilized the ECG wave dataset presented in .csv form which were collected from the Physionet Databank [43] which is an efficient website for datasets regarding health and other purposes. The particular dataset that we have utilized is a famous ECG database called Physikalisch-Technische Bundesanstalt diagnostic ECG Database [44]. This huge source avails the ECG signal of around 300 patients. Among them 148 patients are affected by Myocardial Infarction and the rest 52 are recorded as healthy people also shown in Table 3.1 [44].

Characteristics	**Myocardial Infarction**	**Healthy People**
Minimum Age (years)	36	17
Maximum Age (years)	86	81
Average Age (years)	60.37	43.43
Number of Male	110	39
Number of Female	38	13

Table 3.1: Characteristic Table of the PTB Diagnostic Database

Each dataset of the database includes 12 lead ECG signals of each patient and the signals are represented as m Volts. The MI affected patients are classified as Inferior MI and Anterior MI which makes this database even more valid and user friendly. Since we are working on a system that can detect several classes of MI and indicate later by a forewarning system, this database was undoubtedly useful. However, the datasets are not refined as they include several noises as well. The characteristic Table 3.1 defines the information of the database more precisely.

3.2 Feature Selection

From the database we have used the ECG signals of 90 people. Among them 30 ECG samples belong to Anterior MI section, 30 ECG samples belong to Inferior MI section and 30 ECG samples belong to Healthy People. In the PTB diagnostic Database there are ECG datasets for 12 leads and each lead indicates each corresponding connected position of the heart. To speculate the objective of our project we have utilized the ECG signals of 4 leads among the 12 leads. Since our project will be utilized as a mobile monitoring system rather than professional medical purpose, we have taken the data of the most relevant 4 leads that can indicate the two types of MI.

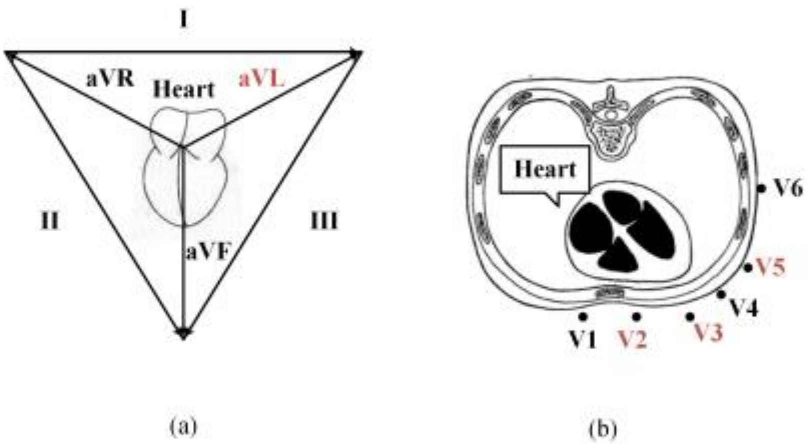


Fig 3.1: Acquisition of Cardiac Electrical Activity via 12-lead ECG (a) 6 Standard Limb Leads, (b) 6 Precordial Leads

Initially, the lead V3 has been selected as it is placed to the closest place to heart for the anterior wall [45]. Secondly, V2 is employed for the septal wall since it provides larger R waves than V1 and therefore easily identifiable by following algorithm [46]. Thirdly, for lateral wall V5 has been selected because it projects the highest sensitivity in detecting Myocardial Ischemia [47]. Finally, aVL is selected because of its consistency as it is a unipolar lead in addition to the precordial lead. Additionally, aVL is more useful while diagnosing MI caused by Left Anterior Descending coronary artery occlusion, especially extensive anterior MI [48]. The entire analysis is visible in Fig 3.1 [6].

3.3 Data Preprocessing

For data preprocessing, we have taken the ECG signal for one second duration for the four leads v3, v2, v5 and avL. This extraction simplified the detection procedure as well as it will activate the detection procedure as immediately as possible. So, in case of both the training sets of data and the test dataset we have articulated each sets of data for one second duration but

Out[3]:

	'0:00.000'	'0:00.001'	'0:00.002'	'0:00.003'	'0:00.004'	'0:00.005'	'0:00.006'	'0:00.007'	'0:00.008'	'0:00.009'	...	'0:03.995'	'0:03.996'	'0:03.997'	'0:03.998'	'0:03
0	0.391	0.413	0.431	0.441	0.450	0.468	0.481	0.491	0.509	0.530	...	0.283	0.290	0.297	0.299	(
1	-0.030	-0.033	-0.041	-0.041	-0.043	-0.051	-0.063	-0.080	-0.085	-0.091	...	-0.636	-0.636	-0.626	-0.626	-(
2	0.269	0.278	0.290	0.299	0.300	0.298	0.294	0.289	0.283	0.276	...	0.172	0.176	0.178	0.184	(
3	-0.602	-0.610	-0.612	-0.618	-0.622	-0.626	-0.633	-0.640	-0.646	-0.654	...	-0.915	-0.920	-0.923	-0.922	-(
4	0.079	0.081	0.083	0.077	0.076	0.073	0.077	0.080	0.075	0.077	...	0.026	0.015	0.024	0.025	(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
85	0.185	0.194	0.199	0.205	0.216	0.221	0.233	0.242	0.247	0.255	...	0.448	0.447	0.446	0.444	(
86	0.117	0.015	-0.070	-0.167	-0.270	-0.353	-0.401	-0.430	-0.446	-0.453	...	-0.256	-0.254	-0.256	-0.260	-(
87	-0.207	-0.205	-0.204	-0.206	-0.205	-0.206	-0.210	-0.212	-0.208	-0.210	...	0.030	0.041	0.035	0.031	(
88	0.407	0.409	0.414	0.420	0.425	0.430	0.435	0.443	0.448	0.452	...	0.165	0.166	0.168	0.170	(
89	-0.043	-0.041	-0.039	-0.041	-0.044	-0.043	-0.040	-0.039	-0.043	-0.041	...	-0.259	-0.258	-0.257	-0.260	-(

90 rows × 4005 columns

Fig 3.2 Training Dataset

point to mention here that the ECG signals in the datasets are so scrutinized that they have been calculated in scale of milliseconds and for the inputs have been applied in millivolts.

The loaded dataset in detection code for training purposes has been shown in Fig 3.2. After formatting the dataset according to the above mentioned procedure, we have processed the by using Standard Scaling Function. Therefore, the exquisite expulsion during the formation of the dataset has prepared the data set to be efficiently applicable for the above mentioned machine learning algorithm.

3.4 Training Dataset

We have trained our model using 90 samples of ECG signal. Since we aim to produce a model which will detect two types of MI, we have taken 30 samples each from Anterior MI and Inferior MI affected patients from PTB diagnostic database [44]. To compare the MI datasets with the healthy ones, we have taken 30 samples of healthy ECG signals. Altogether these 90 samples have been used to train our model using multiple algorithms and later we have observed which algorithm provides the correct detection.

3.5 Testing Dataset

We have manually tested 20 types of ECG signal which have been collected from PTB diagnostic database [44] as well. We have already studied the medical background of each of these 7 patients and thus used them to check if our proposed model is working according to the requirement correctly or not.

Here is the thorough observation of Test Dataset:

Test Dataset Type	Age	Sex	Reason for Admission	Acute Infarction (localization)	Smoker	Previous Infarction
Anterior MI 1	69	Male	MI	Anterior	Unknown	n/a
Anterior MI 2	60	Male	MI	Antero-septal	Unknown	Yes
Anterior MI 3	63	Male	MI	Anterior	No	n/a
Anterior MI 4	36	Male	MI	Antero-lateral	Yes	n/a
Anterior MI 5	49	Male	MI	Antero-septal	Yes	n/a
Anterior MI 6	53	Male	MI	Anterior	Yes	n/a
Anterior MI 7	49	Male	MI	Antero-septal	Yes	n/a
Inferior MI 1	58	Female	MI	Inferior	Unknown	n/a
Inferior MI 2	55	Male	MI	Infero-lateral	Unknown	Yes
Inferior MI 3	61	Female	MI	Inferior	Unknown	n/a
Inferior MI 4	65	Male	MI	Inferior	Yes	n/a

Inferior MI 5	47	Male	MI	Infero-lateral	No	n/a
Inferior MI 6	65	Male	MI	Inferior	Yes	n/a
Inferior MI 7	66	Female	MI	Infero-lateral	No	n/a
Healthy Patient 1	48	Male	Healthy Control	No	Unknown	n/a
Healthy Patient 2	48	Male	Healthy Control	No	Unknown	n/a
Healthy Patient 3	48	Male	Healthy Control	No	Unknown	n/a
Healthy Patient 4	57	Female	Healthy Control	No	Unknown	n/a
Healthy Patient 5	35	Female	Healthy Control	No	No	n/a
Healthy Patient 6	50	Male	Healthy Control	No	No	n/a

Table 3.2: Characteristic Table of Test dataset

Here, Table 3.2 is the visual projection of the characteristic of the test dataset that we have used to detect Myocardial Infarction individually for 20 different cases manually.

Chapter 4

Software Implementation

Several software programs have been utilized to execute our project. The python version we used is 3.9.6, which was released June 28, 2021. There are multiple Integrated Development Environments (IDE) used for the Machine Learning algorithm in python and those are Google Collab, Jupyter notebook and Spyder. In our hardware we ran the python code in Spyder. Moreover, we have used Arduino IDE which is an open source software developed in C and C++ programming language. It is used for writing and uploading codes in our Arduino. Moreover, we have executed multiple machine learning algorithms and our python code to observe and select the suitable classifier.

4.1 Execution of Machine Learning Algorithms using Python

In this project our objective is to distinguish two types of Myocardial Infarction Anterior and Inferior from a 4 lead ECG data using Machine Learning Classifiers. Basically, three kinds of machine learning algorithms are practiced and those are 1. Supervised Learning 2. Unsupervised Learning and 3. Reinforcement Learning. As we destine to classify our test data and we are using supervised learning as it is used for prediction. Supervised learning has to have labelled data as input from where they train themselves using the label as output and feature as input. Whenever any test data is assigned to predict its class our Machine learning algorithm is going to match them with the features assigned in our train data and check their label so that it can classify the test data. Process of deciding the class based on a feature depends upon the algorithm applied for the classification process. To start the classification initially we have to import our dataset for the process. We have used “Pandas”. Which is a publicly available source python library vastly utilized for data analysis. Using this library, we imported the dataset to our python code. We have imported the training and test data together in a single file. Following

this “scikit-learn” library is used to split the dataset between train and test. This process is known as train-test split. However, Train-test split technique is a generic algorithm which varies the train and test data each time it runs. So, to stop that randomization we have set a parameter called “random state” when using train test split. Because we preserved the position of each test data in the same position each time, we only had to set the random state once. In our project we are testing one patient at a time instead of predicting multiple test data once. That is because we have to send a message with the current location of the patient instantly for Myocardial infarction which we are using in our case and a constant 15-minutes cycle when the patient is in healthy condition.

4.1.1 MI Detection using Random Forest

Random forest algorithm is a popular algorithm to use for learning required for supervised system. In our Hardware we have implemented a random forest classifier as our final algorithm to classify the test signal and send the notification accordingly. In [49] prediction of Heart disease is carried out from Kaggle [50] site where they have predicted heart disease of a person using random forest and had a sensitivity value as 90.6% patient predicted heart disease were correct and 82.7% patient without heart disease predicted corrected. Therefore, the Random Forest classifier is a reliable classifier to use. Despite the fact that the data set in [49] is different from the one we are employing. In our dataset which is collected from PTB-Diagnostic ECG Database [44] algorithms will compare the magnitude of voltage from the ECG of a particular time with the magnitude of other patients at that exact time as a feature. To control the predictive power, we have to tune a hyper parameter called n-estimator. It decides the number of trees for the algorithm to take maximum votes. In our case we are going to set the n-estimator which is not divisible by three to avoid an even number of voting for three cases: Healthy, Inferior and Anterior. We experimented with two different n-estimator values: 37 and 43.

4.1.2 MI Detection using KNN

K-Nearest Neighbor is a machine learning algorithm which decides the class of a test data based on k number of closest points of train data. For K with a value of 1 it will choose the closest one value and check the class of that point to classify the test data. In [51] along with normal subjects total eleven myocardial infarctions with above 90% accuracy with data collected from MIT-BIH PTB database [43]. Data preprocessing such as baseline removal, QRS and iso-electric level detection was done before applying k-nearest neighbors. However, we used data directly collected from PTB-Diagnostic ECG database [44] and arranged the data for four leads to train the system. The accuracy and output detection results are discussed in Chapter 6 result analysis.

4.1.3 MI Detection using SVM

SVM is a classifying algorithm we have used from scikit-learn. In our case our data set has three target values Inferior, Anterior and Healthy. Both linear and non-linear problems can be solved using this algorithm. SVM produces a model depending upon its training data for acquiring the best outcome. In [52] 12 lead data used from PTB diagnostic ECG database from Physiobank [44]. And not only ST segment change was considered in this paper but they have also considered Q wave development to detect Myocardial Infarction. In [52] the SVM has a class accuracy of 98.33% where it performed upon 220 parameters. But in our case we have considered four leads as we are trying to propose a portable system. SVM is applied using scikit-learn library after using standard scaling in our dataset which is also a preprocessing library from scikit-learn. The outcome of this algorithm in our system is discussed in Chapter-6 result analysis.

4.1.4 MI Detection using Naive Bayes

Bayes theorem is a statistical law which expresses the probability of an occasion. Naive Bayes classifier is a group of constructive algorithms which is conceptually dependent on Bayes Theorem. In our case we have fit our trained data into the Naive Bayes theory using scikit-learn library for Naive Bayes. A term is used in the Naive Bayes classifier known as Positive predictive value which is the proportion of true result of the test data and tested result it can be calculated using the Bayes theorem. In our project we have experimented the standard scaled 4 lead ECG data into the naive Bayes to predict the correct class of our result. However, this algorithm did not perform well against our training and testing data as expected. In Chapter 6 result demonstration we can see the full analysis of the result using a confusion matrix.

Chapter 5

Hardware Implementation

This chapter revolves around the elaborated execution of the hardware section of the prototype. Each part has been described below with their specification and application in this project with proper reasoning and demonstration procedures.

5.1 Features and Applications of GPS and GSM

The features of GPS and GSM that are significant for the project. They mainly used to get the exact location of the patient through GPS and to send a message with that location through GSM. The features of GPS and GSM that are significant for the project are briefly mentioned below.

5.1.1 Global Positioning System (GPS)

Global Positioning System (GPS) is a widely satellite system of navigating providing particular location, time orientation and velocity.

GPS systems is available everywhere like in cars, smartphones and even your watches. GPS keeps a track from our journey from one point to the other. It is basically a navigation system that uses satellites, a receiver and algorithms to synchronize the location, velocity and time data for air, sea and land travel.

It works through a technique called trilateration that is used to calculate location, velocity and elevation. Trilateration collects signals from satellites to output location information. It also has triangulation that is used to measure angles without the distances [53].

5.1.2 Global System for Mobile Communication (GSM)

Global System for Mobile Communication or GSM is a mobile network which is digitized that is vastly applied in mobile phone application. GSM functions by utilizing a differentiated approach of various accesses to time division. TDMA is basically applied for the three digital wireless communications. GSM sends data through a channel after digitizing and compressing in particular time slot. It normally executes at either the 900 megahertz (MHz) or 1,800 MHz frequency band.

High-Speed Circuit-Switched Data (HSCSD), General Packet Radio Service (GPRS), Enhanced Data GSM Environment (EDGE) and Universal Mobile Telecommunications Service (UMTS) are installed in it. Its limitations are firstly, as GSM uses a pulse-transmission technology it has electronic referencing issues. Sometimes it interferes with functions like hearing aids. Secondly lagging of Bandwidth due to using GSM technologies as various operators and users are availed to the similar bandwidth it may consequent in considerable latency and finally inadequate rate of passing data. GSM avails certain data transfer rate [54].

5.2 Composition of Arduino Uno R3 and Sim808 Module

The system basically alerts any family member or a person related to the patient about the patient's condition as long as the patient is in touch with the system. Here, our prime target is to notify the condition of the patient (whether he is in healthy condition or affected by MI) to his/her family and nearby hospital along with his/her current location and for that reason we have to build a particular system which will able to give the update of the patient constantly after some times when the patient is out of home. So, for obtaining that result we built a system with the help of Arduino and Sim808 module which will notify the condition of the patient through a message along with his/her current location to the phone number of their family member and also to the nearby hospital with the help of the inbuilt GPS and GSM service of the Sim808 module.

So, for the interfacing part of the Arduino and Sim808 module, we connected the TX and RX pin of the Sim808 module with the digital pin 7 and 8 of the Arduino. These RX and TX mainly did the job of receiving and transmitting the data between Arduino Uno and Sim808 modules. Then, we connected the ground of the Sim808 module to the ground pin of the Arduino. After that, we provided the power supply to the Sim808 module through a 9V - 2A adapter and power supply to the Arduino was provided through a Li-Po battery of 7.4V - 1100 mAh capacity. After finishing all of these setups, we compiled our code on the Arduino software using the "SoftwareSerial" library and after compiling we uploaded it onto the Arduino UNO.

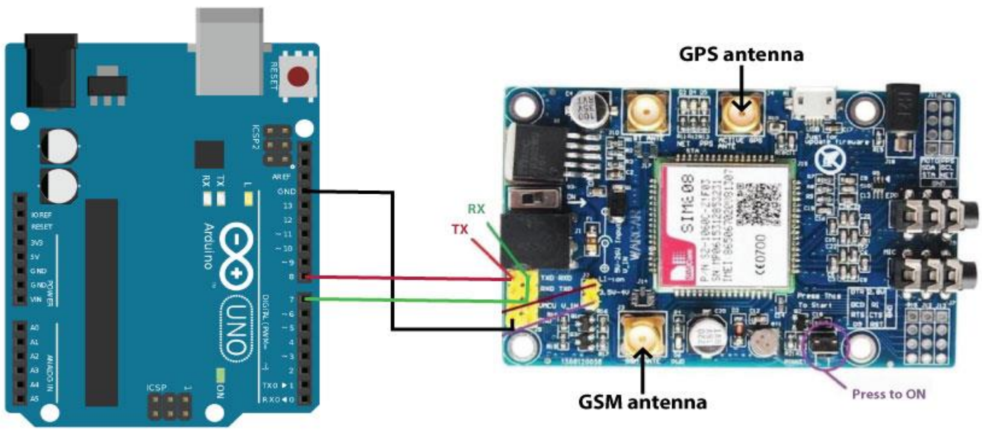


Fig 5.1: Connection Between Arduino Uno R3 and Sim808 Module

Here, Fig 5.1 illustrates the hardware connection between Arduino and Sim 808 module in our project.

After uploading the code successfully, we had to wait for few seconds as the GPS took a little bit of time to display the latitude and longitude value as shown in Fig 5.2. So, as soon as the GPS started to work properly, we will be able to see the below output on the serial monitor of the Arduino software.

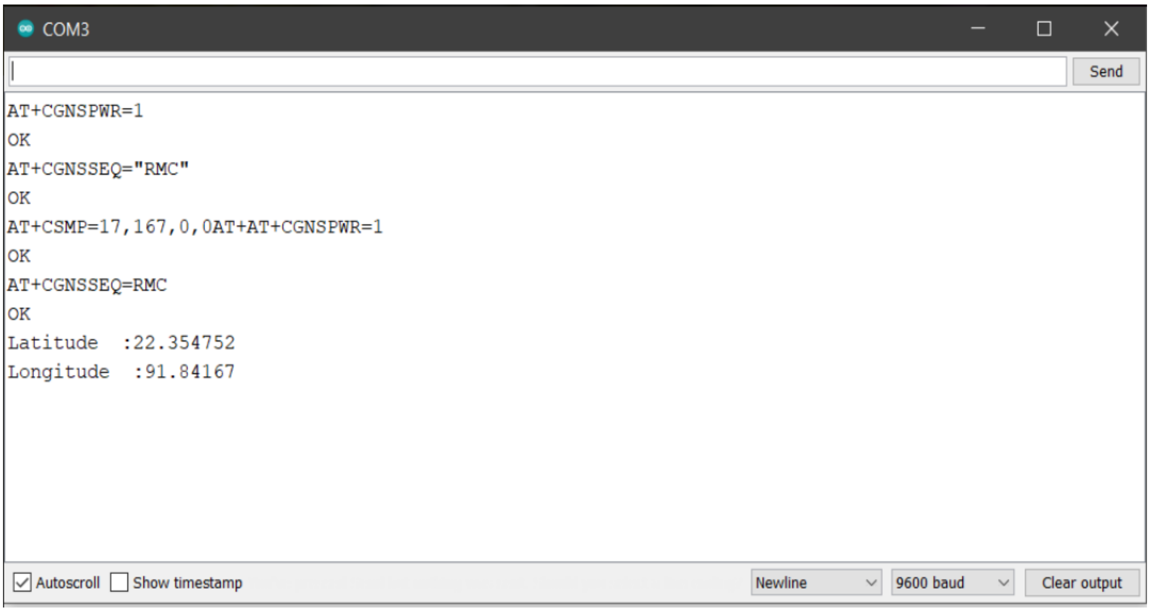


Fig 5.2: Serial Monitor Output

Here, we set the whole system in the code in such a way that it will notify the patient's condition every 15 minutes whether he/she is in good health or affected by MI. Also, if somehow the patient is affected by MI, it will send a message to the family member with a notification sound so that they can easily understand that the patient is in danger currently.

5.3 Integration of Raspberry Pi 4 with Arduino Uno R3 and Sim808 Module

Here, for the classification purpose of MI (Anterior, Inferior, Healthy), we had to use the machine learning algorithm using Python. On the other hand, our hardware interfacing is mainly based on Arduino UNO and Sim808 module. So, to merge both our software data and hardware interfacing together, Raspberry Pi 4 is used as a suitable medium and after merge the whole things together we will be able to generate a complete setup to notify the family and nearby hospital about the patient condition.

So, the hardware connection between Arduino and Sim808 remain as same as before. But, instead of giving the power supply to the Arduino UNO through a Li-Po battery, we will now give the power supply to the Arduino UNO through Raspberry Pi 4. Here, the power to the Raspberry Pi is provided through a Type-C based adapter cable and an Ethernet cable is also used to give the network connection to the Raspberry Pi 4. Moreover, we also used a memory

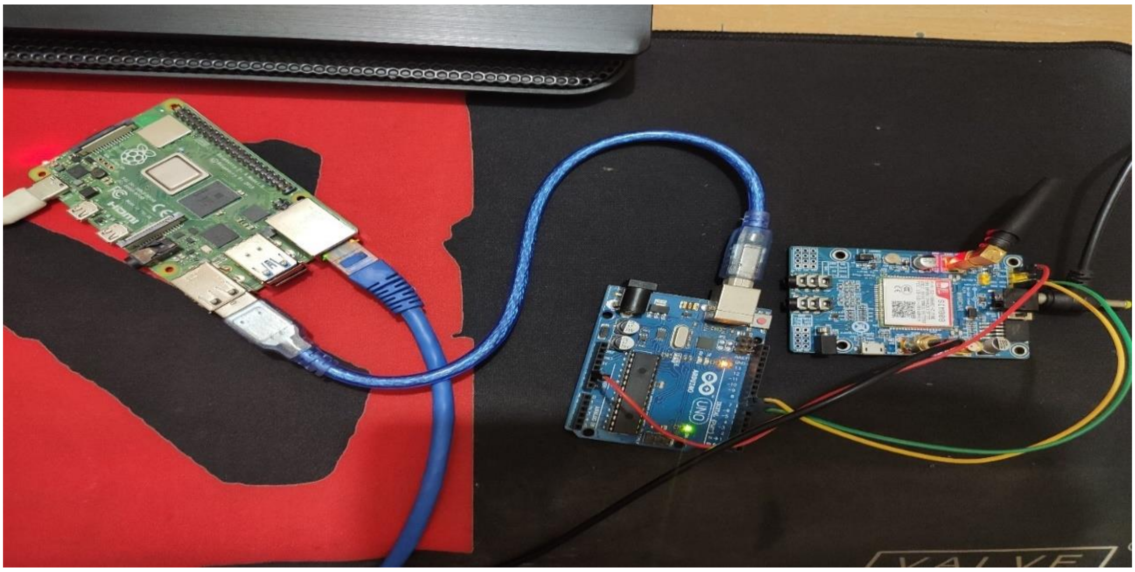


Fig 5.3: Connection of Raspberry Pi 4 with Arduino Uno R3 and Sim808 Module

card of 32 GB mainly to boot our Raspberry Pi 4 and finally we ran our Raspberry Pi by using the remote desktop connection. The hardware connection between Raspberry Pi with Arduino UNO and Sim808 is shown in Fig 5.3. As the connections are quite explicitly shown. Each device is connected with another in corresponding port through multiple wires and cables.

Now the output which is shown in the monitor is visualized below and it is the output of serial communication. Apparently, this particular output shows the outcome of the identified location of the affected patient with latitude and longitude.

