# Classification of Noisy ECG Signals using Chebyshev Filters

A THESIS

*Submitted by*

**M Bhanu Prakash Kumar**
**(CB.EN.P2CEN19008)**

*in partial fulfillment for the award of the degree of*

**MASTER OF TECHNOLOGY IN COMPUTATIONAL ENGINEERING AND NETWORKING**

**Center for Computational Engineering and Networking**

**AMRITA SCHOOL OF ENGINEERING**

**AMRITA VISHWA VIDYAPEETHAM**

COIMBATORE - 641 112 (INDIA)

**May - 2021**

# AMRITA SCHOOL OF ENGINEERING

# AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

## BONAFIDE CERTIFICATE

This is to certify that the thesis entitled **"Classification of Noisy ECG Signals using Chebyshev Filters"** submitted by **M Bhanu Prakash Kumar (Register Number- CB.EN.P2CEN19008),** for the award of the **Degree of Master of Technology** in the **"COMPUTATIONAL ENGINEERING AND NETWORKING"** is a bonafide record of the work carried out by him under our guidance and supervision at Amrita School of Engineering, Coimbatore.

| | | |
|---|---|---|
| Dr. Sowmya V<br>Project Guide<br>Assistant Professor (Sr.Gr.), CEN | Dr. E. A. Gopalakrishnan<br>Project Co-Guide<br>Assistant Professor (SG), CEN | Dr. K. P. Soman<br>Project Co-Guide<br>Head, CEN |

*Submitted for the university examination held on ...15/06/2021 ... ...*

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# AMRITA SCHOOL OF ENGINEERING
# AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

## DECLARATION

I, **M Bhanu Prakash Kumar (CB.EN.P2CEN19008), hereby** declare that this thesis entitled **“Classification of Noisy ECG Signals using Chebyshev Filters”**, is the record of the original work done by me under the guidance of **Dr.Sowmya V**, Assistant Professor, **Dr.E.A.Gopalakrishnan**, Assistant Professor and **Dr.K.P.Soman**, Professor and Head, Centre for Computational Engineering and Networking, Amrita School of Engineering, Coimbatore. To the best of my knowledge this work has not formed the basis for the award of any degree/diploma/ associateship/fellowship/or a similar award to any candidate in any University.

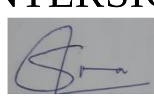
M. Bhanu Prakash

**Place:** Coimbatore

**Signature of the Student**

**Date:** 15/6/2021

COUNTERSIGNED



Dr. K.P.Soman
Professor and Head
Center for Computational Engineering and Networking

# Contents

Acknowledgement

First of all, I thank Almighty for the immeasurable blessings for smooth completion of my project. I am immensely pleased to express my sincere obligation to my project guide **Dr. Sowmya V**, Assistant Professor (Sr.Gr.) and my project co-guides Dr. **E. A. Gopalakrishnan**, Assistant Professor (SG) and **Dr. K.P.Soman**, Professor and Head, Centre for Computational Engineering and Networking, Amrita School of Engineering, Coimbatore, for their valuable guidance, dedication and encouragement for the successful completion of this project.

I am grateful to the entire staff of CEN, and research scholars of the Department for their timely cooperation. I avail this opportunity to thank my friends for their whole-hearted support in the difficult stages of the project.

I am greatly indebted to my loving family members for being the motivating forces behind the completion of this dissertation. Also I am grateful for their invaluable help, moral support and encouragement throughout the course of study.

# List of Figures

# List of Tables

# Abstract

Cardiac disease detection is a tedious process. Classification of electrocardiogram (ECG) signals plays an important role in the diagnosis of heart diseases. The most important factor that limits the detection of cardiac disease is the rare availability of instances of the abnormal condition collected using ECG sensors. And if the signals contain noise, then the classification might become a challenging task. In the first part of this work, we address the problem of cardiac disease detection when the dataset has less number of noisy ECG sensor signals. Here, Chebyshev Type II filter and Chebyshev function, which is termed as Chebfun, are used. The Chebyshev filter is used for high-frequency noise removal and Chebfun is used to approximate the signal with its coefficients. These coefficients known as Chebfun coefficients are used as the features. These features are used for classification. In the proposed work, machine learning algorithms, like SVM, logistic regression, decision tree, and AdaBoost, are used for classifying the features extracted from Chebfun. In the second part of this work, we take two problems into consideration, multiple cardiac disease detection for a noisy dataset. We have split the dataset into small and large. Here, Chebyshev Type II filter and Chebyshev function (Chebfun), are used for noise filtering and feature extraction of the signals. The features which are extracted from the small dataset is classified with SVM, Logistic Regression and Decision tree, while that of the large dataset is converted to spectrogram images and classified using deep neural networks.

