**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELGAUM – 590014**

****

**A Project Report on**

**A REAL TIME APPLICATION TO IDENTIFY ALCOHOLICS**

**FROM ECG SIGNALS**

**Submitted in partial fulfilment of the requirement for the award of degree of**

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS AND COMMUNICATION**

**By**

**Akarsh N Kolekar [1PI13EC009]**

**Apoorv Vatsal [1PI13EC017]**

**Rakshith Vishwanatha [1PI13EC075]**

**Under the Guidance of:**

**Dr. B. Niranjana Krupa**

**Professor, Dept. of ECE,**

**PES Institute of Technology**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

****

**P.E.S. INSTITUTE OF TECHNOLOGY**

**(An Autonomous Institute under VTU, Belgaum)**

**BENGALURU - 560085**

**DECLARATION**

We hereby declare that the project report entitled “**A REAL TIME APPLICATION TO IDENTIFY ALCOHOLICS FROM ECG SIGNALS”** is the bonafide record of the project carried out at **P.E.S. Institute of Technology** in partial fulfilment of the requirements for the award of degree **Bachelor of Engineering** in **Electronics and Communication Engineering** of **Visvesvaraya Technological University, Belgaum** during the academic year 2017. We further declare that the project report is not submitted to any other universities in fulfilment of the requirements for the award of any degree.

By

**AKARSH N. KOLEKAR (1PI13EC009)**

**APOORV VATSAL (1PI13EC017)**

**RAKSHITH VISHWANATHA (1PI13EC075)**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELGAUM – 590014**

****

PES INSTITUTE OF TECHNOLOGY

(An Autonomous Institute under VTU, Belgaum)

BENGALURU – 560085



**CERTIFICATE**

Certified to the project entitled **A REAL TIME APPLICATION TO IDENTIFY ALCOHOLICS FROM ECG SIGNALS** is a bonafide work carried out by **Akarsh N. Kolekar, Apoorv Vatsal** and **Rakshith Vishwanatha** bearing University Seat Number **1PI13EC009, 1PI13EC017 and 1PI13EC075** respectively in partial fulfilment for the award of **Bachelor of Engineering** in **Electronics and communication** of the **Visvesvaraya Technological University**, Belgaum during the academic year 2017. It is certified that all correction/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements with respect to the project work prescribed for the said degree.

**Head of Department** : **Guide:**

**Dr. Chandar T S Dr. B. Niranjana Krupa**

Dept. of ECE. Dept. of ECE

PES Institute of Technology, PES Institute of Technology,

Bengaluru – 560085 Bengaluru – 560085

**Principal**

Dr. K S Sridhar

PESIT, Bengaluru – 560085

**External Viva:**

**Name of the Examiner Signature with Date**

1.

2.

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete, without the mention of people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

Our most sincere and grateful acknowledgement to the P.E.S Institute of Technology for providing us with the opportunity to pursue our degree and thus helping us in shaping our career.

We would like to sincerely thank our project guide, **Mr**. **Purushotham.U** Associate Professor of the Department of Electronics and Communication Engineering for his continuous valuable guidance, advice and persistent encouragement throughout the project work.

We would like to sincerely thank **Dr.T.S.Chandar .** Head of the Department Electronics and Communication for his encouragement and support throughout the project work.

We are thankful to **Dr.Sridhar K.S**. Principal of PES institute of Technology, Bengaluru for his encouragement and support in our endeavour.

We would like to thank all the teaching and non-teaching staff and management of PES institute of Technology, Bengaluru for their cooperation.

Last but not the least we wish to thank our family and friends for all their love support and encouragement.

**ABSTRACT**

The Quad-copter (UAVs) are used to Surveillance and Border Security System. Whenever natural disasters occur to rescue the people who are in danger and require helps .These vehicles carry essential requirements like food medicine to them through quad copter is controlled and observed by remote location. Quad copters can also assist soldiers for security issues borders such as Terrorist activities etc. these device Monitor challenging task for bad weather conditions and life of soldiers is also risky . Designing an unmanned air vehicle which will monitor the border area, remote location, also commercial movie shooting can be a good option. GPS is used to track the position of intruder or our troops or vehicles. This GPS data will be received by Microcontroller processor and conveyed to observer. The Quad-copter is controlled by observer via the IR remote. Observer will fly the Quad-copter from a distance to area which has to be monitored. The Audio-Visual will be transmitted to PC via Wireless camera mounted on assembly.

**Table of Contents**

**CHAPTER -1 PROJECT INTRODUCTION**

1.1 Introduction ....................................................................................................................10

1.2 Problem statements .........................................................................................................10

1.3 Project objectives ............................................................................................................10

1.4 Project scopes / constrains................................................................................................10

**CHAPTER- 2 LITERATURE REVIEW**

**CHAPTER- 3 DESIGN**

3.1 Computer side Diagram...................................................................................................11

3.2 Quad copter side Diagram...............................................................................................11

3.3. Video Recording and Transmission...............................................................................12

3.4. Hardware Components...................................................................................................12

3.5. Software.........................................................................................................................12

**CHATER-4 Hardware Components........................................................................**

4.1 Frames..............................................................................................................................13

4.2 Motors..............................................................................................................................13

4.3 ESC..................................................................................................................................14

4.4 Propellers..........................................................................................................................14

4.5 Fight Control Board.........................................................................................................15

4.6 Radio Transmitter and Receiver.......................................................................................15

4.7 Battery..............................................................................................................................16

4.7.1 Battery Charger..............................................................................................................17

4.8 Wireless Camera.............................................................................................................17

**CHAPTER-5 FLYING MECHANISM**

5.1 Quad copter Moments......................................................................................................18

5.2 Forward and Backward Motion......................................................................................19

5.3 Left and Right Motion...................................................................................................20

5.4 Static Position ................................................................................................................21

**CHAPTER 6 INITIAL SET-UP OF KK 2.1 BOARD**

6.1 Parameter Settings Of Quad copter ..................................................................................33

6.2 Initial Set-Up.....................................................................................................................34

**CHAPTER -7 IMAGE PROCESSING**

7. Block Diagram of Image Processing..................................................................................42

7.1 Algorithm Block Diagram................................................................................................42

7.2 Mat Lab programs........................................................................................................43-48

7.3 Advantages and Disadvantages....................................................................................49

7..4 Applications.................................................................................................................49

**8. CONCLUSION AND SCOPE FOR FUTURE WORK………………………......50**

**9.Reference.....................................................................................................................**

**List of Figures**

Fig3.1 Computer side diagram.................................................................................................11

Fig3.2 Quad copter side diagram.............................................................................................11

Fig4.1 Frames...........................................................................................................................13

Fig4.2 Motors...........................................................................................................................13

Fig4.3 ESC...............................................................................................................................14

Fig4.4 Propellers......................................................................................................................15

Fig4.5 Fight Control Board.....................................................................................................16

Fig4.6 Radio Transmitter and Receiver...................................................................................17

Fig4.7 Battery...........................................................................................................................18

Fig4.8 Wireless Camera.........................................................................................................19

Fig5.2 Forward and Backward Motion...................................................................................20

Fig5.3 Left and Right Motion................................................................................................21

Fig6.2a Connection to Fight Controlled Board......................................................................34

Fig6.2bThe Completed Receiver and Motor Wiring................................................................35

Fig6.2c Connection of ESC to Quad copter Frames................................................................36

Fig6.2d Motor Connections .....................................................................................................37

Fig6.2e Receiver Connections .................................................................................................38

Fig6.2f Battery Connections....................................................................................................39

Fig6.2g LCD Display Quad copter Conditions......................................................................39

Fig 7 Block Diagram of Image Processing.............................................................................40

Fig 7.1 Algorithm Black Diagram............................................................................................43

CHAPTER – 1

**INTRODUCTION/OVERVIEW**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1. Introduction**

**1.1. Problem Statement**

Technology has advanced in various fields at rapid rates, however in an area that concerns the common well-being of humans, technology has remained dormant. Identifying accurately if a person is intoxicated is utmost important to keep public harm and nuisance at bay. The most common device used, the breathalyser has drawbacks that we aim to rectify. The common drawbacks of a breathalyser are:

i) Contamination of SiO2 sensor requires frequent recalibration and replacement

ii) Breathalysers being a medium for propagation of contagious diseases

iii) Interfering components (like acetone) being higher in the breath of dieters and diabetics make them more prone to being detected falsely

iv) Infrared sensors detect the absorbance of the compound as a function of the wavelength of the beam when the infrared beam is passed through the sample breath chamber. The chamber is prone to environmental pollutants and aerosols leading to errors.

These disadvantages mentioned above are addressed with the use of infallible computers and well trained machine learning algorithms.

Heart rate variability [HRV] obtained from Electrocardiogram [ECG] is a useful biomarker and is used extensively in our paradigm to extract features. The features extracted are then used to train the system to classify patients as chronic alcoholic or otherwise. This may be useful in discriminating individuals based on their habits while preventing other external environmental conditions from altering or corrupts the readings.

**1.2. Objective**

The aim is to develop a prototype to read ECG signals, extract features to perform HRV analysis, and classify the person under test as an alcoholic or otherwise, in real time.

