

DETECTION OF ECG ARRHYTHMIA CONDITIONS USING CSVM AND MSVM CLASSIFIERS

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

DASAM SRI SRINIVASA SATHYA MANI PAVAN TEJA (38130051)

DASAM VENKATA RAMA CHANDRA RAO (38130052)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119

MAY - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **DASAM SRI SRINIVASA SATHYA MANI PAVAN TEJA (38130051)**, **DASAM VENKATA RAMA CHANDRA RAO (38130052)** who carried out the project entitled **"DETECTION OF ECG ARRHYTHMIA CONDITIONS USING CSVM AND MSVM CLASSIFIERS"** under my supervision from September 2021 to May 2022.

Internal Guide

Ms. A.SABARIVANI, M. TECH., (Ph.D.)

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **DASAM SRI SRINIVASA SATHYA MANI PAVAN TEJA (38130051)**
and **DASAM VENKATA RAMA CHANDRA RAO (38130052)** hereby declare

That the Project Report entitled **“DETECTION OF ECG ARRHYTHMIA
CONDITIONS USING CSVM AND MSVM CLASSIFIERS”** done by under the
guidance of **Ms. A.SABARIVANI, M. TECH., (Ph.D.)** is submitted in partial fulfillment
of the requirements for the award of Bachelor of Engineering degree in Electronics
and Communication Engineering.

1)

2)

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E, Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E, Ph.D.** Head of the Department, Dept. of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Ms. A.SABARIVANI, M. TECH., (Ph.D.)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

Abstract:

An automatic classification of four beat types Normal (N), premature ventricular contraction (PVC), Supraventricular premature or ectopic beat (SVPB) and Fusion of ventricular and normal beat (FUSION) is implemented using a multi-class Support Vector Machine (MSVM) and Complex support Vector Machine (CSVM) algorithm. Pre-processing and R peak detection is performed with the WFDB Software package which reads the annotation and finds the R (peak) location. R (peak) location is used as a reference to detect peaks in other wave such P and T are extract ECG beat. Discrete Cosine and Sine transforms or Discrete Fourier Transform (DFT) were used for feature extraction and dimensionally reduction of the input vector at the input of the classifier.

The aim of the study was to improve multi-class SVM by extending traditional SVM algorithms to complex spaces so as to simultaneously classify four types of heartbeats. The approach can also account for phase changes in the signal. Accuracy between 86% and 94% are obtained for MSVM and CSVM classification respectively. This system illustrate that the proposed beat classifier is very reliable, and that it may be a useful tool for detection arrhythmia conditions and perform classification automatically.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	V
	LIST OF FIGURES	ix
	LIST OF ABBREVIATION	x
1	INTRODUCTION	1
	1.1 Electrocardiograph	1
2	Literature Survey	3
	2.1 Literature Survey	3
	2.1.1 Literature Survey 1	3
	2.1.2 Literature Survey 2	3
	2.1.3 Literature Survey 3	4
	2.1.4 Literature Survey 4	4
	2.1.5 Literature Survey 5	5
	2.1.6 Literature Survey 6	5
3	Aim And Scope	7
	3.1 Aim	7
	3.2 Scope	7
	3.3 Block Diagram	8
	3.3.1 ECG Signal	9
	3.3.2 Pre-Processing	9
	3.3.3 Noise Cancellation	10
	3.3.3.1 Impulsive Noise	11
	3.3.4 Peak Detection	11
	3.3.5 Measuring Intervals	12

CHAPTER NO.	TITLE	PAGE NO.
	3.3.6 Decision Making	13
4	Materials And Methods	14
	4.1 Flow Chart	14
	4.1.1 Raw ECG Signal	15
	4.1.1.1 Electrocardiogram signals	15
	4.1.2 Pre-Processing	16
	4.1.3 Feature Extraction	17
	4.1.4 Classification	22
	4.1.4.1 Csvm	23
	4.1.4.2 Msvm	24
	4.1.4.3 Wavelet transform	26
	4.1.4.4 R-Wave	27
	4.2 Software's Used	28
	4.2.1 Matlab	28
	4.2.2 Signal Processing	29
	4.2.2.1 Signal Analysis and Measurements	30
	4.2.3 Filter Design and Analysis	31
	4.2.4 De noise Signals	31
	4.2.5 Simulink	31
	4.2.5.1 Simscape	33
	4.2.5.2 Simscape Multibody	33
	4.2.6 Database	34
	4.2.7 Peak Position	34
	4.2.8 Curve Fitting Toolbox	36
	4.2.8.1 Opening the Curve Fitting Tool	37

CHAPTER NO.	TITLE	PAGE NO.
	4.2.8.2 Importing the Data	37
	4.2.8.3 Fitting the Data	38
	4.2.8.4 The Data Fitting Procedure	39
5	RESULT AND DUSSION	40
	5.1 Result	40
6	SUMMARY AND CONCLUSIONS	51
	6.1 Summary	51
	6.2 Conclusions	51
	REFERENCES	52

LIST OF FIGURES

FIGURE NO.	FIGURES TITLE	PAGE NO.
3.1	Block Diagram	8
4.1	Flow Chat	14
4.2	ECG signal	16
4.3	peck detection	35
5.1	Input ECG Signal 1	41
5.2	Input ECG Signal 2	42
5.3	Pre- processing Signal 1	43
5.4	Pre -processing Signal 2	44
5.5	Feature Extraction Signal 1	45
5.6	Feature Extraction Signal 2	46
5.7	Peak detected 1	47
5.8	peak detected 2	48
5.9	Result 1	49
5.10	Result 2	50

LIST OF ABBREVIATION

ECG	ELECTROCARDIOGRAM
PVC	PREMATURE VENTRICULAR CONTRACTION
N	NORMAL
SVPB	SUPRA VENTRICULAR PREMATURE BEAT
FUSION	FUSION OF VENTRICULAR AND NORMAL BEAT
MSVM	MULTI-CLASS SUPPORT VECTOR MACHINE
CSVM	COMPLEX SUPPORT VECTOR MACHINE
SMO	SEQUENTIAL MINIMAL OPTIMIZATION
DCT	DISCRETE COSINE TRANSFORM
DFT	DISCRETE FOURIER TRANSFORM
DST	DISCRETE SINE TRANSFORM
WFDB	WAVEFORM DATABASE
CVD	CARDIOVASCULAR DISEASES
CNN	CONVOLUTIONAL NEURAL NETWORK
HOS	HIGHER ORDER STATISTICS
APC	ARIAL PREMATURE CONTRACTION
RBBB	RIGHT BUNDLE BLOCK BEAT

CHAPTER-1

INTRODUCTION

1.1 ELECTROCARDIOGRAPH

ECGs provide a graphic representation of the electrical activity of the heart muscle. This study evaluates ECG classification using pre-processing routines, feature extraction and selection, and ECG beat classification using MSVM and CSVM. DCT and DST coefficients were used to extract features presented at the input vector of the MSVM classifier, whereas DFT coefficients are used for creating an input vector to the CSVM classifier. The MSVM and CSVM classifier is used to distinguish the four ECG arrhythmias types using the DCT, DST and DFT coefficients as feature vectors at the input vector of the classifier. Sequential minimal optimization (SMO) approach is used to train the CSVM and compute the corresponding complex hyper-plane parameters.

Electrocardiogram (ECG) is widely used for the diagnosis of cardiac arrhythmia conditions. An automatic classification of four beat types Normal (N), premature ventricular contraction (PVC), Supraventricular premature or ectopic beat (SVPB) and Fusion of ventricular and normal beat (FUSION) is implemented using a Multi-class Support Vector Machine (MSVM) and Complex Support Vector Machine (CSVM) algorithms. The ECG signals used in these studies were obtained from the European ST-T Database. A number of beats from different leads and patients were selected for training and classifier performance. Success. ECG arrhythmia classification usually requires optimizing the following procedures: Pre-processing and beat detection, feature extraction and selection, and classifier optimization. Pre-processing and R peak detection is performed with the WFDB Software Package. This reads the annotation

And finds the R (peak) location. R (peak) location used as a reference to detect peaks in other wave such P and T and extract ECG beat. ECG beats are extracted after

windowing the signal using 106 samples before the R and 106 samples after the R-peak. Discrete Cosine and Sine transforms or the Discrete Fourier Transform (DFT) were used for feature extraction and dimensionality reduction of the input vector at the input of the classifier. Studies after selecting either 100 or 50 Fourier coefficients for reconstructing individual ECG beats in the feature selection phase were performed. MATLAB software routines were used to train and validate both the CSVM and the Multi-class Support Vector Machine (MSVM) classifier

A Complex kernel function, (Gaussian RBK) with 5-fold cross validation was used for adjusting the kernel values. Sequential minimal optimization (SMO) was used to train the CSVM and compute the corresponding complex hyper-plane parameters. The aim of the study was to improve multi-class SVM by extending traditional SVM algorithms to complex spaces so as to simultaneously classify four types of heartbeats. Results illustrate that the proposed beat classifier is very reliable, and that it may be adopted for automatic detection of arrhythmia conditions and classification. Accuracies between 86% and 94% are obtained for MSVM and CSVM classification respectively. Using CSVM, a 4 class's problem can be classified rapidly by decomposing it into two distinct SVM tasks. Moreover, the present research confirmed that the use of selected number of Fourier coefficients to approximate the ECG beat signal and compress the input features to the classifier can lead to high classification accuracies and improve the generalization ability of the CSVM classifier

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

2.1.1 Literature Survey 1

Title: ECG analysis and classification using CSVM, MSVM and SIMCA classifiers

Author and Year: Jannah, Najlaa, 2017.

Content: Reliable ECG classification can potentially lead to better detection methods and increase accurate diagnosis of arrhythmia, thus improving quality of care. This thesis investigated the use of two novel classification algorithms: CSVM and SIMCA, and assessed their performance in classifying ECG beats. The project aimed to introduce a new way to interactively support patient care in and out of the hospital and develop new classification algorithms for arrhythmia detection and diagnosis. Wave (P-QRS-T) detection was performed using the WFDB Software Package and meltier solution wavelets.

Drawbacks: less accuracy rate

2.1.2 Literature Survey 2

Title: Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

Author and Year: Amin Ulloa, Syed Muhammad Anwar, Muhammad Bilal and Raja Majid Mahmood, 2020.

Content: The electrocardiogram (ECG) is one of the most extensively employed signals used in the diagnosis and prediction of cardiovascular diseases (CVDs). The

ECG signals can capture the heart's rhythmic irregularities, commonly known as arrhythmias. A careful study of ECG signals is crucial for precise diagnoses of patients' acute and chronic heart conditions. In this study, we propose a two-dimensional (2-D) convolutional neural network (CNN) model for the classification of ECG signals into eight classes.

Drawbacks: they used CNN classification only

2.1.3 Literature Survey 3

Title: Cardiac Arrhythmia Detection and Classification from ECG Signals Using XG Boost Classifier

Author and Year: Saroj Kumar Pandey, Rekh Ram Janghel, Vaibhav Gupta¹³ August, 2021.

Content: In this paper, we propose an effective electrocardiogram (ECG) signal classification method using XG Boost classifier. The ECG signals are passed through four phases of data acquisition, noise filtering, feature extraction, and classification. In first phase, dataset is collected from the MIT-BIH arrhythmia database. In second phase, noise is removed using baseline wandering removal filter. In next phase, 45 descriptors from four prominent features which showed good results in previous work, namely, wavelets, higher order statistics (HOS), morphological descriptors, and R-R intervals, are being employed.

Drawbacks: Used XG boost classifier.

2.1.4 Literature Survey 4

Title: Arrhythmia detection using multi-lead ECG spectra and Complex Support Vector Machine Classifiers

Author and Year: Najlaa Jannah a, Silla's Hadjiloucasb, Jameel Al-Maliki, 2020.

Content: Electrocardiograms (ECG) are extensively used for the diagnosis of cardiac arrhythmias. This work investigates the use of machine learning classification algorithms for ECG analysis and arrhythmia detection. Four beat types, Normal (N), Premature Ventricular Contraction (PVC), Atrial Premature Contraction (APC) and Right Bundle Branch Block Beat (RBBB) are simultaneously presented to a Complex Support Vector Machine (CSVM) classifier.