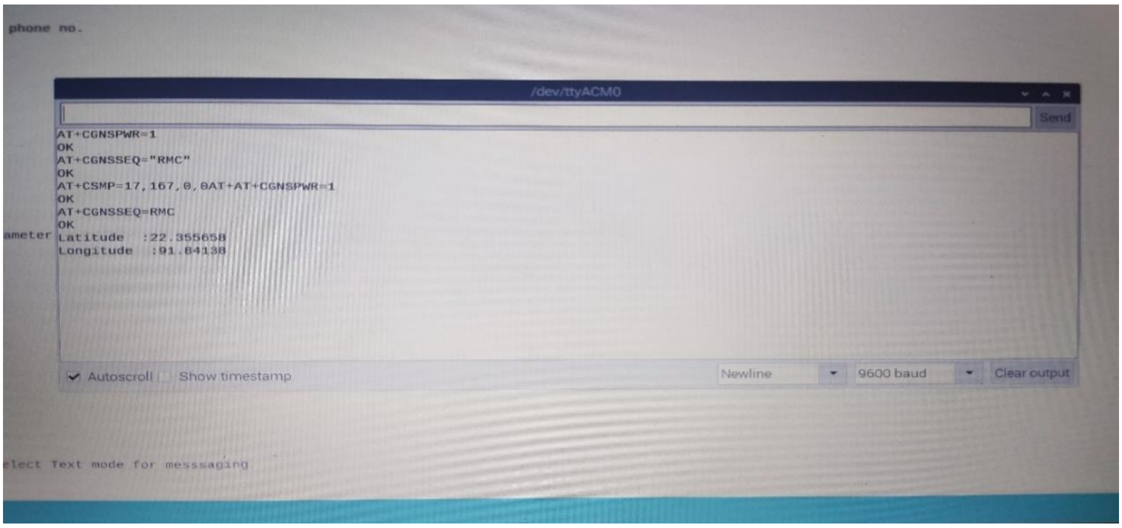


Fig 5.4: Serial Monitor Output (Raspberry Pi 4)

Like the previous one, here in Fig 5.4, we will also see the same output in the serial monitor of the Arduino software after a few seconds of uploading the code. Here, we will also see the continuous update of the patient in terms of location and condition and the system will also continuously notify the family member about the patient condition.

5.4 Incorporating Switch with Arduino Uno R3

A switch was also introduced to our system just to make it easy to start, stop or even pause our system like when the patient is in their home, there is no need to run the system. Again, suppose the patient is affected by Anterior MI; we have to pause the system; otherwise, it will continue to send messages every minute to the patient's family member, which is not required as the system is already notified about the condition of the patient previously.

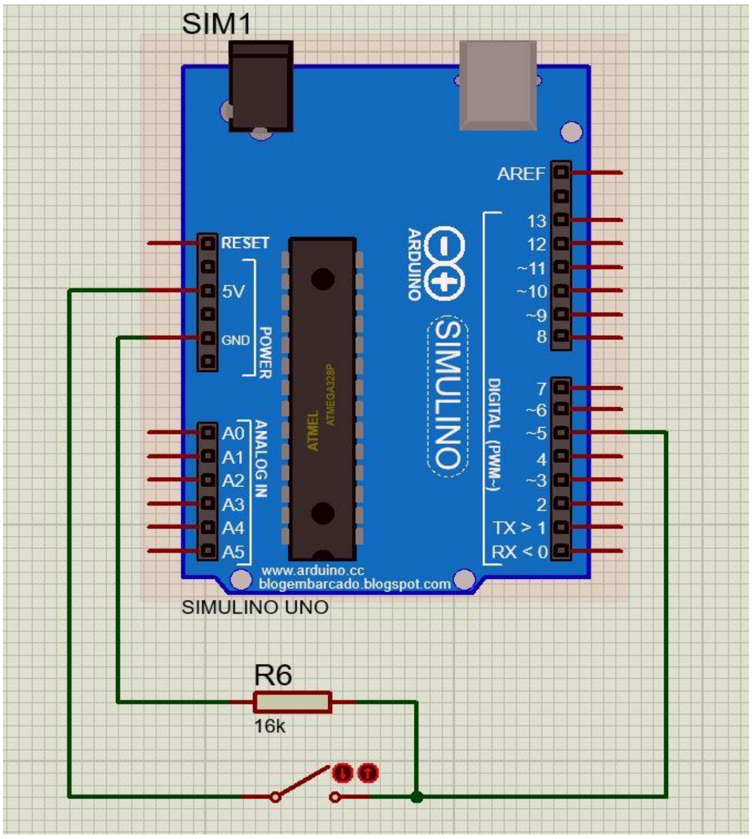


Fig 5.5: Connection Between Arduino Uno R3 and Switch

So, to build the switching part, we first inserted a switch into Veroboard, where one of the outputs of the switch was connected to the ground of the Arduino UNO. But, in between that, we use a 16k resistor to prevent a short circuit between the ground and 5V connection when we turn on the switch, and this resistor was commissioned to the digital pin 5 of the Arduino UNO. After that, the other output part of the switch was commissioned to the 5V supply of power to the Arduino board. Finally, by using this switch, we will be able to start, stop or pause our messaging system. The above Fig 5.5 shows the connection between Arduino UNO and the switch.

5.5 Demonstration

The demonstration of the and test run of the project is briefly explained by the following four steps.

5.5.1 Arduino Uno R3 and Sim808 Module Connection using Arduino IDE

Here, we have to make an interconnection between the Arduino UNO and Sim808 module. The Sim808 module will help us to send our notified message to the patient's family member along with his/her current location [55], which will help the family member to know about the patient's condition whether he/she is in a healthy condition or affected by MI.

To connect the integrated Sim808 GPRS module with Arduino Uno, we connected the TX and RX pin of the Sim808 module with the digital pin 7 and 8 of the Arduino Uno respectively using multiple jumper wires. The RX and TX pin of the module mainly handles the receiving and transmitting of the data between Arduino Uno and Sim808 module. After that the pin indicating as ground of the Sim808 module is connected to the similar pin indicating as ground of the Arduino Uno.

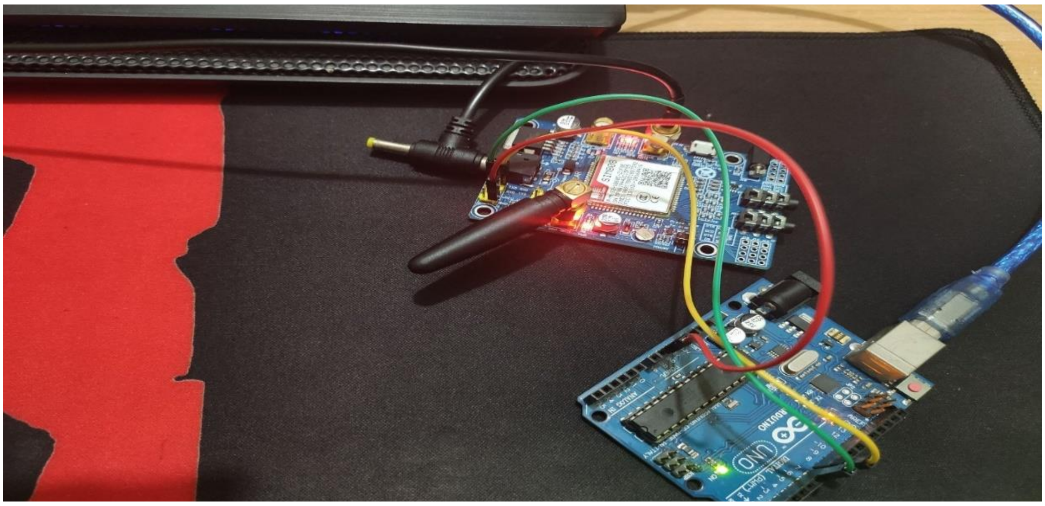


Fig 5.6: Hardware Connection Between Arduino Uno R3 and Sim808 Module

Then we have provided power to the Sim808 module using a power adapter rated 9V - 2A and pressed the switch through which the GPRS module will connect to the network operator that we have provided. We also connected the GPS and GSM antenna to the Sim808 module. The above Fig: 5.6 shows the connection diagram between Arduino UNO and Sim808 Module.

Finally, for the coding purpose, we have used the Arduino IDE software that will operate the Arduino and the Sim808 module. Now, in the Arduino code, we used the “SoftwareSerial” library and also declared the desire phone number that we mainly wanted to notify. After that, we declared different types of AT commands which will help us to get the message along with the location. Such, “AT+CGNSPWR” mainly turned on the GPS power supply. Then, "AT+CGNSSEQ=\RMC\" define the last NMEA sentence that parsed. Again, “AT+CGNSINF” to get the current GPS location, “AT+CMGF” has been applied for choosing the text option for messaging, and also “AT+CMGS” to send the message mainly [56]. After that, we placed three different conditions in our code to determine whether the message sent to the family member would be Anterior, Inferior, or healthy depending on the patient’s health condition.

5.5.2 Connection with Raspberry Pi 4

After completing all the previous setup, we will carry it out to the raspberry pi 4 to merge it with our datasets of different classified versions of MI which is done using python programming.

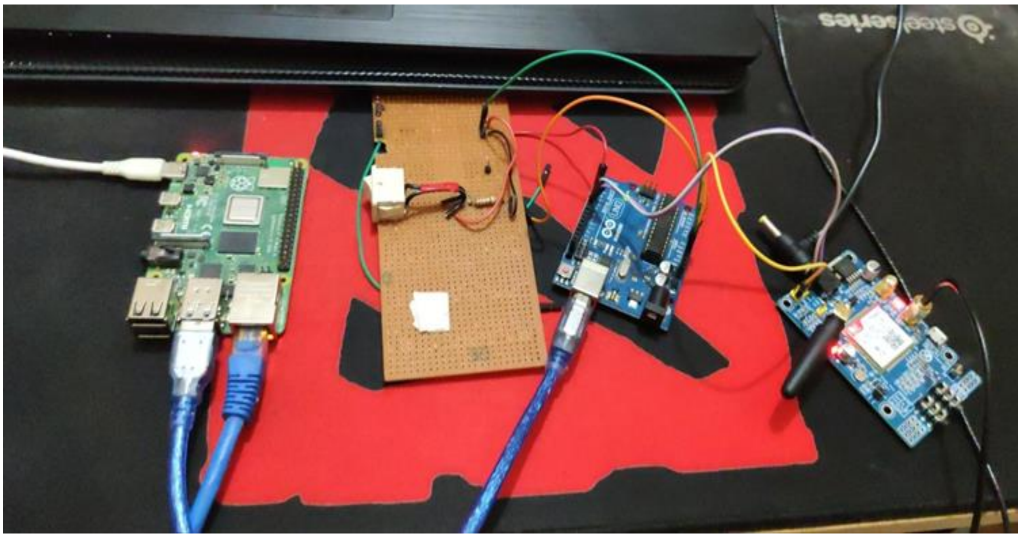


Fig 5.7: Complete Hardware Set-up

For that, we had to set up our Raspberry Pi 4, and to do that we needed a memory card and a card reader.

Here, we used a 32GB memory card, and by using the card reader, we formatted the memory card. Then we installed the latest raspberry pi OS image with desktop and recommended software from the raspberry pi website. After that, we correctly booted the memory card using that OS image. Then, we removed it from the card reader and put it onto the raspberry pi. Now, we can run the raspberry pi by connecting it to a monitor using an HDMI cable. Here, the power supply to the raspberry pi was provided by a 5V adapter of type C cable. Finally, we provided the power supply to the Arduino UNO through our Raspberry Pi. Fig 5.7 shows our complete hardware connection.

5.5.3 Serial Connection of Arduino Uno R3 with Python

To interface our python programming with the Arduino, we are using pyserial, a library used in python programming to create serial communication. Because whenever our machine learning system detects a class, it needs to inform our hardware to act accordingly. So in between python and Arduino, we used a small bridge to send a character assigned to that class. To be precise, whenever our Random Forest classifier detects Anterior, it will send the letter 'A' to the serial connection of our Arduino Uno. Again, the Inferior class will send 'I,' and the Healthy class will send 'H'. As we have said earlier, we are using pyserial to send a signal from our python code to Arduino. Pyserial is driving python code to further execute the program in Arduino and so this works like a bridge between the python code and Arduino Uno.

However, to receive that on the Arduino end, we also have to use a library known as Software Serial, which is a built-in system for the Arduino to support serial connection. The baud rate we are using for this communication is 9600. Whenever the SoftwareSerial detects a character

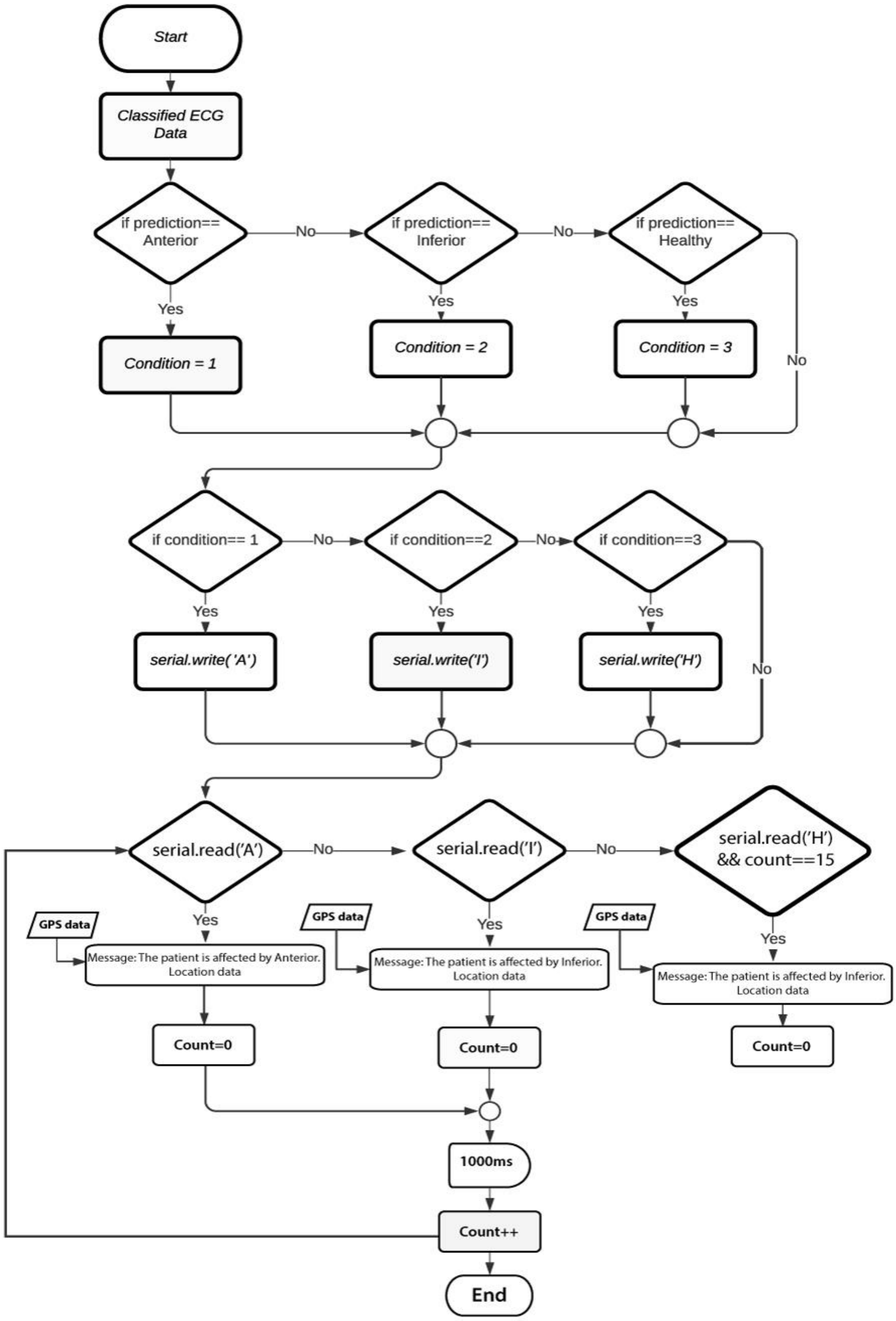


Fig 5.8: Flowchart of Serial Connection Between Python and Arduino

from pyserial it will act according to the execution code assigned to that letter. We have set a delay of 5 seconds in our python code before sending the signal to avoid interrupt with the serial connection of our SIM808 module. The above Fig 5.8, shows a flowchart based on the serial connection between the python code and Arduino.

Chapter 6

Result Analysis

The outcome of the execution and demonstration of this project has been observed and analyzed thoroughly. The individual analysis of software and hardware demonstration is speculated below mentioning each crucial points. Finally, a comparative study has been projected to objectify the efficiency of our prototype.

6.1 Software Result Analysis

Here is the outcome of the detection using four machine learning algorithm and the comparative analysis. Here Correct detection is denoted by “1” and incorrect detection is denoted by “0”.

	Random Forest Classifier		**KNN**	**SVM**	**Naïve Bayes**
Test Data	**n_estimator = 37**	**n_estimator = 43**	**K = 1**		
Anterior_1	1	1	1	1	0
Anterior_2	1	1	0	1	1
Anterior_3	1	1	1	1	0
Anterior_4	0	0	0	1	0
Anterior_5	0	0	0	0	0
Anterior_6	1	0	1	1	0
Anterior_7	1	1	0	1	0
Inferior_1	1	1	1	1	1
Inferior_2	1	1	1	0	0
Inferior_3	1	1	0	0	1

Inferior_4	0	0	0	0	0
Inferior_5	1	1	0	1	1
Inferior_6	1	1	1	0	1
Inferior_7	1	1	1	0	0
Healthy_1	1	1	1	0	0
Healthy_2	1	1	0	0	1
Healthy_3	1	1	0	0	1
Healthy_4	0	0	0	0	0
Healthy_5	1	1	1	0	0
Healthy_6	1	1	1	1	1
Total Correct	16	15	10	9	8

Table 6.1: Test Results for Different Test Data with Separate Classifier

In our project we took 20 sample to go against our train data set one by one and to calculate the accuracy of each algorithm the equation is:

Accuracy [%] = (Total Correct Prediction / Total Test data) * 100 & Total Test data = 20

Algorithm	**Accuracy %**
Random Forest [n_estimator: 37]	80 %
Random Forest [n_estimator: 43]	75 %
K-nearest neighbor [K = 1]	50 %
Support Vector Machine	45 %
Naïve Bayes	40 %

Table 6.2: Accuracy Analysis of Different Algorithms

To have more precise idea about the test's accuracy of multiple classifications we are evaluating the confusion matrix of each classifier and check the F1-score which can be obtained from the precision and recall we can find from the confusion matrix. The Precision: equation 1, Recall: equation 2 and F1 Score: equation 3 [57] are as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive}+\text{False Positive}} \ldots\ldots\ldots(1)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive}+\text{False Negative}} \ldots\ldots\ldots(2)$$

$$\text{F1 Score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \ldots\ldots\ldots..(3)$$

Now, by following the equations of Precision, Recall and F1 Score, we shall formulate confusion matrix for each algorithm and exhibit a comparative analysis of detection of MI.

Now we are going to calculate the corresponding True Positive as 'ATP', True Negative as ATN, False Positive as AFP, False Negative as AFN, Precision as Pre, Recall as Rec and F1 Score as F1S for each case.

6.1.1 Confusion Matrix for Random Forest Classifier (n_estimator 37)

Here, the confusion matrix for n-estimator 37 Random Forest classifier has been shown in Table 6.3 with elaborated calculation and analysis.

Each attribute has been calculated according to the mentioned equations and have been formulated in table for a better understanding purpose.

		Predicted Class		
		Anterior	Inferior	Healthy
Actual Class	Anterior	5	2	0
	Inferior	1	6	0
	Healthy	0	1	5

Table 6.3: Confusion Matrix for MI Detection by Random Forest (n_estimator 37)

Calculation for Anterior:

ATP = 5

ATN = (6+0+1+5) = 12

AFP = (2+0) = 2

AFN = (1+0) = 1

Pre = ATP/(ATP+AFP) = 5/ (5+2) = 0.714

Rec= ATP/(ATP+AFN) = 5/ (5+1) = 0.833

F1S = 2*[(0.833*0.714)/ (0.714+0.833)] = 0.769

Calculation for Inferior:

ATP = 6

ATN = (5+0+0+5) = 10

AFP = (1+0) = 1

AFN = (2+1) = 3

Pre = ATP/(ATP+AFP) = 6/ (6+1) = 0.857

Rec = ATP/(ATP+AFN) = 6/ (6+3) = 0.667

F1S = 2*[(0.857*0.667)/ (0.857+0.667)] = 0.750

<u>Calculation for Healthy:</u>

ATP = 5

ATN = (5+2+1+6) = 14

AFP = (1+0) = 1

AFN = (0+0) = 0

Pre = ATP/(ATP+AFP) = 5/ (5+1) = 0.833

Rec = ATP/(ATP+AFN) = 5/ (5+0) = 1

F1S= 2*[(0.833*1)/ (0.833+1)] = 0.909

Class	Precision	Recall	F1-Score
Anterior	0.714	0.833	0.769
Inferior	0.857	0.667	0.750
Healthy	0.833	1	0.909

Table 6.4: F1-Score of Random Forest Classifier (n_estimator 37)

Therefore, Table 6.4 prospects the precision, recall and F1 Score for Anterior MI, Inferior MI and Healthy patients for n-estimator 37 Random Forest Classifier.

6.1.2 Confusion Matrix for Random Forest Classifier (n_estimator 43)

Here, the confusion matrix for n-estimator 43 Random Forest classifier has been shown in Table 6.5 with elaborated calculation and analysis.

		Predicted Class		
		Anterior	Inferior	Healthy
Actual Class	Anterior	4	3	0
	Inferior	1	6	0
	Healthy	0	1	5

Table 6.5: Confusion Matrix for MI Detection by Random Forest (n_estimator 43)

<u>Calculation for Anterior:</u>

ATP = 4

ATN = (6+0+1+5) = 12

AFP = (3+0) = 3

AFN = (1+0) = 1

Pre = ATP/(ATP+AFP) = 4/ (4+3) = 0.571

Rec = ATP/(ATP+AFN) = 4/ (4+1) = 0.8

F1 Score = 2*[(0.571*0.8)/ (0.571+0.8)] = 0.666

Calculation for Inferior:

ATP = 6

ATN = (4+0+0+5) = 9

AFP = (1+0) = 1

AFN = (3+1) = 4

Pre = ATP/(ATP+AFP) = 6/ (6+1) = 0.857

Rec = ATP/(ATP+AFN) = 6/ (6+4) = 0.6

F1S = 2*[(0.857*0.6)/ (0.857+0.6)] = 0.706

Calculation for Healthy:

ATP = 5

ATN = (4+1+6+3) = 14

AFP = (1+0) = 1

AFN = (0+0) = 0

Pre = ATP/(ATP+AFP) = 5/ (5+1) = 0.833

Rec= ATP/(ATP+AFN) = 5/ (5+0) = 1

F1 Score = 2*[(0.833*1)/ (0.833+1)] = 0.909

Class	Precision	Recall	F1-Score
Anterior	0.571	0.8	0.666
Inferior	0.857	0.6	0.706
Healthy	0.833	1	0.909

Table 6.6: F1-Score of Random Forest Classifier (n_estimator 43)

Hence, Table 6.6 prospects the precision, recall and F1 Score for Anterior MI, Inferior MI and Healthy patients for n-estimator 43 Random Forest Classifier.

6.1.3 Confusion Matrix for K-Nearest Neighbor (K = 1)

Here, the confusion matrix for K-Nearest Neighbor due to the case K=1 has been shown in Table 6.7 with elaborated calculation and analysis.

		Predicted Class		
		Anterior	Inferior	Healthy
Actual Class	Anterior	3	2	2
	Inferior	2	4	1
	Healthy	1	2	3

Table 6.7: Confusion Matrix for MI Detection by K-Nearest Neighbor

<u>Calculation for Anterior:</u>

ATP = 3

ATN = (4+2+1+3) = 10

AFP = (2+2) = 4

AFN = (2+1) = 3

Pre = ATP/(ATP+AFP) = 3/ (3+4) = 0.429

Rec = ATP/(ATP+AFN) = 3/ (3+3) = 0.5

F1S= 2*[(0.429*0.5)/ (0.429+0.5)] = 0.462

<u>**Calculation for Inferior:**</u>

ATP = 4

ATN = (3+1+2+3) = 9

AFP = (2+1) = 3

AFN = (2+2) = 4

Pre = ATP/(ATP+AFP) = 4/ (4+3) = 0.571

Rec = ATP/(ATP+AFN) = 4/ (4+4) = 0.5

F1S = 2*[(0.571*0.5)/ (0.571+0.5)] = 0.533

<u>**Calculation for Healthy:**</u>

ATP = 3

ATN = (3+2+4+2) = 11

AFP = (2+1) = 3

AFN = (2+1) = 3

Pre = ATP/(ATP+AFP) = 3/ (3+3) = 0.5

Rec = ATP/(ATP+AFN) = 3/ (3+3) = 0.5

F1S = 2*[(0.5*0.5)/ (0.5+0.5)] = 0.5

Class	Precision	Recall	F1-Score
Anterior	0.429	0.5	0.462
Inferior	0.571	0.2	0.533
Healthy	0.5	0.5	0.5

Table 6.8: F1-Score of K-Nearest Neighbor (K=1)

So, Table 6.8 prospects the precision, recall and F1 Score for Anterior MI, Inferior MI and Healthy patients for K-nearest neighbor where K=1.