**1.3. Proposed Methodology**

i. The first step is to study the advantages and disadvantages of HRV analysis over other markers for cardiovascular health and the effect of alcohol consumption on ECG signals.

ii. The next step is the familiarisation with the form and structure of the ECG signals, and understanding distortions (artefacts) that can occur in the signal. Two common artefacts seen in ECG signals are the wandering baseline and fuzzy (60 Hz) distortions caused due to probe movement, uneven conductive gel, muscle movement, etc.

iii. A survey of available devices for measuring and recording ECG signals is done. The device made by a previous student connects to a phone via bluetooth and the data is then routed to a server from the phone. This device was found to be noisy and required extra software like Audacity for filtering purposes and hence was not real time. It was proposed to interface the sensor directly with the Raspberry Pi 2 and do all the processing on it, making the overall system more compact and real time.

iv. Parallel to working on the sensor hardware, once the ECG waveforms are understood, work with the training data set is started to decide the feature extraction techniques and classifiers that could be implemented. Survey of tools like KubiosHRV to understand time domain and frequency domain feature extraction for HRV analysis is done. The training data set is obtained from Autonomic Lab, Department of Neurophysiology, NIMHANS and consists of 38 samples of alcoholic and 29 samples of normative subjects.

v. To this data, preprocessing is performed to remove artefacts, and then time domain, frequency domain, and non-linear methods are used for feature extraction from the ECG signals. On extracting the desired features, ELM and SVM are classifiers that are chosen to be trained using the dataset. Kernel functions are also tried in order to check if their usage might improve accuracy.

vi. Accuracy of the classifiers is checked using leave-one-out cross validation and k-fold cross validation, and classifiers providing highest consistent average accuracy in these validation methods is selected.

vii. Based on the algorithms finalised on, the controller board on which the feature extraction and classification algorithms will be implemented are chosen. The target device is the Raspberry Pi 2 currently, as it has a 900 MHz quad-core CPU that is necessary for machine learning applications.

viii. Training of the classifiers to obtain weights is done on MATLAB and the trained parameters are used on the Raspberry Pi 2.

ix. Finally, the sensor is interfaced with the Raspberry Pi 2, which is uploaded with trained parameters, ready to classify real time ECG data.

**1.5. Literature Survey**

[1] In the “A Precise Drunk Driving Detection Using Weighted Kernel Based on Electrocardiogram” paper the authors collected ECGs from 50 volunteers. They applied pre-processing for noise suppression and signal segregation into a number of samples, where each sample represented the heart activity for one heartbeat. To identify characteristics of an alcoholic, features such as Pmax, Pd, means and variances of P, R, S waves and R-R intervals are extracted. The paper used SVM as a classifier and 10-fold cross validation.

[2] The paper “Effects of Alcohol on the Electrocardiogram” took electrocardiograms of 1,000 chronic alcoholic patients and analyzed them to find evidence that excessive consumption of alcohol may produce changes in the electrocardiogram. The predominant abnormalities that were observed by the authors were sinus tachycardia and nonspecific T-wave changes.

[3] The paper by “The Electrocardiogram of alcoholic cardiomyopathy” Evans, William is one of the first few studies that emphasized the need to study the relation between not the just alcohol consumption and the liver but also consider the effect of alcohol consumption on the heart. Various changes were seen in the ECG of alcoholic patients like the dimple T wave, spinous T wave, cloven T waves, etc. This study clearly establishes the fact that alcohol consumption leads to variation in heart beat.

[4] The "Supervised machine learning: A review of classification techniques." gives a comprehensive view of various supervised machine learning classification techniques and provides interesting domains where machine learning can be applied.

[5] In “What are Extreme Learning Machines? Filling the Gap Between

Frank Rosenblatt’s Dream and John von Neumann’s Puzzle”, the author describes the architecture of ELM, how it is different from other neural networks which use back propagation to iteratively tune parameters, and compare it with other classifiers like SVM and Least Squares SVM (LS-SVM).

[6] In “A Meta-Cognitive Learning Algorithm for an Extreme Learning Machine Classifier”, the authors describe an algorithm to build the structure of the ELM neural network during the training phase, instead of fixing the architecture a-priori. This would optimise the number of hidden layer neurons in an intelligent way, instead of using brute force to find the number of hidden layer neurons that gives best accuracy without over fitting.

[7] The “Circuit Design for Front-End Electrocardiograph”, “Techniques for accurate ECG signal processing”, “Electrocardiogram (ECG) circuit for use with oscilloscopes” and “ECG Circuits, Signal Sampling and Digitalization” detailed the circuit elements that would be required in order to amplify the differential signal, filter out the different kinds of noise present in it, like power line interference, low frequency interference from ECG signal baseline drift, et cetera, and described the set up that would be required to test the circuit using oscilloscopes.

[8] “Support Vector Machines” and “The Simplified SMO Algorithm” provide us with an understanding of the Support Vector Machine learning algorithm, and the also talks about Kernels, which allow us to vary the dimensionality of the feature dataset, and the SMO algorithm, which gives us an efficient implementation of SVMs.

CHAPTER – 3

**METHODOLOGY**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

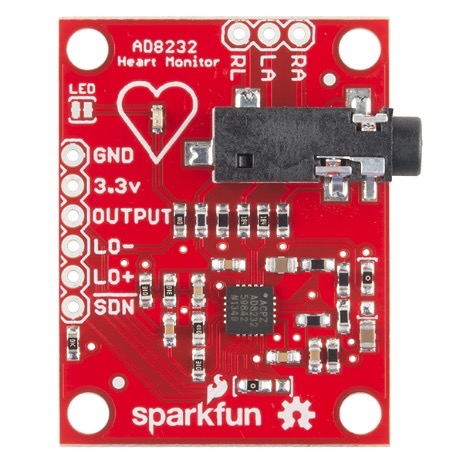
**3.1 INTRODUCTION**

This section describes the overall working of the ECG classification project. The chapter is divided into two sections (Section 3.2 and Section 3.3), and gives details about the hardware and software portions.

**3.2 HARDWARE**

**INTRODUCTION**

**3.1.1 Heart Rate Monitor**  
The Heart Rate Monitor is used to measure the electrical activity of the heart. It uses AD8232 as its core chip which is one of the popular integrated signal conditioning block for ECG and other bio-potential measurement applications.

The AD8232 Heart Rate Monitor breaks out six essential connections from the IC SDN, LO+, LO-, OUTPUT, 3.3V, GND for operation. The Board also has RA(Right Arm),LA(Left Arm) and RL(Right Leg) pins for custom application. Additional features are listed below:

1. Analog Output
2. Leads-Off Detection
3. Shutdown pin
4. LED indicator
5. 3.5mm Jack for Biomedical Pad connection

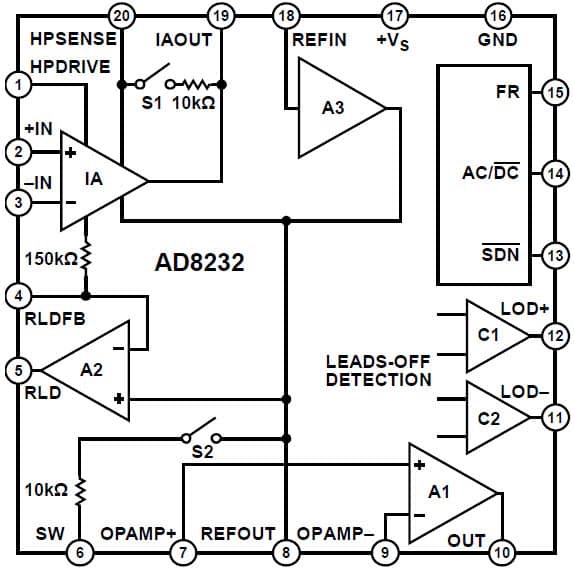
**AD8232 :**  
One of the popular and widely used IC for extraction and filtering of biopotential signals in various bio-medical applications is the AD8232. The features that make AD8232 appropriate for these applications are:   
1. The AD8232 implements a two-high pass filter for eliminating motion artifacts which often affect the small bio potential signals.

2. Three pole low pass filter: The implementation of this filter allows the IC to remove additional noise.

3. Fast restore function: In many applications due to the low cut-off frequency used in the high pass filters, signals may require longer settling time. This long settling time could cause an unwanted delay in obtaining the signal. Hence this function reduces the duration of long settling tails of the high pass filters.

4. Specialized instrumentation Amplifier: The IC has contains a multi-purpose Instrumentation Amplifier which amplifies the ECG signal while rejecting the electrode half-cell potential on the same stage.

5. Right leg drive amplifier: The common mode output at the instrumentation amplifier is inverted by the right leg drive (RLD) amplifier. When the right leg drive output current is injected into the subject, it counteracts common-mode voltage variations, thereby improving the common-mode rejection of the system.



Caption

All the above feature allows AD8232 to extract, amplify, and filter small bio-potential signals in the presence of noisy conditions.

**3.1.2 Rapsberry Pi**  
The Raspberry Pi is a series of small single board computers developed by the Raspberry Pi Foundation. These boards are approximately credit-card sized and have a standard mainline form-factor.

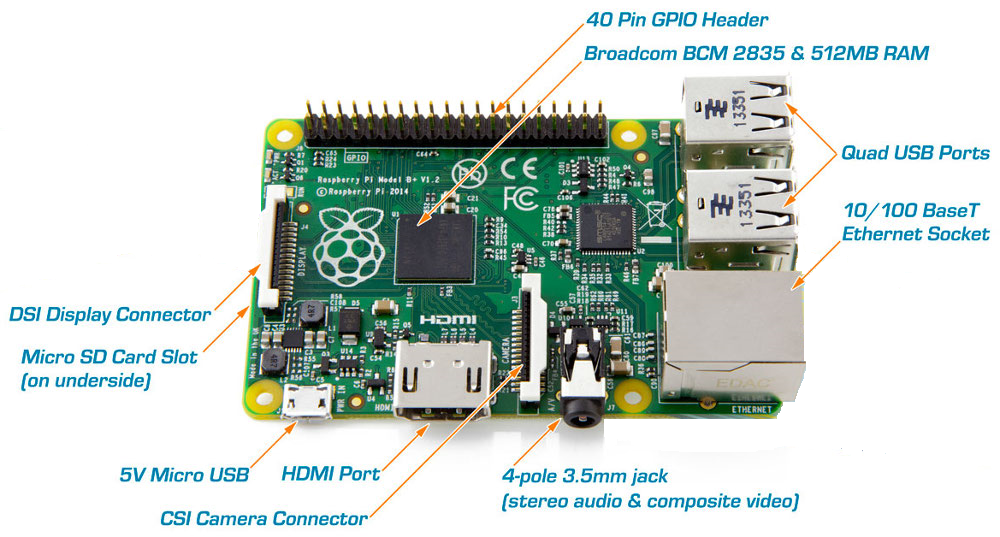


CAPTION

Several generations of Raspberry Pi have been released. The version used in this project is Raspberry Pi 2 Model B.  
All models of Raspberry Pi have a Broadcom system on chip (SoC) which includes an ARM compatible central processing unit (CPU) and on chip graphic processing unit (GPU).  
The specifications (specs) of Raspberry Pi 2 are:

1. Processor: Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor with 256 KB shared L2 cache.
2. Random Access Memory(RAM): 1 GB of RAM
3. Peripherals: 17 General purpose Input Output(GPIO) plus specific function pins have been provided
4. Network: There is separate Ethernet port provided to support 10/100Mbps
5. There are 4 additional Universal Serial Bus(USB) provided for connecting peripherals like keyboard, mouse
6. Storage: There is a separate Secure Digital(SD) card slot provided
7. Power source : 5V via Micro USB or GPIO header
8. Audio output: A 3.5mm audio Jack is provided which can support analog output.
9. Video input: 15-pin MIPI camera Interface(CSI) connector

10) Video Output: High-Definition Multimedia Interface (HDMI) provided for video output



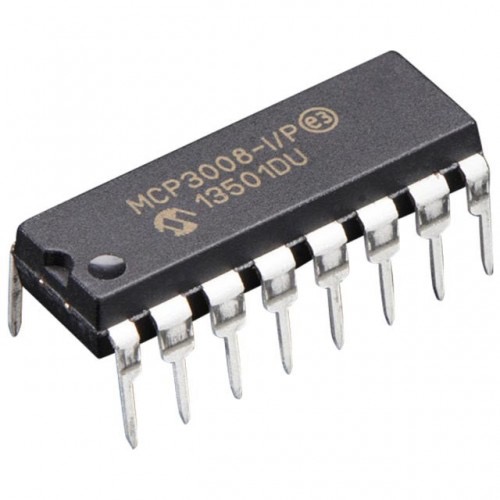
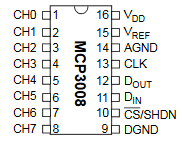
CAPTION

**3.1.3 Analog to Digital Converter (ADC)**

Analog to Digital convertor is a system that converts an analog signal into a digital signal. There are various ADC available in the market of which MCP3008 is used in this project. It is a 10-bit ADC with on board sample and hold circuitry.

Some of the features of MCP3008 are:

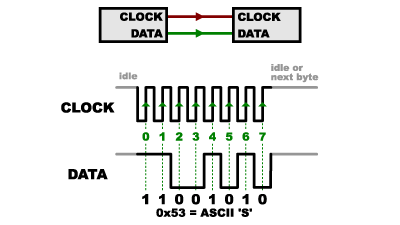
1. 10 bit resolution
2. 8 input channel
3. Serial Peripheral Interface (SPI) serial interface
4. High sampling rate   
   200 kilo samples per second (ksps) at a supply voltage of 5V and 75 kilo samples per second (ksps) at a supply voltage of 3.3V

**3.1.4 Communication protocols**

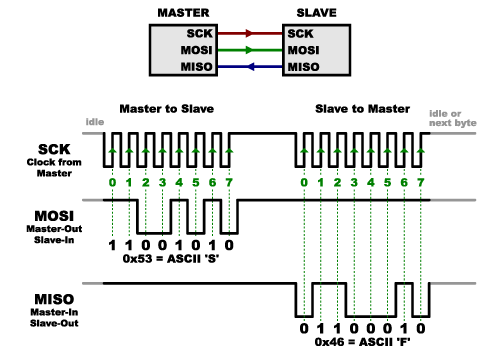
Following are the two communication protocols used in this project:

1. Serial-Peripheral Interface (SPI) protocol:   
   SPI is a synchronous data bus used to send data between microcontrollers and small peripherals such as shift registers and sensors. Here synchronous data bus means that it uses separate clock and data lines.   
   The clock is an oscillating signal that tells the receiver exactly when to sample the bits on the data line. This can be the rising or falling edge of the clock signal.When the receiver detects that edge, it will immediately look at the data line to read the next bit.



CAPTION

In SPI, only one side generates the clock signal.The side that generates the clock is called the “master”, and the other side is called the “slave”. There is always only one master which is the microcontroller,but there can be multiple slaves.



CAPTION

When data is sent from the master to a slave, it is sent on a data line “Master Out / Slave In”(MOSI). If the slave needs to send a response back to the master, the master will continue to generate a prearranged number of clock cycles, and the slave will then put the data onto a third data line called “Master In / Slave Out”(MISO).   
Advantages of SPI:

1. It is faster than asynchronous serial
2. The receive hardware can be a simple shift register
3. It supports multiple slaves

Disadvantages of SPI:

1. It requires more signal lines (wires) than other communications methods
2. The communications must be well-defined in advance
3. The master is in control of all communications
4. Secure Shell:   
   Secure Shell (SSH) is a widely used cryptographic network protocol for operating network services securely over an unsecured network. The best application of it is for remote login to computer systems by users.  
   SSH provides a secure channel over an unsecured network in a client-server architecture, connecting an SSH client application with an SSH server.

Some popular uses of SSH are:  
(a) For login to a shell on a remote host  
(b) For executing a single command on a remote host  
(c) For setting up automatic login to a remote server  
(d) Secure file transfer

**3.2 INTEGRATION:**

**3.2.1 Setting up Raspberry Pi**  
***##ALL THE WORK ON RPI is done on WINDOWS(Have to mention this in some place)***

**(a) SD card setup**

Raspberry Pi 2.0 does not come with a pre-built operating system. Instead the operating system has to be flashed on the SD card which is then inserted into the Raspberry Pi. The operating system on SD card is installed by using the following steps:

1. SD card is inserted into the SD card reader.
2. Win32DiskImager utility is downloaded from the Sourceforge Project page as an installer file, and the software is installed by running it.
3. The Operating System (OS) of choice is downloaded from official raspberry site. For this project Raspbian Wheezy was used.
4. The image of the OS is then extracted from the downloaded file.
5. The Win32DiskImager utility software installed is then opened.
6. The extracted image is then selected and ‘Write’ button is pressed to write the OS to the SD card.
7. On successful completion of the process the SD card is ejected and inserted into the Raspberry Pi.
   1. The Raspberry Pi is ready for use with Raspbian Wheezy installed.
   2. ##The Raspberry Pi is then connected to the Internet using Ethernet cable.

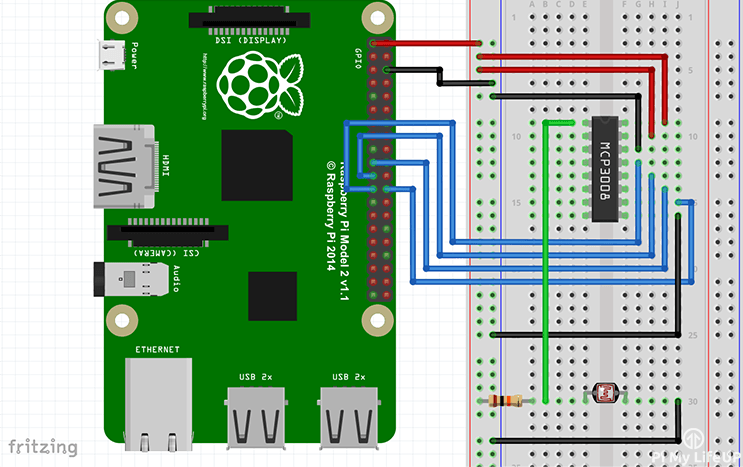
**(b) Internet Protocol (IP) address of Raspberry Pi (Rpi)**

In order to work with the Raspberry Pi the IP address of Pi is needed. There are two ways to obtain this:

1. Working with Rpi using Television (TV) mode:  
   In this method the Rpi is connected to a TV screen using HDMI cable. The Rpi is connected to a router using Ethernet connection for Internet connection.
2. The IP address is then obtained by typing in “ipconfig” in the terminal window of the Rpi.
3. Working with Rpi in “headless” mode :  
   In this case a static IP address is assigned by the user. This is the approach used in this project as it is not always convenient to have a TV screen. The steps to set the static IP address is as follows:  
   (a) SD card is removed from Rpi and then inserted into the SD card reader.  
   (b) The SD card contents are then opened.  
   (c) A file named “cmdline.txt” is opened using Notepad++  
   (d) At the end of the file the following line is appended:  
   Syntax: ip=<client-ip>:<server-ip>:<gw-ip>:<netmask>:<hostname>:<device>:<autoconf>  
   Example:  
   ip=169.254.3.14::169.254.56.85:255.255.0.0:rpi:eth0:off  
   (e) The file is then saved and SD card is safely ejected from the system and inserted back into the Raspberry Pi.  
   The Rpi is then connected to a laptop using Ethernet

**(c) Communicating with the Rpi**  
SSH protocol is used for establishing the communication between the Rpi and the laptop. For this a free and open-source terminal emulator called PuTTY is used. PuTTY supports many variations on the secure remote terminal, and provides user control over the SSH encryption key and protocol version.

**3.2.2 Connecting Heart Rate Monitor to RPi using ADC MCP3008**  
The following is the circuit diagram required for the setup:

****  
  
First a 3v3 pin is connected to the positive rail on the breadboard and a ground pin to the ground rail on the breadboard. The following connections are made:

|  |  |
| --- | --- |
| **MCP3008** | **Rpi** |
| VDD (Pin 16) | 3.3V |
| VREF (Pin 15) | 3.3V |
| AGND (Pin 14) | GROUND |
| CLK (Pin 13) | GPIO11 (Pin 23/SCLK) |
| DOUT (Pin 12) | GPIO9 (Pin 21/MISO) |
| DIN (Pin 11) | GPIO10 (Pin 19/MOSI) |
| CS (Pin 10) | GPIO8 (Pin 24/CE0) |
| DGND (Pin 9) | GROUND |

**3.3 SOFTWARE**

In this section details about the dataset used, pre-processing performed on the dataset, features extracted from the dataset, feature reduction performed and classification algorithms trained has been provided. While pre-processing and feature reduction were important steps in obtaining a well trained classifier, the bulk of the software end of the project focusses on explaining feature extraction and classification algorithms. Details about the various types of features extracted has been explained in section 3.3.3, and some background information about the two classifiers used on the dataset has been provided in sections 3.3.5 and 3.3.6.