# Chapter 1

# Introduction

Nowadays, cardiac diseases have a wide variety of patterns, which change drastically with respect to the individual. ECG is the most reliable method for the detection of these variations in the diseases. The distinct features of the cardiovascular diseases are captured in the QRS complex, P-wave, and T-wave components of the ECG signal. The ECG is a record of the electrical activity of the heart and thus ECG could capture all anomalies that can happen in the cardiac activities. Therefore, the electrical impulses leave the trace of the abnormal activities in its ECG structure. The classification of this type of ECG can become complex and difficult. So, for the proper diagnosis of ECG such types of noises should be removed.

The major types of noises in ECG signals are electromyographic (EMG) interface caused by the electrical activity of muscles, baseline drift caused by respiration and movement of body, power line interference caused by the electromagnetic interferences of the power line and electrode contact noise caused when there is no contact between the skin and electrode.

Spectrogram, an electronic or visual representation of the spectrum of frequencies for a signal that differs with time produced by a Fourier transform where frequency and time are visually represented, was also used in this work for pre-processing of the large dataset. The amplitude of the spectrum is represented by different colors.

## 1.1 Literature Review

In [3], the Chebyshev type I and Chebyshev type II filters have been used for noise reduction of ECG. In this work, both types I and II filters were designed with order 5, cut-off frequency 116Hz. Both the filters were found to be effective in reducing the low and high-frequency noise to a good extent. In [4], the Chebyshev Type II digital filter was implemented to improve degraded ECG signal quality for clinical diagnosis. Here the Type II filter was designed with order 5 and with a passband of 0.5 to 100 Hz was used for high-frequency noise removal. It was observed that the filter reduces the high-frequency noise components up to a satisfactory level. In [11], it is found that removal of high-frequency noise from the ECG, with Butterworth filter, has helped in the proper diagnosis of a fetus. In [12], FIR filters and IIR filters were used for noise filtering of ECG and K. S. Kumar et al. found them to be effective in eliminating the noise in the signals.

The above techniques were implemented for noise removal in signals. But the classification of small numbers of samples cannot be done using deep learning methods. In [1], the analysis of the variation of features captured by the model in case of conventional features and the Chebyshev features using less number of ECG samples was done. The classification was performed using Machine Learning algorithms. It was concluded that Chebyshev features extracted from the ECG signal have outperformed all other conventional feature extraction techniques, but in case of noisy data, the Chebyshev features couldn't outperform other conventional feature extraction techniques. In [2], B. Ganga et al. used the Chebfun for the accurate epoch extraction of the telephonic speech signal. The proposed method gained an improvement in performance by incorporating changes in the prevailed zero frequency filtering by including the Chebyshev interpolation method. In [9], the information about power signals is taken using VMD as modes. Each mode is then given to the Chebyshev function for approximation. This method gave a better accuracy for the estimation of components in power signals. In [13], Om Prakash Yadav et al. have used chebfun for compression and approximation of ECG signal. This method has worked very well and has given better results than other existing approximation techniques.

The variety of features that the model captures was classified in [14] when the critical frequency of the filter and the Chebyshev features were varied for a smaller number of ECG samples. Machine Learning algorithms were used to complete the classification. It was concluded that chebfun extracted features from the ECG signal in case of noisy signals, when filtered with Chebyshev type-II filter gave promising results.

Sujadevi.et.al. found that using a convolutional neural network (CNN) on the physionet datasets for raw signal classification generated better results without any denoising or trivial pre-processing techniques. Similarly, they declared CNN to be the stronger network for classifying Phonocardiogram signals in [15]. Similarly, it was declared that CNN is the stronger network for classifying Phonocardiogram signals in [18].

Hence, the objective of our work was to check the possibility of classification of a small noisy ECG dataset with more than 2 classes, using Chebyshev features and to check the classification performance of deep neural networks for the Noisy ECG dataset, if the dataset is large.