Drawbacks: used only CSVM.

2.1.5 Literature Survey 5

Title: "Complex Support Vector Machines for Regression and Quaternary Classification"

Author and Year: P. Boubou Lis, S. Theodoridis, C. Mavroforakis and L. Evaggelatos-Dalla 2014

Content: An automatic classification of four beat types Normal (N), premature ventricular contraction (PVC), Supraventricular premature or ectopic beat (SVPB) and Fusion of ventricular and normal beat (FUSION) is implemented using a Multi-class Support Vector Machine (MSVM) and Complex Support Vector Machine (CSVM) algorithm

Drawbacks: they use only DFT

2.1.6 Literature Survey 6

Title: "On the application of optimal wavelet filter banks for ECG signal classification"

Author and Year: S. Hadjiloucas, N. Jannah, F. Hwang and R. K. H. Galvano, 2014

Content: Future work on wavelet pre-processing to further compress the input space of the classifier by generating wavelets on the basis of higher order moment criteria

[3] as well as alternative approaches for extending the CSVM input and output spaces to arbitrary dimension using Clifford algebra SVM [4] will be discussed at the conference.

Drawbacks: used only CSVM less accuracy

CHAPTER-3

AIM AND SCOPE

3.1 AIM

Detection of ECG using MSVM and CSVM classifier is used to distinguish the four ECG arrhythmias types using the DCT, DST and DFT coefficients as feature vectors at the input vector of the classifier.

3.2 SCOPE

The proposed work provides a useful methodology for multi-lead ECG analysis and classification of arrhythmia conditions based on CSVM classification following DFT signal pre-filtering. The study focused on the use of four types of beats obtained from the INCARTDB database although other databases with well-annotated files could have also been used. Single-beat classification was performed. SMO was used to compute the hyper-plane parameters and threshold values for CSVM during the training phase. Results illustrate that the proposed beat classifier is very reliable, and that it may be adopted for automatic detection of arrhythmia conditions and classification. Using CSVM, a 4-class problem can be classified rapidly by decomposing it into two distinct SVM tasks. The CSVM algorithm demonstrated excellent performance, mapping the extracted ECG features to a higher dimensional space through the use of two complex hyper-planes. Using the CSVM algorithm indicated a significant improvement in multi-class classification over MSVM. The overall accuracy using ECG beats extracted from six precordial. In terms of using CSVM, the major limitation is that only four types of ECG beats can be classified, this provides motivation to further develop Clifford algebra based ECG analysis classifiers.

3.3 BLOCK DIAGRAM

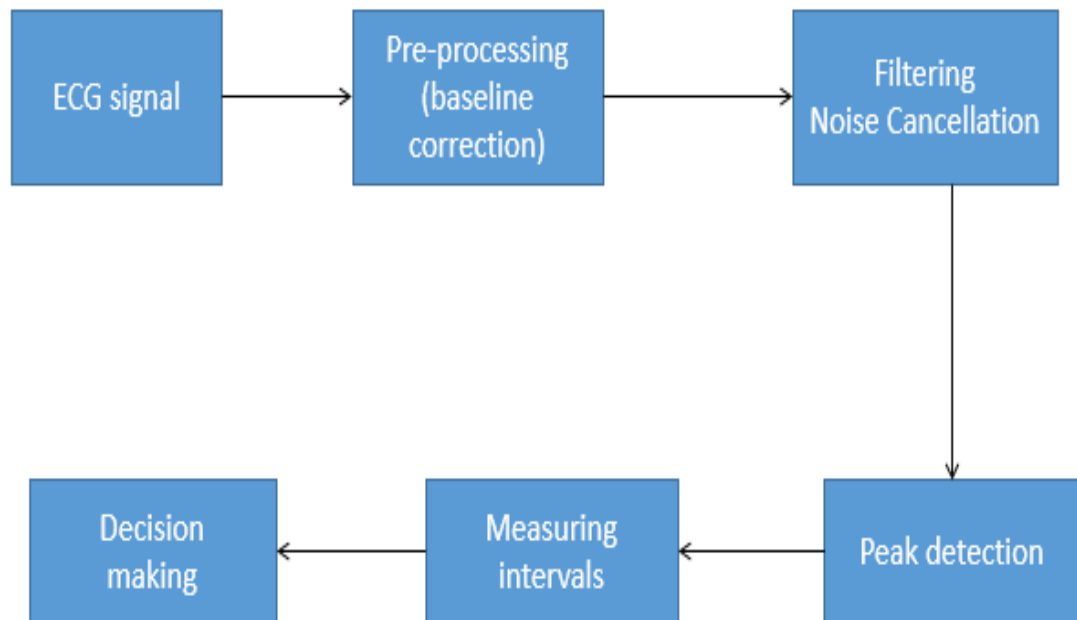


Fig: 3.1 Block Diagram

3.3.1 ECG Signal

An electrocardiograph is a graph recorded by electric potential changes occurring between electrodes placed on a patient's torso to demonstrate cardiac activity. An ECG signal tracks heart rhythm and many cardiac diseases, such as poor blood flow to the heart and structural abnormalities. The action potential created by contractions of the heart wall spreads electrical currents from the heart throughout the body. The spreading electrical currents create different potentials at points in the body, which can be sensed by electrodes placed on the skin. The electrodes are biological transducers made of metals and salts. In practice, 10 electrodes are attached to different points on the body.

3.3.2 Pre Processing

Electrocardiography is a well-established method to monitor and record heart activity. The electrocardiogram (ECG) is obtained by measuring the difference in potential between two points on the body surface. However, surface measurements include various types of noise. The challenge of ECG preprocessing is to filter and minimize these noise components, therefore making it easier to diagnose various heart disorders. In order to create such methods, within this work the most important noise sources, baseline wandering, frequency noise, muscle noise and interference with impulse character are modelled. Subsequently de noising methods were implemented to cancel out the modelled noise components. To evaluate each method, the signal-to-noise ratios (SNR) of the original and filtered signals were calculated. We showed that the so called morphological filtering works best for baseline wander and impulsive noise whereas band stop filtering leads to good results when dealing with narrow-band frequency noise. Finally the modelling and de noising methods were integrated into an existing Mat lab GUI.

Baseline wandering is a type of slow frequency noise which adds to the ECG signal, resulting in a varying baseline. This noise is caused by changes in the impedance between the electrode and the patient's skin. These changes might be caused by movement, breathing or by perspiration, leading to a change of distance between the source of the signal, the heart, and the electrodes. The ST-segment is especially affected by this kind of noise which is particularly relevant in exercise testing. The frequency components of the baseline wander usually lie below 0.5 Hz but, in case of stress tests, this limit can be higher. Thus, these components can be in the same range as low frequency ECG components like the ST segment or the T-wave. This type of interference can be modelled by adding a single sine shaped half-wave to the ECG signal,

$$X_n = A \cdot \sin(2\pi \cdot f \cdot t).$$

3.3.3 Noise Cancellation

Muscle Noise Muscle disturbances occur when skeletal muscles contract, the patient moves or because of shivering due to inappropriate temperature of the environment. The main problem with these interferences is the spectral overlap between the ECG and the muscle noise. Modelling muscle disturbances is quite difficult as it appears at random and therefore has no exact deterministic description and can only be characterized by stochastic terms. In practice, either white Gaussian noise or databases for noise models are used to add muscle disturbances to the ECG signal. For adding those from databases see section 3.1.6. In this section we modelled muscle disturbances by adding white Gaussian noise at random positions to the ECG signal with user-defined start point, length and amplitude of the noise.

3.3.3.1 Impulsive Noise

Muscle noise cannot be described with white Gaussian noise only, because sometimes it displays impulse character too. Additionally impulsive noise may be produced by diathermy, which is a form of heating tissue, when using surgical devices or in occupational therapy. These impulses have a short duration and a high peak, and can therefore be shaped as a triangle function with adjustable height and width, which is added to the signal. The triangle is created by using linear time vectors, and is modelled with the function

$$X(i) = \{k \cdot i + d1 \quad \text{for } (i \leq \text{duration}/2) \\ -k \cdot i + d2 \quad \text{for } (i > \text{duration}/2)\}$$

Where k and d depend on the duration and location of the impulse. In Figure 11 we added five example impulses to the signal.

Noise Data Another method to model noise was implemented in this work by adding noise vectors given from a database such as the MIT-BIH noise stress test data base the recordings showing baseline wander, electrode motion artefacts and muscle noise can be downloaded by the user and then added to the ECG signal with this method. Before adding the noise to the signal, the amplitude of the added vector has to be adapted to the signal. The MIT-BIH Noise Stress Test Database does not just provide noise-only vectors but also noisy ECG signals from real patient recordings. De noising is used during preprocessing of the input signal. Since real-world ECG signals are often disturbed by noise and strongly depend on the individual patient, it is mandatory to perform de noising to get a robust recognition model.

3.3.4 Peak Detection

The R-peak detection is crucial in all kinds of electrocardiogram (ECG) applications. However, almost all existing R-peak detectors suffer from the non-stationarity of both QRS morphology and noise.

To combat this difficulty, we propose a new R-peak detector, which is based on the new. Preprocessing technique and an automated peak-finding logic. In this paper, we first demonstrate that the proposed preprocessor with a Shannon energy envelope (SEE) estimator is better able to detect R-peaks in case of wider and small QRS complexes, negative QRS polarities, and sudden changes in QRS amplitudes over that using the absolute value, energy value, and Shannon entropy features. Then we justify the simplicity and robustness of the proposed peak-finding logic using the Hilbert-transform (HT) and moving average (MA) filter. The proposed R-peak detector is validated using the first-channel of the ECG records of the MIT-BITH arrhythmia database, and achieves average detection accuracy .Various experimental results show that the proposed R-peak detection method significantly outperforms other well-known methods in case of noisy or pathological signals.

3.3.5 Measuring Intervals

Normal Duration Times for the 3 Waves As well as being able to recognize the 3 different parts of the cardiac cycle, each stage should be completed within a specific time period to be considered normal. Although these measurements are in fractions of a second, the ECG paper allows you to count the time in small squares. Measurement in small squares is more universally used than tenths and hundredths of seconds. The P-R Interval The first measurement is known as the "P-R interval" and is measured from the beginning of the upslope of the P wave to the beginning of the QRS wave. This measurement should be 0.12-0.20 seconds, or 3-5 small squares in duration. The second measurement is the width of the QRS which should be less than 3 small squares, or less than 0.12 seconds in duration

3.3.6 Decision Making

The aim of the ECG simulator is to produce the typical ECG waveforms of different leads and as many arrhythmias as possible. My ECG simulator is a mat lab based simulator and is able to produce normal lead II ECG waveform.

The use of a simulator has many advantages in the simulation of ECG waveforms. First one is saving of time and another one is removing the difficulties of taking real ECG signals with invasive and noninvasive methods. The ECG simulator enables us to analyze and study normal and abnormal ECG waveforms without actually using the ECG machine. One can simulate any given ECG waveform using the ECG simulator. The way by which my simulator differs from other typical ECG simulators is that i have used the principle of Fourier series. The calculations used and other necessary descriptions are included in the file attached.