6.1.4 Confusion Matrix for Support Vector Machine

Here, the confusion matrix for Support Vector Machine has been shown in Table 6.9 with elaborated calculation and analysis.

		Predicted Class		
		Anterior	Inferior	Healthy
Actual Class	Anterior	6	1	0
	Inferior	4	2	1
	Healthy	4	1	1

Table 6.9: Confusion Matrix for MI Detection by Support Vector Machine

Calculation for Anterior:

ATP = 6

ATN = (2+1+1+1) = 5

AFP = (1+0) = 1

AFN = (4+4) = 8

Pre = ATP/(ATP+AFP) = 6/ (6+1) = 0.857

Rec = ATP/(ATP+AFN) = 6/ (6+8) = 0.429

F1S = 2*[(0.857*0.429)/ (0.857+0.429)] = 0.572

Calculation for Inferior:

ATP = 2

ATN = (6+4+0+1) = 11

AFP = (4+1) = 5

AFN = (1+1) = 2

Pre = ATP/(ATP+AFP) = 2/ (2+5) = 0.286

Rec= ATP/(ATP+AFN) = 2/ (2+2) = 0.5

F1S= s2*[(0.286*0.5)/ (0.286+0.5)] = 0.364

Calculation for Healthy:

ATP = 1

ATN = (6+4+1+2) = 13

AFP = (4+1) = 5

AFN = (0+1) = 1

Pre = ATP/(ATP+AFP) = 1/ (1+5) = 0.167

Rec= ATP/(ATP+AFN) = 1/ (1+1) = 0.5

F1S = 2*[(0.167*0.5)/ (0.167+0.5)] = 0.250

Class	Precision	Recall	F1-Score
Anterior	0.857	0.429	0.572
Inferior	0.286	0.5	0.364
Healthy	0.167	0.5	0.250

Table 6.10: F1-Score of Support Vector Machine

Here, Table 6.10 prospects the precision, recall and F1 Score for Anterior MI, Inferior MI and Healthy patients for Support Vector Machine.

6.1.5 Confusion Matrix for Naive Bayes

Here, the confusion matrix for Naive Bayes has been shown in Table 6.11 with elaborated calculation and analysis.

Each attribute has been calculated following the above-mentioned equations and have been projected in table for a better understanding purpose.

		Predicted Class		
		Anterior	Inferior	Healthy
Actual Class	Anterior	1	4	2
	Inferior	0	4	3
	Healthy	2	1	3

Table 6.11: Confusion Matrix for MI Detection by Naive Bayes

Calculation for Anterior:

ATP = 1

ATN = (4+1+3+3) = 11

AFP = (4+2) = 6

AFN = (0+2) = 2

Pre = ATP/(ATP+AFP) = 1/ (1+6) = 0.143

Rec = ATP/(ATP+AFN) = 1/ (1+2) = 0.333

F1S = 2*[(0.143*0.333)/ (0.143+0.333)] = 0.20

Calculation for Inferior:

ATP = 4

ATN = (1+2+2+3) =8

AFP = (0+3) = 3

AFN = (4+1) = 5

Pre = ATP/(ATP+AFP) = 4/ (4+3) = 0.571

Rec = ATP/(ATP+AFN) = 4/ (4+5) = 0.444

F1 Score = 2*[(0.571*0.444)/ (0.571+0.444)] = 0.5

<u>Calculation for Healthy:</u>

ATP = 3

ATN = (1+0+4+4) = 9

AFP = (1+2) = 3

AFN = (3+2) = 5

Pre = ATP/(ATP+AFP) = 3/ (3+3) = 0.5

Rec= ATP/(ATP+AFN) = 3/ (3+5) = 0.375

F1S = 2*[(0.5*0.375)/ (0.5+0.375)] = 0.436

Class	Precision	Recall	F1-Score
Anterior	0.143	0.333	0.20
Inferior	0.571	0.444	0.5
Healthy	0.5	0.375	0.436

Table 6.12: F1-Score of Naive Bayes

Thus, Table 6.12 prospects the precision, recall and F1 Score for Anterior MI, Inferior MI and Healthy patients for Naïve Bayes.

6.1.6 Comparative Result Analysis of Different Classifiers

From the F1 scores of all the algorithms it is easier to select which algorithm we should apply to our hardware to avoid any false alarm and accurate result. In terms of selecting Anterior Class the Random Forest algorithm with 37 n estimator provides the best F1 score in Table 6.3 which is 0.769 and K-nearest neighbor have the lowest F1-score in Table 6.5. For the Inferior and Healthy class in Table 6.3 Random forest classifier has the best F1-score which is consecutively 0.750 and 0.909. As we can see the f1-score of healthy class using Random forest is more than 0.9 it will avoid any false alarm of Myocardial Infarction.

Moreover, from the accuracy analysis from Table 6.2, the visualization clearly shows the differences of efficiency of the system. Here Fig 6.1 is the visual representation by histogram of the comparative accuracy analysis between the detection procedure using Random Forest, KNN, SVM and Naïve Bayes Machine Learning algorithms.

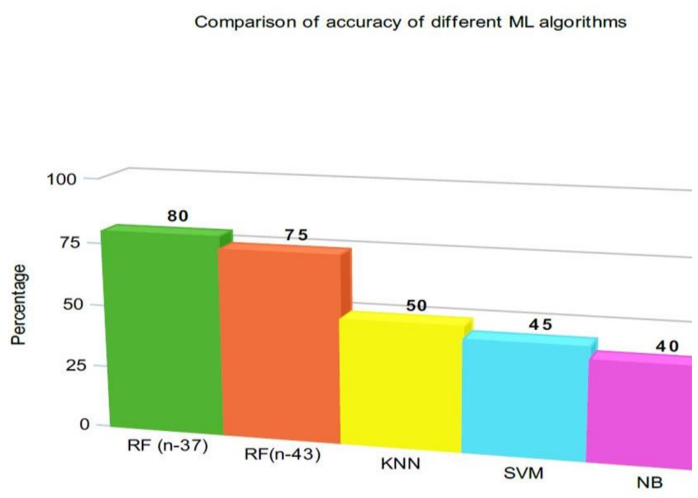


Fig 6.1: A Histogram Comparing the Accuracy of Different Machine Learning Algorithms

Therefore, after examining F1-scores of all four algorithms we can conclude that for our system and dataset Random Forest Classifier is the best algorithm to choose.

6.2 Hardware Result Analysis

In this hardware result analysis part, we will see the demonstration result like what type of messages we will receive in case of Anterior, Inferior and Healthy detection along with the Google map location and direction of the patient. Moreover, we will also figure out some delay analysis between all these messages that are received.

6.2.1 Display of Demonstration Result

So, in our system GPS receives the data for the location in the latitude and longitude values depending on the patient's current location and send the location along with a message to the family member through the GSM system. Here, for this GPS and GSM purpose we used the sim808 module which mainly operates based on the different AT command in the Arduino program. Like, to give the GPS power supply we use the AT+CGNSPWR command and also to get the current location data we used the AT+CGNSINF command.

Then, after getting the GPS location we have to send this data with a message on patient current condition. In this purpose we also used the sim808 module and sent the message to the family member using a sim card and setting different AT command in the Arduino to get the message in the desired phone number. Like, we used AT+CMGF command to select the text mode for message and also AT+CMGS command to send the message to the family members phone number. Again, we set the code in a way that the latitude and longitude value will go as a hyperlink in the Google map. So, just clicking into the hyperlink the family member can easily see the patient's current location through Google map.

Now, according to our prototype, we are supposed to send three different types of messages along with the Google map location depending on the patient's health condition. These messages are Anterior, Inferior, and Healthy. So, in our case the message will be sent depending on the dataset test result in the python program whether it detects Anterior, Inferior or Healthy. Moreover, a position indicating hyperlink is also attached with these messages to opportune the family member to be able to easily find the patient's current location displayed from Google map. Now, we will see the three possible types of messages that we can receive through our system.

Here, Fig 6.2 displays the format of notification text that will be sent after a person gets affected by Anterior Myocardial Infarction with the corresponding location of the patient. After clicking the link, the user will be able to see the direction to the patient's location as well.

If the patient is affected by Anterior MI, the message will be like below in Fig 6.2:

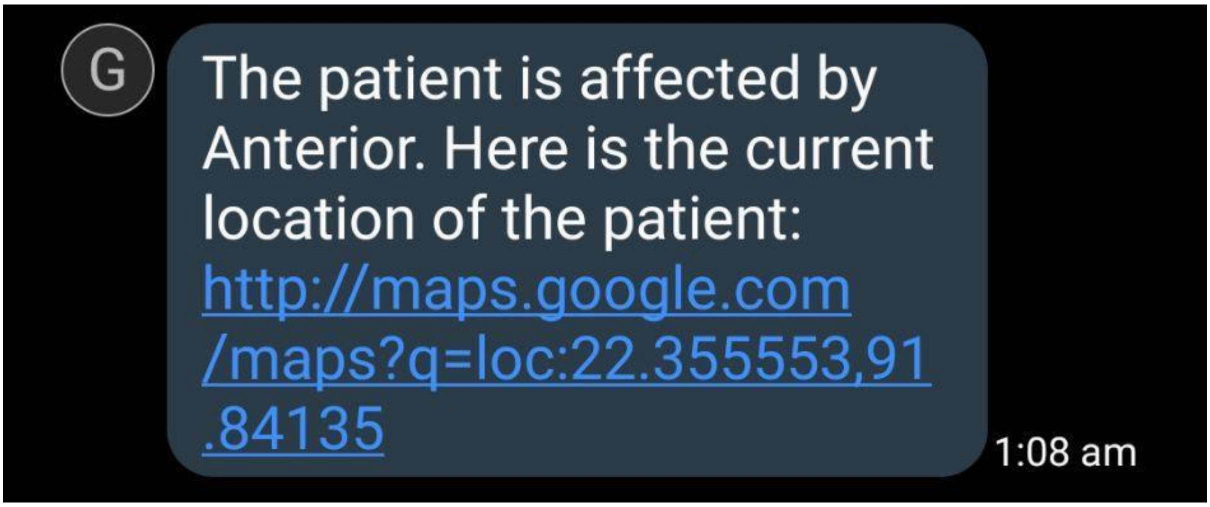


Fig 6.2:Notifying Message for Anterior MI Affected Patient

Similarly, the notifying text indicating that a person is affected by Inferior Myocardial Infarction with corresponding location has been illustrated in Fig 6.3. So, the user will be able to track the position of the patient immediately by going through the link.

If the patient is affected by Inferior MI, the message will be like below in Fig 6.3:

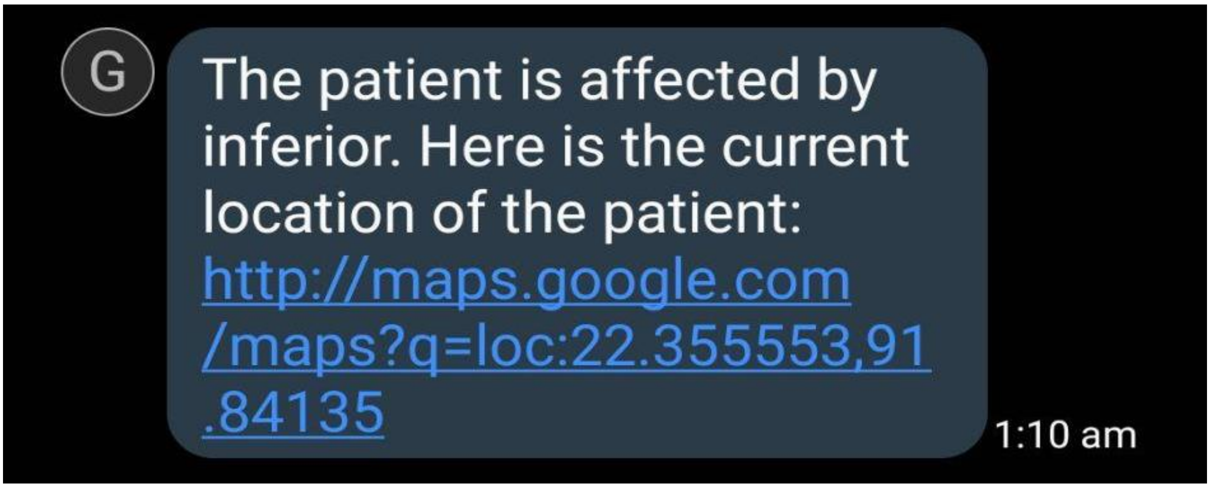


Fig 6.3 Notifying Message for Inferior MI Affected Patient

So, these are the notifying text format which will aware the MI affected patient's nearest kin mentioning the type of Myocardial Infarction. This classification will not only avail the ease of patient but also assist the acting physician while diagnosing the type of heart attack. Hence, this will reduce the time of treatment procedure.

However, in case the patient is healthy the system will send a text which has been shown below.

If the patient is Healthy, the message will be like below in Fig 6.4:

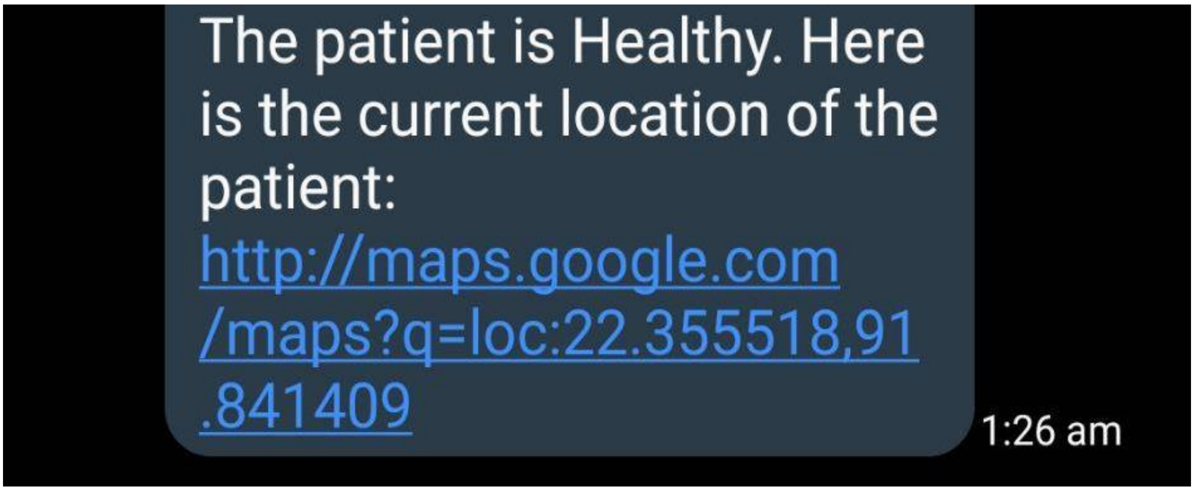
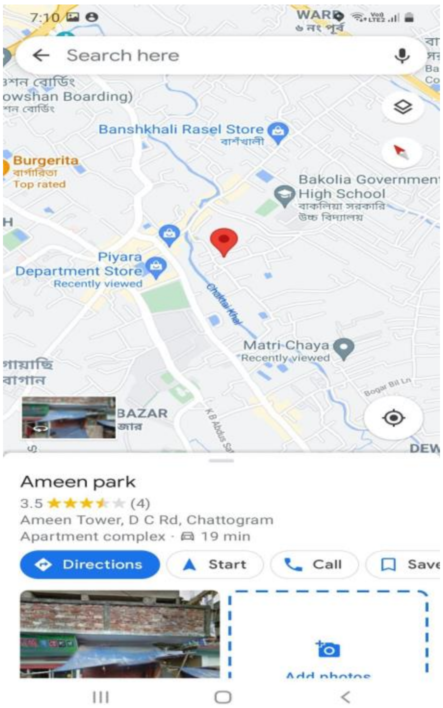


Fig 6.4: Notifying Message for Healthy Person

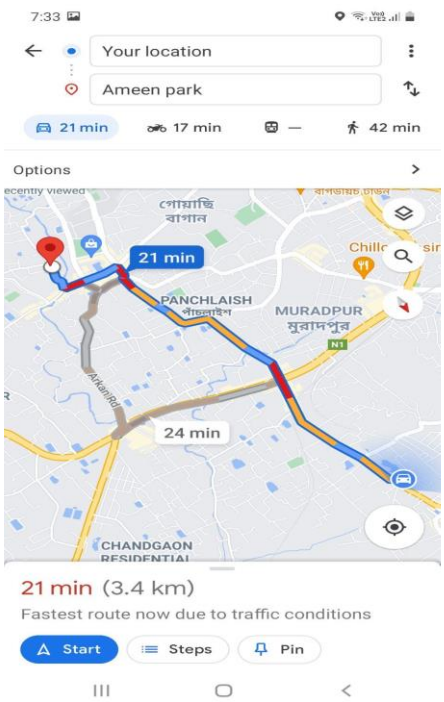
So, as we can see from these figures, we can receive these three types of messages with a Google map location link depending on the patient's health condition.

6.2.2 Visualization of Location and Direction

Now, if we click on one of these links from the messages, we can see the sender's (the patient's) current location through Google map just like the below Fig 6.5(a) shows. The link will display the prime location of the patient and corresponding direction from the receiver to the patient. After that, using the Google map, the receiver (family member) will be able to find out the direction of the location to reach the patient immediately. Fig 6.5(b) shows a direction between the patient and the receiver (family member). Thus this forewarning system avails the opportunity to reduce the death rate due to the inability to take immediate action after getting affected by Myocardial Infarction or Heart Attack. The joint venture of Global Positioning System (GPS) and Global System for Mobile Communication (GSM) have been done by GSRM and the forewarning system is driven by this execution.



a. Location



b. Direction

Fig 6.5: Google Map Showing a. Location and b. Direction

Hence, the forewarning system displaying current location of the MI affected patient via text message will minimize the causalities due to heart attack. The forewarning system is quite efficient as it operates with a minimal delay which is another crucial factor to reduce causalities.

6.2.3 Delay Analysis

At the time of testing, we also observed the delay analysis. Here we observed the delay time that we set in the code for our python and Arduino interfacing will actually matches or not with the delay we receive between messages during testing the system. So, for that different sets of timing data are recorded to complete the delay analysis.

However, both the inherent delay and technical delay of the system have been analyzed thoroughly in the following with proper reasoning and evidence.

6.2.3.1 Inherent Delay in Python and Arduino Uno R3 Interfacing

In our project we have send a signal to Arduino as a recognition of types of ECG detected by our machine learning algorithm. However, we have observed that when we send the signal from python with a 0.2 second interval most of the time at the start of simulation our GPS stops initialization. However, when we tried to initialize the SIm808 with Arduino code it works fine. After analyzing the whole situation, we have introduced 40 second delay for sending signal from our python code that the GPS initialization issue solved. We have introduced 40 second as we have a 1-minute cycle in our Arduino loop we wanted the python signal to go in before a single loop occurs. After observing the issue is solved now, we started to reduce the python signal input interval to 30 second, 20 second, 10 and 5 second. However, when we reduce the delay less than 5 second the GPS initialization issue occurs again. Therefore, we have introduced the lowest delay which is 5 second.

6.2.3.2 Technical Delay Between Receiving Messages

In our project, while demonstrating the hardware, we have monitored the delay between the message sent and how fast they've reacted and if the delay caused is the normal delay or not. During the execution of the project, we have noticed and observed that there are some technical

Step No	**Step Name**	**Stopwatch Timer**	**Time Difference with previous step**
		mm:ss:ms	**mm:ss:ms**
1	System Start	00:07:45	00:07:45
2	Anterior Signal Input	00:49:53	00.42.08
3	Anterior Message Received	01:56:20	01:06:67
4	Inferior Input	02:43:07	00:46:87
5	Inferior Message Received	03:09:58	00:26:51
6	System Paused	4:01:17	3:34:26

Table 6.13: Timer and Delay Table 1

delays due to the delay in execution of both hardware and software parts. So, we have performed an elaborate analysis on this issue and notified the crucial points explicitly. The crucial factors behind the technical delays have been explained properly with appropriate causes and results.

In Table 6.13, at first, we started the signal at 7.45 seconds, and without any serial provided from the Raspberry Pi 4, the system will not send any message. We have simulated an anterior signal after 42.08 seconds, so now the system has a serial signal from the Raspberry Pi 4. In the Arduino, there is a 1:03:00 minute delay introduced. So in each 1:03:00 minute, a message will be sent except when the condition is healthy, in that case, the delay is 15:45:00 minute. So after 1:06:67 minute, the Anterior signal fed into the hardware a message with the anterior condition and the location received in our mobile phone. After 46.87 sec, we have tested an interior signal which is detected by the Machine Learning system and sends a serial signal to our hardware after 1:13:38 minute [In Table 6.14] from the first message, we have received another message. This time with Inferior condition instead of Anterior.

Delay Between Inferior and Anterior Message	01:13:38

Table 6.14: Timer and Delay Table 2

We can see that there has been a delay of approximately 10 seconds. While working with a similar kind of system like us, many of the users also faced the same situation. They also have raised the question regarding the delay which was answered by the experts working on the similar field. According to the experts in [58], we can find out that the cause of the delay is actually similar to the delay that we face on a daily basis while texting on the mobile phone. However, from [59] we have gathered some of the causes of text delay in mobile phones that regular users face. The reasons are:

- The location of the sender or the receiver might not be suitable for example in the urban areas, the transmission requires time due to tall buildings.

- Communicating between different networks is also a reason for the delay because different networks have different communicating ranges that depend on the availability.

- Network Traffic can also be reason because at a time several networks are working that creates signal overlapping and disturbance and delivery

After that, we turned on the pause switch and waited for 14 minutes to see whether there were any kind of messages. So, as we did not receive any messages, we turned off the pause switch after 14 minutes, and then in Table 6.15 we can see the delays. Before turning off the pause switch, we simulated a Healthy signal to send the serial signal in our Arduino for a healthy text.

Step No	**Step Name**	**Stopwatch Timer**	**Time Difference with previous step**
		mm:ss:ms	**mm:ss:ms**
1	System Pause off	18:02:45	14:01:28
2	Healthy Message 1	20:08:32	02:05:47
3	Healthy Message 2	22:16:44	02:18:12

Table 6.15: Timer and Delay Table After Pause

Here Table 6.16 clarifies the delay between the incoming time of two messages which indicates that the patient is healthy.

Delay Between two Healthy Message	02:18:12

Table 6.16: Delay Between Two Notifying Messages of Healthy Person

Here, to make our demonstration purpose and video easy, we used a delay of just 2 minutes between two healthy messages. But, in our prototype, we will use a delay of 15 minutes between two healthy messages so that we can continuously monitor the patient condition in every 15 minutes whether he/she is healthy or there is any kind of abnormality.

Now, the visual illustration of the two text messages indicating the healthy person with 15 minutes' interval which is essentially programed to be working as a cycled process is shown below with receiving times.