## 1.2 Problem statement

In the literature, there are various methods used for cardiac disease detection. In most cases, the signals used are not noisy. And the deep learning models are not suggestable in cases in which the number of ECG samples are less.

Deep learning architectures are proven to be good to classify ECG when there is a large dataset of samples. But efficiency of the deep learning architectures decreases for datasets with less number of samples. So Chebyshev features used with the conventional machine learning algorithms have been proved to effectively classify less number of samples, if they are not noisy. Hence, for noisy signals first noise reduction should be done for achieving good classification because the presence of noise can hinder the classification.

## 1.3 Objectives

- To check the possibility of Chebyshev filter to reduce noise, so that the Chebyshev features can be used to detect cardiac diseases.
- To analyze the change in the classification performance of the algorithms after the denoising of signals using Chebyshev filter.
- To check the possibility of classification of a small noisy ECG dataset with more than 2 classes, using Chebyshev features.
- To check the classification performance of deep neural networks with the Noisy ECG dataset, if it is large.

# Chapter 2

# Background

## 2.1 Chebyshev Filters

Chebyshev filters are analogue or digital filters having a steeper roll-off and having more passband ripple (type I) or stopband ripple (type II) than Butterworth filters. These filters are named in honor of Pafnuty Chebyshev because their mathematical characteristics have been derived from Chebyshev polynomials.

### 2.1.1 Chebyshev Type I filter

The amplitude response of a low pass Chebyshev type I filter is

$$G_n(\omega) = 1/\sqrt{(1 + \varepsilon^2 T_n{}^2(\omega/\omega_0))} \tag{1}$$

Where ε is the ripple factor, $\omega_0$is the cutoff frequency and $\mathbf{T}_n$ is a Chebyshev polynomial of the $n^{th}$ order. The amplitude response of a type I filter is shown in fig. 2.1.

### 2.1.2 Chebyshev Type II Filter

The amplitude response of a low pass Chebyshev type II filter is

$$G_n(\omega, \omega_0) = 1/\sqrt{(1 + 1/( \varepsilon^2 T_n{}^2 \frac{\omega_0}{\omega}))} \qquad (2)$$

Where ε is the ripple factor, $\omega_0$is the cutoff frequency and $\mathbf{T}_n$ is a Chebyshev polynomial of the $n^{th}$ order. The amplitude response of a type II filter is shown in fig.2.2.

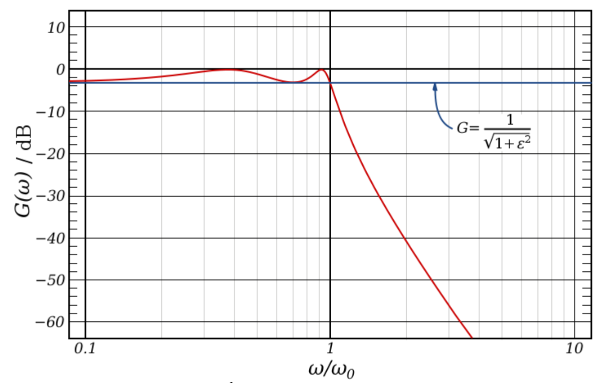



**Fig. 2.1**: The amplitude response of a $4^{th}$ order type I Chebyshev low-pass filter when ε=1. [7]

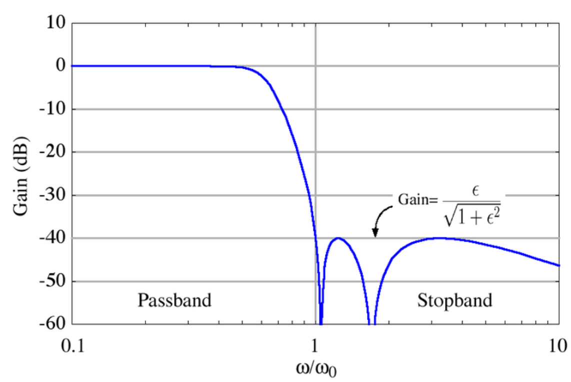



**Fig. 2.2**: The amplitude response of a $5^{th}$ order type II Chebyshev low-pass filter when ε =0.01. [7]

The response of type II filter has mostly flat passband and equiripple stopband; which is quite opposite to that of type I filter response. This is why the type II Chebyshev filters are known as inverse Chebyshev filters.