CHAPTER-4

MATERIALS AND METHODS

4.1 FLOW CHART

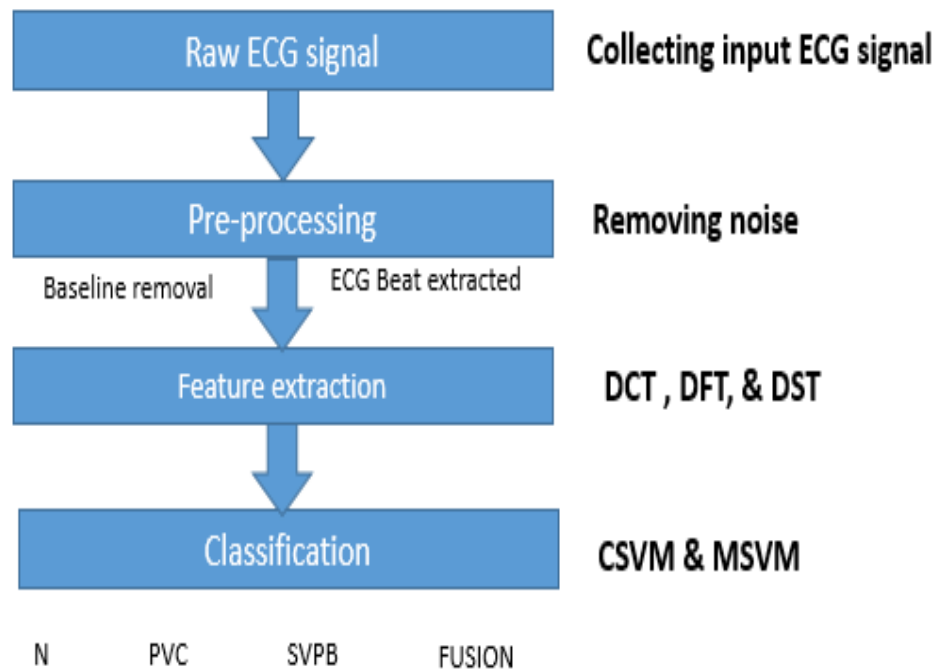


Fig: 4.1 Flow Chart

4.1.1 Raw ECG signal

Collecting input ECG signal from internet. The raw ECG signals are rather noisy and contain both high and low frequency noise components. Electrocardiography (ECG) is the measurement of the bioelectrical activity during the contraction and relaxation stages of the atria and ventricles of the heart, the stimulation of the heart and the conduction of the stimulus. ECG signal has some specific waves which correspond to particular conditions.

4.1.1.1 Electrocardiogram signals

Electrocardiogram (ECG) is an essential medium to measure health and detect diseases for all human beings. It plays a major medicinal role in screening and diagnosis of many illnesses. ECGs consist of time recordings of the electrical activities of an active heart obtained through electrodes on the human body. The ECGs' electrical signal variation describes the rhythm of the heart. As described for a healthy person, they consist of different phases.

- P wave representing atrial depolarisation.
- PR interval representing the time that takes the electrical activity to move between the atria and ventricles.
- QRS complex describing the depolarisation of the ventricles.
- *ST segment which describes the time between de- polarization and repolarization of the ventricles.*
- T wave representing ventricular repolarisation.
- U wave.

These waves may not always be seen for their reduced size.

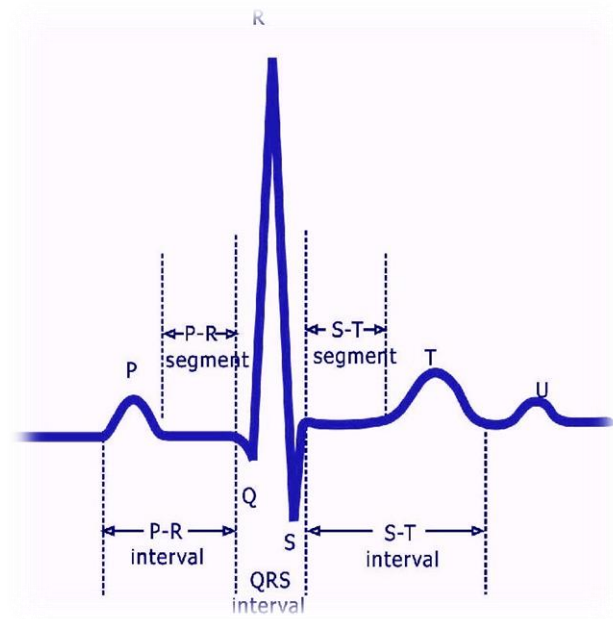


Fig: 4.2 ECG signal.

4.1.2 Pre-Processing

Preprocessing this stage includes artifact such as ECG. Removal, noise filtering, and resampling the signal to comply with detector input specifications. A low pass filter along with an artifact removal algorithm using adaptive signal processing techniques were implemented for this purpose Some of the point processing techniques include: contrast stretching, global thresholding, histogram equalization, log transformations

and power law transformations. Some mask processing techniques include averaging filters, local thresholding.

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. Enhances some image features important for further processing. This noise is caused by changes in the impedance between the electrode and the patient's skin. These changes might be caused by movement, breathing or by perspiration, leading to a change of distance between the source of the signal, the heart, and the electrodes. The ST-segment is especially affected by this kind of noise which is particularly relevant in exercise

4.1.3 Feature Extraction

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. These features are easy to process, but still able to describe the actual data set with the accuracy and originality. Discrete Cosine and Sine transforms or Discrete Fourier Transform (DFT) were used for feature extraction and dimensionally reduction of the input vector at the input of the classifier.

The discrete Fourier transform (DFT), implemented by one of the computationally efficient fast Fourier transform (FFT) algorithms, has become the core of many digital signal processing systems. These systems can perform general time domain signal processing and classical frequency domain processing. This chapter describes some signal processing applications that use the DFT to perform specific time domain filtering tasks. Two basic filtering tasks can be performed with DFT: (1) convolution or correlation between two arbitrary arrays and (2) narrowband channelization. The DFT gives access to the computational efficiency of the FFT. Some perspectives have evolved that allow classical filtering operations to be implemented by the FFT. The

chapter highlights these perspectives. It describes the way in which a judicious choice of window, overlap, block size, and circular indexing, coupled with simple pre- and Post-processing tasks, leads to use of the DFT to perform general time domain processing. It examines the way by which simple pre- and post-processing tasks significantly enhance the degrees of freedom available to the designer of multichannel processors, the way by which processing parameters are coupled, and the way by which a choice of processing parameters impacts important system considerations such as total computational burden and classical fidelity measures such as channel crosstalk and noise levels.

Discrete cosine transforms (DCTs) and discrete sine transforms (DSTs) are members of the class of sinusoidal unitary transforms. A sinusoidal unitary transform is an invertible linear transform whose kernel is defined by a set of complete, orthogonal/orthonormal discrete cosine and/or sine basis functions. The well-known Karhunen-Loève transform (KLT) generalized discrete Fourier transform (DFT), generalized discrete Hartley transform (DHT) or equivalently generalized discrete W transform, and various types of the DCT and DST are members of this class of unitary transforms. The complete set of DCTs and DSTs, so-called discrete trigonometric transforms, consists of eight versions of DCT and corresponding eight versions of DST. Each transform is classified as even or odd and of type I, II, III and IV. Since all present applications involve only even DCT and DST, this chapter considers only four types of even DCT and DST. The crucial aspect for the applicability of DCT and DST is the existence of fast algorithms that allow their efficient computation compared to the direct matrix–vector multiplication. The algorithmic history of DCT and DST is parallel to that of the DFT. Over three decades many fast algorithms for the efficient computation of one-/two-dimensional (1-D/2-D) DCT and DST have been developed. In general, they are based on indirect computation (via other discrete orthogonal transforms) or direct computation (recursive sparse matrix factorization of the transform matrix), and they are generally classified as radix-2, split-radix, mixed-radix, odd-length, composite-length and prime-

factor algorithms. However, one of the most important requirements is that fast DCT/DST algorithms possess excellent numerical stability.

This chapter discusses the fast rotation-based DCT/DST algorithms. Almost all are direct algorithms defined by the recursive (if exists) sparse matrix factorization of the transform matrix. Definitions, basic mathematical properties and relations between corresponding DCT and DST which are important in deriving fast algorithms are recalled. Then the explicit forms of orthonormal DCT and DST matrices for $N = 2, 4$ and 8 are presented in Section. The fast and numerically stable DCT/DST algorithms with regular structure employing only real arithmetic are presented in Section. In particular, these rotation-based algorithms are very convenient for the construction of integer transforms. The generalized signal flow graphs corresponding to the sparse matrix decomposition of the transform matrix are also provided. Finally, in Section how 2-D DCT/DST fast algorithms can be derived from the corresponding 1-D ones is discussed. The chapter concludes with a summary, problems and exercises, and bibliography. The discrete cosine transform (DCT) represents an image as a sum of sinusoids of varying magnitudes and frequencies. The `dct2` function computes the two-dimensional discrete cosine transform (DCT) of an image. The DCT has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT.

For this reason, the DCT is often used in image compression applications. For example, the DCT is at the heart of the international standard glossy image compression algorithm known as JPEG.

The discrete Fourier transform, or DFT, is the primary tool of digital signal processing. The foundation of the product is the fast Fourier transform (FFT), a method for computing the DFT with reduced execution time. Many of the toolbox functions (including Z-domain frequency response, spectrum and cepstrum analysis, and some filter design and implementation functions) incorporate the FFT.

The MATLAB® environment provides the functions FFT and IFFT to compute the discrete Fourier transform and its inverse, respectively. For the input sequence x and its transformed version X (the discrete-time Fourier transform at equally spaced frequencies around the unit circle), the two functions implement the relationships. Pre-processing and R peak detection is performed with the WFDB Software Package. This reads the annotation and finds the R (peak) location. R (peak) location used as a reference to detect peaks in other wave such P and T and extract ECG beat. ECG beats are extracted after windowing the signal using samples before the R and samples after the R-peak.

Discrete Cosine and Sine transforms or the Discrete Fourier Transform (DFT) were used for feature extraction and dimensionality reduction of the input vector at the input of the classifier Reconstruction after deleting coefficient below threshold. Discrete Fourier Transform (DFT) and Inverse DFT Using DCT coefficients for creating an input vector to MSVM provides better classification than DST

Like any Fourier-related transform, discrete sine transforms (DSTs) express a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. Like the discrete Fourier transform (DFT), a DST operates on a function at a finite number of discrete data points. The obvious distinction between a DST and a DFT is that the former uses only sine functions, while the latter uses both cosines and sines (in the form of complex exponentials). However, this visible difference is merely a consequence of a deeper distinction: a DST implies different boundary conditions than the DFT or other related transforms. The Fourier-related transforms that operate on a function over a finite domain, such as the DFT or DST or a Fourier series, can be thought of as implicitly defining an extension of that function outside the domain. That is, once you write a function display style $f(x)$ $f(x)$ as a sum of sinusoids, you can evaluate that sum at any display style x , even for display style x where the original display style $f(x)$ $f(x)$ was not specified. The DFT, like the Fourier series, implies a periodic extension of the original function. A DST, like a sine transform, implies an odd extension of the original function. However, because DSTs

operate on finite, discrete sequences, two issues arise that don't apply for the continuous sine transform. First, one has to specify whether the function is even or odd at both the left and right boundaries of the domain (i.e. the min-n and max-n boundaries in the definitions below, respectively). Second, one has to specify around what point the function is even or odd. In particular, consider a sequence (a,b,c) of three equally spaced data points, and say that we specify an odd left boundary. There are two sensible possibilities: either the data is odd about the point prior to a , in which case the odd extension is $(-c,-b,-a,0,a,b,c)$, or the data is odd about the point halfway between a and the previous point, in which case the odd extension is $(-c,-b,-a,a,b,c)$. These different boundary conditions strongly affect the applications of the transform, and lead to uniquely useful properties for the various DCT types. Most directly, when using Fourier-related transforms to solve partial differential equations by spectral methods, the boundary conditions are directly specified as a part of the problem being solved. The DFT and Short-Time Fourier Transform have been used to identify and extract features from ECG beat signals in the frequency domain. The ECG wave signal is transformed into the time–frequency domain using STFT. The selected coefficients form the elements of a row vector that defines the classifier input feature space. DFT coefficients were used to create the training as well as the testing datasets used for ECG classification. Initially, the DFT is applied to the ECG beat extracted using the input signal amplitude in the time domain, while the frequency domain output expression. Is used to find a row vector for DFT coefficients for each beat using the DFT matrix, whereas) is used to reconstruct the original ECG beat from fewer DFT coefficients. DFT coefficients are transformed back to the time domain using an inverse DFT.