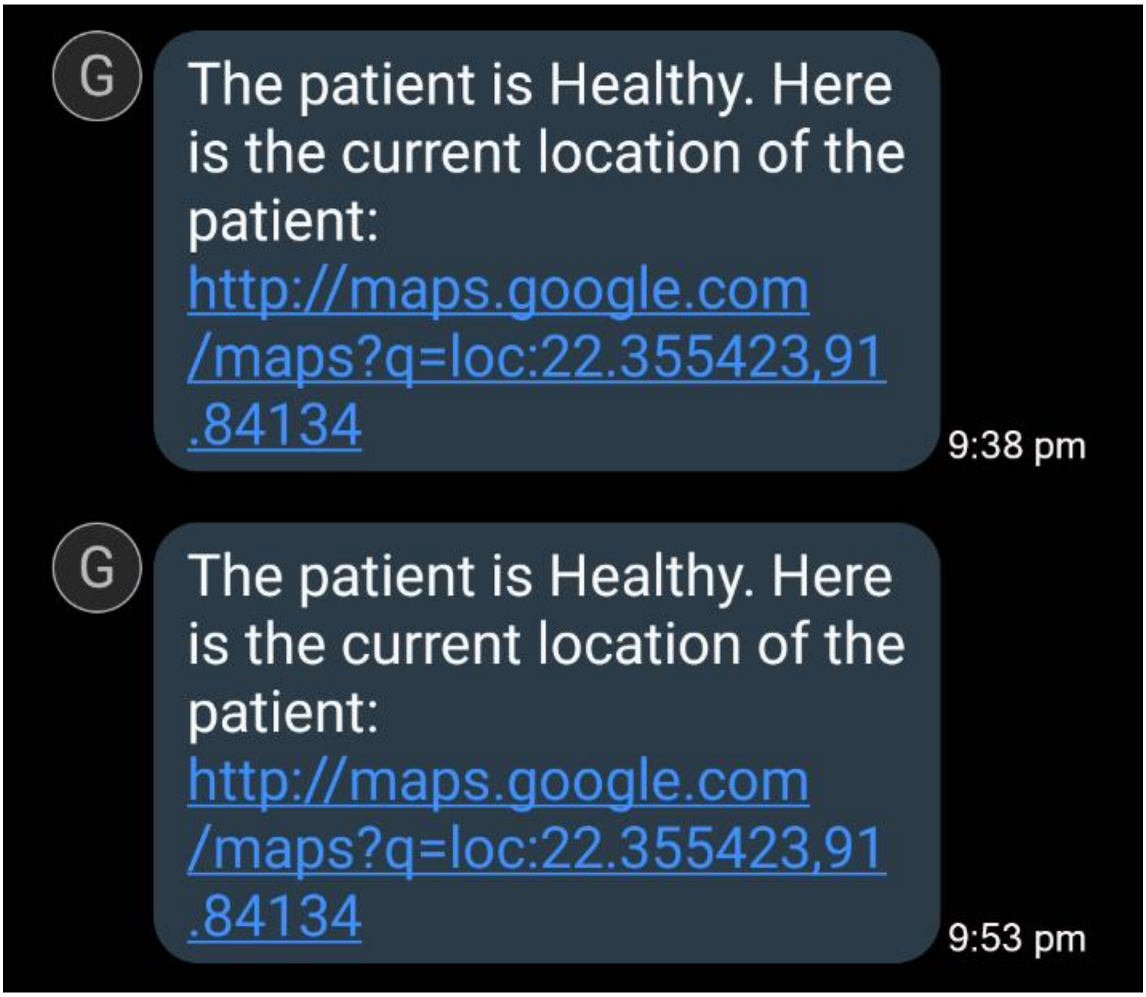


Fig 6.6: Cycled Interval of 15 Minutes between Two Notifying Messages for Healthy Person

So, as we can see from this above Fig 6.6 there is a clear 15 minutes of interval between these messages as the first one, we received at 9.38 pm and the second one at 9.53 pm. In this way the continuous record of the patient in healthy condition can be observed and the family members will be assured of the patient's wellbeing.

Chapter 7

Sustainability and Durability Evaluation

Apparently, the prototype contains the detection of Myocardial Infarction and generates a forewarning system which accumulates a greater purpose for humankind. Hence, the sustainability and durability analysis are a significant discussion for this project.

7.1 Environmental Sustainability and Impacts

The components that we have used in this project are mostly environment friendly and do not cause any harm to the animal ecosystem. Moreover, most of the parts are biodegradable and reusable which can be used for multipurpose. So, if the components are preserved properly, they can be used later in another project. However, as the objective of Health Care Without Harm(HCWH) organization is to ensure the reduction of environmental degradation caused by any medical device, we have been extremely careful while executing the operation [60]. As mentioned before, all the parts are reusable, the prototype does not cause any potential harm to the environment.

7.2 Economic Sustainability and Impacts

The objective of this project was to implement a device which is functional for humankind directly connected to health. Heart being the most crucial part of the human body, requires a proper maintenance as well and our prototype is applicable for the maintenance of hear. Despite being in the preliminary level, the prototype is cost effective since the overall costing is so far around ten thousand Taka. Nevertheless, if the project is commercialized in future by installing an embedded system it will be more affordable for people regardless of economic solvency. The intention of implementing this project directly relates to those kinds of people who have primary difficulties in heart or prone to heart attack suggested by particular physicians. Therefore, if we put the verdict with respect to individual families, this device will be affordable for the people of developing countries like ours.

Hence, the fundamental concerns about developing a medical diagnosing device are directly related to some significant factors like waste management, reusability, recycling process and awareness [61]. Accordingly, the prototype that we have built follows the generalized criteria as each equipment can be easily reusable and recycled. The embedded system can be used again and again depending upon the maintenance of the user.

7.3 Social Sustainability and Impacts

Death rate is a crucial yardstick of measuring the standard of health of any society. Our society is an integration of numerous families accommodating family members with different names and roles. From the nuclear point of view, each member of the family is important to each other and the demise of any of them causes adverse effects on each family. From the beginning, we have been very much focused on implementing something that can minimize the death rate of the human race. Moreover, this device can be reusable and if possible they can be donated to the less opportune countries and proper awareness is extremely necessary before using such devices which are directly connected to human body [61]. Heart attack being one of the crucial reasons for millions of deaths every year, we have focused on implementing a prototype being able to decrease the death rate due to heart attack at least to some extent.

Besides working on the prototype, the implementation process requires educative campaigns before exposing the usage to mass people [61]. In order to ensure that we are going to run awareness programs and organize tutorials so that general people can use this device without any difficulty.

7.4 Ethical Persistency and Risk Assessment

It has been clinically advised that any device which is directly or indirectly related to health and is applied in medical purpose accommodates seven individual steps. However, according to that concept, our project is at the very initial step and that involves the procedure of developing the concept and preliminary implementation. At this level, it is necessary to demonstrate the project thoroughly and clinically multiple times and observe the variables like accuracy, verification, error calculations etc. Moreover, the trials should be engineered in such a way that the procedure will execute without causing any minimal harm to any living being [62]. By keeping this in mind, we have demonstrated the software and hardware part individually multiple times and many times after incorporation as well. Since, the project is at development level, we have not applied any execution on any living being and that ensures the overall health ethics at this level. Accordingly, since the ethical issues have been maintained so far, our project did not cause any risk to the living ecosystem.

In Bangladesh, Directorate General of Drug Administration (DGDA) is the responsible authority for approving any medical device before applying on human body or even manufacturing it [63]. Therefore, before applying this prototype on any living being for trial procedure, we are going to take permission from the responsible authority to preserve the ethical concerns regarding this project.

Chapter 8

Conclusion and Future Work

Finally, here is the brief of all the operations mentioned above and the future schemes that we aspire to fulfill by continuing this exertion. This endeavor has been a meticulous learning expedition where we explored the difficulties in both hardware and software demonstrations and overcame them eventually.

8.1 Summary

The incorporation of the software portion with hardware portion of this project is the crucial accomplishment of this prototype. Detecting Myocardial Infarction from ECG signal and generating a forewarning system at the same time including the immediate location of the affected person are the milestones of this research endeavor. Hence, the particular objective of this project was to implement such a device that might be able to assist human beings in health purposes. By keeping that objective in front, we have developed this prototype after one year of hard work and dedication. Nevertheless, the genre of this project focuses on developing an application based system which will be able to work efficiently in real life as well. This prototype has accomplished that criteria and it will be able to assist people who have the tendency of affecting MI or those who want to be on regular virtual checkup too. In brief, this project will assist in reducing the death rate due to Myocardial Infarction and establish a healthy balanced lifestyle.

8.2 Future Work

Since the project is completed as a prototype at present, we are looking forward to improving some aspects to increase the efficiency. Initially, we shall focus on increasing the accuracy level of the detection procedure. Increasing accuracy involves the improvement in preprocessing of the dataset. For instance, we intend to work with noise and without noise ECG signals to increase the variety of our training dataset and standardize both of them applying R peak detection. Then we shall execute z-score normalization to minimize the issues in scaling the amplitudes of the ECG signals and diminish the effects of offset [19]. This will make the system more efficient for multiple ECG signals and the detection procedure will be amplified eventually. Since our project is able to show only the accuracy level of detection, we would like to introduce more regulatory outcome based observation like sensitivity and specificity. These observations will make the detection procedure more meticulous for both patients and physicians. Nevertheless, the accuracy level of our project is 80% currently and undoubtedly we would like to introduce some developments here too. Basically, we are going to apply a deep Convolutional Neural Network for the detecting procedure since some studies have shown a significant amount of accuracy such as 93%+ [19].

However, apart from the investigative developments, we aim to evolve some commercialized developments as well. For instance, we plan to add more features like a heartbeat monitoring system which will make the project more efficient for regular purposes. After that, to establish the portability of this prototype we shall make an embedded system including an ECG sensor so that anyone can easily carry the device comfortably. Then we intend to build a mobile App which will accustom the forewarning system and will be able to accommodate more features. Hopefully, the success in the above mentioned points may lead this project to be commercialized and hence help mankind.

References

[1] O. Mechanic, M. Gavin and S. Grossman, "Acute Myocardial Infarction", *Ncbi.nlm.nih.gov*, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK459269/. [Accessed: 19- Sep- 2021].

[2] "Acute Myocardial Infarction: Causes, Symptoms, and Treatment", *Healthline*. [Online]. Available: https://www.healthline.com/health/acute-myocardial-infarction#causes. [Accessed: 19- Sep- 2021].

[3] K. Bansal, M. Gore and P. Nalabothu, "Anterior Myocardial Infarction", *Ncbi.nlm.nih.gov*, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK562234/. [Accessed: 19- Sep- 2021].

[4] I. Kusama, K. Hibi, M. Kosuge, N. Nozawa, H. Ozaki, H. Yano, S. Sumita, K. Tsukahara, J. Okuda, T. Ebina, S. Umemura and K. Kemura, "Impact of Plaque Rupture on Infarct Size in ST-Segment Elevation Anterior Acute Myocardial Infarction", *Journal of the American College of Cardiology*, vol. 50, no. 13, pp. 1230-1237, 2007.

[5] W. MJ and T. VS, "Inferior Myocardial Infarction", *PubMed*, 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29262146/. [Accessed: 19- Sep- 2021].

[6] W. Liu, M. Jhang, Y. Zhang, Y. Liao, Q. Huang, S. Chang, H. Wang and J. He,"Real-Time Multilead Convolutional Neural Network for Myocardial Infarction Detection," in *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1434-1444, Sept. 2018.

[7] A. Diker, Z. Cömert, E. Avci and S. Velappan, "Intelligent system based on Genetic Algorithm and support vector machine for detection of myocardial infarction from ECG signals," *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 2018.

[8] N.A. Bhaskar, "Performance Analysis of Support Vector Machine and Neural Networks in Detection of Myocardial Infarction," *Procedia Computer Science*, vol. 26, pp. 20-30, 2015.

[9] D. Shadhukhan, S. Pal and M. Mitra, "Automated Identification of Myocardial Infarction Using Harmonic Phase Distribution Pattern of ECG Data," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no, 10, pp. 2303-2313, 2018.

[10] U.R. Acharya, H. Fujita, V.K. Sudarshan, S.L. Oh, M. Adam, J.E.W. Koh, J.H. Tan, D.N. Ghista, R.J. Martis, C.K. Chua, C.K. Poo, R.S. Tan, "Automated detection and localization of myocardial infarction using electrocardiogram: a comparative study of different leads," *Knowledge-Based Systems*, vol. 99, pp. 146-156, 2016.

[11] Mohit Kumar, R.B. Pachori, U.R. Acharya, "Automated Diagnosis of Myocardial Infarction ECG Signals Using Sample Entropy in Flexible Analytic Wavelet Transform Framework," *Entropy* 19, 2017.

[12] L. Sun, Y. Lu, K. Yang, S. Li, "ECG analysis using multiple instance learning for myocardial infarction detection," *IEEE transactions on bio-medical engineering*, vol. 59, pp. 2248-3356, 2012.

[13] U.R. Acharya, H. Fujita, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals", *Information Sciences*, 2017.

[14] K. M. Dewarshi and J. R. Panchal, “Heart Attack Warning System Using ECG Processing,” *International Journal of Advances in Electronics and Computer Sciences*, vol. 2, no. 6, June, 2015.

[15] N. Patel, P. Patel and N. Patel, “Heart Attack Detection and Heart Rate Monitoring Using IoT,” *International Journal of Innovations & Advancement in Computer Sciences*, vol. 7, no. 4, April, 2018.

[16] P. Sihombing, M. Manullang, D. Sitompul and I. S. Dumayanti, “The Heart Attack Detection by ESP8266 Data Communication at a Real Time to Avoid Sudden Death,” *the 3rd International Conference on Computing and Applied Informatics*, 2018.

[17] J. P. Bassand, C. W. Hamm, D. Ardissino, "Guidelines for the diagnosis and treatment of non-ST-segment elevation acute coronary syndromes: the Task Force for the Diagnosis and Treatment of Non-ST-Segment Elevation Acute Coronary Syndromes of the European Society of Cardiology," *European Heart Journal*, vol. 28, no. 13, pp. 1598-1660, 2007.

[18] "Cardiovascular Diseases (CVDs)," *Who.int*, 2021. [Online] Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). [Accessed: 22- Sep- 2021].

[19] U. Rajendra Acharya , Hamido Fujita , Oh Shu Lih , Yuki Hagiwara , Jen Hong Tan and Muhammad Adam , "Application of Deep Convolutional Neural Network for Automated Detection of Myocardial Infarction Using ECG Signals", *Information Sciences*, 2017.

[20] D. L. Kulick, "Heart Attack Myocardial Infarction," *Medicinenet.com*, 2020. [Online]. Available: https://www.medicinenet.com/heart_attack/article.htm. [Accessed: 19- Sep- 2021].

[21] Z. N. Hatmi, S. Tahvildari, M. A.Gafarzadeh, K. A. Sabouri, "Prevalence of coronary artery disease risk factors in Iran: a population based survey," *BMC Cardiovasc Disord*, vol. 7, no. 32, 2007.

[22] "Introduction to Random Forest in Machine Learning", *Engineering Education (EngEd) Program | Section*, 2021. [Online]. Available: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/. [Accessed: 22- Sep- 2021].

[23] "K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint", *www.javatpoint.com*, 2021. [Online]. Available: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning. [Accessed: 22- Sep- 2021].

[24] A. Navlani, "KNN Classification using Scikit-learn," 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn. [Accessed: 22- Sep- 2021].

[25] S. Ray, "6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R" *Analytics Vidhya*, 2017. [Online]. [Accessed: 22- Sep- 2021].

[26] "1.9. Naive Bayes — scikit-learn 0.24.2 documentation", *Scikit-learn.org*, 2027-20. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html. [Accessed: 22- Sep- 2021].

[27] A. Navlani, "Support Vector Machines with Scikit Learn," 2019. [Online]. Available: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python. [Accessed: 22- Sep- 2021].

[28] "Support Vector Machine (SVM) Algorithm - Javatpoint", *www.javatpoint.com*. [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm. [Accessed: 22- Sep- 2021].

[29] "Google Colab", *Research.google.com*. [Online]. Available: https://research.google.com/colaboratory/faq.html. [Accessed: 22- Sep- 2021].

[30] "Welcome to Spyder’s Documentation — Spyder 5 documentation", *Docs.spyder-ide.org*. [Online]. Available: https://docs.spyder-ide.org/current/index.html. [Accessed: 21- Sep- 2021].

[31] "Arduino IDE 1.8.16", *Arduino.cc*. [Online]. Available: https://www.arduino.cc/en/software. [Accessed: 21- Sep- 2021].

[32] "Arduino - ReleaseNotes", *Arduino.cc*. [Online]. Available: https://www.arduino.cc/en/Main/ReleaseNotes. [Accessed: 21- Sep- 2021].

[33] "Arduino UNO R3 Microcontroller, Specifications, and Pin Diagram", *ElProCus - Electronic Projects for Engineering Students*. [Online]. Available: https://www.elprocus.com/what-is-arduino-uno-r3-pin-diagram-specification-and-applications/. [Accessed: 22- Sep- 2021].

[34] "SIM808 GSM&GPRS + GPS Module", *Seeedstudio.com*. [Online]. Available: https://www.seeedstudio.com/SIM808-GSM-GPRS-GPS-Module-p-2523.html. [Accessed: 22- Sep- 2021].

[35] *Raspberrypi.org*. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/. [Accessed: 22- Sep- 2021].

[36] *Bstelectronic.com*. [Online]. Available: http://www.bstelectronic.com/Admin/upload/pimg/70520141120154556.jpg. [Accessed: 22- Sep- 2021].

[37] *Cdn-thumbs.freeart.com*. [Online]. Available: https://cdn-thumbs.freeart.com/network-ethernet-cable-with-rj45_fa28383071.jpg. [Accessed: 22- Sep- 2021].

[38] "BD SHOP | Online Shopping BD | YouTube Gadget & Gear Shop", *Bdshop.com*. [Online]. Available: https://www.bdshop.com/. [Accessed: 22- Sep- 2021].

[39] *Encrypted-tbn0.gstatic.com*. [Online]. Available: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRxPoWP796tb9-nnw_bIrztnU1kSlncK4-QQehgisEQUQIbgfkF91ZuM93l_M8Z9KJd-mw&usqp=CAU. [Accessed: 21- Sep- 2021].

[40] *Leetechbd.com*. [Online]. Available: *https://projectpoint.in/image/cache/catalog/CFR-25JB-16K-672x378.jpg*. [Accessed: 21- Sep- 2021].

[41] tech.alpsalpine.com. [Online]. Available: https://tech.alpsalpine.com/prod/common/images2018/category/03_switches_e.png [Accessed: 21- Sep- 2021]

[42] *Mcdn01.gittigidiyor.net*. [Online]. Available: https://mcdn01.gittigidiyor.net/69101/691018871_0.jpg. [Accessed: 21- Sep- 2021].

[43] "PhysioNet", *Archive.physionet.org*. [Online]. Available: https://archive.physionet.org/. [Accessed: 21- Sep- 2021].

[44] "PTB Diagnostic ECG Database v1.0.0", *Physionet.org*, 2004. [Online]. Available: https://www.physionet.org/content/ptbdb/1.0.0/. [Accessed: 21- Sep- 2021].

[45] J. Yu, Z. Wang and C. Yang, "Comparison of Electrocardiographic Criteria for Diagnosis of Left Ventricular Hypertrophy," *Chin. Circ. J.*, vol.19, no.1, pp.28-30, February, 2004.

[46] A. L. Goldberger, "Clinical Electrocardiography: A Simplified Approach," *New York, NY, USA: Elsevier Health Sciences*, 2012.

[47] R. J. Gibbons. "ACC/AHA Guidelines for Ambulatory Electrocardiography," *J. Am. Coll. Cardiol.*, vol.34, no.3, pp.912-945, September, 1999.

[48] B. Surawicz and T. K. Knilans, "Chou's Electrocardiography in Clinical Practice," *Philadelphia, PA: Saunders Elsevier*, 2008.

[49] M. Pal and S. Parija, "Prediction of Heart Diseases using random Forest," *Journal of Physics: Conference Series 1817*, 2021.

[50] "Kaggle: Your Machine Learning and Data Science Community", *Kaggle.com*. [Online]. Available: https://www.kaggle.com/. [Accessed: 21- Sep- 2021].

[51] M. Arif, I. A. Malagore and F. A. Afsar, "Detection and Localization of Myocardial Infarction using K-nearest Neighbor Classifier," *Journal of Medical Systems*, February, 2012.

[52] A. K. Dohare, V. Kumar and R. Kumar, " Detection of myocardial infarction in 12 lead ECG using support vector machine," *Applied Soft Computing*, vol. 64, pp. 138-147, March, 2018.

[53] "Fleet Tracking and Management | Geotab", *Geotab*. [Online]. Available: https://www.geotab.com/. [Accessed: 21- Sep- 2021].

[54] "Enterprise Mobile Computing news and information – SearchMobileComputing", *Searchmobilecomputing.techtarget.com*. [Online]. Available: https://searchmobilecomputing.techtarget.com/. [Accessed: 21- Sep- 2021].

[55] N. N. S. Hlaing, M. Naing and S. S. Naing, "GPS and GSM Based Vehicle Tracking System," *International Journal of Trend in Scientific Research and Development (IJTSRD)*, vol. 3, no. 4, May-June, 2019.

[56] "SIM808_GPS_Application Note_V1.00,"1st ed., Shanghai SIMCom Wireless Solution Ltd., 2014.

[57] J. Mohajon, "Confusion Matrix for Your Multi-Class Machine Learning Model", Medium, 2020. [Online]. Available: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826. [Accessed: 22-Sep- 2021].

[58] "Send SMS/Text From Arduino Using SIM800L GSM/GPRS Module", *Arduino Project Hub*. [Online]. Available: https://create.arduino.cc/projecthub/SurtrTech/send-sms-text-from-arduino-using-sim800l-gsm-gprs-module-228975. [Accessed: 21- Sep- 2021].

[59] "What causes a delay in delivering SMS messages?", *Vonage*. [Online]. Available: https://help.nexmo.com/hc/en-us/articles/204014893-What-causes-a-delay-in-delivering-SMS-messages-. [Accessed: 21- Sep- 2021].

[60] "History and Victories", *Health Care Without Harm*. [Online]. Available: https://noharm-uscanada.org/content/us-canada/history-and-victories. [Accessed: 26- Sep- 2021].

[61] S. Elabed, A. Belal and A. Shamayleh, "Sustainability of Medical Equipment in the Healthcare Industry: An Overview," *2019 Fifth International Conference on Advances in Biomedical Engineering (ICABME)*, 2019.

[62] M. Cheng, *Medical Device Regulations Global overview and guiding principles*. World Health Organization Geneva, 2003.

[63] "Bangladesh Medical Device Regulations | RegDesk", *RegDesk*. [Online]. Available: https://www.regdesk.co/resource-library/bangladesh/. [Accessed: 26- Sep- 2021].