**3.3.1 Dataset Description**

In order to classify test subjects as alcoholics or normative, with a reasonable accuracy, the classifier has to be trained with several samples of the test subjects, and subsequently tested with more samples. These samples along with the appropriate labelling constitute the dataset.

**3.2.1.1.** The data in the dataset is an array of values which represent the ECG signal of the test subject.

**3.2.1.2.** The ECG data used has been recorded at the Autonomic Lab, Department of Neurophysiology, NIMHANS, Bengaluru. The data was recorded after taking informed consent adhering to Helsinki’s declaration.

**3.2.1.3.** At the Autonomic Lab , HRV is done using proprietary hardware and software setup by AD instruments, Australia. The product used in the lab premises for observing and recording HRV is PowerLab.

**3.2.1.4.** This device uses high sample frequency (of the order of 1kHz) to record electronic activity of the heart and other signals pertaining to other functions such as respiratory functions et cetera. The raw ECG data was extracted as five minute samples in European Data Format [EDF].

**3.2.1.3.** The dataset comprises of 67 samples, out of which 38 are ECG recordings of alcoholic test subjects and 29 are of normative test subjects.

**3.3.2 Pre-processing**

**ADD clear subheadings to distinguish different techniques?**

ECG data acquired from a patient contains various types of disturbances and noise, which makes it difficult to extract features from the signal. The most common types of disturbances are [ADD REF *ecg\_noise\_sources\_removal\_IJAIEM*]:

1. Power-line Interference: ECG signals are measured as the differential voltage that exists between two points on the body and turn out to have very small voltage amplitudes (in the mV range). The small voltage amplitude of ECG signals makes it susceptible to interference from AC signals present at any point in the circuit. The primary source of power-line interference is the AC power-line used as the power source for the ECG recording devices and display monitors or CROs. The frequency of the signal in the power-line in India is 50Hz, leading disturbances on the ECG signals also with the same frequency.
2. Baseline Wandering: Gradual changes in the skin impedance of the patient and the patient’s breathing lead to low frequency baseline wandering. Baseline wandering is seen as an overall rise and fall of the PQRST complexes of the ECG signal.
3. Motion Artefacts: Movements by the patient or the electrode cause mild to severe disturbances in the baseline of the ECG signal. While mild movements like breathing or slight movement of limbs not connected with the probes do not completely ‘submerge’ the ECG signal, abrupt movements by the patient can completely mask the QRS complex with meaningless noise.

In the dataset used, all three forms of artefacts are observable and have been dealt with to obtain cleaner waveforms for feature extraction.

PHOTO of unfiltered noise

ADD post filtered noise photos later in corsp sections

Different filtering techniques have been used to obtain suitable signals for time domain, frequency domain and non-linear frequency extractions.Time domain and non-linear feature extraction rely primarily on the RR intervals of the ECG signal. Thus, we focus on filtering out artefacts and other sections of the ECG sequence while maintaining a high amplitude for the QRS complex. Such results have been obtained using a first order, low pass butterworth filter, having a stop band attenuation of 0.2 and cutoff frequencies of 5Hz and 7Hz. The filter was designed for values mentioned previously since, [ADD REF kubios\_hrv\_ref1] indicates motion artefacts to be present in the 0Hz to 5Hz range. The same filter also helped rid the signal of baseline wandering. Removing the motion artefacts and the baseline wandering essentially ensured that the required R peaks remained the prominent portion of the signal. Finally, the RR intervals were obtained by setting a threshold and measuring the time difference between the occurrences of maximas in those sections of the signal that crossed the threshold. A simple first order IIR filter sufficiently filtered the signal to make the R peaks prominent and allow the calculation of RR intervals.

The same IIR filtered signal was not sufficient clean to accurately extract frequency domain features, as it still contained some low amplitude noise at its baseline. Frequency domain features are obtained primarily by taking the power spectral density (PSD) of the signal over different frequency ranges. To obtain good PSD values and accurate frequency domain features, a clean signal is required. Wavelet Transform has been used to achieve high precision filtering.

**ADD SHORTER VERSION WAVELET TRANSFORMS for preproc for Freq Dom FE**

**Add photos of all the sequences**

**Add photo of final sequence**

ALSO, Brief Description:

It also gave RR intervals which matched IIR filter and threshold method, 8 sequence wavelet transform,

Which sequence was used

Photo of original signal and the final filtred signla

* + 1. **Feature Extraction**

**ADD clear subheadings to distinguish different techniques? … IIR filter, Wavelet, ARX**

Feature extraction is a method of extracting useful information from an otherwise meaningless ECG signal dataset. These features (ie. ‘useful information’) that are extracted from data samples are used by the classifer algorithms to learn how to categorize and classify samples belonging to different classes. Four types of features have been extracted from each of the data samples, and they are:

1. Time Domain features: This kind of feature extraction works on calculations made on the RR interval sequence. Seven time domain features have been extracted.
2. Non-Linear features: This kind of feature extraction also works on calculations made on the RR interval sequence. Here however, the mathematical equations that are used are obtained via graphical analysis. Three time domain features have been extracted.
3. Frequency Domain features: This kind of feature extraction applies Power Spectral Density (PSD) on different sections of the signal, to come up with features. Thirteen frequency domain features have been extracted.

1. Exogenous Input Auto Regressive (ARX) Model Coefficients: Variable number of features based on order taken. Order chosen according to minimal misfit

Time domain and non-linear features utilize Heart Rate Variability (HRV) of a person ... which is … and is a legit way to differeciate ALC and Non ALC as per references [][][][][] …

**All features extracted for are from the filtered or preprocessed form of the original dataset’s signals. Fro t and non lin, and wavelet for frq dom and orig data for arx**

Time domain feature extraction is the simplest of the four types of features that have been extracted from the ECG signals. All calculations for time domain feature extraction have been performed on the IIR filtered signal, which was described in section 3.3.2. In this type of feature extraction, the time instants of occurrences of R peaks is noted and the RR intervals (the time passed between two R peaks) is calculated. The series of RR intervals if plotted against time gives a visual representation of HRV of the ECG signal, and is called a tachogram. After obtaining the RR intervals, values as given below have been calculated and used as features:

1. Mean of RR intervals (RR\_mean)

This feature is the arithmetic mean calculated on the RR interval series and was calculated as follows

EQN

1. Standard Deviation of RR intervals (RR\_std)

This feature is the standard deviation of the RR interval series and was calculated using

EQN

1. Mean Heart Rate (HR\_mean)

From the RR interval series, the average frequency of occurrence of the RR intervals per minute was calculated using [ADD EQN NO]. The inverse was then taken which resulted in the mean heart rate.

EQNS

1. Standard Deviation of Heart Rate (HR\_std)

The standard deviation of the heart rate was calculated in a similar manner as the mean heart rate. First the standard deviation of the of the frequency of occurrence of the RR intervals per minute was calculated. Then, the inverse was taken to obtain the standard deviation of the heart rate.

EQNS

1. Root Mean Square of RR intervals (RR\_rms)

The square root of the mean of the sum of the squares of all the entries in the RR interval series results in the RMS of the RR interval series. The formula for the same is as follows:

EQN

1. Number of Intervals Varying by Larger than a Threshold (RR\_50)

This feature is slightly different from the rest, in the sense that it involves an additional step of taking differences. While obtaining the RR interval series required taking successive differences of the time instants at which R peaks occurred, here, successive differences are taken for the values in the RR interval series itself. On this new series of difference, the number of time differences that are larger than 50ms are counted to yield the RR\_50 feature.

1. Relative Number of Intervals Varying Larger than a Threshold (RR\_r50)

The previous feature obtained divided by the total length of the the RR interval series gives rise to the final time domain feature. This can be represented by the following equation:

EQN

Three non-linear features have been extracted for each of the samples in the dataset. There were two types of analysis done, one of which yielded two features while the other gave rise to a single feature. These two methods are given below:

1. Poincare Plot

This feature is extracted ‘graphically’. A graph of Poincare points is plotted, where the horizontal axis value of all the points is some i-th value in the RR interval series, while the vertical axis value of the point is the (i+1)-th value of the RR interval series. Then, the standard deviations of the points along two different axes are calculated to yield two of three non-linear features denoted by SD1 and SD2 respectively. The axes along which the standard deviations are calculated are the x = y line and the x = -y line. A sample of the Poincare plot obtained for the first alcoholic sample in the dataset has been shown below.

POINCARE PLOT and axes

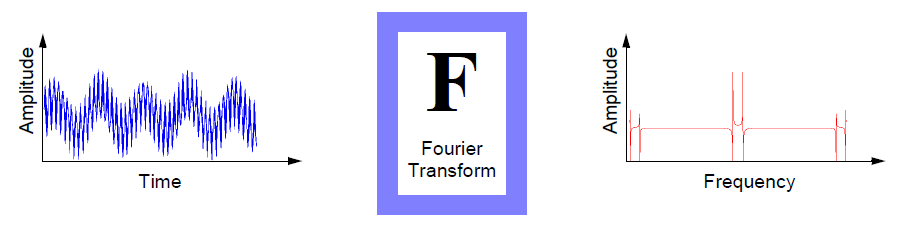
1. Approximate Entropy

This feature is essentially a measure of how much irregularity exists within the RR interval series.

Explain the rest with EQNS

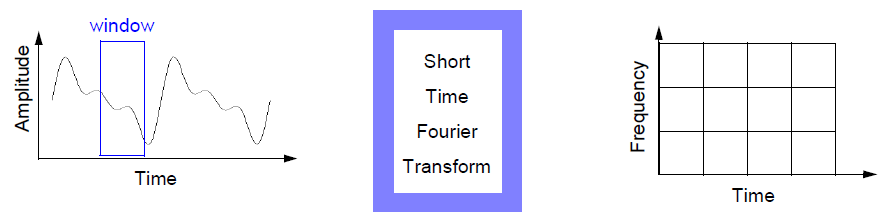
**ADD all wavelet brief intro again, theory and equations**

**Signal Analysis**  
One of the most well know signal analysis tools used by analysts all around the world is the Fourier Analysis. Fourier Analysis breaks the signal down into sinusoids of different frequencies.