4.1.4 Classification

The MSVM and CSVM classifier is used to distinguish the four ECG arrhythmias types using the DCT, DST and DFT coefficients as feature vectors at the input vector of the classifier. A multioutput support vector machine is used to map the complex, nonlinear relationship between the in situ stress, elastic parameters, and borehole pressure. Sequential minimal optimization approach is used to train the CSVM and compute the corresponding complex hyper-plane parameters. MATLAB software routines were used to train and validate both the CSVM and the Multi-class Support Vector Machine (MSVM) classifier. A Complex kernel function with 5-fold cross validation was used for adjusting the kernel values. Sequential minimal optimization (SMO) was used to train the CSVM and compute the corresponding complex hyper-plane parameters. The aim of the study was to improve multi-class Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors. The main function of the SVM is to check for that hyperplane that is able to distinguish between the two classes SVM by extending traditional SVM algorithms to complex spaces so as to simultaneously classify four types of heartbeats. Results illustrate that the proposed beat classifier is very reliable, and that it may be adopted for automatic detection of arrhythmia conditions and classification. Accuracies are obtained for MSVM and CSVM classification respectively. Using CSVM, a 4 class's problem can be classified rapidly by decomposing it into two distinct SVM tasks. Moreover, the present research confirmed that the use of selected number of Fourier coefficients to approximate the ECG beat signal and compress the input features to the classifier can lead to high classification accuracies and improve the generalization ability of the CSVM classifier.

4.1.4.1 CSVM

The purpose of a traditional binary SVM classification task is to separate two different data classes using a maximum margin hyper-plane. In other words, elements belonging to the first class satisfy as a truth statement, whereas elements belonging to the second class satisfy where the parameters that control the function are and the decision rule is given from: f, w . The adopted formulation for the current study follows the general ideas discussed which generalized the SVM formulation to complex spaces. The approach defines a complex space separated into four parts by using a pair of complex hyperplanes. For clarity a notation where integers, real, and complex numbers are denoted by N, R and c respectively is adopted and denotes the complex conjugate of z . A complex reproducing Kernel Hilbert Space will be denoted by $\mathbb{H}$ while a real RKHS by $\mathcal{H}$. The complex hyperplane is composed of two orthogonal hyperplanes that will be referred to as real and imaginary hyperplanes

The ECG arrhythmia classification scheme is composed of pre-processing and beat extraction, and feature extraction, and classification as shown in First, the baseline offset needs to be removed in order to scale better the ECG signal and perform de noising so individual beats can be extracted. Second, feature extraction is performed using DFT and the DFT coefficients are used to form feature vectors, which are subsequently used as input to the CSVM classifier.

The ECG features can be extracted either in the time the frequency domain. There are several dimensionality reduction and feature extraction algorithms that may be applied successfully to ECG signals such as Wavelet transforms (WT) and Fourier transforms. The features of the P, QRS complex, and T waves are associated with the location, duration, amplitudes, and shapes of the waves. In this study, the ECG features from each beat were extracted in the frequency domain and the complex components of the transform were also used as complex datasets for CSVM. As mentioned earlier, DFT is used for feature extraction from ECG signals. Selected

Fourier coefficients were subsequently used as input to a classifier; this approach has been extensively used in a large number of biomedical applications. In the associated DFT, the complex values in the signals were retained as vectors. The DFT complex values were used as the input feature vectors for the training and testing of the CSVM. Fifty Fourier coefficient were selected to reconstruct each beat and create input features for classification. The dimension of each ECG beat sample was reduced from 301 to 50 features, which was found as optimal number of the Fourier coefficient (avoiding over-filtering and distortion or under-filtering and noise propagation that would compromise the performance of the classifier

4.1.4.2 MSVM

We can find out that the final decision function can be considered as the combination of two parts, the support - 4 - vectors and the class-median vectors. This fact leads to a simplified implementation of MSVM, which can be obtained through the modification of the standard SVM result. That is, after getting the weight of a standard SVM classifier, we can adjust the hyperplane by adding the difference vector between the two class-medians with certain factor, new SVM 1 where SVM w is the weight obtained by standard SVM and is a constant controlling the balance between the classification information contained in the SVs and in the class medians. By taking this short cut, previous algorithms for SVM can be utilized. In practice, λ can be selected according the a priori estimation about our beliefs in the specific samples and in the class medians. For example, if there is severe noise in sample set, a larger λ can be chosen to make the class medians play a more important role. If $\lambda = 0$, the standard SVM can be obtained. In practice, since SVM w is scaled in standard SVM, may need scaling to be comparable with SVM. When kernel is adopted for the nonlinear transform, we can use the image of the median vectors in attribute space as the “median” in feature space such that, this is our simplified implementation of kernel version of MSVM.

In order to evaluate the performance of MSVM and compare MSVM with SVM and CSVM, we design two toy data sets. Experiment results and analysis are presented

as follows. The first example is to compare the performance of MSV with that of CSVM and SVM under the existence of outliers not far from the class center. In there are two outliers in the training set and the SVM separation hyperplane is obvious not the “optimal” one. Shows the optimal SVM result when the training set is noiseless and the separation hyperplane changes a lot from the one in is the result obtained by CSVM, using its simplified implementation. Illustrates the result obtained by MSVM, using its simplified implementation, with the same. We can find that although the outliers exist in the training set, the optimal result of MSVM is almost the same as the one obtained when the training set is clean. Note that here we did not attempt to identify and remove the outlier, but simply made the result less sensitive to outliers by taking the class-median information into account. It can also be seen that when the outlier is not far from class-centers, MSVM and CSVM achieve similar performance. A comparison experiment of SVM, CSVM and MSVM. Pluses and circles denote the training samples of two classes. The standard SVM result at the existence of outliers; the standard SVM result excluding the outliers, the hyperplanes of CSVM and MSVM when the outliers exist, respectively. A comparison experiment between CSVM and MSVM. Pluses and circles denote the training samples of two classes. Are the results of CSVM and MSVM when the outliers are far from class-centers, respectively shows the experiment results of the comparison between CSVM and MSVM under the existence of outliers far from the class-center shows the result obtained by CSVM and outlier the result by MSVM It can be seen that when the outliers are far from class center, they change the class-center greatly so that the results obtained by CSVM is far from the really “optimal” one. But the class median is less sensitive to such outliers and therefore the result is more reliable. So it can be concluded that MSVM is a more robust algorithm.

4.1.4.3 WAVELET TRANSFORM

The wavelet transform is a fine-grained approach that seeks to achieve an optimal balance between frequency resolution and time resolution. At higher frequencies, the transform gains temporal information in exchange for a loss in frequency information, while at lower frequencies, the transform gains frequency information in exchange for a loss in temporal information. This fine-grained approach in handling the tradeoff is useful for digital signal and music applications, since transients normally occur at high frequencies (thus needing a higher time resolution), and lower frequencies usually require a higher frequency resolution. Like the Fourier transform, the wavelet transform can be represented as an integral: $\int_{-\infty}^{\infty} W(x, u, s) = \int_{-\infty}^{\infty} x(t) \Psi\left(\frac{t-u}{s}\right) dt$ (), In the integral above, the input signal $x(t)$ is correlated with the wavelet with translation parameter u and dilation parameter s . This transform converts a signal into coefficients that represent both time and frequency information, with more time resolution at high frequencies, and more frequency resolution at low frequencies. The dilation of the wavelet enables the fine-grained tradeoff to occur. As with the Fourier transform, there are fast discrete ways of computing the wavelet transform. One fast way of computing a wavelet transform is with a cascade of filters. The input signal is fed into two filters, H and G . The filters produce two sets of coefficients which are both down-sampled by a factor of 2. This procedure is recursively applied to the set of coefficient that comes out of the H filter. One assumption of the wavelet transform is that the number of samples in the input signal is a power of 2. If the number of samples is not a power of 2, the signal can be zero padded to achieve this criterion.

4.1.4.4 R-Wave

The detection of R-waves in ECG signals presents many challenges, such as EMG interference, power-frequency interference, and baseline drift, which can affect signal primitiveness. These noises are the difficulties in the automatic detection of ECG signals. Due to the large body swing and electrode friction caused by movement, the noise is particularly prominent in the motion ECG signals. This paper proposes an adaptive MMT difference algorithm to detect R-wave, which includes the following

steps: the whole algorithm process of preprocessing baseline correction for R-wave detection. Firstly, in order to eliminate the noise embedded in the ECG signal and enhance the ECG signal, digital filters with appropriate parameters and adaptive filters are usually used to eliminate the noise. In this paper, an FIR filter and Notch filter are used to eliminate EMG interference and power-frequency interference in the ECG signal. This process has a large amount of ECG processing capacity and a lot of detail processing, but it does not affect the detection of R-wave. In addition, the calculation cost of the filtering process is lower and the space occupancy is less. Secondly, a baseline drift can greatly interfere with overall R-wave detection, for which a sliding window is used to overcome the baseline drift. The ECG baseline was extracted from the ECG through the sliding window, and the difference between the original signal and the ECG baseline was calculated to obtain the ECG signal after the baseline was more positive. Finally, the amplitude of the R-wave varies from person to person, and the amplitude of the R-wave varies greatly from person to person at different times. Therefore, it is very important to find the appropriate threshold. In this paper, through adaptive multi threshold to cope with different scenarios, different populations, and different collection patterns, the R-wave is accurately calculated.

4.2 SOFTWARES USED

4.2.1 MATLAB

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and FORTRAN.

Mat lab is a data analysis and visualization tool which has been designed with powerful support for matrices and matrix operations. As well as this, Mat lab has excellent graphics capabilities, and its own powerful programming language. One of the reasons that Mat lab has become such an important tool is through the use of sets of Mat lab programs designed to support a particular task. These sets of programs are called toolboxes, and the particular toolbox of interest to us is the image processing toolbox. Rather than give a description of all of Mat lab's capabilities, we shall restrict ourselves to just those aspects concerned with handling of images. We shall introduce functions, commands and techniques as required. A Mat lab function is a keyword which accepts various parameters, and produces some sort of output: for example a matrix, a string, a graph. Examples of such functions are sin, imread, imclose. There are many functions in Mat lab, and as we shall see, it is very easy (and sometimes necessary) to write our own.

Mat lab's standard data type is the matrix _all data are considered to be matrices of some sort. Images, of course, are matrices whose elements are the grey values (or possibly the RGB values) of its pixels. Single values are considered by Mat lab to be matrices, while a string is merely a matrix of characters; being the string's length. In this chapter we will look at the more generic Mat lab commands, and discuss images in further chapters.

When you start up MATLAB, you have a blank window called the Command

Window_ in which you enter commands. Given the vast number of Mat lab's functions, and the different parameters they can take, a command line style interface is in fact much more efficient than a complex sequence of pull-down menus.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement financial modeling and analysis. Add-on toolboxes (collections of special-purpose MATLAB functions) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

When working with images in Mat lab, there are many things to keep in mind such as loading an image, using the right format, saving the data as different data types, how to display an image, conversion between different image formats.

Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image de blurring, feature detection, noise reduction, image segmentation, spatial transformations, and image registration. Many functions in the toolbox are multithreaded to take advantage of multicore and multiprocessor computers.

4.2.2 Signal Processing

Signal processing engineers use MATLAB® and Simulink® at all stages of development—from analyzing signals and exploring algorithms to evaluating design implementation tradeoffs for building real-time signal processing systems. MATLAB and Simulink offer:

Built-in functions and apps for analysis and preprocessing of time-series data,

spectral and time-frequency analysis, and signal measurements

Apps and algorithms to design, analyze, and implement digital filters (FIR and IIR) from basic FIR and IIR filters to adaptive, multirate, and multistage designs

An environment to model and simulate signal processing systems with a combination of programs and block diagrams

Capabilities to model fixed-point behavior and automatically generate C/C++ or HDL code for deploying on embedded processors, FPGAs, and ASICs

Tools for developing predictive models on signals and sensor data using machine learning and deep learning workflows. Tracking the DNA of Sound with MATLAB.

Tracking the DNA of Sound with MATLAB Hear how u be unlocks the potential of voice-based interfaces with MATLAB, signal processing, artificial intelligence, and support from the Math Works Startup Program.