## 2.2 Chebyshev polynomial interpolation

The Chebyshev function is defined in an interval [a, b]. It is an open-source package available in Matlab. Chebfun is defined based on the fact that the smooth function can be symbolized by using the polynomial interpolations in Chebyshev points. The number of Chebyshev points is stored automatically to machine precision using an adaptive technique which computes the function with 15 or 16 digits of relative accuracy. The Chebfun system only stores minimum Chebyshev points for representing a signal even with a huge number of samples.

The N+1 Chebyshev points in the interval [-1, 1] are defined by

$$x(j) = -\cos(j\pi/N),\ 0 \leq j \leq N \tag{3}$$

A good approximation of f(x) with reduced coefficients is obtained by evaluating the Chebyshev polynomial series at the Chebyshev points. The Chebyshev approximation of f(x) is obtained by truncating the polynomial series and is given by

$$P_N(x) = \sum_{k=0}^{\infty} a_k T_K(x) \tag{4}$$

$a_k$ is known as the Chebyshev coefficients.

The Chebfun of the signal 's' is computed using the command

$$f=chebfun(s) \tag{5}$$

f is the Cheb function then, the command

$$C=chebcoeffs\ (f) \tag{6}$$

Returns its Chebyshev coefficient. The truncation of the valid coefficient is computed by using the command

$$t = chebfun\ (f, 'trunc', m) \tag{7}$$

‘m’ represents the number of the coefficients to which the function is to be truncated.

## 2.3 Spectrogram

A Fourier transform is used to create a spectrogram, which visually reflects the range of frequencies of a signal as they change over time. In a formed visual image, frequency and time are horizontals and verticals, respectively, and different colours indicate the magnitude of the spectrum [17].

For a given signal of x with a length of N, there are consecutive segments of the signal of m, where m ≤ ≤ N, and the $x \in R^{m\times\ (N-m+1)}$ where, in the formed matrix, rows and columns of x are indexed by time.

$\dot{x}$ = F × x and x = (1/m) × F × $\dot{x}$, of size m and matrix F, which are DFT columns of x, and F is the Fourier matrix with Fi, being its complex conjugate. $\dot{x}$ is shown in equation (2).

Rows and columns of $\dot{x}$ are indexed by the frequency and time, respectively, and their location corresponds to the point in frequency and time. The spectrogram is a visualized matrix where the matrix image with the $i^{th}$ and $j^{th}$ entry in the matrix, corresponds to the intensity or colour of the $i^{th}$ and $j^{th}$ pixel in the visually represented image general the bright colours denote the strong frequencies in a spectrogram. Figure 2.3 shows the sample spectrogram images of all classes.

$$\begin{bmatrix} 1 & 1 & 1 & .. & 1 \\ 1 & e^{i2\pi} & e^{i\frac{4\pi}{N}} & .. & e^{i2\pi\frac{(N-1)}{N}} \\ 1 & e^{i\frac{4\pi}{N}} & e^{i\frac{8\pi}{N}} & .. & e^{i2\pi\frac{2(N-1)}{N}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & e^{i2\pi\frac{(N-1)}{N}} & e^{i2\pi\frac{2(N-1)}{N}} & .. & e^{i2\pi\frac{(N-1)^2}{N}} \end{bmatrix} \tag{8}$$

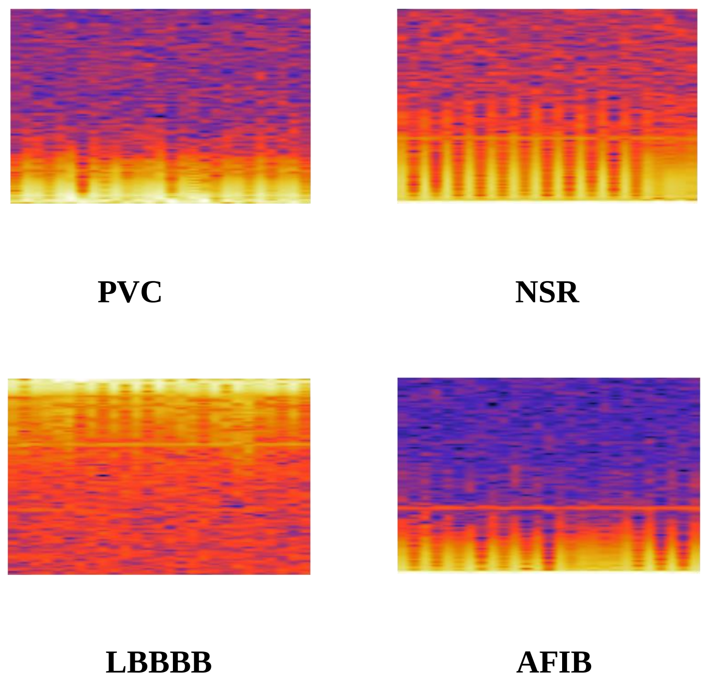



**Fig. 2.3.** Spectrographic Images of the Dataset used in Chapter 4 containing the 4 classes PVC, NSR, LBBBB and AFIB respectively.