Appendix A

1. Arduino Code for GPS and GSM Interfacing

```
#include <SoftwareSerial.h>
SoftwareSerial sim808(7,8);

char phone_no[] = "01*********"; // replace with your phone no.
String data[5];
#define DEBUG true
String state,timegps,latitude,longitude;

void setup() {
 sim808.begin(9600);
 Serial.begin(9600);
 delay(50);

sim808.println("AT+CGNSPWR=1");
delay(2000);
sim808.println("AT+CGNSSEQ=\"RMC\"");
delay(2000);

sim808.print("AT+CSMP=17,167,0,0");  // set this parameter if empty SMS received
delay(100);
sim808.print("AT+CMGF=1\r");
delay(400);
sendData("AT+CGNSPWR=1",1000,DEBUG);
delay(50);
sendData("AT+CGNSSEQ=RMC",1000,DEBUG);
delay(150);
```

```
}

void loop() {
        sendTabData("AT+CGNSINF",1000,DEBUG);

  if (state !=0) {

          Serial.println("Latitude  :"+latitude);
          Serial.println("Longitude  :"+longitude);

   sim808.println("AT+CMGF=1");              // Select Text mode for messsaging
   delay(100);

   sim808.print("AT+CMGS=\"");
    sim808.print(phone_no);
    sim808.println("\"");

    delay(300);

    sim808.print("The patient is affected by ‘________’. Here is the current location of the patient:\nhttp://maps.google.com/maps?q=loc:");
    sim808.print(latitude);
    sim808.print(",");
    sim808.print (longitude);
    delay(200);
    sim808.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(200);
    sim808.println();
```

```
    delay(900000);
    sim808.flush();
  }
else {
    Serial.println("GPS Initialising...");
  }
}

void sendTabData(String command , const int timeout , boolean debug){

  sim808.println(command);
  long int time = millis();
  int i = 0;

  while((time+timeout) > millis()){
    while(sim808.available()){
      char c = sim808.read();
      if (c != ',') {
         data[i] +=c;
         delay(100);
      } else {
        i++;
      }
      if (i == 5) {
        delay(100);
        goto exitL;
      }
    }
  }exitL:
```

```
  if (debug) {
    state = data[1];
  timegps = data[2];
    latitude = data[3];
    longitude =data[4];
  }
}
String sendData (String command , const int timeout ,boolean debug){
  String response = "";
  sim808.println(command);
  long int time = millis();
  int i = 0;

  while ( (time+timeout ) > millis()){
    while (sim808.available()){
      char c = sim808.read();
      response +=c;
    }
  }
  if (debug) {
     Serial.print(response);
     }
     return response;
}
```

2.Arduino Code for GPS and GSM Interfacing along with Pause Switch Operation and Different MI conditions

```
#include <SoftwareSerial.h>
SoftwareSerial sim808(7,8);

int incomingByte;
const int ledPin  = 13;
int condition;
int inp = 5;
int i;
int val = 0; // Initiating the switch value to zero[Pull down]
int count;

char phone_no[] = "01*********"; // replace with your phone no.
String data[5];
#define DEBUG true
String state,timegps,latitude,longitude;

void setup() {
 //python setup
 pinMode(ledPin, OUTPUT);
 //python setup end

 sim808.begin(9600);
 Serial.begin(9600);
 delay(50);

 sim808.println("AT+CGNSPWR=1");
```

```
delay(2000);
sim808.println("AT+CGNSSEQ=\"RMC\"");
delay(2000);


 sim808.print("AT+CSMP=17,167,0,0");  // set this parameter if empty SMS received
 delay(100);
 sim808.print("AT+CMGF=1\r");
 delay(400);
 sendData("AT+CGNSPWR=1",1000,DEBUG);
delay(50);
sendData("AT+CGNSSEQ=RMC",1000,DEBUG);
delay(150);
}

void loop() {
  val = digitalRead(inp);
  //trial python code
  //trial python code end

  sendTabData("AT+CGNSINF",1000,DEBUG);

  if  ( state !=0 && val == 0) {

    //check signal
    if(Serial.available()>0){
    incomingByte = Serial.read();
    if (incomingByte == 'A'){
      digitalWrite(ledPin, HIGH);
```

```
      condition = 1;
    }
    if (incomingByte == 'I'){
      //digitalWrite(ledPin, LOW);
      condition = 2;
      }
    if (incomingByte == 'H'){
     //digitalWrite(ledPin, LOW);
     condition = 3;
      }
    }
    //check signal

//    Serial.println("Latitude  :"+latitude);
//    Serial.println("Longitude  :"+longitude);

    sim808.println("AT+CMGF=1");                // Select Text mode for messsaging
    delay(100);

    sim808.print("AT+CMGS=\"");
    sim808.print(phone_no);
    sim808.println("\"");

    delay(300);
    if(condition ==1){
      sim808.print("The patient is affected by Anterior. Here is the current location of the patient:\nhttp://maps.google.com/maps?q=loc:");
      sim808.print(latitude);
      sim808.print(",");
```

```
sim808.print (longitude);
delay(200);
sim808.println((char)26); // End AT command with a ^Z, ASCII code 26
delay(200);
sim808.println();

count = 0;
}
if(condition ==2){
sim808.print("The patient is affected by inferior. Here is the current location of the patient:\nhttp://maps.google.com/maps?q=loc:");
sim808.print(latitude);
sim808.print(",");
sim808.print (longitude);
delay(200);
sim808.println((char)26); // End AT command with a ^Z, ASCII code 26
delay(200);
sim808.println();

count = 0;
}
if(condition ==3 && count == 2){
sim808.print("The patient is Healthy. Here is the current location of the patient:\nhttp://maps.google.com/maps?q=loc:");
sim808.print(latitude);
sim808.print(",");
sim808.print (longitude);
delay(200);
sim808.println((char)26); // End AT command with a ^Z, ASCII code 26
delay(200);
```

```
      sim808.println();
      sim808.flush();
      count = 0;
      }

      delay(60000);
      count = count + 1;

    //delay(60000);
    sim808.flush();

  } else {
    Serial.println("GPS Initialising...");
    for(i=1;i<=3;i++){
      digitalWrite(LED_BUILTIN,HIGH);
      delay(250);
      digitalWrite(LED_BUILTIN,LOW);
      delay(250);
    }
  }
}

void sendTabData(String command , const int timeout , boolean debug){
  delay(3000);
  sim808.println(command);
  long int time = millis();
  int i = 0;

  while((time+timeout) > millis()){
```

```
    while(sim808.available()){
      char c = sim808.read();
      if (c != ',') {
         data[i] +=c;
         delay(100);
      } else {
        i++;
      }
      if (i == 5) {
        delay(100);
        goto exitL;
      }
    }
  }exitL:
  if (debug) {
    state = data[1];
  timegps = data[2];
    latitude = data[3];
    longitude =data[4];
  }
}
String sendData (String command , const int timeout ,boolean debug){

  String response = "";
  sim808.println(command);
  long int time = millis();
  int i = 0;

  while ( (time+timeout ) > millis()){
```

```
    while (sim808.available()){
      char c = sim808.read();
      response +=c;
    }
  }
  if (debug) {
     Serial.print(response);
     }
     return response;
}
```

Appendix B

1. Machine Learning Code for Dataset Processing and Detection of MI

```
import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model_selection import train_test_split


dataset = pd.read_csv("Anterior_1.csv")

#dataset = pd.read_csv("Anterior_2.csv")

#dataset = pd.read_csv("Anterior_3.csv")

#dataset = pd.read_csv("Anterior_4.csv")

#dataset = pd.read_csv("Anterior_5.csv")

#dataset = pd.read_csv("Anterior_6.csv")

#dataset = pd.read_csv("Anterior_7.csv")

#dataset = pd.read_csv("Inferior_1.csv")

#dataset = pd.read_csv("Inferior_2.csv")

#dataset = pd.read_csv("Inferior_3.csv")

#dataset = pd.read_csv("Inferior_4.csv")

#dataset = pd.read_csv("Inferior_5.csv")

#dataset = pd.read_csv("Inferior_6.csv")

#dataset = pd.read_csv("Inferior_7.csv")

#dataset = pd.read_csv("Healthy_1.csv")

#dataset = pd.read_csv("Healthy_2.csv")

#dataset = pd.read_csv("Healthy_3.csv")

#dataset = pd.read_csv("Healthy_4.csv")

#dataset = pd.read_csv("Healthy_5.csv")

#dataset = pd.read_csv("Healthy_6.csv")

#Calibration Purpose Check with index column added

#This section is to be run to assign correct Random_state to get the test data

#Start
```

```
for i in range(1,300):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.003333,random_state=i)
    if X_test[0,0]==300:
        print(i)
#Stop
dataset.pop('Index')
X = dataset.iloc[:, :4003].values
 y = dataset.iloc[:, 4004].values
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test
#Knn
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=4)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print('knn')
print(y_pred)

#random forest
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=37)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print('Random Forest')
print(y_pred)
```

```
#SVM
from sklearn import svm
classifier = svm.SVC(kernel='linear')
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print("SVM")
print(y_pred)

#Naive Baes
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print('Naive Bayes')
print(y_pred)
```

2.Code for Interfacing with Raspberry Pi 4

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
import serial
import time



dataset = pd.read_csv("Anterior_1.csv") #For Anterior Test
#dataset = pd.read_csv("4.Inferior_1.csv")#For Inferior Test
#dataset = pd.read_csv("7.Healthy_1.csv")#For Healthy Test
#This section is to be run to assign correct Random_state to get the test data
#Start
```

```
for i in range(1,300):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.003333,random_state=i)
    if X_test[0,0]==300:
        print(i)
#Stop
dataset.pop('Index')

X = dataset.iloc[:, :4003].values
y = dataset.iloc[:, 4004].values

#Train Test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.0109,random_state=44)

#Check Test Data
#print(X_test[0,0])

#Scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

#Random forest
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=37)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
#print('Random Forest')
#print(y_pred)
```

```
#Knn
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=1)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
#print('knn')
#print(y_pred)
#classifier.score(X_test,y_test)

#Bridge

#int condition
#Anterior = 1
#Inferior = 2
#Healthy = 3
ser = serial.Serial('/dev/ttyACM0', 9600)
conditon = int(3)

if(y_pred=='Anterior MI'):
   condition = 1
elif(y_pred=='Inferior MI'):
   condition = 2
elif(y_pred=='Healthy '):
   condition = 3

print(condition)
#Function for making the bridge
def bridge():
```

```
    if condition == 1:
        time.sleep(5)
        ser.write(b'A')
        time.sleep(0.1)
        bridge()
    elif condition == 2:
        time.sleep(5)
        ser.write(b'I')
        time.sleep(0.1)
        bridge()
    elif condition == 3:
        time.sleep(5)
        ser.write(b'H')
        time.sleep(0.1)
        bridge()