4.2.2.1 Signal Analysis and Measurements

MATLAB and Simulink help you analyze signals using built-in apps for visualizing and preprocessing signals in time, frequency, and time-frequency domains to detect patterns and trends without having to manually write code. You can characterize signals and signal processing systems using domain-specific algorithms across different applications such as communications, radar, audio, medical devices, and IoT Model-Based Design for Signal Processing

When designing signal processing systems, you can use a combination of block diagrams and language-based programming. You can use Simulink to apply Model-Based Design to signal processing systems for modeling, simulation, early verification, and code generation. You can use libraries of blocks with application-specific algorithms for baseline signal processing, audio, analog mixed-signal and RF, wire line and wireless communications, and radar systems. You can visualize live signals during simulations using virtual scopes, including spectrum and logic analyzers, constellations, and eye diagrams

4.2.3 Filter Design and Analysis

Design, analyze, and implement digital and analog filters. Use the Filter Designer app or the Design Filter live editor task to design a variety of digital FIR and IIR filters, such as low pass, high pass, and band stop.

4.2.4 De noise Signals

To de noise one or more selected signals, on the Analyzer tab, expand the Preprocessing gallery and click the De noise icon. Signal Analyzer uses the de noise (Wavelet Toolbox) function to perform signal de noising with these parameters:

- Wavelet family
- De noising method
- Thresholding rule

For an example, see De noise Noisy Doppler Signal. You must have a Wavelet Toolbox™ license to de noise signals in the app.

Feature Extraction and Signal Measurements Measure and extract distinctive features in signals including peaks, power, bandwidth, distortion, and signal statistics. Compute metrics related to pulses and transitions. Extract features for an entire dataset using the Signal Labeler app.

4.2.5 Simulink

Simulink is a MATLAB-based graphical programming environment for modeling, simulating and analyzing multi domain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in automatic control and

digital signal processing for multi domain simulation and model-based design.

Math Works and other third-party hardware and software products can be used with Simulink. For example, State flow extends Simulink with a design environment for developing state machines and flow charts. Math Works claims that, coupled with another of their products, Simulink can automatically generate C source code for real-time implementation of systems. As the efficiency and flexibility of the code improves, this is becoming more widely adopted for production systems,[5][6] in addition to being a tool for embedded system design work because of its flexibility and capacity for quick iteration[citation needed]. Embedded Coder creates code efficient enough for use in embedded systems.

Simulink Real-Time (formerly known as x PC Target), together with x86-based real-time systems, is an environment for simulating and testing Simulink and State flow models in real-time on the physical system. Another Math Works product also supports specific embedded targets. When used with other generic products, Simulink and State flow can automatically generate synthesizable VHDL and Verilog [citation needed].

Simulink Verification and Validation enables systematic verification and validation of models through modeling style checking, requirements traceability and model coverage analysis. Simulink Design Verifier uses formal methods to identify design errors like integer overflow, division by zero and dead logic, and generates test case scenarios for model checking within the Simulink environment.

Sim Events is used to add a library of graphical building blocks for modeling queuing systems to the Simulink environment, and to add an event-based simulation engine to the time-based simulation engine in Simulink.

4.2.5.1 Simscape

Simscape™ enables you to rapidly create models of physical systems within the Simulink® environment. With Simscape, you build physical component models based on physical connections that directly integrate with block diagrams and other modeling paradigms. You model systems such as electric motors, bridge rectifiers, hydraulic actuators, and refrigeration systems, by assembling fundamental components into a schematic. Simscape add-on products provide more complex components and analysis capabilities.

Simscape helps you develop control systems and test system-level performance. You can create custom component models using the MATLAB® based Simscape language, which enables text-based authoring of physical modeling components, domains, and libraries. You can parameterize your models using MATLAB variables and expressions, and design control systems for your physical system in Simulink. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape supports C-code generation.

4.2.5.2 Simscape Multibody

Simscape Multibody (formerly SimMechanics) provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear. You can model multibody systems using blocks representing bodies, joints, constraints, force elements, and sensors. Simscape Multibody formulates and solves the equations of motion for the complete mechanical system. You can import complete CAD assemblies, including all masses, inertias, joints, constraints, and 3D geometry, into your model. An automatically generated 3D animation lets you visualize the system dynamics.

Simscape Multibody helps you develop control systems and test system-level performance. You can parameterize your models using MATLAB® variables and

expressions, and design control systems for your multibody system in Simulink®. You can integrate hydraulic, electrical, pneumatic, and other physical systems into your model using components from the Simscape™ family of products. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape Multibody supports C-code generation.

4.2.6 Database

The Physionet database includes a variety of digitized recording for ECG signals divided by class, patient, acquisition date and recording purposes. The data is available in a binary format. Each recording includes a header file (.hea) containing detailed clinical information about the subject, a binary file (.dat) holding the ECG signal, and sometimes annotation files with different suffixes (.ari,.atr ...). They are stored in Waveform Database (WFDB) standard format. For reading, writing and processing purposes, Physionet provides the WFDB software package. This package includes tools highly useful for reading, writing, and analyzing the files they store. Also, they offer a jointed toolbox (WFDB Matlab toolbox) for Matlab programming and another package for python programming (WFDB Python package). For this project, the Python package¹ is used for the manipulation of the raw data and the arrangement of the dataset intended to be fed to the Neural Network.

4.2.7 Peak Position

When analysing heart rate, the main crux lies in the accuracy of the peak position labeling being used. When extracting instantaneous heart rate (BPM), accurate peak placement is not crucial. The BPM is an aggregate measure, which is calculated as the average beat-beat interval across the entire analysed signal (segment). This makes it quite robust to outliers.

However, when extracting heart rate variability (HRV) measures, the peak positions are crucial. Take as an example two often used variability measures, the RMSSD (root mean square of successive differences) and the SDSD (standard deviation of successive differences). Given a segment of heart rate data as displayed in the figure below, the RMSSD is calculated as shown. The SDSD is the standard deviation between successive differences.

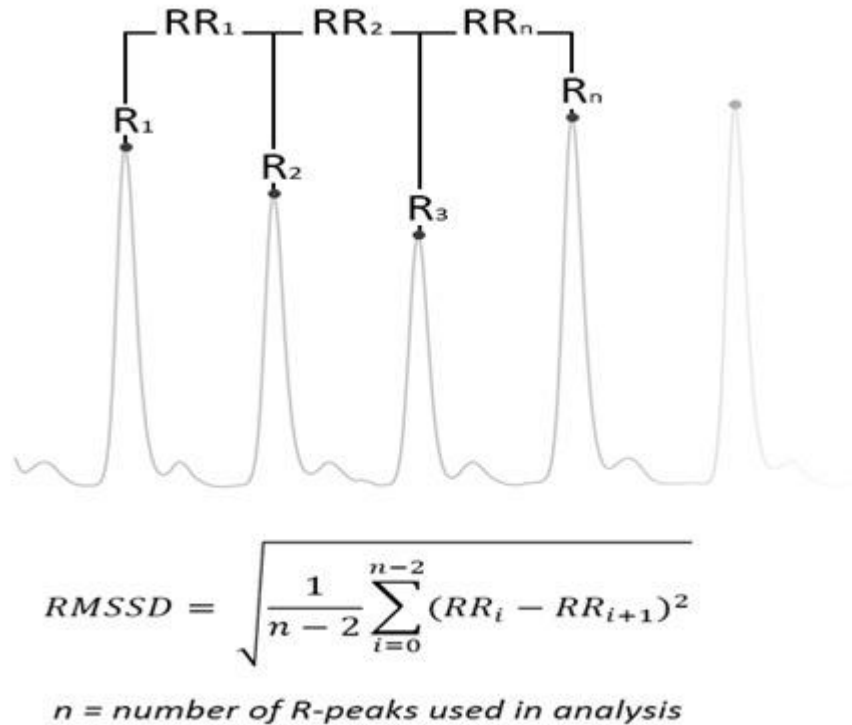


Fig: 4.3 peck detection

Image displaying the desired peak detection result, as well as the calculation of the RMSSD measure. The SDSD measure is the standard deviation between successive differences.

Now consider that two mistakes are possible: either a beat is not detected at all (missed), or a beat is placed at an incorrect time position (incorrectly placed). These

will have an effect on the calculated HRV output measures, which are highly sensitive to outliers as they are designed to capture the slight natural variation between peak-peak intervals in the heart rate signal. To illustrate the problem we have run a few simulations. We took a sample of a heart rate signal which was annotated manually, and introduced two types of errors. We randomly dropped $n\%$ of peaks from the signal, then re-ran the analysis considering only intervals between two peaks where no missing value occurred in between. We introduced a random position error ($0.1\% - 10\%$ of peak position, meaning between about 1ms and 100ms deviation) in $n\%$ of peaks.

4.2.8 Curve Fitting Toolbox

The Curve Fitting Toolbox is a collection of graphical user interfaces (GUIs) and M-file functions built on the MATLAB® technical computing environment. The toolbox provides you with these main features: Data preprocessing such as sectioning and smoothing. Parametric and nonparametric data fitting: You can perform a parametric fit using a toolbox library equation or using a custom equation. Library equations include polynomials, exponentials, rational, sums of Gaussians, and so on. Custom equations are equations that you define to suit your specific curve fitting needs. - You can perform a nonparametric fit using a smoothing spline or various interpolants. Standard linear least squares, nonlinear least squares, weighted least squares, constrained least squares, and robust fitting procedures Fit statistics to assist you in determining the goodness of fit Analysis capabilities such as extrapolation, differentiation, and integration A graphical environment that allows you to: - Explore and analyze data sets and fits visually and numerically - Save your work in various formats including M-files, binary files, and workspace variables The Curve Fitting Toolbox consists of two different environments: • The Curve Fitting Tool, which is a graphical user interface (GUI) environment The MATLAB command line environment You can explore the Curve Fitting Tool by typing `c tool` Click the GUI Help buttons to learn how to proceed. Additionally, you can follow the examples in the tutorial

sections of this guide, which are all GUI oriented. What Is the Curve Fitting Toolbox?
1-3 To explore the command line environment, you can list the toolbox functions by typing `help curvefit` To view the code for any function, type `type function_name` To view the help for any function, type `help function_name` You can change the way any toolbox function works by copying and renaming the M-file, and then modifying your copy. However, these changes will not be reflected in the graphical environment. You can also extend the toolbox by adding your own M-files, or by using it in combination with other products such as the Statistics Toolbox or the Optimization Toolbox.

4.2.8.1 Opening the Curve Fitting Tool

The Curve Fitting Tool is a graphical user interface (GUI) that allows you to

- Visually explore one or more data sets and fits as scatter plots.
- Graphically evaluate the goodness of fit using residuals and prediction bounds.
- Access additional interfaces for
 - Importing, viewing, and smoothing data
 - Fitting data, and comparing fits and data sets
 - Marking data points to be excluded from a fit
 - Selecting which fits and data sets are displayed in the tool
 - Interpolating, extrapolating, differentiating, or integrating fits

You open the Curve Fitting Tool with the `cftool` command.

4.2.8.2 Importing the Data

Before you can import data into the Curve Fitting Tool, the data variables must exist in the MATLAB workspace. For this example, the data is stored in the file `census.mat`, which is provided with MATLAB. Load `census`. The workspace now contains two new variables, `cdate` and `pop`:

- `Cdate` is a column vector containing the years 1790 to 1990 in 10-year increments.
- `Pop` is a column vector with the US population figures that correspond to the years

in cdate.

You can import data into the Curve Fitting Tool with the Data GUI. You open this GUI by clicking the Data button on the Curve Fitting Tool. As shown below, the Data GUI consists of two panes: Data sets and Smooth. The Data Sets pane allows you to

- Import predictor (X) data, response (Y) data, and weights. If you do not import weights, then they are assumed to be 1 for all data points.
- Specify the name of the data set.
- Preview the data.

To load cdate and pop into the Curve Fitting Tool, select the appropriate variable names from the X Data and Y Data lists. The data is then displayed in the Preview window.

4.2.8.3 Fitting the Data

You fit data with the Fitting GUI. You open this GUI by clicking the Fitting button on the Curve Fitting Tool. The Fitting GUI consists of two parts: the Fit Editor and the Table of Fits. The Fit Editor allows you to

- Specify the fit name, the current data set, and the exclusion rule.
- Explore various fits to the current data set using a library or custom equation, a smoothing spline, or an interpolant.
- Override the default fit options such as the coefficient starting values.
- Compare fit results including the fitted coefficients and goodness of fit statistics. The Table of Fits allows you to
- Keep track of all the fits and their data sets for the current session.
- Display a summary of the fit results.
- Save or delete the fit results.