However Fourier Analysis has a serious drawback that when a signal is transformed from time domain to frequency domain, the time information is lost. This drawback is not important for stationary signals. But most real world signals contain numerous non-stationary or transitory characteristics: drift, trends, abrupt changes, and beginnings and ends of events. These characteristics are often the most important part of the signal, and Fourier analysis is not suited to detecting them.

Another analysis method that overcomes the drawback of Fourier Transform is the Short Fourier Transform (STFT). The STFT is a sort of compromise between the time and frequency-based views of a signal. A signal is mapped into two dimensional function of time and frequency.



The drawbacks of STFT are:

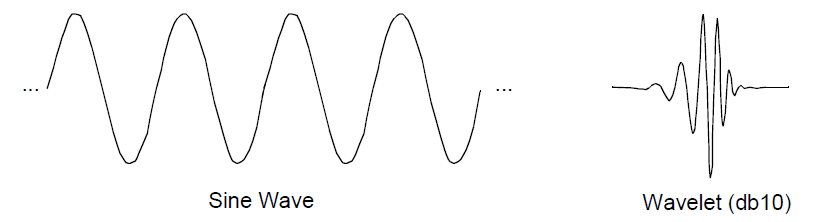
(a) Limited precision depending on the size of window

(b) Once a particular size for the time window is selected, that window is the same for all frequencies

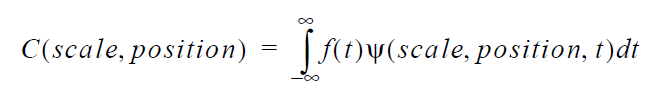
Hence there is a need for an analysis technique that represents a windowing technique with variable-sized regions. Wavelet analysis allows the use of long time intervals where more precise low frequency information is needed, and shorter regions where high frequency information is needed.

**Wavelet Analysis**

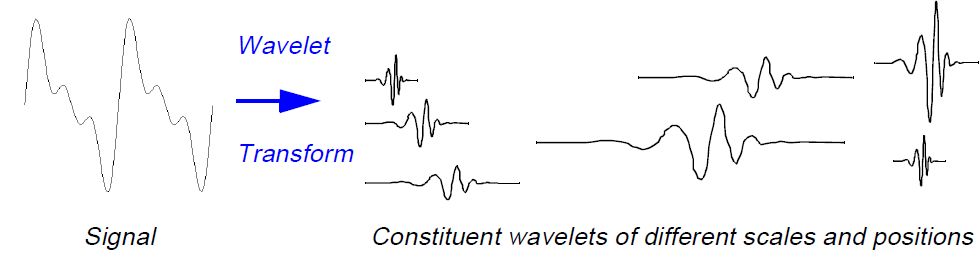
Wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original wavelet. A wavelet is a waveform of effectively limited duration that has an average value of zero.



**Types of Wavelet Transforms:**

1. Continuous Wavelet Transform (CWT)   
   The continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function ψ:  
   

The output of the CWT are many wavelet coefficients C, which are a function of scale and position. These coefficients on multiplication by scaled and shifted wavelets yield the constituent wavelets of the original signal.



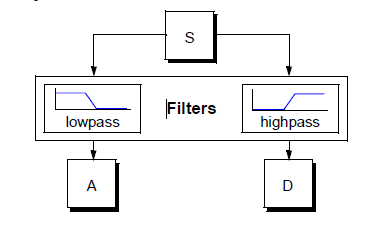
1. Discrete Wavelet Transform (DWT)  
   The Discrete Wavelet Transform(DWT) is a wavelet analysis technique in which the scale and position of the wavelets are varied in powers of two which is called dyadic scales and positions.

The wavelet transform method used in this project is DWT as the results can be more accurately determined by processing less number of data sets.

**Methodology used for performing DWT**

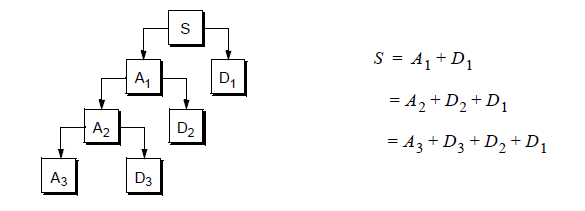
For most real world signal the low frequency content forms the most important part. For example consider human voice. If high frequency component is removed the voice sounds different but what is being said can still be understood. Now if low frequency components is removed then only gibberish is heard. Hence it can be said that the high frequency component imparts the flavor to the signal whereas the low frequency component forms the identity of the signal.

It is for this reason that discrete wavelet transform is divided into approximations and details. The approximations (A) are the high-scale, low-frequency components of the signal whereas the details (D) are the low-scale, high-frequency components of the signal.



The original signal, S, passes through two complementary filters and emerges as two signals.

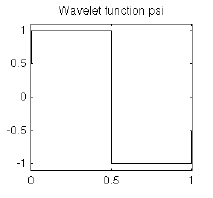
**Wavelet Decomposition Tree:**  
The decomposition process is iterated, with successive approximations being decomposed in turn. In this way the original signal is broken down into many lower-resolution components. This is called the wavelet decomposition tree.

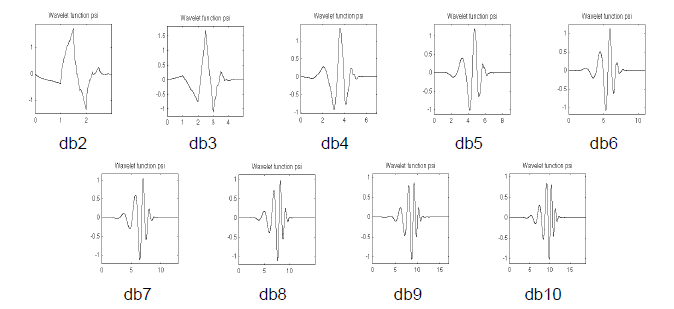
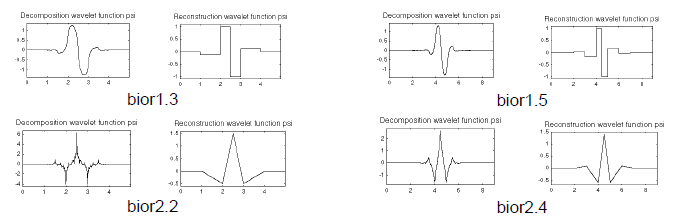
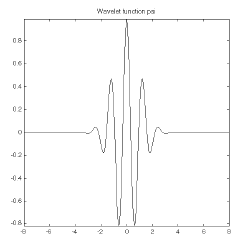
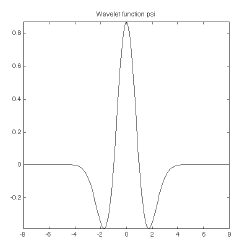


The detail coefficients (D) consist mainly of the high-frequency noise, while the approximation coefficients (A) contains much less noise than does the original signal. Hence in this way the decomposition coefficients of a signal are obtained.  
**Wavelet Families:**

There are various Wavelet Families from which the mother wavelet for analysis is chosen. Some of them are

1. Haar  
   The simplest of Wavelets is the Haar wavelet which resembles a step function.



1. Daubechies  
   Often written as dbN, where N is the order, and db the “surname” of the wavelet, was invented by pioneer of Wavelet research, Ingrid Daubechies  
     
   
2. Biorthogonal  
   This family of wavelets exhibits the property of linear phase, which is needed for signal and image reconstruction.  
   
3. Morlet  
   This wavelet has no scaling function, but is explicit.  
   
4. Mexican Hat  
   This wavelet has no scaling function and is derived from a function that is proportional to the second derivative function of the Gaussian probability density function.  
   

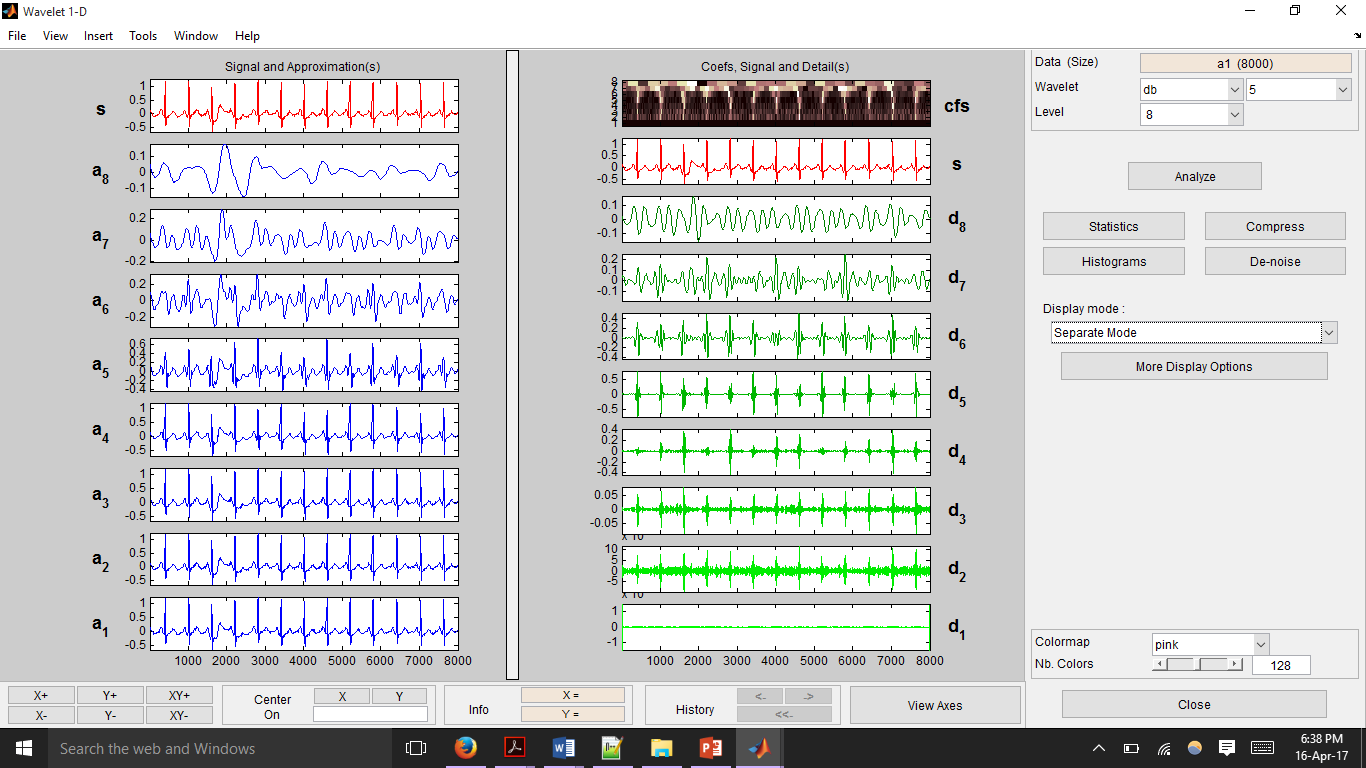
**Wavelet Transform on ECG signals:**  
ECG signals plays and important role in primary diagnosis and analysis of heart diseases. When an ECG is recorded many kinds on unwanted noise is also recorded with it. These noises cause an alternate shift in baseline of the ECG signal. A process of removing the baseline drift of a signal is called as de-trending.  
Wavelet transform is used in this project to de-trend the ECG signal that is obtained from the sensor.  
**Wavelet Selection:**  
The Daubechies wavelet of the DWT wavelet family is selected because the shape of the ECG signal and that of db5 is same. Also Daubechies wavelet families are similar in shape to QRS complex and their energy spectrums are concentrated around lower frequencies.  
**Methodology:**  
The signal is loaded into MATLAB’s Wavelet Analysis and Design Toolbox and the following results were obtained.  
 