time.sleep(2)
bridge()
```

Appendix C

In our paper, there are plenty of hardware connections that are necessary for building up the desired prototype. However, the primary idea that we had gained for setting up the connections are based on the following catalogue.

1. Arduino Uno R3:

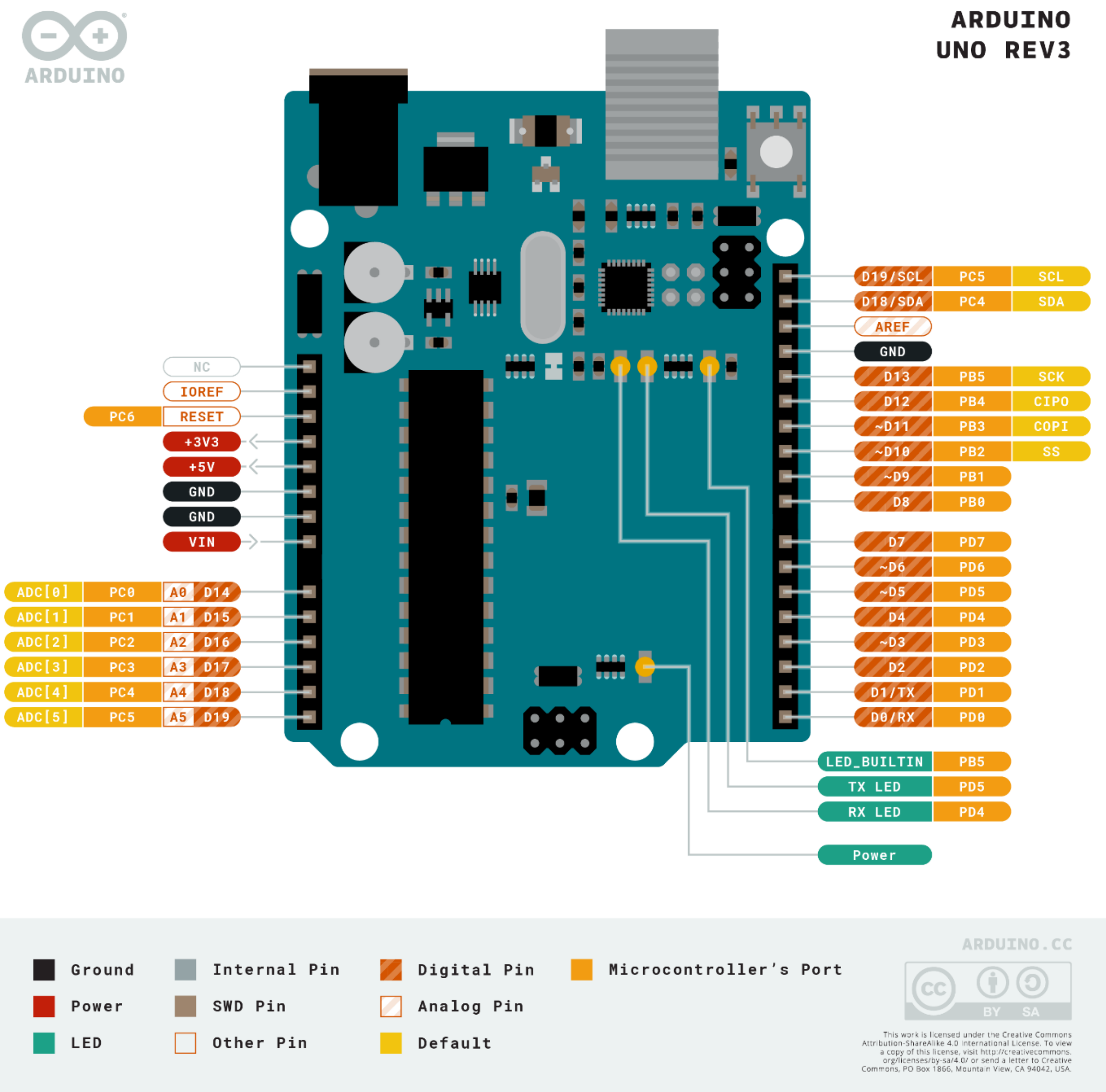


Fig: Diagram of Arduino Uno

Board	Name	Arduino UNO R3
	SKU	A000066
Microcontroller	ATmega328P	
USB connector	USB-B	
Pins	Built-in LED Pin	13
	Digital I/O Pins	14
	Analog input pins	6
	PWM pins	6
Communication	UART	Yes
	I2C	Yes
	SPI	Yes
Power	Circuit operating voltage	5V
	Input voltage (recommended)	7-12V
	Input voltage (limit)	6-20V
	Supported battery	9V battery
	Battery connector	Power jack
	DC Current per I/O Pin	20 mA
	DC Current for 3.3V Pin	50 mA
Clock speed	Main Processor	ATmega328P 16 MHz
	USB-Serial Processor	ATmega16U2 16 MHz
Memory	ATmega328P	2KB SRAM, 32KB FLASH, 1KB EEPROM
Dimensions	Weight	25 g
	Width	53.4 mm
	Length	68.6 mm

Fig: Features of Arduino Uno

Website Link: https://docs.arduino.cc/hardware/uno-rev3

2. Sim808 GPRS Module

The following figure shows a functional diagram of SIM808:

- The GSM baseband engine
- The GPS engine
- Flash
- The GSM radio frequency part
- The antenna interface
- The other interfaces

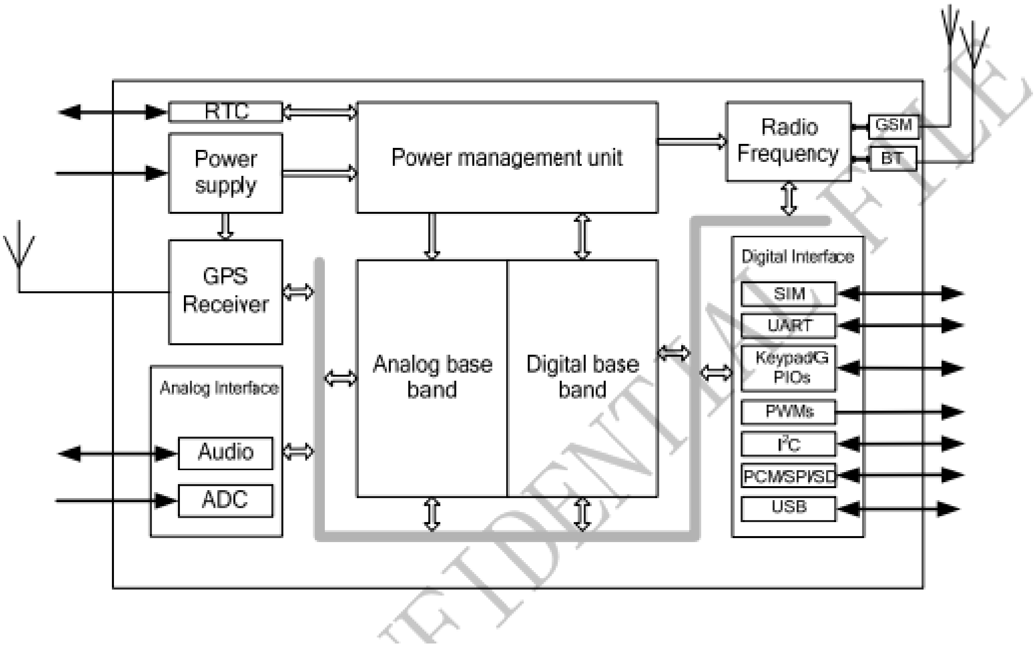


Fig: Functional Diagram of Sim808

Grammar	Description
AT+CGPSPWR=1 OK	Open GPS
AT+CGPSSTATUS? +CGPSSTATUS: Location 3D Fix OK	Read the GPS fix status GPS has fixed with 3D status
AT+CGPSINF=0 0,3113.317683,12121.244232,51.563730,011400 27022006.085,16,13,0.072580,0.000000 OK AT+ CGPSOUT =8 OK $GPGSA,A,3,02,09,05,193,29,08,26,15,04,07,10 ,,1.62,0.95,1.31*3C $GPGSA,A,3,02,09,05,193,29,08,26,15,04,07,10 ,,1.62,0.95,1.31*3C	Read GPS information GPS NMEA data output from AT uart
AT+CGPSRST=0 OK $GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30 …… $GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30 $GPGSA,A,3,02,15,10,05,26,09,29,04,07,,,,1.27, 0.95,0.84*01	RESET GPS(COLD Start)
AT+CGPSRST=1 OK $GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,1.00*01 $GPGSA,A,3,193,02,05,26,29,15,10,09,04,08,07 ,,1.23,0.93,0.80*34	RESET GPS(HOT Start)
AT+CGPSPWR=0 OK	Close GPS

Fig: Features of Sim808

Website Link:

https://cdn-shop.adafruit.com/datasheets/SIM808_Hardware+Design_V1.00.pdf

https://cdn-shop.adafruit.com/datasheets/SIM808_GPS_Application_Note_V1.00.pdf

3. Raspberry Pi 4:

Specification

Processor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	1GB, 2GB, 4GB or 8GB LPDDR4 (depending on model) with on-die ECC
Connectivity:	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO:	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & sound:	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia:	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics
SD card support:	Micro SD card slot for loading operating system and data storage
Input power:	5V DC via USB-C connector (minimum 3A[1]) 5V DC via GPIO header (minimum 3A[1]) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit https://www.raspberrypi.org/documentation/hardware/raspberrypi/conformity.md
Production lifetime:	The Raspberry Pi 4 Model B will remain in production until at least January 2026.

Fig: Specifications of Raspberry Pi 4

WARNINGS

- This product should only be connected to an external power supply rated at 5V/3A DC or 5.1V/ 3A DC minimum[1]. Any external power supply used with the Raspberry Pi 4 Model B shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance and result in damage to the unit and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors and mice when used in conjunction with the Raspberry Pi.
- Where peripherals are connected that do not include the cable or connector, the cable or connector must offer adequate insulation and operation in order that the relevant performance and safety requirements are met.

SAFETY INSTRUCTIONS

To avoid malfunction or damage to this product please observe the following:

- Do not expose to water, moisture or place on a conductive surface whilst in operation.
- Do not expose it to heat from any source; Raspberry Pi 4 Model B is designed for reliable operation at normal ambient room temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Avoid handling the printed circuit board whilst it is powered and only handle by the edges to minimise the risk of electrostatic discharge damage.

---

[1] A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

Fig: Warning and Safety Instructions of Using Raspberry Pi 4

Website Link: https://datasheets.raspberrypi.org/rpi4/raspberry-pi-4-product-brief.pdf

Appendix D

1. Partial Train Data [From V3]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Index	'0:00.000'	'0:00.001'	'0:00.002'	'0:00.003'	'0:00.004'	'0:00.005'	'0:00.006'	'0:00.007'	'0:00.008'	'0:00.009'	'0:00.010'	'0:00.011'	'0:00.012'	'0:00.013'	'0:00.014'	'0:00.015'	'0:00.016'	'0:00.017'
2	1	0.391	0.413	0.431	0.441	0.45	0.468	0.481	0.491	0.509	0.53	0.543	0.552	0.577	0.604	0.617	0.638	0.661	0.679
3	2	-0.03	-0.033	-0.041	-0.041	-0.043	-0.051	-0.063	-0.08	-0.085	-0.091	-0.113	-0.123	-0.137	-0.141	-0.135	-0.13	-0.113	-0.095
4	3	0.269	0.278	0.29	0.299	0.3	0.298	0.294	0.289	0.283	0.276	0.272	0.27	0.265	0.269	0.279	0.285	0.289	0.294
5	4	-0.602	-0.61	-0.612	-0.618	-0.622	-0.626	-0.633	-0.64	-0.646	-0.654	-0.661	-0.669	-0.679	-0.688	-0.696	-0.703	-0.712	-0.719
6	5	0.079	0.081	0.083	0.077	0.076	0.073	0.077	0.08	0.075	0.077	0.082	0.09	0.093	0.093	0.093	0.095	0.096	0.097
7	6	0.281	0.281	0.288	0.289	0.291	0.292	0.292	0.289	0.284	0.286	0.298	0.302	0.298	0.292	0.281	0.281	0.273	0.257
8	7	1.198	1.197	1.2	1.2	1.198	1.193	1.19	1.197	1.194	1.188	1.187	1.183	1.184	1.187	1.192	1.188	1.185	1.188
9	8	0.348	0.348	0.348	0.347	0.347	0.35	0.35	0.351	0.351	0.348	0.346	0.347	0.348	0.347	0.349	0.35	0.347	0.346
10	9	0.365	0.362	0.371	0.378	0.377	0.379	0.383	0.386	0.392	0.396	0.397	0.398	0.395	0.393	0.396	0.398	0.401	0.4
11	10	0.896	0.887	0.883	0.889	0.891	0.893	0.891	0.88	0.876	0.882	0.883	0.876	0.871	0.879	0.89	0.889	0.884	0.887
12	11	0.081	0.086	0.081	0.076	0.073	0.072	0.073	0.071	0.07	0.074	0.075	0.077	0.079	0.08	0.079	0.081	0.082	0.077
13	12	-0.185	-0.264	-0.358	-0.428	-0.485	-0.549	-0.613	-0.673	-0.726	-0.786	-0.836	-0.894	-0.959	-1.005	-1.05	-1.095	-1.153	-1.236
14	13	-1.633	-1.645	-1.662	-1.713	-1.786	-1.867	-1.919	-1.962	-2.055	-2.112	-2.142	-2.171	-2.166	-2.123	-2.096	-2.088	-2.06	-2.03
15	14	-0.131	-0.13	-0.133	-0.126	-0.126	-0.116	-0.11	-0.112	-0.106	-0.106	-0.102	-0.101	-0.102	-0.106	-0.105	-0.104	-0.106	-0.102
16	15	-0.117	-0.117	-0.117	-0.117	-0.12	-0.12	-0.12	-0.121	-0.119	-0.121	-0.121	-0.122	-0.126	-0.129	-0.132	-0.132	-0.133	-0.139
17	16	-0.494	-0.496	-0.495	-0.494	-0.498	-0.502	-0.5	-0.499	-0.501	-0.504	-0.502	-0.502	-0.505	-0.504	-0.506	-0.508	-0.509	-0.509
18	17	-1.143	-1.147	-1.147	-1.146	-1.147	-1.149	-1.15	-1.149	-1.149	-1.149	-1.151	-1.153	-1.153	-1.149	-1.148	-1.149	-1.151	-1.152
19	18	-0.363	-0.359	-0.357	-0.36	-0.361	-0.357	-0.357	-0.357	-0.354	-0.354	-0.352	-0.351	-0.348	-0.345	-0.348	-0.347	-0.345	-0.345
20	19	-0.013	-0.011	-0.014	-0.011	-0.011	-0.015	-0.013	-0.015	-0.015	-0.013	-0.018	-0.017	-0.018	-0.022	-0.024	-0.02	-0.02	-0.022
21	20	-0.753	-0.692	-0.625	-0.558	-0.502	-0.466	-0.438	-0.402	-0.354	-0.312	-0.276	-0.239	-0.211	-0.186	-0.151	-0.125	-0.116	-0.11
22	21	0.939	0.941	0.941	0.942	0.943	0.944	0.943	0.942	0.94	0.945	0.943	0.938	0.941	0.941	0.942	0.948	0.955	0.955
23	22	0.373	0.374	0.372	0.377	0.381	0.381	0.379	0.383	0.381	0.371	0.379	0.389	0.393	0.384	0.378	0.38	0.384	0.391
24	23	-0.863	-0.921	-0.975	-1.059	-1.141	-1.203	-1.218	-1.135	-1.03	-0.932	-0.852	-0.804	-0.746	-0.693	-0.643	-0.606	-0.587	-0.542
25	24	-0.077	-0.074	-0.075	-0.07	-0.067	-0.059	-0.053	-0.051	-0.041	-0.038	-0.039	-0.036	-0.032	-0.028	-0.022	-0.02	-0.024	-0.024
26	25	1.516	1.573	1.633	1.686	1.728	1.766	1.801	1.831	1.855	1.879	1.903	1.922	1.941	1.96	1.974	1.986	1.994	2.001
27	26	0.106	0.106	0.104	0.097	0.101	0.093	0.092	0.088	0.087	0.09	0.086	0.086	0.081	0.086	0.085	0.087	0.089	0.084
28	27	-0.958	-0.964	-0.97	-0.975	-0.981	-0.988	-0.994	-1.004	-1.01	-1.013	-1.02	-1.024	-1.028	-1.033	-1.033	-1.035	-1.038	-1.042
29	28	0.083	0.077	0.077	0.077	0.073	0.066	0.062	0.059	0.061	0.064	0.064	0.059	0.059	0.058	0.052	0.046	0.038	0.036
30	29	0.186	0.185	0.182	0.181	0.18	0.182	0.182	0.179	0.182	0.185	0.184	0.183	0.185	0.184	0.187	0.189	0.189	0.19
31	30	1.383	1.39	1.39	1.382	1.381	1.385	1.392	1.395	1.393	1.395	1.403	1.404	1.399	1.395	1.397	1.401	1.393	1.392
32	116	-0.056	-0.051	-0.053	-0.051	-0.05	-0.053	-0.054	-0.051	-0.05	-0.054	-0.054	-0.053	-0.052	-0.051	-0.052	-0.054	-0.057	-0.056
33	117	-0.932	-0.929	-0.936	-0.942	-0.945	-0.95	-0.954	-0.964	-0.97	-0.973	-0.979	-0.983	-0.992	-1	-1.002	-1.002	-1.012	-1.017
34	118	0.936	0.954	0.957	0.962	0.975	0.975	0.958	0.955	0.96	0.949	0.949	0.943	0.931	0.951	0.96	0.957	0.965	0.97
35	119	-0.339	-0.338	-0.332	-0.335	-0.334	-0.328	-0.317	-0.314	-0.322	-0.328	-0.334	-0.326	-0.323	-0.33	-0.334	-0.332	-0.331	-0.338
36	120	0.846	0.847	0.848	0.849	0.851	0.848	0.85	0.853	0.853	0.857	0.859	0.859	0.859	0.86	0.861	0.864	0.867	0.868
37	121	0.151	0.14	0.123	0.113	0.098	0.086	0.074	0.068	0.058	0.043	0.03	0.017	0.01	-0.004	-0.014	-0.021	-0.029	-0.04
38	122	0.056	0.119	0.194	0.251	0.317	0.406	0.517	0.633	0.777	0.959	1.139	1.327	1.506	1.695	1.863	2.006	2.108	2.134
39	123	-0.235	-0.253	-0.248	-0.252	-0.254	-0.249	-0.254	-0.251	-0.241	-0.239	-0.243	-0.251	-0.254	-0.251	-0.244	-0.238	-0.24	-0.244
40	124	0.093	0.101	0.105	0.106	0.104	0.104	0.105	0.102	0.107	0.108	0.103	0.099	0.102	0.107	0.106	0.104	0.102	0.102
41	125	0.037	0.038	0.04	0.04	0.041	0.043	0.043	0.044	0.042	0.041	0.044	0.045	0.047	0.046	0.044	0.042	0.04	0.041
42	126	-0.022	-0.021	-0.019	-0.019	-0.018	-0.015	-0.014	-0.015	-0.019	-0.025	-0.028	-0.03	-0.033	-0.035	-0.041	-0.046	-0.052	-0.057
43	127	-0.059	-0.064	-0.064	-0.064	-0.064	-0.065	-0.067	-0.066	-0.065	-0.065	-0.067	-0.068	-0.068	-0.066	-0.066	-0.072	-0.073	-0.071
44	128	2.15	2.151	2.149	2.151	2.151	2.15	2.151	2.152	2.149	2.147	2.145	2.146	2.148	2.147	2.151	2.151	2.147	2.147
45	129	0.626	0.623	0.62	0.612	0.574	0.532	0.505	0.468	0.406	0.341	0.271	0.175	0.104	0.044	-0.029	-0.101	-0.169	-0.246
46	130	-0.157	-0.159	-0.164	-0.169	-0.177	-0.175	-0.173	-0.174	-0.176	-0.179	-0.183	-0.193	-0.204	-0.214	-0.21	-0.223	-0.229	-0.223
47	131	-1.776	-1.775	-1.771	-1.769	-1.758	-1.758	-1.772	-1.76	-1.758	-1.768	-1.782	-1.782	-1.777	-1.774	-1.767	-1.768	-1.775	-1.775
48	132	-1.459	-1.46	-1.461	-1.464	-1.464	-1.461	-1.458	-1.458	-1.457	-1.454	-1.452	-1.454	-1.46	-1.46	-1.46	-1.46	-1.461	-1.462
49	133	0.026	0.019	0.013	0.009	0.011	0.015	0.019	0.026	0.026	0.029	0.024	0.017	0.019	0.026	0.029	0.024	0.024	0.024
50	134	-0.014	-0.012	-0.013	-0.013	-0.015	-0.013	-0.014	-0.013	-0.013	-0.015	-0.017	-0.02	-0.019	-0.018	-0.015	-0.013	-0.014	-0.017
51	135	0.037	0.029	0.034	0.041	0.04	0.041	0.05	0.051	0.05	0.051	0.05	0.054	0.058	0.052	0.053	0.054	0.057	0.062
52	136	0.237	0.238	0.235	0.235	0.239	0.238	0.243	0.245	0.25	0.254	0.257	0.263	0.259	0.262	0.265	0.266	0.272	0.275
53	137	0.043	0.041	0.043	0.044	0.043	0.038	0.039	0.04	0.037	0.037	0.036	0.041	0.042	0.045	0.045	0.038	0.036	0.035
54	138	0.234	0.236	0.237	0.238	0.236	0.238	0.238	0.237	0.237	0.235	0.234	0.234	0.233	0.233	0.235	0.237	0.236	0.234
55	139	0.402	0.403	0.405	0.406	0.402	0.405	0.402	0.399	0.401	0.403	0.409	0.411	0.41	0.407	0.408	0.409	0.401	0.402
56	140	0.568	0.564	0.563	0.563	0.564	0.562	0.563	0.562	0.56	0.559	0.558	0.559	0.56	0.559	0.559	0.56	0.558	0.558
57	141	0.127	0.141	0.151	0.146	0.152	0.162	0.16	0.168	0.179	0.175	0.176	0.173	0.171	0.176	0.18	0.185	0.19	0.184
58	142	0.092	0.087	0.085	0.084	0.083	0.085	0.084	0.086	0.091	0.103	0.115	0.124	0.137	0.15	0.165	0.175	0.184	0.2
59	143	-1.591	-1.601	-1.55	-1.512	-1.48	-1.448	-1.419	-1.387	-1.36	-1.345	-1.337	-1.323	-1.297	-1.271	-1.26	-1.242	-1.229	-1.226
60	144	-0.823	-0.798	-0.786	-0.773	-0.755	-0.733	-0.706	-0.683	-0.664	-0.647	-0.638	-0.631	-0.619	-0.607	-0.596	-0.594	-0.599	-0.598
61	145	0.059	0.055	0.054	0.055	0.045	0.036	0.027	0.021	0.021	0.018	0.019	0.013	0.004	-0.001	-0.011	-0.019	-0.029	-0.036
62	234	0.573	0.574	0.576	0.576	0.577	0.581	0.583	0.581	0.583	0.586	0.588	0.591	0.591	0.595	0.595	0.597	0.6	0.599
63	235	0.168	0.201	0.231	0.266	0.3	0.328	0.354	0.382	0.41	0.435	0.459	0.482	0.498	0.513	0.53	0.538	0.548	0.557
64	236	-1.739	-1.742	-1.744	-1.744	-1.75	-1.756	-1.762	-1.768	-1.773	-1.776	-1.78	-1.78	-1.782	-1.782	-1.782	-1.782	-1.78	-1.778
65	237	0.3	0.302	0.303	0.301	0.303	0.307	0.306	0.306	0.308	0.307	0.306	0.306	0.308	0.307	0.308	0.308	0.306	0.309
66	238	0.599	0.609	0.617	0.627	0.635	0.644	0.655	0.659	0.666	0.672	0.679	0.692	0.7	0.707	0.713	0.72	0.724	0.73
67	239	0.141	0.14	0.139	0.134	0.132	0.133	0.133	0.132	0.13	0.131	0.132	0.129	0.13	0.129	0.127	0.128	0.129	0.13
68	240	0.158	0.156	0.161	0.162	0.164	0.167	0.172	0.172	0.174	0.179	0.184	0.188	0.188	0.189	0.191	0.197	0.196	0.195
69	241	0.774	0.776	0.779	0.78	0.782	0.783	0.783	0.784	0.787	0.788	0.789	0.79	0.791	0.793	0.795	0.796	0.8	0.804
70	242	-1.016	-1.015	-1.016	-1.016	-1.014	-1.014	-1.016	-1.014	-1.01	-1.012	-1.01	-1.008	-1.008	-1.009	-1.01	-1.01	-1.009	-1.009
71	243	0.027	0.027	0.025	0.025	0.021	0.021	0.02	0.018	0.018	0.018	0.017	0.021	0.025	0.021	0.019	0.021	0.025	0.026
72	244	-0.062	-0.06	-0.06	-0.064	-0.065	-0.067	-0.07	-0.069	-0.067	-0.062	-0.058	-0.051	-0.041	-0.034	-0.031	-0.029	-0.028	-0.025
73	245	0.244	0.245	0.245	0.247	0.247	0.245	0.244	0.242	0.244	0.245	0.243	0.241	0.241	0.243	0.243	0.246	0.247	0.245
74	246	0.838	0.848	0.855	0.857	0.866	0.873	0.878	0.885	0.891	0.9	0.907	0.914	0.922	0.93	0.936	0.94	0.944	0.946
75	247	1.156	1.164	1.169	1.165	1.171	1.18	1.189	1.2	1.203	1.206	1.217	1.224	1.231	1.238	1.236	1.248	1.25	1.247
76	248	0.314	0.314	0.311	0.308	0.313	0.31	0.305	0.307	0.306	0.304	0.308	0.311	0.31	0.312	0.314	0.308	0.31	0.312
77	249	-0.089	-0.094	-0.093	-0.093	-0.092	-0.086	-0.085	-0.081	-0.079	-0.079	-0.076	-0.073	-0.067	-0.062	-0.065	-0.068	-0.066	-0.067
78	250	0.622	0.622	0.623	0.632	0.636	0.634	0.628	0.626	0.635	0.649	0.665	0.67	0.668	0.673	0.675	0.676	0.666	0.666
79	251	1	0.994	0.999	1.004	1.012	1.006	1.011	1.012	1.01	1.028	1.03	1.028	1.032	1.036	1.04	1.041	1.044	1.048
80	252	0.026	0.029	0.028	0.027	0.026	0.023	0.022	0.023	0.018	0.018	0.019	0.018	0.017	0.018	0.019	0.021	0.022	0.019
81	253	0.22	0.226	0.228	0.232	0.236	0.238	0.239	0.246	0.247	0.25	0.25	0.247	0.248	0.247	0.251	0.254	0.26	0.26
82	254	0.219	0.221	0.225	0.23	0.231	0.232	0.236	0.242	0.249	0.249	0.252	0.252	0.256	0.258	0.261	0.266	0.266	0.268
83	255	0.063	0.062	0.063	0.064	0.063	0.063	0.068	0.07	0.072	0.073	0.075	0.077	0.079	0.081	0.083	0.086	0.086	0.086
84	256	-0.202	-0.203	-0.205	-0.204	-0.208	-0.21	-0.21	-0.208	-0.21	-0.214	-0.214	-0.216	-0.217	-0.217	-0.219	-0.218	-0.218	-0.22
85	257	0.036	0.045	0.057	0.068	0.076	0.084	0.093	0.102	0.11	0.12	0.132	0.141	0.153	0.159	0.165	0.178	0.187	0.196
86	258	-0.181	-0.184	-0.182	-0.179	-0.177	-0.176	-0.175	-0.177	-0.176	-0.177	-0.179	-0.179	-0.181	-0.179	-0.179	-0.18	-0.175	-0.174
87	259	0.185	0.194	0.199	0.205	0.216	0.221	0.233	0.242	0.247	0.255	0.259	0.267	0.277	0.285	0.293	0.299	0.305	0.315
88	260	0.117	0.015	-0.07	-0.167	-0.27	-0.353	-0.401	-0.43	-0.446	-0.453	-0.456	-0.448	-0.42	-0.364	-0.302	-0.241	-0.193	-0.159
89	261	-0.207	-0.205	-0.204	-0.206	-0.205	-0.206	-0.21	-0.212	-0.208	-0.21	-0.212	-0.21	-0.209	-0.204	-0.208	-0.212	-0.212	-0.216
90	262	0.407	0.409	0.414	0.42	0.425	0.43	0.435	0.443	0.448	0.452	0.46	0.467	0.472	0.476	0.482	0.491	0.499	0.507
91	263	-0.043	-0.041	-0.039	-0.041	-0.044	-0.043	-0.04	-0.039	-0.043	-0.041	-0.037	-0.039	-0.04	-0.041	-0.042	-0.04	-0.043	-0.045

2. Partial Train Data [From V2]

	ALO	ALP	ALQ	ALR	ALS	ALT	ALU	ALV	ALW	ALX	ALY	ALZ	AMA	AMB	AMC	AMD	AME	AMF	AMG
1	'0:01.001'	'0:01.002'	'0:01.003'	'0:01.004'	'0:01.005'	'0:01.006'	'0:01.007'	'0:01.008'	'0:01.009'	'0:01.010'	'0:01.011'	'0:01.012'	'0:01.013'	'0:01.014'	'0:01.015'	'0:01.016'	'0:01.017'	'0:01.018'	'0:01.019'
2	0.333	0.33	0.33	0.319	0.316	0.314	0.311	0.305	0.307	0.311	0.303	0.298	0.308	0.314	0.302	0.311	0.324	0.326	0.328
3	-0.035	-0.038	-0.051	-0.059	-0.07	-0.091	-0.116	-0.141	-0.154	-0.171	-0.205	-0.234	-0.257	-0.279	-0.291	-0.302	-0.31	-0.32	-0.322
4	0.187	0.192	0.204	0.214	0.217	0.211	0.204	0.199	0.194	0.186	0.179	0.178	0.173	0.178	0.189	0.195	0.2	0.204	0.206
5	-0.614	-0.623	-0.629	-0.64	-0.646	-0.655	-0.665	-0.673	-0.685	-0.696	-0.708	-0.721	-0.734	-0.746	-0.757	-0.769	-0.782	-0.792	-0.804
6	-0.043	-0.043	-0.04	-0.048	-0.049	-0.051	-0.048	-0.047	-0.051	-0.052	-0.049	-0.04	-0.041	-0.038	-0.035	-0.034	-0.028	-0.022	-0.022
7	-0.064	-0.065	-0.055	-0.052	-0.054	-0.054	-0.053	-0.053	-0.058	-0.059	-0.046	-0.04	-0.043	-0.049	-0.056	-0.054	-0.061	-0.079	-0.088
8	1.066	1.068	1.07	1.073	1.067	1.065	1.064	1.067	1.064	1.06	1.058	1.055	1.056	1.058	1.064	1.06	1.059	1.062	1.06
9	0.171	0.172	0.17	0.172	0.172	0.173	0.176	0.176	0.175	0.174	0.172	0.172	0.176	0.175	0.175	0.175	0.173	0.174	0.173
10	0.404	0.402	0.41	0.419	0.417	0.418	0.421	0.423	0.429	0.428	0.433	0.434	0.433	0.433	0.431	0.43	0.431	0.43	0.421
11	0.819	0.811	0.809	0.816	0.817	0.816	0.816	0.81	0.805	0.811	0.814	0.807	0.8	0.808	0.818	0.81	0.806	0.812	0.82
12	-0.288	-0.285	-0.29	-0.295	-0.296	-0.299	-0.3	-0.297	-0.3	-0.297	-0.293	-0.294	-0.291	-0.288	-0.291	-0.291	-0.29	-0.292	-0.292
13	0.018	-0.037	-0.112	-0.159	-0.201	-0.267	-0.329	-0.381	-0.431	-0.498	-0.565	-0.646	-0.73	-0.796	-0.869	-0.935	-1.001	-1.077	-1.16
14	-1.43	-1.48	-1.521	-1.579	-1.639	-1.715	-1.753	-1.774	-1.855	-1.91	-1.939	-1.958	-1.929	-1.855	-1.795	-1.778	-1.761	-1.734	-1.698
15	0.081	0.081	0.078	0.084	0.082	0.087	0.095	0.093	0.096	0.097	0.101	0.103	0.101	0.095	0.094	0.093	0.089	0.095	0.097
16	-0.227	-0.227	-0.226	-0.228	-0.234	-0.234	-0.235	-0.237	-0.237	-0.24	-0.243	-0.246	-0.25	-0.255	-0.26	-0.262	-0.262	-0.272	-0.281
17	-0.404	-0.404	-0.403	-0.404	-0.407	-0.409	-0.408	-0.406	-0.408	-0.409	-0.407	-0.406	-0.407	-0.408	-0.411	-0.412	-0.412	-0.412	-0.411
18	-1.018	-1.02	-1.02	-1.02	-1.022	-1.021	-1.024	-1.024	-1.023	-1.024	-1.024	-1.024	-1.025	-1.024	-1.021	-1.022	-1.024	-1.026	-1.026