4.2.8.4 The Data Fitting Procedure

For this example, begin by fitting the census data with a second degree polynomial. Then continue fitting the data using polynomial equations up to sixth degree, and a single-term exponential equation. The data fitting procedure follows these general steps:

- From the Fit Editor, click New Fit. Note that this action always defaults to a linear polynomial fit type. You use New Fit at the beginning of your curve fitting session, and when you are exploring different fit types for a given data set.
- Because the initial fit uses a second degree polynomial, select quadratic polynomial from the Polynomial list. Name the fit poly2.
- Click the Apply button or select the Immediate apply check box. The library model, fitted coefficients, and goodness of fit statistics are displayed in the Results area
- Fit the additional library equations. For fits of a given type (for example, polynomials), you should use Copy Fit instead of New Fit because copying a fit retains the current fit type state thereby requiring fewer steps than creating a new fit each time.

CHAPTER-5

RESULT AND DUSSION

5.1 RESULT

Signal processing was carried out using MATLAB and the Wave Form Database software packages. The ECG beats were obtain from the ST-T Database with samples around the R-peak. In feature selection Fourier coefficients were selected for reconstructing individual ECG beats, these form the input vector to the classifier for the classifier implementation, after feature selection is performed, the datasets are divided into two groups with beats for training and testing purposes. The optimal hyper-plane parameters are identified on the basis of the training data set, class label. The output of the SMO algorithm was used for training. Finally, on the basis of the results, test data are imported to the CSVM and MSVM classifiers to perform unknown beat classification

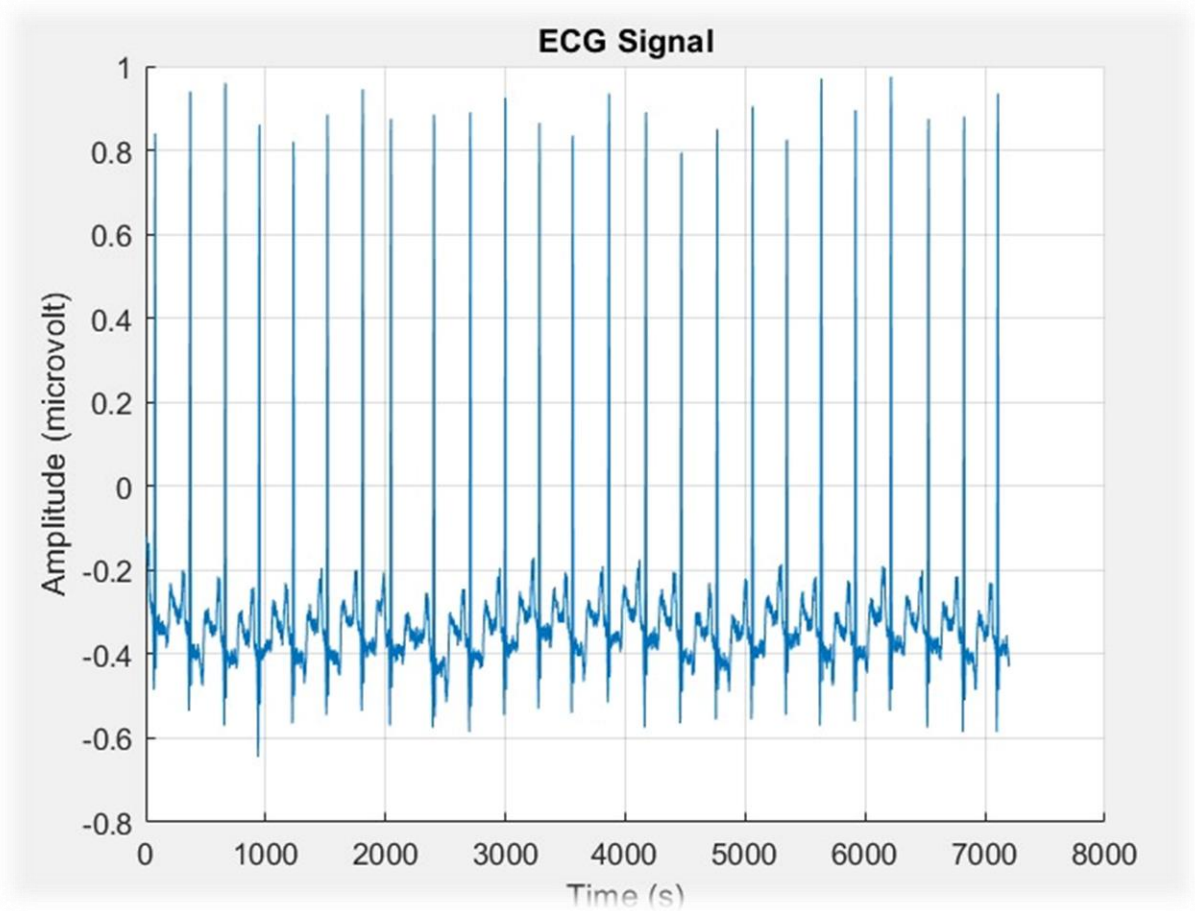


Fig: 5.1 Input ECG Signal 1

The above figure is the raw ECG signal it contain both high and low frequency noise components.

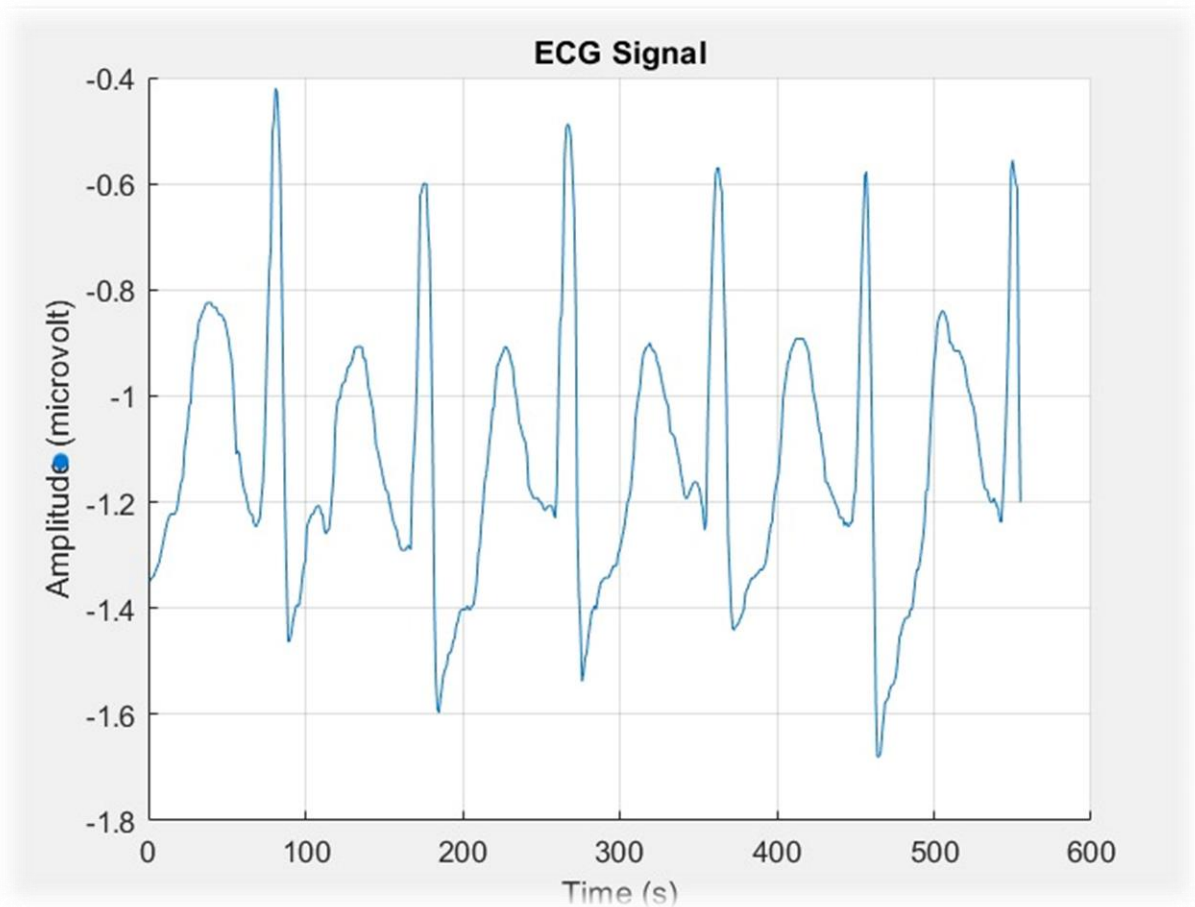


Fig: 5.2 Input ECG Signal 2

The above figure is the raw ECG signal it contain both high and low frequency noise components

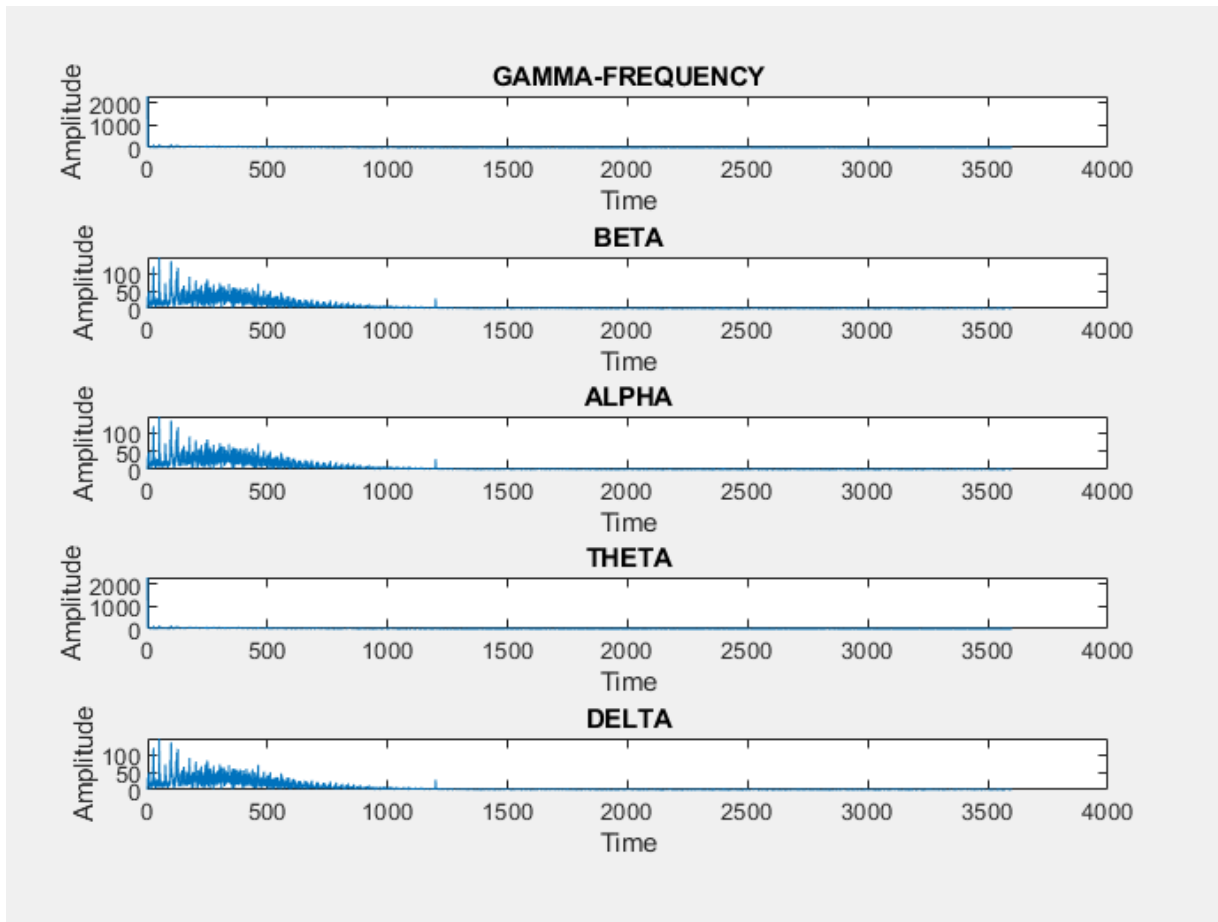


Fig: 5.3 Pre- processing Signal 1

The above figure is pre-processing signal here removal of noise and baseline correction.