Fig1

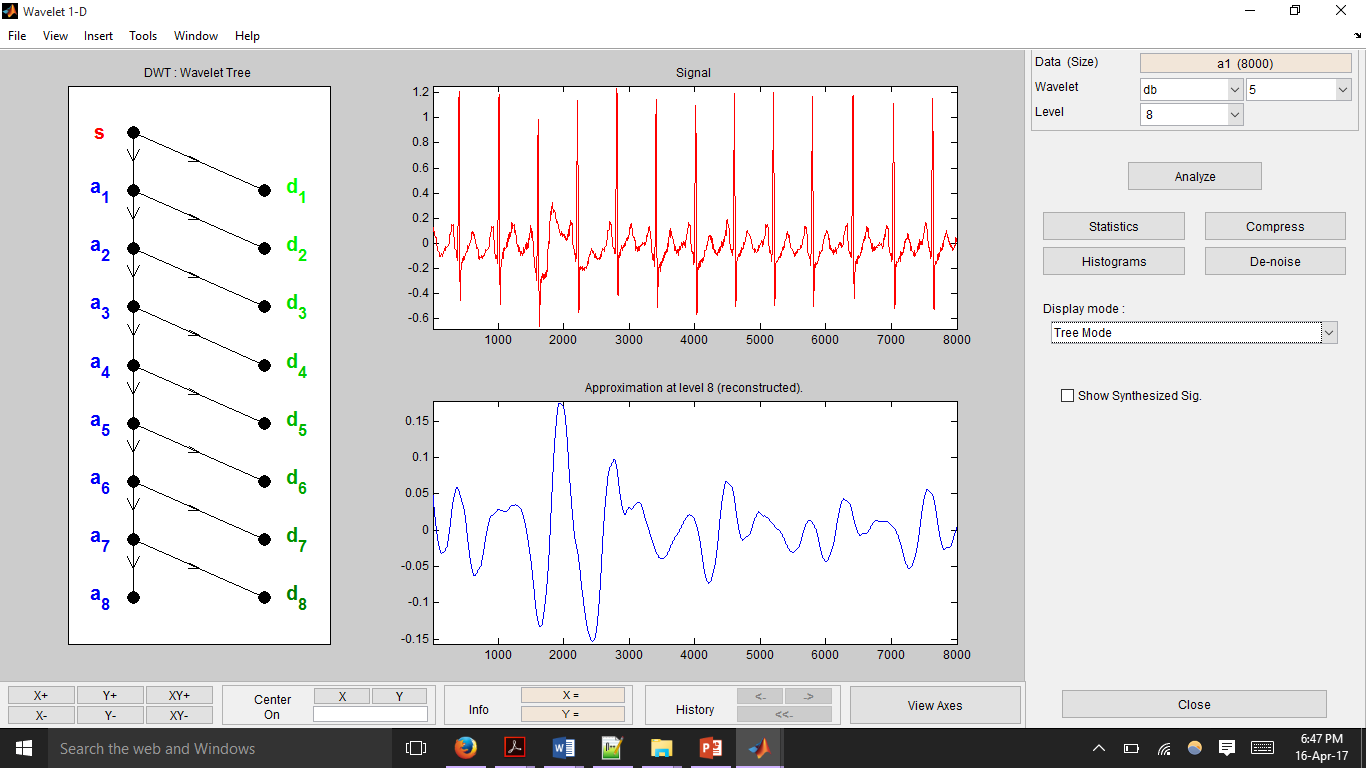


Fig 2

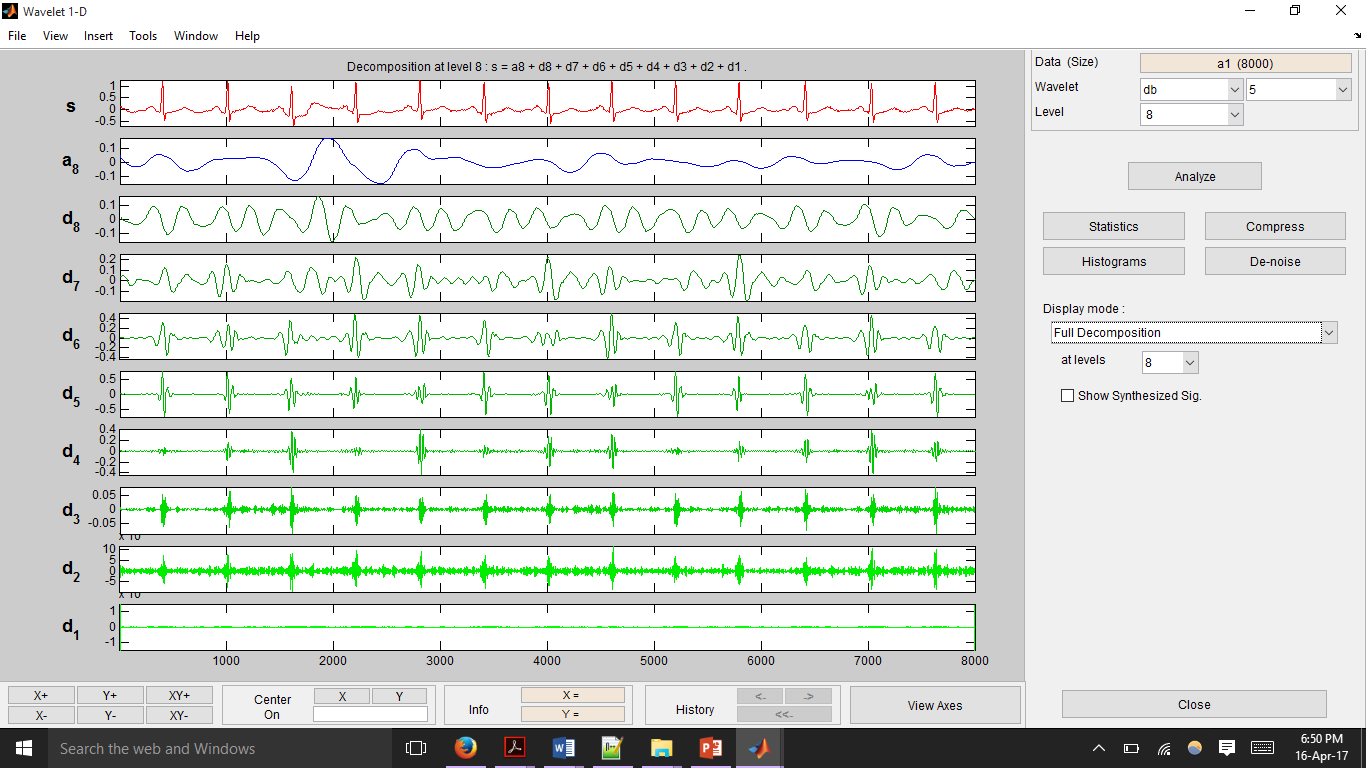


Fig 3

Fig1 shows the decomposition of the signal in separate mode. The right hand side depicts the high frequency decomposition components of the signal whereas the left hand side depicts the low frequency decomposition components of the original signal.

Fig2 shows the decomposition in tree mode. The wavelet tree is shown on the left. The original signal and the last low frequency approximation of the signal is shown on the right.

Fig3 shows the full decomposition of the signal according to the wavelet tree.  
  
**Methodology:**

From the decomposition of ECG signal it is seen that the low frequency component causes the baseline wandering. Hence these components have to be removed from the original signal to get a clean signal free from baseline wandering.  
From the above figures we can see that the low frequency component of the decomposed signal is a8.Therefore this component is subtracted from the original signal to get a de-trended signal.  
De-trended Signal = Original Signal – (A8)

Thus in this way Wavelet Transform is used to remove the baseline wandering present in the ECG signal.

**Output obtained:**

****

fig 4

****

fig5

****

fig 6

****

fig 7

fig4 shows the original ECG signal recorded from the sensor

fig5 shows the level8 approximation obtained after Wavelet analysis on the ECG signal

fig6 shows the de-trended signal

fig7 shows the RR-peaks detected after the detrending of the signal

**ADD ARX intro for the first time, theory and equations**

In conclusion, seven time domain, three non-linear, 13 frequency domain and five/three (since two coeffs don’t have any variation) ARX features were obtained from each sample. These 26 features are then fed into the classifiers to train them and obtain the optimal set of weights for classification of any new input data.