19	-1.482	-1.479	-1.476	-1.482	-1.481	-1.478	-1.478	-1.478	-1.476	-1.475	-1.476	-1.475	-1.473	-1.47	-1.468	-1.467	-1.468	-1.468	-1.464
20	0.146	0.148	0.145	0.147	0.147	0.144	0.147	0.145	0.144	0.144	0.141	0.142	0.141	0.136	0.135	0.139	0.139	0.134	0.133
21	-0.621	-0.556	-0.486	-0.416	-0.357	-0.319	-0.291	-0.254	-0.206	-0.169	-0.137	-0.104	-0.082	-0.059	-0.027	-0.006	0	-0.001	-0.006
22	0.961	0.963	0.963	0.964	0.966	0.967	0.967	0.965	0.964	0.966	0.965	0.96	0.966	0.966	0.965	0.974	0.979	0.981	0.983
23	0.402	0.403	0.401	0.405	0.409	0.408	0.407	0.407	0.408	0.403	0.4	0.412	0.418	0.409	0.404	0.405	0.409	0.414	0.413
24	-1.008	-1.03	-1.042	-1.067	-1.087	-1.105	-1.091	-0.997	-0.889	-0.777	-0.693	-0.649	-0.598	-0.554	-0.512	-0.472	-0.456	-0.423	-0.368
25	-0.234	-0.231	-0.236	-0.234	-0.233	-0.23	-0.228	-0.226	-0.222	-0.224	-0.222	-0.223	-0.224	-0.222	-0.218	-0.217	-0.221	-0.223	-0.216
26	1.393	1.452	1.516	1.567	1.607	1.647	1.682	1.713	1.738	1.762	1.786	1.806	1.825	1.842	1.858	1.872	1.881	1.887	1.891
27	0.103	0.105	0.103	0.101	0.105	0.096	0.101	0.1	0.095	0.098	0.098	0.095	0.087	0.093	0.095	0.096	0.101	0.1	0.102
28	-1.001	-1.006	-1.012	-1.02	-1.024	-1.029	-1.038	-1.046	-1.053	-1.059	-1.063	-1.067	-1.069	-1.069	-1.067	-1.067	-1.071	-1.073	-1.075
29	0.006	0.004	0.006	0.006	0.003	-0.001	-0.005	-0.007	-0.003	-0.003	-0.005	-0.005	-0.005	-0.01	-0.012	-0.018	-0.026	-0.027	-0.035
30	0.095	0.093	0.09	0.087	0.089	0.089	0.09	0.087	0.084	0.089	0.087	0.086	0.092	0.091	0.089	0.091	0.092	0.092	0.09
31	1.326	1.333	1.331	1.321	1.321	1.327	1.331	1.335	1.335	1.334	1.343	1.343	1.339	1.333	1.333	1.337	1.333	1.333	1.333
32	-0.12	-0.117	-0.118	-0.117	-0.117	-0.121	-0.121	-0.117	-0.118	-0.12	-0.123	-0.122	-0.119	-0.12	-0.119	-0.119	-0.122	-0.121	-0.119
33	1.287	1.284	1.282	1.278	1.276	1.27	1.268	1.262	1.25	1.246	1.245	1.24	1.234	1.228	1.226	1.228	1.22	1.218	1.216
34	0.787	0.801	0.807	0.811	0.823	0.821	0.8	0.8	0.809	0.8	0.801	0.792	0.788	0.806	0.81	0.814	0.814	0.822	0.826
35	-0.187	-0.185	-0.178	-0.178	-0.179	-0.179	-0.166	-0.162	-0.171	-0.175	-0.181	-0.175	-0.172	-0.175	-0.177	-0.177	-0.175	-0.185	-0.189
36	0.604	0.608	0.608	0.608	0.608	0.603	0.603	0.605	0.606	0.605	0.607	0.607	0.607	0.608	0.611	0.616	0.619	0.62	0.621
37	0.223	0.209	0.193	0.181	0.165	0.154	0.139	0.132	0.123	0.102	0.087	0.077	0.07	0.057	0.047	0.037	0.028	0.016	0.001
38	0.203	0.27	0.334	0.38	0.443	0.498	0.581	0.675	0.78	0.893	1.012	1.157	1.288	1.442	1.565	1.649	1.696	1.679	1.617
39	-0.113	-0.131	-0.128	-0.131	-0.13	-0.127	-0.131	-0.127	-0.118	-0.117	-0.124	-0.131	-0.135	-0.134	-0.13	-0.123	-0.126	-0.13	-0.12
40	0.169	0.172	0.177	0.179	0.176	0.176	0.178	0.174	0.177	0.18	0.175	0.171	0.172	0.178	0.176	0.175	0.175	0.174	0.176
41	0.09	0.09	0.09	0.091	0.09	0.091	0.093	0.093	0.092	0.093	0.095	0.095	0.095	0.096	0.096	0.094	0.093	0.092	0.095
42	0.003	0.002	0.002	0.003	0.003	0.006	0.004	0.003	0	-0.008	-0.011	-0.013	-0.017	-0.016	-0.018	-0.022	-0.028	-0.032	-0.036
43	-0.057	-0.056	-0.058	-0.059	-0.059	-0.059	-0.061	-0.06	-0.059	-0.059	-0.06	-0.063	-0.064	-0.062	-0.061	-0.066	-0.067	-0.066	-0.064
44	1.363	1.361	1.362	1.364	1.363	1.362	1.362	1.365	1.362	1.358	1.357	1.357	1.359	1.361	1.362	1.362	1.361	1.359	1.357
45	-0.178	-0.207	-0.236	-0.27	-0.329	-0.386	-0.425	-0.477	-0.546	-0.6	-0.649	-0.731	-0.792	-0.835	-0.893	-0.948	-0.996	-1.045	-1.105
46	-0.447	-0.447	-0.456	-0.46	-0.467	-0.463	-0.459	-0.458	-0.456	-0.456	-0.459	-0.468	-0.478	-0.488	-0.487	-0.499	-0.505	-0.501	-0.504
47	-1.94	-1.941	-1.938	-1.935	-1.925	-1.921	-1.935	-1.925	-1.923	-1.931	-1.943	-1.946	-1.94	-1.939	-1.931	-1.933	-1.94	-1.941	-1.933
48	-1.831	-1.827	-1.83	-1.833	-1.833	-1.829	-1.823	-1.822	-1.821	-1.819	-1.817	-1.816	-1.822	-1.821	-1.82	-1.819	-1.819	-1.82	-1.823
49	0.144	0.141	0.137	0.13	0.132	0.135	0.138	0.14	0.141	0.144	0.143	0.141	0.143	0.15	0.148	0.142	0.144	0.146	0.142
50	-0.06	-0.059	-0.057	-0.06	-0.06	-0.06	-0.059	-0.059	-0.059	-0.06	-0.063	-0.066	-0.065	-0.064	-0.061	-0.059	-0.06	-0.061	-0.059
51	0.143	0.14	0.143	0.151	0.151	0.151	0.158	0.157	0.158	0.163	0.163	0.165	0.161	0.158	0.165	0.159	0.16	0.164	0.158
52	0.44	0.44	0.435	0.44	0.442	0.444	0.45	0.451	0.454	0.463	0.468	0.466	0.464	0.469	0.47	0.472	0.478	0.481	0.485
53	-0.378	-0.376	-0.378	-0.376	-0.375	-0.38	-0.38	-0.379	-0.381	-0.383	-0.385	-0.38	-0.377	-0.374	-0.374	-0.382	-0.386	-0.383	-0.382
54	-1.543	-1.543	-1.543	-1.542	-1.541	-1.54	-1.539	-1.54	-1.54	-1.541	-1.542	-1.541	-1.545	-1.545	-1.543	-1.541	-1.542	-1.546	-1.547
55	0.361	0.358	0.362	0.363	0.356	0.359	0.358	0.355	0.356	0.357	0.365	0.365	0.363	0.361	0.362	0.364	0.353	0.356	0.365
56	0.242	0.241	0.24	0.24	0.238	0.24	0.242	0.24	0.239	0.237	0.236	0.238	0.238	0.236	0.236	0.236	0.235	0.235	0.233
57	0.275	0.294	0.299	0.299	0.305	0.304	0.314	0.325	0.328	0.327	0.328	0.33	0.334	0.338	0.337	0.339	0.342	0.344	0.336
58	-0.032	-0.029	-0.031	-0.029	-0.029	-0.022	-0.016	-0.004	0.012	0.041	0.07	0.096	0.126	0.153	0.182	0.202	0.223	0.253	0.281
59	-1.141	-1.133	-1.066	-1.032	-0.996	-0.974	-0.948	-0.912	-0.889	-0.866	-0.87	-0.854	-0.827	-0.813	-0.797	-0.778	-0.764	-0.762	-0.755
60	-0.926	-0.903	-0.896	-0.879	-0.862	-0.85	-0.826	-0.806	-0.79	-0.772	-0.763	-0.752	-0.741	-0.725	-0.716	-0.715	-0.717	-0.721	-0.718
61	0.178	0.172	0.169	0.165	0.153	0.143	0.133	0.127	0.121	0.117	0.116	0.104	0.091	0.083	0.068	0.054	0.042	0.03	0.025
62	0.258	0.259	0.259	0.259	0.258	0.262	0.265	0.264	0.264	0.265	0.266	0.269	0.271	0.273	0.273	0.275	0.278	0.279	0.281
63	0.649	0.669	0.69	0.717	0.743	0.759	0.78	0.804	0.824	0.841	0.854	0.867	0.872	0.877	0.884	0.886	0.89	0.892	0.892
64	-1.056	-1.06	-1.061	-1.062	-1.066	-1.073	-1.081	-1.087	-1.091	-1.095	-1.101	-1.102	-1.103	-1.1	-1.098	-1.098	-1.093	-1.091	-1.088
65	0.312	0.314	0.316	0.315	0.318	0.319	0.316	0.315	0.32	0.32	0.316	0.315	0.319	0.316	0.317	0.321	0.318	0.32	0.322
66	0.321	0.328	0.332	0.341	0.344	0.352	0.359	0.361	0.365	0.364	0.369	0.38	0.385	0.389	0.391	0.396	0.397	0.402	0.407
67	0.124	0.126	0.125	0.123	0.121	0.121	0.123	0.124	0.121	0.123	0.124	0.124	0.128	0.126	0.128	0.13	0.131	0.135	0.136
68	0.133	0.128	0.137	0.138	0.138	0.138	0.145	0.145	0.148	0.153	0.157	0.158	0.157	0.158	0.16	0.168	0.165	0.165	0.169
69	0.794	0.796	0.8	0.802	0.804	0.805	0.805	0.805	0.807	0.811	0.811	0.812	0.815	0.816	0.815	0.817	0.822	0.826	0.828
70	-0.798	-0.796	-0.799	-0.799	-0.794	-0.797	-0.798	-0.794	-0.793	-0.794	-0.793	-0.792	-0.793	-0.793	-0.792	-0.793	-0.794	-0.794	-0.795
71	0.133	0.134	0.131	0.131	0.127	0.126	0.128	0.124	0.123	0.125	0.122	0.126	0.129	0.124	0.121	0.124	0.128	0.13	0.134
72	-0.069	-0.066	-0.065	-0.067	-0.069	-0.071	-0.076	-0.075	-0.072	-0.068	-0.065	-0.062	-0.05	-0.043	-0.042	-0.04	-0.037	-0.036	-0.036
73	0.483	0.486	0.485	0.486	0.486	0.487	0.486	0.487	0.487	0.486	0.486	0.486	0.489	0.487	0.488	0.489	0.486	0.487	0.486
74	0.872	0.876	0.881	0.88	0.885	0.889	0.895	0.902	0.906	0.913	0.92	0.925	0.931	0.937	0.941	0.941	0.94	0.939	0.938
75	1.571	1.577	1.575	1.567	1.563	1.569	1.571	1.577	1.573	1.567	1.574	1.571	1.58	1.579	1.575	1.582	1.581	1.577	1.567
76	0.127	0.125	0.122	0.117	0.122	0.119	0.114	0.116	0.115	0.116	0.118	0.122	0.121	0.122	0.127	0.123	0.124	0.127	0.126
77	-0.203	-0.204	-0.204	-0.203	-0.2	-0.195	-0.191	-0.186	-0.184	-0.181	-0.177	-0.174	-0.166	-0.163	-0.166	-0.169	-0.166	-0.166	-0.165
78	0.502	0.496	0.494	0.505	0.513	0.51	0.506	0.502	0.509	0.518	0.532	0.54	0.54	0.548	0.554	0.559	0.55	0.552	0.559
79	1.512	1.509	1.511	1.51	1.516	1.513	1.515	1.511	1.512	1.53	1.524	1.522	1.523	1.519	1.526	1.526	1.526	1.526	1.53
80	-0.119	-0.118	-0.118	-0.116	-0.117	-0.12	-0.119	-0.12	-0.122	-0.124	-0.124	-0.124	-0.124	-0.121	-0.121	-0.121	-0.119	-0.119	-0.122
81	0.165	0.175	0.177	0.183	0.186	0.188	0.187	0.193	0.195	0.195	0.198	0.196	0.195	0.194	0.199	0.202	0.206	0.205	0.202
82	0.093	0.095	0.096	0.1	0.101	0.1	0.103	0.108	0.114	0.115	0.112	0.111	0.116	0.116	0.119	0.117	0.117	0.12	0.116
83	0.029	0.03	0.03	0.031	0.031	0.03	0.033	0.034	0.035	0.036	0.037	0.039	0.041	0.042	0.045	0.045	0.044	0.044	0.043
84	-0.059	-0.06	-0.062	-0.062	-0.066	-0.07	-0.07	-0.067	-0.068	-0.07	-0.07	-0.071	-0.07	-0.071	-0.073	-0.074	-0.072	-0.073	-0.074
85	0.16	0.168	0.176	0.184	0.191	0.199	0.205	0.211	0.219	0.227	0.234	0.241	0.252	0.255	0.26	0.27	0.274	0.28	0.283
86	-0.166	-0.166	-0.166	-0.161	-0.159	-0.157	-0.156	-0.158	-0.155	-0.157	-0.157	-0.16	-0.16	-0.158	-0.158	-0.157	-0.157	-0.155	-0.155
87	0.046	0.05	0.052	0.059	0.066	0.066	0.074	0.084	0.088	0.092	0.095	0.1	0.108	0.114	0.117	0.12	0.124	0.13	0.14
88	-0.326	-0.314	-0.29	-0.298	-0.306	-0.316	-0.319	-0.316	-0.311	-0.288	-0.262	-0.238	-0.209	-0.17	-0.133	-0.102	-0.076	-0.06	-0.04
89	-0.098	-0.096	-0.094	-0.095	-0.096	-0.097	-0.1	-0.1	-0.098	-0.1	-0.1	-0.101	-0.101	-0.096	-0.097	-0.101	-0.101	-0.102	-0.102
90	0.402	0.404	0.407	0.412	0.416	0.42	0.423	0.429	0.433	0.435	0.443	0.448	0.452	0.454	0.459	0.469	0.474	0.48	0.484
91	0.265	0.263	0.265	0.263	0.26	0.26	0.264	0.264	0.26	0.263	0.268	0.267	0.267	0.264	0.263	0.266	0.262	0.264	0.267

3. Partial Train Data [From V5]

	BYB	BYC	BYD	BYE	BYF	BYG	BYH	BYI	BYJ	BYK	BYL	BYM	BYN	BYO	BYP	BYQ	BYR	BYS	BYT
1	'0:02.002'	'0:02.003'	'0:02.004'	'0:02.005'	'0:02.006'	'0:02.007'	'0:02.008'	'0:02.009'	'0:02.010'	'0:02.011'	'0:02.012'	'0:02.013'	'0:02.014'	'0:02.015'	'0:02.016'	'0:02.017'	'0:02.018'	'0:02.019'	'0:02.020'
2	-0.384	-0.36	-0.341	-0.33	-0.319	-0.305	-0.293	-0.286	-0.272	-0.253	-0.242	-0.233	-0.212	-0.191	-0.175	-0.157	-0.141	-0.126	-0.113
3	-0.261	-0.255	-0.252	-0.242	-0.224	-0.213	-0.21	-0.208	-0.204	-0.202	-0.201	-0.198	-0.196	-0.182	-0.164	-0.143	-0.107	-0.071	-0.022
4	0.011	0.019	0.032	0.038	0.037	0.035	0.032	0.025	0.015	0.009	0.004	0	-0.006	-0.004	0.007	0.011	0.019	0.026	0.029
5	-0.882	-0.881	-0.879	-0.879	-0.877	-0.875	-0.873	-0.872	-0.871	-0.868	-0.868	-0.868	-0.869	-0.868	-0.866	-0.867	-0.868	-0.866	-0.866
6	0.212	0.211	0.209	0.207	0.205	0.204	0.207	0.206	0.207	0.208	0.209	0.214	0.216	0.218	0.218	0.222	0.224	0.224	0.226
7	0.221	0.217	0.223	0.226	0.229	0.229	0.232	0.229	0.222	0.222	0.234	0.235	0.227	0.225	0.218	0.22	0.218	0.208	0.196
8	1.208	1.208	1.212	1.212	1.208	1.204	1.2	1.203	1.204	1.201	1.2	1.196	1.196	1.198	1.202	1.202	1.202	1.202	1.204
9	0.171	0.171	0.17	0.17	0.17	0.172	0.173	0.171	0.172	0.17	0.17	0.171	0.17	0.169	0.171	0.173	0.171	0.172	0.173
10	0.303	0.302	0.31	0.317	0.318	0.319	0.324	0.327	0.33	0.332	0.333	0.329	0.322	0.319	0.319	0.322	0.324	0.322	0.321
11	0.623	0.616	0.61	0.613	0.613	0.616	0.613	0.604	0.598	0.602	0.606	0.599	0.593	0.602	0.613	0.614	0.609	0.611	0.616
12	0.159	0.158	0.155	0.15	0.147	0.147	0.146	0.144	0.142	0.142	0.141	0.142	0.142	0.14	0.137	0.137	0.139	0.139	0.137
13	0.346	0.349	0.351	0.341	0.354	0.388	0.415	0.43	0.446	0.467	0.48	0.497	0.509	0.525	0.544	0.559	0.579	0.593	0.6
14	0.082	0.123	0.174	0.216	0.236	0.261	0.269	0.276	0.318	0.345	0.329	0.297	0.237	0.141	0.035	-0.046	-0.118	-0.192	-0.246
15	-0.14	-0.138	-0.142	-0.136	-0.136	-0.128	-0.122	-0.114	-0.11	-0.114	-0.11	-0.113	-0.116	-0.12	-0.119	-0.121	-0.128	-0.127	-0.132
16	0.113	0.108	0.109	0.111	0.106	0.108	0.11	0.109	0.113	0.115	0.119	0.121	0.12	0.121	0.121	0.123	0.126	0.127	0.125
17	1.49	1.491	1.489	1.49	1.488	1.481	1.484	1.484	1.48	1.482	1.484	1.483	1.482	1.479	1.477	1.477	1.479	1.478	1.48
18	-0.855	-0.857	-0.858	-0.859	-0.86	-0.861	-0.86	-0.861	-0.862	-0.862	-0.861	-0.86	-0.86	-0.856	-0.856	-0.857	-0.857	-0.859	-0.856
19	-0.182	-0.18	-0.182	-0.181	-0.18	-0.176	-0.175	-0.178	-0.176	-0.175	-0.177	-0.176	-0.173	-0.173	-0.176	-0.175	-0.176	-0.176	-0.175
20	0.15	0.152	0.149	0.148	0.15	0.147	0.147	0.147	0.146	0.149	0.147	0.146	0.146	0.143	0.14	0.138	0.136	0.133	0.131
21	-0.112	-0.101	-0.085	-0.069	-0.052	-0.043	-0.036	-0.025	-0.007	0.011	0.029	0.052	0.066	0.075	0.093	0.106	0.11	0.111	0.107
22	1.365	1.365	1.363	1.361	1.363	1.361	1.358	1.351	1.345	1.347	1.343	1.337	1.341	1.341	1.339	1.343	1.345	1.339	1.335
23	0.099	0.096	0.098	0.102	0.105	0.103	0.103	0.104	0.105	0.098	0.098	0.107	0.11	0.104	0.098	0.099	0.103	0.107	0.107
24	0.245	0.12	0.037	-0.054	-0.139	-0.2	-0.237	-0.23	-0.217	-0.222	-0.221	-0.217	-0.207	-0.207	-0.206	-0.207	-0.209	-0.189	-0.161
25	-0.049	-0.045	-0.041	-0.036	-0.03	-0.019	-0.009	0.001	0.007	0.011	0.015	0.019	0.027	0.034	0.042	0.048	0.047	0.049	0.057
26	1.79	1.82	1.85	1.873	1.893	1.913	1.931	1.944	1.952	1.962	1.974	1.981	1.99	1.995	1.998	2.005	2.007	2.01	2.01
27	-0.061	-0.06	-0.064	-0.072	-0.07	-0.078	-0.08	-0.085	-0.085	-0.082	-0.085	-0.079	-0.085	-0.084	-0.086	-0.089	-0.088	-0.089	-0.082
28	-0.852	-0.853	-0.858	-0.862	-0.865	-0.869	-0.872	-0.878	-0.88	-0.878	-0.882	-0.884	-0.886	-0.891	-0.894	-0.895	-0.9	-0.905	-0.909
29	0.214	0.215	0.21	0.206	0.205	0.201	0.201	0.202	0.192	0.188	0.194	0.19	0.184	0.185	0.183	0.177	0.167	0.166	0.163
30	0.118	0.121	0.119	0.112	0.108	0.111	0.109	0.104	0.107	0.11	0.107	0.107	0.112	0.109	0.109	0.112	0.113	0.114	0.115
31	0.75	0.752	0.754	0.747	0.745	0.748	0.752	0.752	0.75	0.75	0.753	0.758	0.755	0.748	0.748	0.747	0.742	0.743	0.745
32	0.197	0.202	0.201	0.201	0.2	0.196	0.194	0.199	0.201	0.198	0.199	0.198	0.198	0.199	0.199	0.198	0.196	0.198	0.201
33	1.614	1.619	1.619	1.619	1.621	1.621	1.621	1.619	1.617	1.618	1.618	1.618	1.615	1.613	1.615	1.621	1.619	1.619	1.623
34	0.818	0.834	0.833	0.839	0.861	0.859	0.842	0.84	0.846	0.832	0.832	0.832	0.823	0.842	0.849	0.851	0.857	0.87	0.877
35	-0.188	-0.187	-0.182	-0.186	-0.186	-0.181	-0.167	-0.162	-0.171	-0.175	-0.178	-0.17	-0.168	-0.178	-0.181	-0.178	-0.178	-0.186	-0.184
36	0.778	0.778	0.777	0.779	0.783	0.781	0.781	0.78	0.781	0.783	0.785	0.786	0.785	0.784	0.783	0.787	0.79	0.79	0.791
37	-0.155	-0.155	-0.16	-0.157	-0.158	-0.158	-0.162	-0.161	-0.162	-0.168	-0.17	-0.173	-0.175	-0.181	-0.182	-0.183	-0.181	-0.181	-0.187
38	-0.151	-0.123	-0.085	-0.048	-0.018	0.031	0.097	0.166	0.241	0.34	0.432	0.508	0.588	0.654	0.718	0.781	0.842	0.883	0.891
39	-0.084	-0.098	-0.092	-0.09	-0.092	-0.087	-0.091	-0.088	-0.075	-0.074	-0.08	-0.084	-0.086	-0.084	-0.076	-0.069	-0.071	-0.075	-0.067
40	0.128	0.135	0.138	0.141	0.139	0.138	0.139	0.136	0.14	0.143	0.138	0.134	0.136	0.141	0.14	0.139	0.139	0.138	0.14
41	0.198	0.195	0.198	0.198	0.198	0.2	0.2	0.2	0.198	0.198	0.2	0.201	0.203	0.202	0.198	0.195	0.194	0.194	0.196
42	-0.061	-0.059	-0.056	-0.055	-0.052	-0.05	-0.051	-0.048	-0.048	-0.051	-0.051	-0.052	-0.053	-0.054	-0.059	-0.061	-0.064	-0.067	-0.07
43	-0.21	-0.211	-0.209	-0.207	-0.208	-0.205	-0.207	-0.206	-0.205	-0.203	-0.204	-0.203	-0.202	-0.208	-0.206	-0.212	-0.213	-0.209	-0.209
44	2.022	2.023	2.025	2.026	2.026	2.025	2.026	2.027	2.023	2.022	2.021	2.022	2.022	2.022	2.024	2.023	2.022	2.022	2.022
45	0.35	0.418	0.477	0.52	0.557	0.588	0.621	0.651	0.653	0.627	0.593	0.562	0.53	0.477	0.413	0.362	0.3	0.224	0.141

	BYB	BYC	BYD	BYE	BYF	BYG	BYH	BYI	BYJ	BYK	BYL	BYM	BYN	BYO	BYP	BYQ	BYR	BYS	BYT
46	-0.182	-0.18	-0.18	-0.18	-0.182	-0.176	-0.172	-0.172	-0.175	-0.178	-0.181	-0.191	-0.203	-0.213	-0.21	-0.224	-0.232	-0.224	-0.229
47	-1.53	-1.526	-1.522	-1.521	-1.512	-1.51	-1.522	-1.511	-1.511	-1.521	-1.533	-1.536	-1.53	-1.527	-1.522	-1.522	-1.528	-1.53	-1.521
48	-1.91	-1.91	-1.914	-1.917	-1.917	-1.915	-1.915	-1.917	-1.919	-1.919	-1.918	-1.921	-1.924	-1.927	-1.929	-1.929	-1.929	-1.927	-1.928
49	0.152	0.147	0.145	0.145	0.148	0.152	0.155	0.156	0.154	0.154	0.147	0.142	0.146	0.151	0.151	0.147	0.148	0.151	0.153
50	-0.028	-0.029	-0.027	-0.028	-0.029	-0.029	-0.029	-0.029	-0.027	-0.028	-0.033	-0.036	-0.035	-0.035	-0.033	-0.029	-0.029	-0.03	-0.03
51	-0.068	-0.067	-0.064	-0.06	-0.059	-0.056	-0.05	-0.05	-0.05	-0.051	-0.05	-0.045	-0.047	-0.05	-0.048	-0.048	-0.044	-0.041	-0.049
52	0.344	0.342	0.334	0.333	0.334	0.331	0.331	0.333	0.334	0.334	0.333	0.334	0.33	0.328	0.331	0.327	0.327	0.33	0.331
53	2.804	2.803	2.803	2.805	2.804	2.801	2.805	2.805	2.802	2.801	2.801	2.803	2.803	2.803	2.803	2.8	2.8	2.804	2.805
54	0.081	0.08	0.08	0.083	0.083	0.084	0.086	0.084	0.084	0.082	0.082	0.082	0.081	0.081	0.081	0.084	0.082	0.08	0.08
55	0.318	0.321	0.326	0.328	0.325	0.326	0.324	0.321	0.323	0.327	0.333	0.335	0.334	0.332	0.334	0.334	0.327	0.327	0.335
56	0.413	0.413	0.412	0.413	0.414	0.413	0.414	0.415	0.414	0.411	0.411	0.411	0.41	0.409	0.41	0.411	0.409	0.407	0.408
57	-0.124	-0.114	-0.102	-0.106	-0.11	-0.107	-0.11	-0.108	-0.103	-0.11	-0.116	-0.121	-0.126	-0.125	-0.126	-0.123	-0.117	-0.119	-0.119
58	-0.499	-0.5	-0.501	-0.504	-0.513	-0.518	-0.523	-0.532	-0.541	-0.554	-0.571	-0.588	-0.602	-0.614	-0.626	-0.636	-0.649	-0.663	-0.677
59	-0.758	-0.795	-0.782	-0.77	-0.771	-0.772	-0.774	-0.768	-0.766	-0.771	-0.777	-0.772	-0.759	-0.745	-0.738	-0.728	-0.719	-0.717	-0.712
60	-0.901	-0.889	-0.882	-0.875	-0.859	-0.838	-0.816	-0.793	-0.779	-0.765	-0.757	-0.756	-0.749	-0.739	-0.734	-0.736	-0.732	-0.729	-0.732
61	-0.788	-0.792	-0.79	-0.786	-0.791	-0.798	-0.803	-0.802	-0.799	-0.796	-0.786	-0.782	-0.784	-0.781	-0.781	-0.785	-0.79	-0.794	-0.788
62	-0.383	-0.384	-0.381	-0.382	-0.382	-0.379	-0.377	-0.38	-0.377	-0.376	-0.374	-0.37	-0.37	-0.368	-0.369	-0.365	-0.363	-0.363	-0.359
63	-0.192	-0.167	-0.143	-0.118	-0.093	-0.073	-0.058	-0.038	-0.019	-0.002	0.015	0.032	0.043	0.054	0.066	0.071	0.077	0.083	0.085
64	-0.528	-0.535	-0.537	-0.54	-0.545	-0.55	-0.554	-0.556	-0.557	-0.56	-0.558	-0.558	-0.56	-0.56	-0.562	-0.562	-0.561	-0.564	-0.565
65	0.088	0.086	0.085	0.086	0.087	0.089	0.088	0.088	0.087	0.087	0.087	0.087	0.091	0.092	0.092	0.089	0.086	0.089	0.093
66	0.322	0.326	0.331	0.336	0.337	0.342	0.348	0.354	0.36	0.36	0.363	0.369	0.377	0.381	0.386	0.394	0.397	0.403	0.41
67	0.206	0.204	0.204	0.202	0.199	0.199	0.198	0.197	0.193	0.19	0.19	0.189	0.189	0.184	0.18	0.18	0.177	0.175	0.172
68	-0.117	-0.114	-0.109	-0.107	-0.106	-0.104	-0.102	-0.101	-0.098	-0.093	-0.088	-0.083	-0.081	-0.078	-0.079	-0.076	-0.075	-0.076	-0.075
69	0.729	0.73	0.733	0.733	0.734	0.734	0.733	0.733	0.735	0.737	0.736	0.738	0.74	0.742	0.743	0.741	0.745	0.747	0.746
70	-0.716	-0.719	-0.719	-0.72	-0.718	-0.717	-0.72	-0.718	-0.715	-0.716	-0.717	-0.717	-0.715	-0.715	-0.715	-0.716	-0.715	-0.716	-0.717
71	0.095	0.089	0.086	0.086	0.085	0.085	0.082	0.076	0.078	0.08	0.076	0.082	0.087	0.084	0.084	0.086	0.089	0.09	0.091
72	-0.192	-0.19	-0.192	-0.193	-0.196	-0.2	-0.203	-0.204	-0.203	-0.199	-0.193	-0.187	-0.18	-0.176	-0.176	-0.176	-0.174	-0.173	-0.173
73	-2.005	-2.003	-2.003	-2.002	-2.002	-2	-1.999	-2	-1.998	-1.998	-2	-2	-1.998	-1.996	-1.998	-2	-1.998	-1.998	-1.997