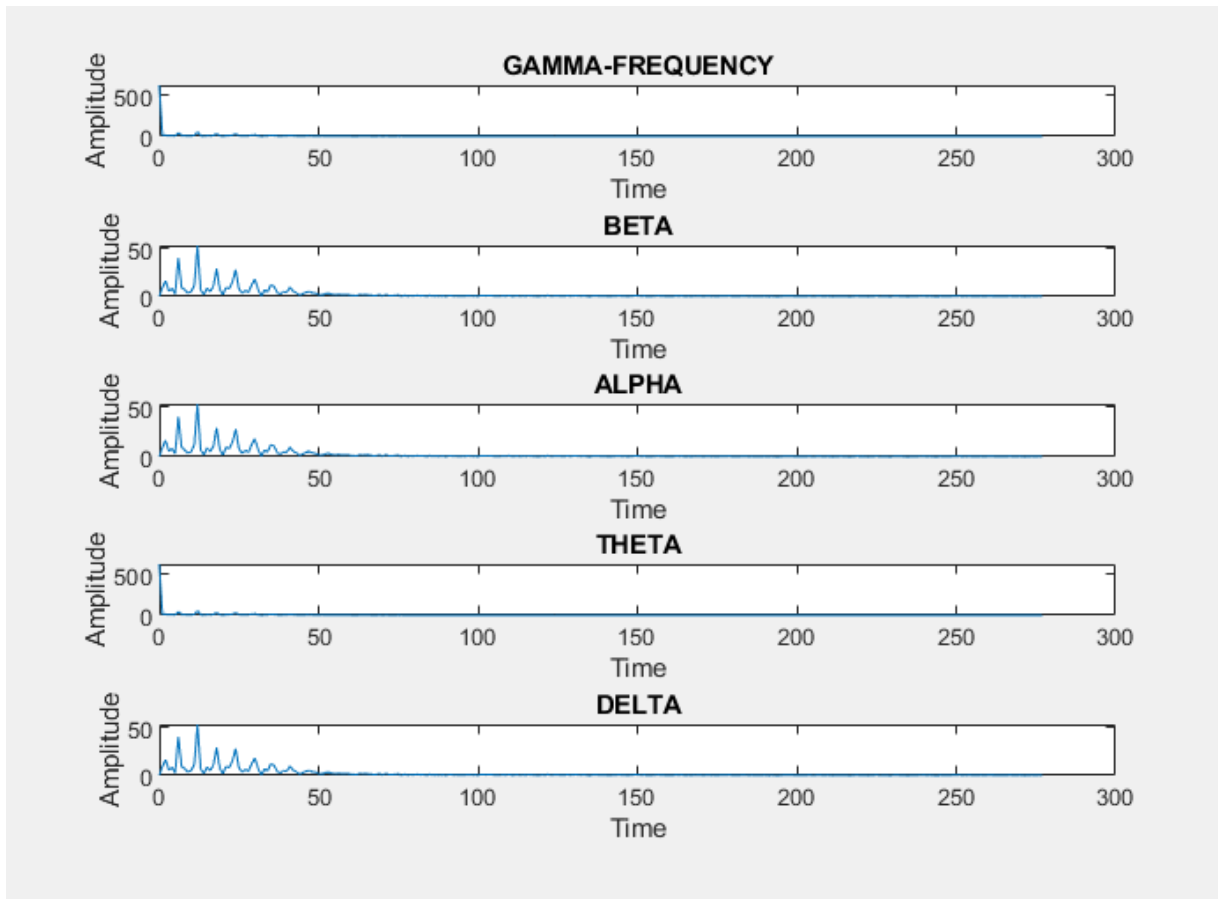


Fig: 5.4 Pre -processing Signal 2

The above figure is pre-processing signal here removal of noise and baseline correction.

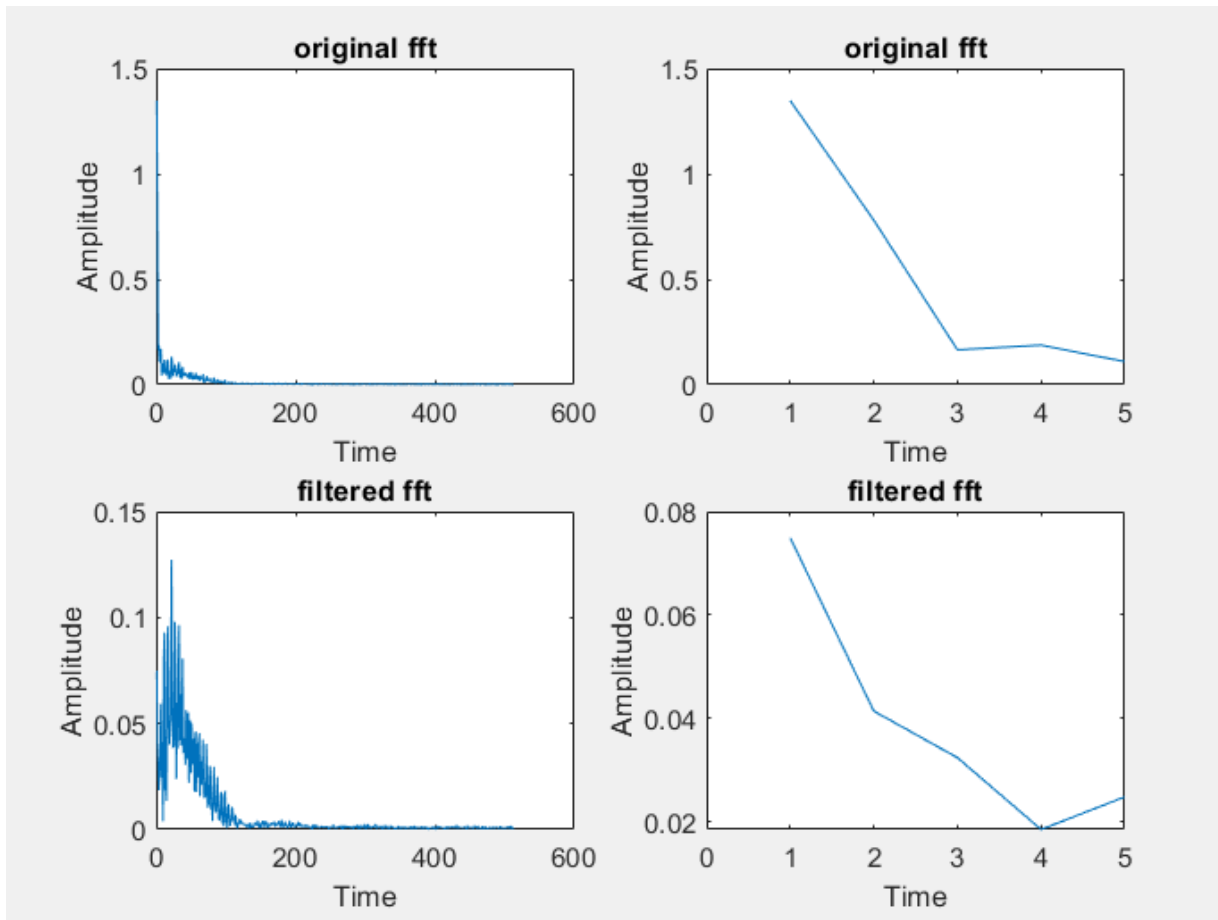


Fig: 5.5 Feature Extraction Signal 1

The above figure is showing converting original signal to filtered signal.

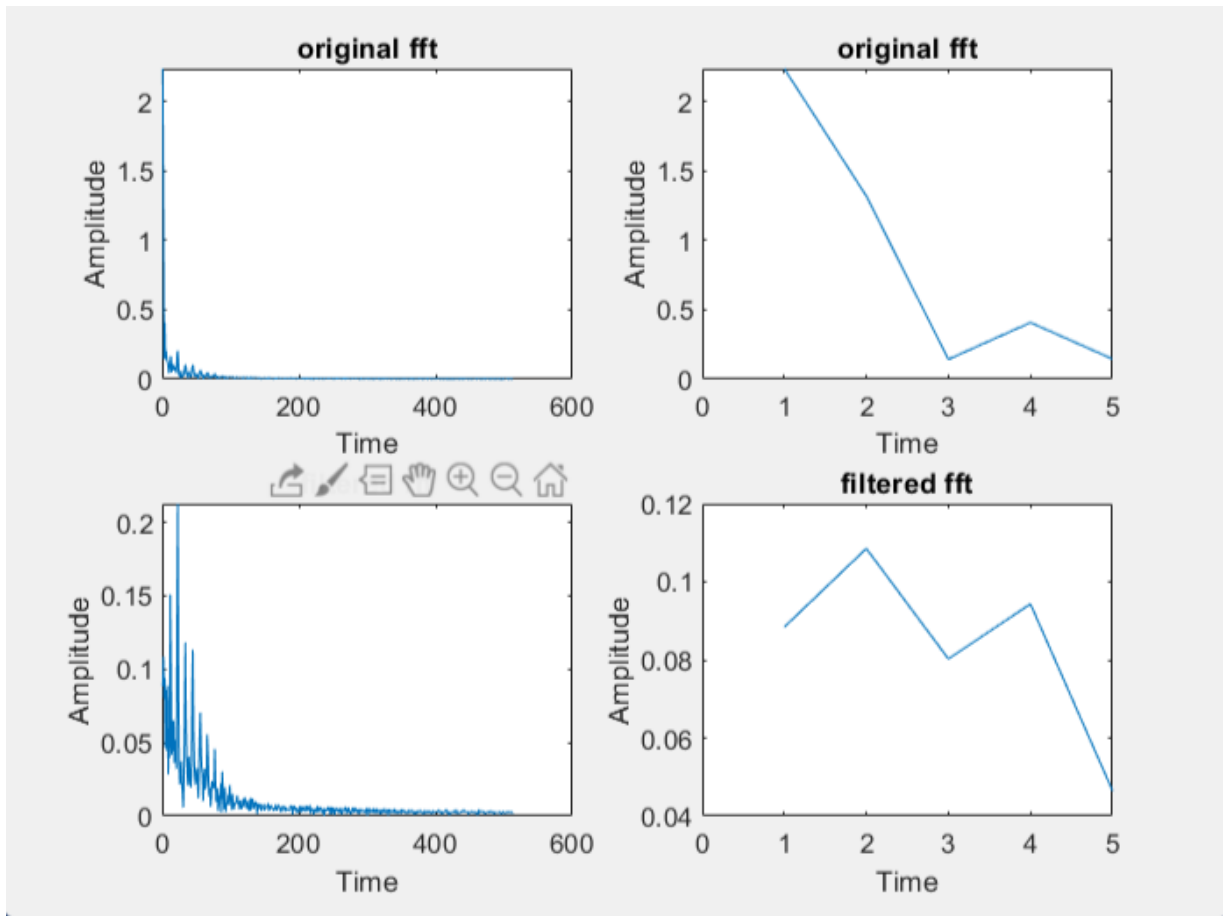


Fig: 5.6 Feature Extraction Signal 2

The above figure is showing converting original signal to filtered signal.