**3.3.4 Feature Reduction**

In order to train the algorithm to achieve a high training accuracy and to obtain a good confusion matrix, a large number of features were extracted from the ECG signals as mentioned in section 3.3.3. While this method of using a large number of features works to obtain desired results, the feature set can be optimized further to remove redundant features if any, or remove those features which contribute minimally toward the training accuracy of the algorithm.

FRAGMENTED SENTENCES. CONSIDER REVISING this paragraph:

Feature reduction is an important step to reduce computation cost while training the classifier algorithm, as reducing the number of features reduces the dimension of the input dataset matrix (or input layer to hidden layer mapping for the ELM classifier) and reduce the number of multiplicative and additive operations being performed. Feature reduction also contributes in reducing the computational cost in the feature extraction stage. Some features (such as the non-linear features) consume a lot of time to calculate due to the use of loops computation cost multiple times before obtaining the final result. By applying feature reduction techniques we come to see that the such a feature does not contribute much toward the accuracy of the algorithm, then calculation of that feature from the input sample can be removed entirely to speed up the real time application process on the Raspberry Pi. Finally, reducing the number of features helps maintain cleaner, slimmer background literature and computer code, for the overall process of training and classification. This allows for easier debugging and scope for modification and optimization if needed in the future.

To obtain at the minimal set of features, MATLAB combnk function was made use of. This function provides a matrix containing all the combinations possible when ‘k’ entries are selected out of an array of ‘n’ elements. This function was applied to the feature set being used as obtained from section 3.3.3 by setting n to 26 (which is the total number of features) and setting k to 25. Combnk was used to obtain all the 25 combination of features possible from the total of 26 features. For each combination of features, the classifiers were trained and k-fold cross validation accuracy was calculated. Seeing that selecting 25 of the 26 features still provided a training accuracy comparable to the original feature set, k was reduced to 24 and the algorithm was retrained for all 24 combinations of features taken from the full set of 26 features. Again, a training accuracy comparable to that obtained while using all the features was obtained. Thus, combinations of 23 features were used to train the algorithms. In this manner, the value of ‘k’ (the number of features selected to train the algorithm) was reduced until a noticeable change in the training accuracy was observed.

The training accuracy remained consistent with the full feature training accuracy for all combinations of features when considered 17 at a time. However, a significant drop in accuracy was observed when lesser than 16 features were used.

\*VERIFY all n and k values mentioned above

\*ADD TABLE WITH ACCURACY RESULTS for k values for BOTH algos

\*ADD a final note with actual training accuracy that was there before and after the feature reduction was performed … IN RESULTS AND DISCUSSION

**3.3.5.Classifiers**

**And intro of this, say two types of classifiers exist. Descision boundary type and nueral network type. Mention need/adv/disadv for each.**

**3.3.5.1.Extreme Learning Machine[ELM]**

Classification is the process of identifying which sub category (or class) a certain sample belongs to. Several classification algorithms have been developed and tested for a number of datasets, for example, Support Vector Machines, Naive Bayes Classifier and Neural Networks. Some classifiers perform better than others for a certain application, or for a specific dataset. There are a number of parameters that are looked into which selecting a certain classifier for a certain application, like accuracy, training time, testing time and for neural networks, the number of hidden layers, and the number of neurons in each layer. One of the classifiers that is chosen for this dataset is the Extreme Learning Machine [ELM].

**3.2.5.1.1.** A neural network is a processing unit consisting of sub units called neurons, which are interconnected to each other. These interconnects are assigned weights, representing the acquired knowledge, which may or may not be changed as the classifier is trained using the training samples. A neural network consists of an input and output layer, along with one or more hidden layers.

**3.2.5.1.2.** In machine learning, if the classifier is being trained using labelled data, that is, the corresponding target output is given for a certain input, it is known as supervised learning. If the classifier is being trained using unlabelled data, and clustering algorithms are required, it is known as unsupervised learning.

**3.2.5.1.3.** A feed forward network is one in which the values at the input are propagated towards the output through the hidden layers without being looped back to any preceding layers as an input.

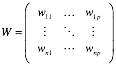
**3.2.5.1.4.** Back propagation consists of values that are fed forward through to the output, the error is calculated, and fed back to the preceding layers in order to correct the weights.

**3.2.5.1.5.** ELM is a single hidden layer, feed forward neural network. The weights connecting the input nodes to the hidden layer nodes are assigned randomly and never updated. The weights between the hidden nodes and output nodes are learnt in a single step. These models can learn several times faster than neural networks trained using the back-propagation algorithm. They are also known to produce decent generalisation capability.

**3.2.5.1.6.** The architecture of an ELM model is fixed a-priori, that is, the number of neurons in each of the layers is fixed before training. The number of input layer nodes is equal to the size of the feature set being input to the classifier. The number of output neurons is equal to the number of classes of samples being fed. The number of hidden layer neurons is chosen based on which value gives highest accuracy without over-learning.

**3.2.5.1.7.** Kernels are a computationally efficient way of changing the dimensions of the feature set of a sample using dot products. If a kernel is used before the classifier, the dimensions of the input layer will change to the size of feature set after kernel is applied. Examples of some popular kernels include Radial Basis Function [RBF] Kernel, Polynomial Kernel, Fisher Kernel, et cetera. The kernel in use in this project is the RBF kernel. It is given by equation( ), where x is the feature set of a training sample, 𝝁 is one of the kernel centres, and σ is the distance metric that is varied to check which gives best accuracy.

**3.2.5.1.8.** The dataset is split into training data and testing data. The training data is used to train the classifier, after which the test data is used to test the accuracy of the classifier. The percentage of training and testing data is varied in order to prevent under-learning and over-learning and obtain best accuracy.

**3.2.5.1.9.** The Input Weights and Biases of Hidden Neurons are generated and assigned randomly. These values are never changed. Let the Input Weights be designated as W and the Bias as B. The dimensions of Wis (n,k), where n is the number of hidden layer neurons and k is the dimension of the feature set of sample after passing it through a kernel. The dimensions of B is (n,p), where p is the number of input training samples given as a batch. If so, each column vector of matrix B will be identical.

**3.2.5.1.10.** The Input Weights are multiplied with the input training data and the Bias is added to it. Let the training data vector be designated as X. The output here is designated as Htemp, given in equation( ).

**3.2.5.1.11.** The Activation Function is used to calculate the output of each neuron in the hidden layer. Examples of Activation functions include Sigmoid, Sinusoidal, Hard Limit, Triangular Basis function, Radial Basis functions. Htemp is the input argument to the activation function, which results in H, as shown in equation( )

**3.2.5.1.12.** The activation function that is used in this project is the Sigmoid function, given by equation( ).

It is to be noted that H has the same dimensions as Htemp.

**3.2.5.1.13.** The Output Weights, denoted by Wo, are calculated by multiplying the Moore-Penrose pseudo-inverse of H with the targets of training data, denoted by T, whose elements represent the target values of each class of training data. This is represented in equation( ).



**3.2.5.1.14.** Once the classifier is trained, that is, all weights and biases are obtained, it is tested with the test sample data set.

First, the Input Weights, W, are multiplied with the Input Feature dataset (after passing through kernel function with centres as the same as that of training phase), given by Xtest, and added with the Bias, B, to obtain Htest as given by the equation( ).

**3.2.5.1.15.** The output of activation function is calculated by passing Htest as an argument to the activation function that was used in the training phase as well.

**3.2.5.1.15.** The actual output of testing data, Ytest is obtained by multiplying the output of activation function, H, with the Output Weights, Wo, which was assigned during the training phase. This is represented by equation( ).

**3.2.5.1.16.** The predicted output class is the index of the maximum value in the output vector Y, as given by the equation( ).

**3.2.5.1.17.** ELM has been found to be good at learning easy functions and performing well for small number of labelled data. They are also incredibly fast to train and have fewer parameters that need to be trained. They provide comparable accuracies to that of other classifiers like Support Vector Machines [SVM], and other deep neural network architectures, for a highly reduced training period and reduction in size of the neural network, owing to the single hidden layer and random initial assignment of input weights, which are not modified during the rest of the training process.

**3.3.5.2 Support Vector Machine**

A support vector machine (SVM) is a classifier that works on finding a decision boundary that can separate the classes of a dataset. Similar to a two dimensional problem where lines and polynomial curves are used to separate the data belonging to different classes, SVMs use separating surfaces/planes of higher dimensions called hyperplanes. SVMs have a primary hyperplane that behaves as the actual decision boundary for classification, but also uses two other hyperplanes in order to achieve the optimized primary hyperplane. Optimization of the primary hyperplane is done generally solving a Lagrangian dual to the actual geometric equation that needs to be solved.

**Change numbers as per the final result of combnk**

As the number of features that have been extracted for each samples in the dataset is twenty-six, all the data points and hyperplanes exist in the twenty-sixth dimension. This means that twenty-six coordinates exist to describe each data point. A hyperplane in such a dimension is given by (eqn 1), where , and have 26 coordinates (eqn0)

(eqn 0)

(eqn 1)

(eqn 2.1)

(eqn 2.2)

If (eqn 1) is considered to be primary hyperplane that takes on the role of being the decision boundary, there exist two other hyperplanes lying on either side of the primary hyperplane given by (eqn 2.1) and(eqn 2.2). The purpose of the two adjacent decision boundaries is to aid in arriving at a geometric optimization problem. What needs to be ensured is that the adjacent hyperplanes lie as far away from each other as possible without misclassifying any of the data samples. In other words, the margin between the primary hyperplane and the adjacent hyperplanes needs to be maximized. Talsk about support vectors/ SV points. The margin/distance that exists between the adjacent hyperplanes can be given by (eqn 3.1), and this is the value that needs to be maximized.