# Chapter 3

# Noise Reduction of ECG using Chebyshev filter and Classification using Machine Learning Algorithms

## 3.1 Dataset Description

In the present work, two classes of ECG signals namely normal and Atrial Fibrillation (AF) are used. So, the dataset given in the 2017 PhysioNet/CinC challenge is taken [5]. The summary of the whole dataset taken for this work is shown in table 3.1.

**Table 3.1:** The summary of the Dataset taken for Classification

| **Class Name** | **No. of signals** |
|---|---|
| Normal Sinus rhythm | 5050 |
| Atrial Fibrillation | 738 |
| Other rhythms | 2456 |
| Too noisy to classify | 284 |
| **Total** | 8528 |

Here we are considering only the Normal sinus rhythm and atrial fibrillation classes. From each of the classes, 50 records are considered for the present work. Making our dataset of 100 signals.

## 3.2 Methodology

The main aim of the present work is to analyze the change in the classification performance of the algorithms after denoising the ECG signals using a Chebyshev filter. As we considered the experiments in a scenario where the availability of the data is limited we have conducted the analysis using state-of-art machine learning algorithms like SVM, Logistic Regression, Decision Tree, and AdaBoost. For better validation of the result using a small dataset, instead of conventionally splitting the dataset into training, testing, and validation sets we consider the leave-one-out cross-validation (LOOCV) and 10-fold cross-validation. In LOOCV, the model is validated against each of the data samples. The workflow of the validation set up is that the model is trained for all the available data samples except one. The trained model is tested using untrained data. Similarly, this procedure is repeated for all the data samples by shifting one after the other. The model accuracy is the average accuracy of the model prediction for all the data samples.

The methodology followed in the present work is shown in Fig 3.1. The signals are first given to Chebyshev filters for removing the high-frequency noise components if it is present. Next, the signals are given to chebfun for extracting the features of the ECG signals. Finally, the extracted features of the signals are fed to the classifiers for classification.

Also, the raw signals are given for classification and these are compared with the classification of the denoised signals. This comparison is done so that we can analyze the

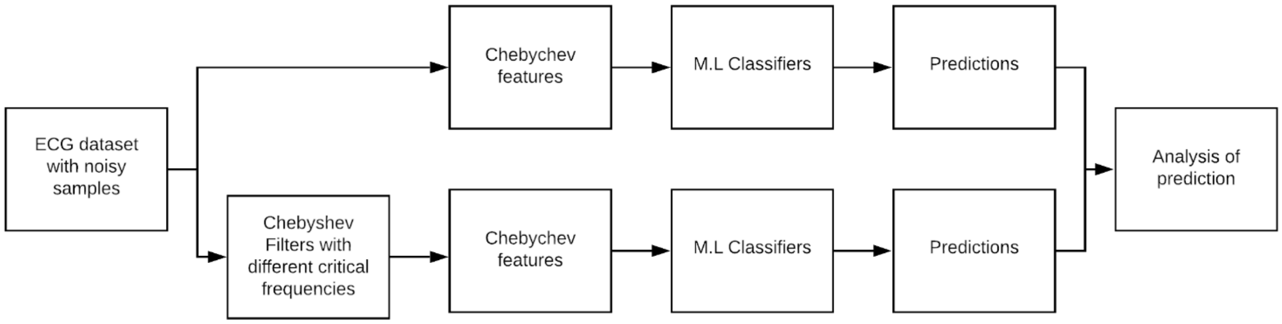



**Fig.3.1.** Proposed Methodology for the Noise Removal in ECG signals using Chebyshev Filter

extent to which the classification results are improving by using the Chebyshev filter.

## 3.3 Experiments and Results

The number of Chebyshev coefficients required to capture all the information of the ECG signal is fixed according to the findings in [1].