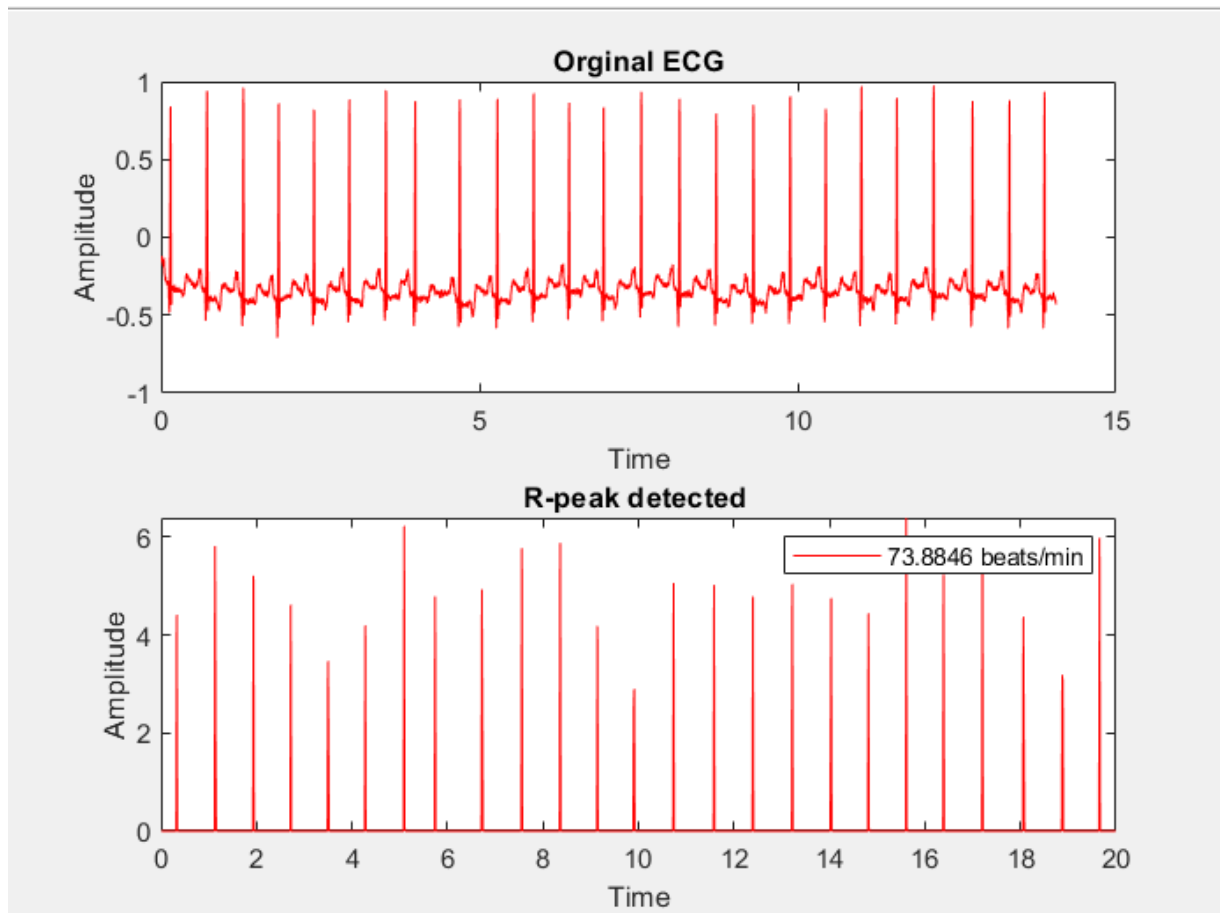


Fig: 5.7 Peak detected 1

The above figure is detecting the peck of ECG signal.

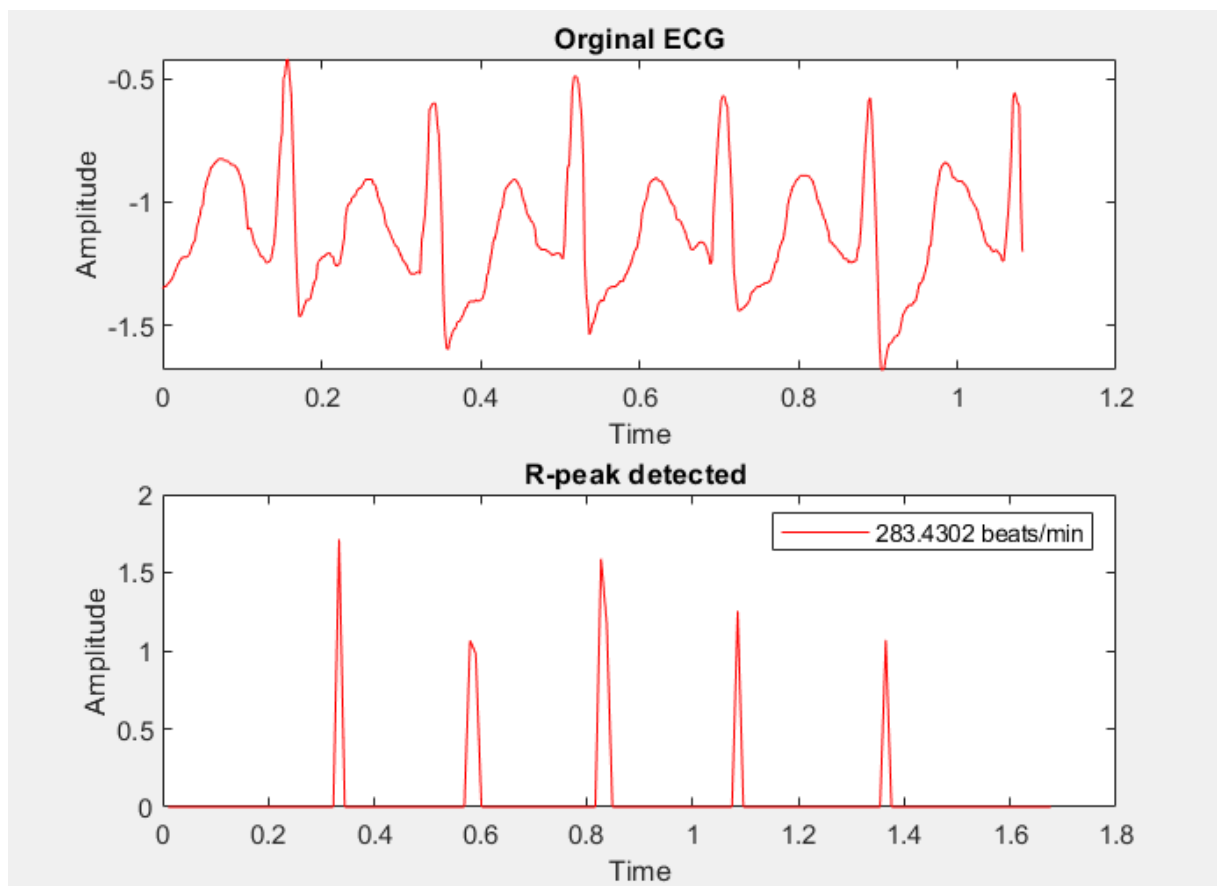


Fig: 5.8 peak detected 2

The above figure is detecting the peck of ECG signal.

```
Command Window
New to MATLAB? See resources for Getting Started.

0.0006 + 0.0009i
-0.0008 - 0.0009i
-0.0025 + 0.0029i
-0.0013 - 0.0004i
0.0031 + 0.0067i
0.0075 + 0.0030i
0.0018 + 0.0024i
-0.0042 + 0.0020i
0.0012 + 0.0021i
-0.0035 + 0.0012i
-0.0005 + 0.0079i
0.0028 + 0.0048i
0.0088 + 0.0046i
0.0106 + 0.0004i
-0.0297 + 0.0000i

Delta:Maximum occurs at 49.000000 Hz.
sampling rate is 360.05
recording_time=19.9972sec
Normal
beat rate for input signal is detected 73 beats per minute
fx >> |
```

Fig: 5.9 Result 1

The above figure is final output of the project it shows the ECG signal is normal or abnormal.

```
Command Window
New to MATLAB? See resources for Getting Started.

-0.2590 - 0.1200i
-0.1494 + 0.1560i
-0.0861 + 0.0879i
-0.1983 + 0.0341i
-0.3520 + 0.2820i
 0.0407 + 0.1399i
-0.1831 - 0.2269i
-0.3705 + 0.1093i
-0.0515 - 0.0705i
-0.1274 - 0.0867i
-0.2739 - 0.1278i
-0.1355 + 0.1062i
-0.2509 - 0.0212i
-0.3898 + 0.1330i
-0.0355 - 0.0720i

Delta:Maximum occurs at 12.000000 Hz.
sampling rate is 330.9481
recording_time=1.677sec
Abnormal
beat rate for input signal is detected 283 beats per minute
fx >> |
```

Fig: 5.10 Result 2

The above figure is final output of the project it shows the ECG signal is normal or abnormal.

CHAPTER-6

SUMMARY AND CONCLUSIONS

6.1 SUMMARY

Using CSVM and MSVM, a 4 class's problem can be classified rapidly by decomposing it into two distinct SVM tasks. The present research confirmed that the use of selected number of Fourier coefficients to approximate the ECG beat signal and further compress the input features to the classifier can lead to high classification accuracies and improve the generalization ability of the CSVM and MSVM classifier. Using the CSVM and MSVM algorithm indicated a significant improvement in overall classification.

6.2 CONCLUSIONS

Using CSVM and MSVM, a 4 class's problem can be classified rapidly by decomposing it into two distinct SVM tasks. Moreover, the present research confirmed that the use of selected number of Fourier coefficients to approximate the ECG beat signal and compress the input features to the classifier can lead to high classification accuracies and improve the generalization ability of the CSVM classifier. Future work on wavelet pre-processing to further compress the input space of the classifier by generating wavelets on the basis of higher order moment criteria as well as alternative approaches for extending the CSVM AND MSVM input and output spaces to arbitrary dimension using Clifford algebra will be discussed at the conference . The heart beat detection is normal or abnormal.

REFERENCES

- 1) Bouboulis P, Theodoridis S, Mavroforakis C, et al. "Complex Support Vector Machines for Regression and Quaternary Classification". *IEEE Transactions on Neural Networks and Learning Systems* 2015; 26(6): 1260–1274.
- 2) Froese T, Hadjiloucas S, Galvao RKH, et al. "Comparison of extra systolic ECG signal classifiers using discrete wavelet transforms". *Pattern Recognition Letters* 2006; 27(5): 393–407.
- 3) Gacke A. "Preprocessing and analysis of ECG signals - A self-organizing maps approach". *Expert Systems with Applications* 2011; 38(7): 9008–9013.
- 4) Goldberger AL, Amaral LAN, Glass L, et al. "Physio Bank, Physio Toolkit, and Physio Net: Components of a New Research Resource for Complex Physiologic Signals". *Circulation* 2000; 101(23): e215-e220.
- 5) Jannah N, Hadjiloucas S. "A Comparison between ECG Beat Classifiers Using Multiclass SVM and SIMCA with Time Domain PCA Feature Reduction". In: *Proceedings - 2017 UKSim-AMSS 19th International Conference on Modelling and Simulation, UKSim 2017*. Cambridge, 2017, pp. 126–131.
- 6) Kanani P, Padole M. ECG Heartbeat Arrhythmia Classification Using Time-Series Augmented Signals and Deep Learning Approach. *Procedia Computer Science* 2020; 171: 524–531.
- 7) Latif G, Al Anezi FY, Zikria M, et al. "EEG-ECG Signals Classification for Arrhythmia Detection using Decision Trees". In: *Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC 2020)*. Coimbatore, 2020, pp. 192–196.
- 8) Monasterio V, Laguna P, and Martinez JP. "Multilead estimation of T-wave alterations in the ECG using principal component analysis". In: *16th European Signal Processing Conference (EUSIPCO 2008)*. 2008, pp. 1–5.

- 9) Rahhal MM Al, Bazi Y, Alhichri H, et al. “ Deep learning approach for active classification of electrocardiogram signals”. Information Sciences 2016; 345: 340–354.
- 10) Übeyli ED. “Combining recurrent neural networks with eigenvector methods for classification of ECG beats”. Digital Signal Processing 2009; 19(2): 320–329.
- 11) Wang J-S, Chiang W-C, Hsu Y-L, et al. “ ECG arrhythmia classification using a probabilistic neural network with a feature reduction method”. Neuro computing 2013; 116: 38–45.
- 12) Wu J, Li F, Chen Z, et al. “Patient-specific ECG classification with integrated long short-term memory and convolutional neural networks”. IEICE Transactions on Information and Systems 2020; E103D (5): 1153–1163.
- 13) Yu SN, and Chou KT. “Selection of significant independent components for ECG beat classification”. Expert Systems with Applications 2009; 36(2): 2088–2096.
- 14) Zhang J, Tian J, Cao Y, et al. “Deep time–frequency representation and progressive decision fusion for ECG classification”. Knowledge Based Systems.2020;190:105402.