(eqn 3.1)

(eqn 3.2)

Maximizing (eqn 3.1) is the same as minimizing (eqn 3.2). Equation (eqn 3.2) can be solved directly or, as per [ADD REF svm-cs229] the Lagrangian’s dual can be solved. The Lagrangian dual for (eqn 3.2) is given by (eqn 4).

(eqn 4.1)

Subject to constraints (as per Karush-Kuhn-Tucker);

(eqn 4.2)

(eqn 4.3)

(eqn 4.4)

These yields the final Lagrangian optimization equation:

(eqn 4.5)

In equation (eqn 4.5), represents the total number of training samples supplied to the classifier, stands for the i-th or j-th training sample and stands for the class label of the i-th or j-th sample. is the Lagrangian multiplier for the i-th sample. Lagrangian multiplier values are assigned based on whether they are support vectors or not, and stands for the kernel function that is applied on the input dataset. More information on the application of and need for kernels is given later on in this section.

**ADD f(z) equation … decision boundary equation. Also add what exactly a SV is.**

Various algorithms exist to solve the Lagrangian dual. Here, a simplified version of the Sequential Minimal Optimization (SMO) algorithm proposed in (ADD REF svm\_smo.pdf) has been used. **EXPLAIN what SMO is doing**

Earlier, in equation (eqn 4.5) was used to represent the use of a kernel on the dataset. Kernel provide ways of changing the dimension of a dataset using simple dot products, which remain computationally efficient. By changing the dimension of the dataset the present feature space is mapped to another feature space. Using the right kernel an allows some data points that were not separable by hyperplanes in the original dimension to becoming separable in the new dimension that they are mapped to. Thus, this becomes a tremendously powerful method by which a larger set of data points can be classified correctly, helping increase the training accuracy of the algorithm. Some of the common kernels that are used are the polynomial kernel, Gaussian (also called the Radial Basis Function (RBF)) kernel and the wavelet kernel. Multiple kernels have been used with the SVM classifier and based on the training accuracy obtained, the Gaussian kernel (given by eqn 5) has been selected.

(eqn 5)

ADD other kernel equations?

Another important concept about decision boundary type of classifiers like SVMs is regularization. It is common for a dataset to have a few outliers and anomalies. Since the algorithm attempts to classify all the samples correctly, sometimes the algorithm can overfit or go out of the way to include an anomalous sample to the class it had been assigned. As a result, the decision boundary that the algorithm comes up with does not remain generic to new test data that is fed in. Regularization is a method by which some equation parameter/coefficient can be tweaked and tuned to control the algorithm’s sensitivity to outliers and anomalies. This control over how closely the algorithm should fit to the training data supplied to the algorithm is very important for real world applications, and has been implemented in the SVM classifier as the parameter ‘C’. The use of a regularization parameter puts an additional constraint on the Lagrangian coefficients . That is, apart from the constraints given by (eqn 4.2), there is an additional constraint as given by (eqn 6)

(eqn 6)

A final thought on SVMs is how the SVM, once trained with the weights of the decision boundary, is used to classify a new data sample. This process of classification of some i-th sample after obtaining a trained model is very straightforward. The (eqn 7) is calculated to obtain a number that represents which class the sample belongs to. If the number calculated is positive, then the sample belongs to the first class. In case the number is negative, it is classified to the other class.

**Prediction = class 1 if pv < 0 and class 2 if pv > 0**

**Pv is prediction value**

To summarize all that was explained above and piece together how all the steps fall into place to train and test the SVM, the following steps have been provided:

1. The input matrix dataset is fed to the algorithm, and all the features are normalized to ensure that all the features contribute equally to the learning algorithm.
2. A combination of regularization parameter and Gaussian variance value are chosen.
3. To this normalized data set, the desired kernel function is applied. If the kernel being used is a Gaussian kernel, then the chosen in the previous step is used.
4. Now, the SVM is actually ‘trained’ using some optimization algorithm like Sequncial Minimal Optimization (SMO) to solve the Lagrangian dual and obtain a set of weights for the hyperplane. Further reference for the working of the SMO algorithm is provided in the appendix.
5. Once the weights for the hyperplane is obtained, the same kernel is applied to the normalized validation dataset which is sent to the trained classifier to calculate the accuracy.
6. Accuracy calculation may be performed either on a single validation dataset, or averaged over multiple folds of data using a technique like k-fold cross validation. For each fold the algorithm is retrained and the weights are recalculated
7. All the above steps are performed for new pairs of values of the regularization parameter and Gaussian variance , and the pair that yields the best accuracy is chosen to train the final SVM model.

**Generically mention how C and sigma is obtained and then mention:**

**HOW actual training and testing is done**

**3.3.5.3. Meta-Cognitive Extreme Learning Machine [McELM]**

In the case of ELM, the architecture of the neural network is fixed a-priori. The number of neurons in the hidden layer is fixed, and this number determines to a large extent the amount of over-learning or over-fitting that occurs during training. Also, determining the number of fixed hidden layer neurons before training the classifier is a challenge and is largely done by trial and error, or by the brute force method where training and testing accuracies are calculated for a varied range of number of hidden layer neurons and the number which gives highest accuracy is chosen. The meta-cognitive approach to ELMs provides a smart way of obtaining this number of hidden layer neurons, and builds the network as it is being trained.

**3.3.6. Validation**

Validation is an incredibly important part of testing how well a classifier works. Having trained the classifier for a certain portion of the dataset, it is essential to find out how well the classifier performs when exposed to a set of test data it has not seen before. Sometimes, the data that a classifier is trained for might result in the classifier being biased towards a certain class and might perform poorly when given test data that belongs to a different class. So it is important to test the classifier using different test data sets or combinations of data for a classifier which has been trained for a corresponding different combination of training data set, so that upon averaging out the accuracies obtained from each train-test combination, we obtain an overall generalised picture of how the classifier performs in the real world and allows us to prevent over-fitting of the classifier. In cross-validation, the entire dataset is split into subsets of training and testing data, which are complementary to each other, and several rounds of training and testing are done, and results averaged over all the rounds of validation.

**3.2.6.1. Types of Validation**

There are two types of cross validation, namely Exhaustive Validation and Non-Exhaustive Validation. Exhaustive cross-validation consists of Leave-P-Out Cross-Validation and Leave-One-Out Cross-Validation. In exhaustive cross validation, all possible combination of samples of data set are chosen and used to train and test the classifier. Non-Exhaustive cross-validation consists of K-Fold Cross-Validation, Hold-Out Method and Repeated Random Sub-Sampling Validation. Non-exhaustive cross validation methods do not include all possible combinations of samples from the dataset for training and testing the classifier. These methods are an approximation of the leave-p-out cross validation method that falls under exhaustive cross-validation.

**3.2.6.2. Leave-One-Out Cross-Validation**

This is a type of exhaustive cross-validation and an explicit example of leave-p-out cross-validation for p=1, in which one sample from the dataset is chosen for testing and all other samples are chosen to train the classifier per iteration, and this procedure is iterated for as many times as there are samples in the dataset, only that the sample chosen for testing is different every single iteration. The accuracies over every single iteration are averaged over and an average accuracy is obtained. This method is better than leave-p-out cross-validation in that the number of combinations of train-test data are numerically equal to the number of samples in the dataset. For example, in a dataset having 100 samples, in each iteration, 1 sample is chosen as the test data while the other 99 samples are chosen as the training data.

**3.2.6.3. K-Fold Cross-Validation**

This is a type of non-exhaustive cross-validation in which the dataset is randomly partitioned into k number of equally sized subsets, and in each of k number of iterations, one of the subset is chosen as the test data set while the rest of the k-1 subsets constitute the training data set, and the subset chosen to test and the subsets chosen to train the classifier are unique in every iteration of training and testing. The accuracies obtained in each kth “fold” are averaged out to obtain an average accuracy that gives us an idea about the performance of the classifier. The advantage of this method of cross validation is that every single subset is used for both training and testing and a certain subset is used for testing only one time. The value of k determines the percentage of the dataset that is used for training and testing. For example, if k=4, for a dataset having 100 samples, the dataset is split into 4 subsets of 25 samples each. In each iteration, 75 samples are used for training and 25 samples are chosen as the test dataset.

This procedure happens 3 more times for the other 3 subsets of 25 samples, while the rest of the 75 samples in each fold participate as the training set. The accuracies over the 4 folds are averaged and presented as the accuracy of the classifier. When k equals the number of samples in the dataset, it becomes nothing but leave-one-out cross-validation.

**3.2.6.4. Confusion Matrix and Table of Confusion**

A confusion matrix is a table of values which provides us with a way to visualise the performance of an algorithm, or in the case of pattern classification problems, the performance of a classifier. The rows represent instances of the actual class while the columns represent instances of the predicted class. From this matrix, we can get to know how many times the classifier is classifying the sample correctly and and how many times a certain class of sample is misclassified as another class, and the number for each class. The table of confusion (also called the confusion matrix) is one consisting of 2 rows and 2 columns. The 4 elements of this matrix give us the number of True Positives, False Negatives, False Positives and True Negatives. True positives are those samples which were classified correctly for a specific class. False negatives are those in which the actual label was that of the class we are calculating the matrix for, and the classifier predicted wrongly. False positives are those in which the predicted label was that of the class for which we were finding the matrix for, but the actual label was that of another class. True negatives are those in which all the other classes apart from the class we were calculating the matrix for, were classified correctly.

|  |  |  |
| --- | --- | --- |
| Total Population | Prediction Positive | Prediction Negative |
| Condition Positive | True Positive | False Negative |
| Condition Negative | False Positive | True Negative |

**3.2.6.5. Sensitivity and Specificity**

Sensitivity is a measure of the proportion of positives that are correctly classified as positives. It is also known as recall, hit rate or true positive rate.

Specificity is a measure of the proportion of negatives that are correctly classified as negatives. It is also known as true negative rate.

Both of these are measures of the performance of a classifier for a certain dataset.

**Need to keep citing sources in different sections of the report … Do so at the end while reading through and figuring out where all the citation is required.**

CHAPTER – 4

**RESULTS AND DISCUSSION**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SVM**

**Training multiple times/different combos of C and sigma, to obtain C and sigma …. In results and discussion**