74	0.268	0.279	0.289	0.293	0.299	0.303	0.306	0.314	0.32	0.327	0.334	0.34	0.347	0.355	0.366	0.376	0.382	0.389	0.395
75	0.146	0.16	0.169	0.178	0.192	0.204	0.217	0.234	0.248	0.263	0.284	0.301	0.313	0.321	0.33	0.343	0.352	0.359	0.365
76	0.263	0.261	0.26	0.256	0.263	0.258	0.252	0.255	0.252	0.252	0.256	0.26	0.262	0.262	0.261	0.256	0.259	0.261	0.261
77	-0.18	-0.186	-0.188	-0.192	-0.191	-0.189	-0.189	-0.186	-0.185	-0.188	-0.188	-0.185	-0.181	-0.179	-0.181	-0.184	-0.182	-0.183	-0.18
78	0.349	0.35	0.351	0.358	0.357	0.356	0.357	0.357	0.363	0.369	0.378	0.384	0.386	0.393	0.396	0.398	0.399	0.399	0.403
79	-0.043	-0.044	-0.044	-0.032	-0.024	-0.028	-0.018	-0.015	-0.014	-0.001	0.005	0.007	0.015	0.018	0.019	0.031	0.031	0.036	0.046
80	-0.467	-0.463	-0.465	-0.466	-0.469	-0.472	-0.473	-0.475	-0.477	-0.478	-0.479	-0.479	-0.478	-0.477	-0.477	-0.476	-0.476	-0.479	-0.479
81	0.288	0.295	0.301	0.303	0.307	0.306	0.31	0.318	0.319	0.321	0.321	0.32	0.322	0.321	0.323	0.328	0.333	0.331	0.33
82	0.065	0.067	0.073	0.077	0.082	0.086	0.091	0.099	0.104	0.105	0.109	0.113	0.116	0.119	0.126	0.129	0.132	0.135	0.138
83	0.001	0.001	0.002	0.002	0.002	0.004	0.007	0.011	0.013	0.013	0.015	0.018	0.02	0.022	0.025	0.026	0.025	0.025	0.024
84	-0.026	-0.033	-0.026	-0.026	-0.028	-0.028	-0.03	-0.029	-0.029	-0.032	-0.032	-0.032	-0.032	-0.033	-0.033	-0.033	-0.034	-0.034	-0.035
85	-0.293	-0.287	-0.281	-0.276	-0.27	-0.263	-0.261	-0.256	-0.252	-0.247	-0.238	-0.233	-0.224	-0.221	-0.217	-0.208	-0.203	-0.199	-0.196
86	-0.186	-0.188	-0.189	-0.191	-0.19	-0.189	-0.188	-0.188	-0.189	-0.189	-0.19	-0.19	-0.19	-0.189	-0.189	-0.191	-0.192	-0.191	-0.191
87	0.25	0.254	0.257	0.26	0.269	0.273	0.277	0.276	0.282	0.291	0.291	0.294	0.3	0.307	0.311	0.314	0.319	0.327	0.335
88	0.23	0.123	0.023	-0.065	-0.145	-0.209	-0.242	-0.262	-0.267	-0.273	-0.28	-0.281	-0.272	-0.247	-0.224	-0.192	-0.169	-0.154	-0.133
89	-0.084	-0.082	-0.084	-0.086	-0.082	-0.081	-0.085	-0.085	-0.081	-0.075	-0.079	-0.077	-0.074	-0.073	-0.07	-0.07	-0.065	-0.068	-0.073
90	0.587	0.589	0.592	0.592	0.595	0.597	0.599	0.607	0.609	0.609	0.616	0.621	0.622	0.622	0.626	0.632	0.636	0.64	0.643
91	0.27	0.272	0.274	0.271	0.267	0.268	0.27	0.271	0.272	0.274	0.275	0.276	0.276	0.271	0.272	0.274	0.271	0.271	0.272

4. Partial Train Data [From aVL]

	DKO	DKP	DKQ	DKR	DKS	DKT	DKU	DKV	DKW	DKX	DKY	DKZ	DLA	DLB	DLC	DLD	DLE	DLF	DLG
1	'0:03.003'	'0:03.004'	'0:03.005'	'0:03.006'	'0:03.007'	'0:03.008'	'0:03.009'	'0:03.010'	'0:03.011'	'0:03.012'	'0:03.013'	'0:03.014'	'0:03.015'	'0:03.016'	'0:03.017'	'0:03.018'	'0:03.019'	'0:03.020'	'0:03.021'
2	0.291	0.305	0.308	0.321	0.336	0.342	0.34	0.353	0.352	0.342	0.349	0.356	0.356	0.366	0.366	0.36	0.366	0.367	0.364
3	-0.633	-0.618	-0.612	-0.618	-0.621	-0.627	-0.625	-0.618	-0.608	-0.593	-0.585	-0.572	-0.566	-0.564	-0.551	-0.547	-0.534	-0.517	-0.493
4	-0.557	-0.559	-0.554	-0.549	-0.545	-0.538	-0.546	-0.561	-0.572	-0.577	-0.577	-0.581	-0.581	-0.584	-0.581	-0.583	-0.585	-0.574	-0.574
5	-0.897	-0.898	-0.897	-0.9	-0.9	-0.899	-0.901	-0.902	-0.902	-0.903	-0.905	-0.909	-0.913	-0.913	-0.913	-0.917	-0.921	-0.917	-0.92
6	0.017	0.011	0.012	0.01	0.01	0.004	-0.001	-0.003	-0.006	-0.004	-0.007	-0.01	-0.006	-0.003	-0.001	0.005	0.007	0.007	0.013
7	-0.163	-0.137	-0.168	-0.187	-0.194	-0.197	-0.188	-0.177	-0.164	-0.163	-0.171	-0.181	-0.19	-0.186	-0.179	-0.195	-0.208	-0.205	-0.211
8	0.351	0.361	0.358	0.354	0.346	0.348	0.347	0.346	0.353	0.35	0.344	0.343	0.344	0.345	0.344	0.349	0.349	0.341	0.343
9	0.349	0.346	0.345	0.349	0.347	0.347	0.353	0.352	0.352	0.349	0.347	0.349	0.349	0.348	0.35	0.348	0.346	0.345	0.342
10	-0.096	-0.096	-0.087	-0.088	-0.093	-0.089	-0.086	-0.083	-0.085	-0.084	-0.082	-0.083	-0.084	-0.086	-0.084	-0.087	-0.097	-0.097	-0.087
11	0.665	0.667	0.661	0.641	0.646	0.639	0.643	0.657	0.669	0.676	0.667	0.671	0.688	0.68	0.659	0.66	0.657	0.657	0.666
12	-0.119	-0.119	-0.124	-0.128	-0.133	-0.138	-0.139	-0.134	-0.129	-0.121	-0.123	-0.127	-0.126	-0.128	-0.129	-0.132	-0.128	-0.125	-0.122
13	-0.04	-0.053	-0.052	-0.013	0.018	0.04	0.068	0.097	0.136	0.187	0.25	0.286	0.321	0.358	0.376	0.402	0.429	0.47	0.52
14	0.878	0.906	0.957	0.998	1.026	1.056	1.081	1.122	1.147	1.157	1.156	1.143	1.155	1.187	1.209	1.216	1.226	1.232	1.223
15	-0.015	-0.013	-0.016	-0.011	-0.006	0.013	0.008	-0.003	0.007	0.003	-0.009	-0.007	0.001	-0.005	-0.024	-0.021	-0.015	-0.013	-0.003
16	-0.264	-0.263	-0.259	-0.261	-0.262	-0.261	-0.258	-0.256	-0.252	-0.249	-0.251	-0.254	-0.254	-0.249	-0.248	-0.252	-0.254	-0.25	-0.245
17	0.397	0.398	0.402	0.406	0.399	0.397	0.4	0.395	0.389	0.387	0.388	0.385	0.383	0.382	0.384	0.389	0.386	0.383	0.385
18	1.226	1.226	1.228	1.228	1.223	1.222	1.224	1.222	1.222	1.224	1.226	1.224	1.222	1.23	1.235	1.232	1.23	1.231	1.234
19	-0.529	-0.525	-0.52	-0.531	-0.54	-0.534	-0.537	-0.539	-0.536	-0.534	-0.537	-0.534	-0.532	-0.542	-0.535	-0.532	-0.533	-0.529	-0.531
20	0.101	0.105	0.107	0.107	0.113	0.11	0.11	0.116	0.115	0.108	0.104	0.101	0.099	0.101	0.103	0.1	0.096	0.096	0.093
21	-0.156	-0.158	-0.154	-0.163	-0.164	-0.154	-0.158	-0.18	-0.191	-0.192	-0.185	-0.181	-0.2	-0.214	-0.212	-0.2	-0.194	-0.185	-0.182
22	1.56	1.556	1.55	1.568	1.577	1.563	1.563	1.558	1.549	1.546	1.55	1.558	1.565	1.562	1.56	1.565	1.568	1.576	1.574
23	0.191	0.185	0.185	0.191	0.184	0.183	0.186	0.177	0.161	0.169	0.174	0.158	0.167	0.183	0.189	0.185	0.194	0.204	0.2
24	1.139	1.02	0.941	0.879	0.806	0.743	0.689	0.657	0.648	0.634	0.611	0.584	0.555	0.54	0.534	0.53	0.524	0.523	0.526
25	-0.336	-0.338	-0.342	-0.344	-0.347	-0.348	-0.344	-0.34	-0.346	-0.349	-0.344	-0.34	-0.332	-0.33	-0.331	-0.334	-0.34	-0.334	-0.33
26	1.946	1.925	1.915	1.891	1.877	1.855	1.83	1.817	1.808	1.806	1.808	1.802	1.802	1.806	1.804	1.806	1.804	1.806	1.81
27	-0.532	-0.528	-0.531	-0.532	-0.531	-0.523	-0.516	-0.516	-0.522	-0.52	-0.517	-0.521	-0.518	-0.527	-0.524	-0.516	-0.528	-0.538	-0.535
28	-0.169	-0.163	-0.167	-0.169	-0.165	-0.159	-0.162	-0.162	-0.163	-0.174	-0.175	-0.177	-0.181	-0.171	-0.166	-0.17	-0.168	-0.165	-0.165
29	0.328	0.327	0.327	0.327	0.322	0.326	0.317	0.309	0.309	0.316	0.322	0.32	0.317	0.315	0.318	0.315	0.312	0.315	0.322
30	0.026	0.025	0.02	0.019	0.019	0.018	0.016	0.006	0.005	0.013	0.017	0.015	0.016	0.017	0.022	0.026	0.025	0.025	0.019
31	1.255	1.256	1.251	1.24	1.246	1.246	1.241	1.248	1.258	1.265	1.26	1.25	1.253	1.255	1.242	1.241	1.246	1.244	1.245
32	-0.13	-0.126	-0.124	-0.126	-0.118	-0.11	-0.106	-0.117	-0.118	-0.105	-0.104	-0.113	-0.119	-0.122	-0.128	-0.126	-0.117	-0.119	-0.122
33	-0.445	-0.428	-0.439	-0.448	-0.451	-0.459	-0.464	-0.464	-0.458	-0.461	-0.468	-0.475	-0.489	-0.489	-0.481	-0.484	-0.493	-0.493	-0.491
34	-0.122	-0.117	-0.117	-0.122	-0.117	-0.12	-0.124	-0.123	-0.141	-0.157	-0.167	-0.174	-0.166	-0.142	-0.148	-0.146	-0.136	-0.141	-0.139
35	-0.147	-0.143	-0.141	-0.143	-0.144	-0.134	-0.112	-0.11	-0.122	-0.129	-0.133	-0.13	-0.133	-0.141	-0.141	-0.137	-0.141	-0.139	-0.126
36	0.604	0.613	0.613	0.612	0.615	0.612	0.608	0.604	0.599	0.603	0.606	0.597	0.594	0.595	0.597	0.6	0.599	0.597	0.595
37	0.201	0.196	0.19	0.188	0.185	0.178	0.176	0.177	0.174	0.169	0.165	0.158	0.159	0.16	0.159	0.16	0.154	0.151	0.144
38	-0.641	-0.62	-0.611	-0.592	-0.567	-0.558	-0.548	-0.531	-0.518	-0.497	-0.477	-0.456	-0.45	-0.438	-0.419	-0.407	-0.396	-0.405	-0.407
39	-0.024	-0.001	-0.031	-0.025	-0.001	0.005	-0.009	-0.031	-0.013	0.009	0.004	0.004	0.004	0.007	0.003	-0.004	-0.013	-0.021	-0.011
40	-0.061	-0.059	-0.05	-0.043	-0.052	-0.052	-0.051	-0.054	-0.046	-0.041	-0.046	-0.055	-0.051	-0.046	-0.051	-0.051	-0.051	-0.05	-0.044
41	-0.255	-0.255	-0.256	-0.252	-0.251	-0.252	-0.25	-0.246	-0.246	-0.249	-0.247	-0.243	-0.244	-0.246	-0.247	-0.253	-0.256	-0.252	-0.248
42	-0.234	-0.234	-0.232	-0.231	-0.228	-0.228	-0.228	-0.226	-0.222	-0.22	-0.216	-0.215	-0.212	-0.207	-0.203	-0.202	-0.203	-0.202	-0.205
43	0.1	0.098	0.098	0.098	0.098	0.098	0.092	0.096	0.1	0.099	0.097	0.096	0.096	0.099	0.1	0.094	0.095	0.097	0.097
44	1.203	1.205	1.205	1.206	1.208	1.204	1.2	1.198	1.197	1.2	1.198	1.198	1.202	1.202	1.207	1.208	1.204	1.196	1.196
45	0.066	0.122	0.175	0.211	0.237	0.275	0.316	0.357	0.383	0.403	0.426	0.444	0.458	0.486	0.522	0.547	0.569	0.581	0.589
46	0.467	0.467	0.471	0.482	0.491	0.495	0.495	0.491	0.495	0.483	0.472	0.485	0.498	0.509	0.48	0.523	0.555	0.523	0.51
47	0.697	0.711	0.717	0.72	0.729	0.729	0.718	0.736	0.753	0.718	0.689	0.694	0.69	0.689	0.698	0.705	0.716	0.722	0.72
48	-3.418	-3.416	-3.41	-3.425	-3.433	-3.424	-3.421	-3.423	-3.419	-3.418	-3.419	-3.419	-3.421	-3.42	-3.421	-3.426	-3.447	-3.449	-3.437
49	-0.069	-0.079	-0.072	-0.077	-0.074	-0.066	-0.07	-0.073	-0.071	-0.07	-0.078	-0.081	-0.076	-0.067	-0.067	-0.073	-0.076	-0.075	-0.069
50	0.242	0.249	0.257	0.262	0.263	0.262	0.258	0.257	0.256	0.254	0.254	0.248	0.245	0.242	0.234	0.235	0.251	0.262	0.265
51	-0.512	-0.512	-0.513	-0.508	-0.509	-0.513	-0.512	-0.513	-0.51	-0.509	-0.514	-0.515	-0.514	-0.517	-0.519	-0.518	-0.513	-0.51	-0.514
52	0.2	0.198	0.189	0.188	0.19	0.19	0.195	0.195	0.198	0.198	0.2	0.201	0.195	0.194	0.197	0.197	0.198	0.199	0.198
53	-0.402	-0.415	-0.418	-0.418	-0.412	-0.412	-0.414	-0.414	-0.422	-0.424	-0.422	-0.421	-0.422	-0.423	-0.424	-0.427	-0.426	-0.427	-0.424
54	-0.629	-0.626	-0.622	-0.621	-0.621	-0.626	-0.628	-0.621	-0.618	-0.618	-0.619	-0.622	-0.63	-0.632	-0.628	-0.628	-0.629	-0.63	-0.633
55	-0.287	-0.29	-0.276	-0.271	-0.276	-0.275	-0.285	-0.291	-0.288	-0.287	-0.277	-0.276	-0.284	-0.29	-0.281	-0.27	-0.281	-0.28	-0.269
56	-0.519	-0.523	-0.522	-0.518	-0.519	-0.519	-0.518	-0.519	-0.52	-0.519	-0.517	-0.517	-0.514	-0.515	-0.521	-0.518	-0.52	-0.522	-0.517
57	-0.101	-0.095	-0.086	-0.084	-0.084	-0.091	-0.09	-0.089	-0.093	-0.091	-0.092	-0.095	-0.094	-0.087	-0.076	-0.08	-0.08	-0.083	-0.084
58	-0.168	-0.174	-0.173	-0.175	-0.179	-0.182	-0.191	-0.198	-0.205	-0.223	-0.244	-0.265	-0.28	-0.298	-0.316	-0.33	-0.354	-0.369	-0.379
59	-0.153	-0.205	-0.225	-0.233	-0.259	-0.275	-0.273	-0.281	-0.28	-0.274	-0.277	-0.273	-0.268	-0.27	-0.266	-0.26	-0.257	-0.259	-0.266
60	-1.079	-1.07	-1.065	-1.071	-1.074	-1.069	-1.067	-1.067	-1.071	-1.073	-1.077	-1.087	-1.084	-1.079	-1.077	-1.085	-1.097	-1.107	-1.116
61	-0.679	-0.681	-0.673	-0.672	-0.679	-0.682	-0.683	-0.682	-0.688	-0.688	-0.679	-0.676	-0.682	-0.685	-0.681	-0.688	-0.696	-0.696	-0.698
62	0.606	0.604	0.603	0.602	0.605	0.609	0.61	0.61	0.609	0.606	0.606	0.608	0.605	0.608	0.609	0.607	0.609	0.608	0.611
63	-0.603	-0.606	-0.606	-0.6	-0.587	-0.578	-0.572	-0.56	-0.552	-0.543	-0.531	-0.519	-0.513	-0.505	-0.5	-0.496	-0.493	-0.492	-0.491
64	-1.056	-1.059	-1.058	-1.06	-1.062	-1.063	-1.064	-1.062	-1.062	-1.065	-1.063	-1.056	-1.056	-1.056	-1.054	-1.055	-1.051	-1.05	-1.054
65	-0.297	-0.291	-0.294	-0.293	-0.292	-0.29	-0.284	-0.284	-0.287	-0.288	-0.291	-0.291	-0.288	-0.284	-0.288	-0.292	-0.291	-0.293	-0.293
66	-0.011	-0.011	-0.006	-0.005	-0.009	-0.006	-0.003	-0.001	0.006	0.01	0.006	0.005	0.007	0.009	0.013	0.015	0.017	0.018	0.022
67	-0.043	-0.042	-0.04	-0.042	-0.038	-0.038	-0.046	-0.052	-0.053	-0.05	-0.043	-0.043	-0.052	-0.054	-0.059	-0.069	-0.073	-0.08	-0.084
68	0.002	-0.008	-0.003	-0.005	0.003	0.011	0.011	0.015	0.018	0.019	0.018	0.021	0.023	0.029	0.032	0.028	0.029	0.021	0.016
69	0.508	0.505	0.506	0.506	0.507	0.509	0.512	0.512	0.507	0.51	0.514	0.513	0.512	0.511	0.515	0.515	0.512	0.512	0.514
70	-1.106	-1.108	-1.103	-1.102	-1.102	-1.103	-1.102	-1.099	-1.095	-1.098	-1.103	-1.103	-1.103	-1.101	-1.098	-1.097	-1.103	-1.105	-1.103
71	0.104	0.1	0.098	0.093	0.088	0.088	0.086	0.088	0.088	0.087	0.089	0.092	0.097	0.096	0.093	0.089	0.088	0.089	0.091
72	-0.012	-0.008	-0.007	-0.007	-0.011	-0.013	-0.021	-0.024	-0.021	-0.026	-0.022	-0.012	-0.011	-0.007	-0.009	-0.013	-0.011	-0.011	-0.013
73	1.417	1.416	1.418	1.417	1.418	1.417	1.417	1.416	1.415	1.417	1.417	1.415	1.413	1.416	1.417	1.417	1.418	1.417	1.415
74	0.211	0.211	0.208	0.212	0.212	0.21	0.209	0.207	0.213	0.214	0.216	0.22	0.218	0.219	0.22	0.223	0.225	0.221	0.221
75	-1.34	-1.33	-1.329	-1.321	-1.316	-1.316	-1.31	-1.315	-1.311	-1.301	-1.3	-1.301	-1.306	-1.302	-1.291	-1.282	-1.282	-1.274	-1.269
76	0.048	0.045	0.046	0.046	0.045	0.045	0.046	0.044	0.045	0.043	0.046	0.051	0.049	0.045	0.047	0.046	0.043	0.045	0.045
77	-0.333	-0.333	-0.338	-0.338	-0.331	-0.328	-0.327	-0.327	-0.326	-0.323	-0.323	-0.322	-0.325	-0.33	-0.331	-0.334	-0.335	-0.332	-0.332
78	0.195	0.202	0.204	0.206	0.21	0.204	0.202	0.208	0.215	0.219	0.221	0.215	0.209	0.201	0.19	0.196	0.203	0.198	0.192
79	1.806	1.808	1.808	1.806	1.799	1.796	1.805	1.796	1.788	1.796	1.8	1.802	1.803	1.804	1.806	1.807	1.811	1.819	1.812
80	0.417	0.415	0.416	0.416	0.412	0.412	0.413	0.411	0.41	0.413	0.412	0.412	0.416	0.417	0.421	0.422	0.421	0.42	0.419
81	0.081	0.078	0.076	0.079	0.074	0.076	0.081	0.081	0.076	0.074	0.072	0.076	0.08	0.08	0.08	0.079	0.083	0.082	0.082
82	0.169	0.17	0.175	0.172	0.17	0.168	0.17	0.173	0.168	0.165	0.171	0.172	0.167	0.167	0.171	0.17	0.171	0.171	0.174
83	-0.145	-0.145	-0.143	-0.143	-0.143	-0.141	-0.14	-0.139	-0.141	-0.141	-0.14	-0.139	-0.137	-0.139	-0.141	-0.143	-0.142	-0.142	-0.144
84	0.076	0.073	0.072	0.072	0.07	0.076	0.08	0.077	0.079	0.08	0.079	0.076	0.075	0.076	0.078	0.083	0.079	0.077	0.077
85	1.107	1.107	1.105	1.106	1.107	1.107	1.109	1.109	1.107	1.111	1.111	1.105	1.101	1.106	1.109	1.109	1.111	1.113	1.121
86	-0.021	-0.021	-0.022	-0.023	-0.021	-0.021	-0.021	-0.018	-0.017	-0.014	-0.014	-0.013	-0.013	-0.015	-0.015	-0.019	-0.019	-0.018	-0.018
87	0.511	0.507	0.508	0.512	0.516	0.513	0.506	0.514	0.516	0.51	0.513	0.514	0.516	0.512	0.51	0.512	0.519	0.522	0.518
88	-0.26	-0.268	-0.28	-0.287	-0.294	-0.293	-0.278	-0.26	-0.242	-0.235	-0.226	-0.212	-0.195	-0.175	-0.154	-0.138	-0.134	-0.131	-0.135
89	0.02	0.015	0.013	0.017	0.017	0.017	0.018	0.015	0.015	0.017	0.019	0.019	0.024	0.028	0.028	0.026	0.023	0.021	0.019
90	0.177	0.176	0.179	0.182	0.182	0.181	0.181	0.186	0.186	0.184	0.189	0.19	0.186	0.186	0.185	0.189	0.193	0.194	0.194
91	-0.299	-0.3	-0.301	-0.298	-0.3	-0.301	-0.299	-0.297	-0.294	-0.296	-0.298	-0.297	-0.294	-0.293	-0.291	-0.292	-0.295	-0.295	-0.293

Appendix E

ORIGINALITY REPORT

6%	4%	5%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ecorobotics.com.na Internet Source	<1%
2	www.rapidonline.com Internet Source	<1%
3	www.elprocus.com Internet Source	<1%
4	Wenhan Liu, Mengxin Zhang, Yidan Zhang, Yuan Liao, Qijun Huang, Sheng Chang, Hao Wang, Jin He. "Real-time Multilead Convolutional Neural Network for Myocardial Infarction Detection", IEEE Journal of Biomedical and Health Informatics, 2017 Publication	<1%
5	"Proceedings of ICETIT 2019", Springer Science and Business Media LLC, 2020 Publication	<1%
6	Submitted to Southern University And A & M College Student Paper	<1%

7	Deboleena Sadhukhan, Saurabh Pal, Madhuchhanda Mitra. "Automated Identification of Myocardial Infarction Using Harmonic Phase Distribution Pattern of ECG Data", IEEE Transactions on Instrumentation and Measurement, 2018 Publication	<1 %
8	btep.net Internet Source	<1 %
9	Submitted to KDU College Sdn Bhd Student Paper	<1 %
10	Submitted to University of Hull Student Paper	<1 %
11	www.arrivalelectronics.co.uk Internet Source	<1 %
12	www.journaltocs.ac.uk Internet Source	<1 %
13	hdl.handle.net Internet Source	<1 %
14	Submitted to Maryville University Student Paper	<1 %
15	www.ncbi.nlm.nih.gov Internet Source	<1 %
16	www.slideshare.net Internet Source	<1 %

17	"Sustainable Communication Networks and Application", Springer Science and Business Media LLC, 2020 Publication	<1 %
18	"2020 7th International Conference on Smart Structures and Systems (ICSSS) - Full Conference Proceedings", 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020 Publication	<1 %
19	doctorpenguin.com Internet Source	<1 %
20	U. Rajendra Acharya, Hamido Fujita, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, Muhammad Adam. "Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals", Information Sciences, 2017 Publication	<1 %
21	www.hkshanhai.com Internet Source	<1 %
22	Submitted to International University - VNUHCM Student Paper	<1 %
23	Upendra Kaul, K.K. Sethi, Jamshed Dalal, Keyur Parikh, M.S. Hiremath, Ajit S. Mullasari, Tiny Nair, P.S. Sandhu, Bhavesh Kotak. "A	<1 %

multicentre retrospective study to understand anti-platelet treatment patterns and outcomes of acute coronary syndrome patients in India (TRACE)", Indian Heart Journal, 2014
Publication

24	"Advanced Computing and Intelligent Technologies", Springer Science and Business Media LLC, 2022 Publication	<1 %
25	res.mdpi.com Internet Source	<1 %
26	searchmobilecomputing.techtarget.com Internet Source	<1 %
27	21centurytoxicology.com Internet Source	<1 %
28	Submitted to Trinity Valley Community College Student Paper	<1 %
29	Junming Zhang, Yan Wu, Jing Bai, Fuqiang Chen. "Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers", Transactions of the Institute of Measurement and Control, 2015 Publication	<1 %

Submitted to University of Sheffield

30	Student Paper	<1%
31	www.mdpi.com Internet Source	<1%
32	Mohit Kumar, Ram Pachori, U. Acharya. "Automated Diagnosis of Myocardial Infarction ECG Signals Using Sample Entropy in Flexible Analytic Wavelet Transform Framework", Entropy, 2017 Publication	<1%
33	Muhammad Arshad Awan, Zheng Guangbin, Shin-Dug Kim. "A Dynamic Approach to Recognize Activities in WSN", International Journal of Distributed Sensor Networks, 2017 Publication	<1%
34	Sun, L., Y. Lu, K. Yang, and S. Li. "ECG Analysis Using Multiple Instance Learning for Myocardial Infarction Detection", IEEE Transactions on Biomedical Engineering, 2012. Publication	<1%
35	Submitted to University of Northampton Student Paper	<1%
36	digital.library.unt.edu Internet Source	<1%

37	"Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2", Springer Science and Business Media LLC, 2021 Publication	<1 %
38	ijisrt.com Internet Source	<1 %
39	"Internet of Things and Big Data Technologies for Next Generation Healthcare", Springer Science and Business Media LLC, 2017 Publication	<1 %
40	123dok.com Internet Source	<1 %
41	Rahman Atiqur. "Radio frequency identification based smart parking system using Internet of Things", IAES International Journal of Robotics and Automation (IJRA), 2021 Publication	<1 %
42	ir.lib.uwo.ca Internet Source	<1 %
43	www.ldnews.com Internet Source	<1 %
44	A.K. Jain, S. Prabhakar, Lin Hong. "A multichannel approach to fingerprint classification", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999 Publication	<1 %

45	www.geotab.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85