### 3.3.1 Designing the Chebyshev Type II filter

For the current work, the Chebyshev type II filter was designed according to the findings from [3], [4] and [6]. The order of the filter is 5, the minimum attenuation required in the stopband is 3db and the critical frequency used initially is 116Hz.

### 3.3.2 Chebfun feature extraction

The signal is down sampled from the sampling frequency of 1000 Hz to a sampling frequency of 300Hz. Next, the signal is transformed into a Chebyshev function. The Chebfun coefficients in the Chebyshev function will reconstruct the signal. Here the Chebfun coefficients considered are fixed according to the minimum coefficient required for the signal reconstruction. The Chebfun coefficients are fixed to 4994 coefficients for this work. The sample reconstruction of atrial fibrillation signal can be seen in fig.3.2 and the normal signal can be seen in fig.3.3.

The performance analysis of the different machine learning algorithms using different filters of varying critical frequencies are performed. Tables 3 to 10 show the performance analysis of the different machine learning algorithms using different filters by reducing critical frequencies with both cross-validations.

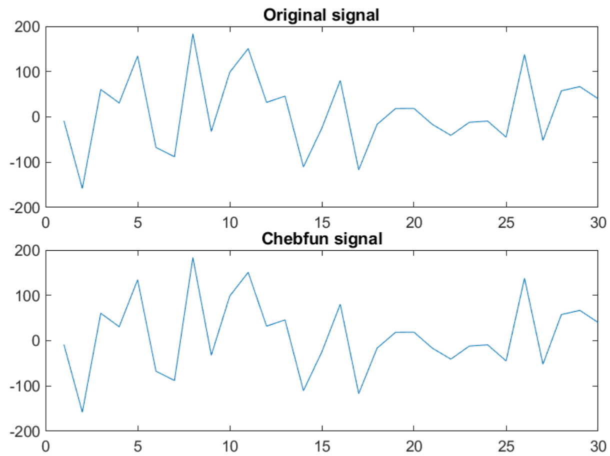



**Fig.3.2**: Reconstruction of atrial fibrillation signal using chebfun

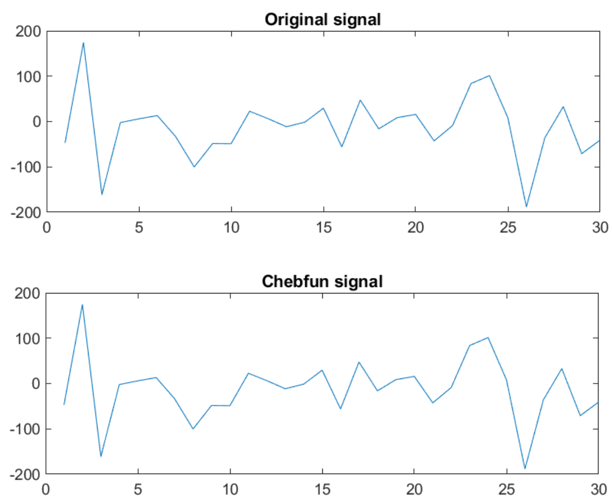



**Fig.3.3**: Reconstruction of the normal signal using chebfun

**Table 3.2**: The confusion matrix of the SVM model predictions using Chebyshev filters with different critical frequencies evaluated using leave one out cross-validation

| | | Raw Signal | | Critical frequency 116Hz | | Critical frequency 100Hz | | Critical frequency 80Hz | | Critical frequency 60Hz | | Critical frequency 50Hz | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PREDICTED VALUES | 0 | 37 | 13 | 31 | 19 | 27 | 23 | 27 | 23 | 28 | 22 | **39** | **11** |
| | 1 | 29 | 21 | 29 | 21 | 29 | 21 | 26 | 24 | 27 | 23 | **21** | **29** |
| | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | | Actual Values | | | | | | | | | | | |

For initial evaluation of the experiments, we have included the confusion matrix. The confusion matrices for the SVM model using different filters with reducing critical frequencies can be seen in table 3.2. Here class 0 is the atrial fibrillation class and class 1 is the normal class. Here it is evident that if we are removing the higher frequency components the classifier performance is also improving. Also, mainly the true positives in the normal class are improving from 21 to 29, which were concluded to be noisy in [1].

**Table 3.3**: Results obtained for SVM classifier using leave one out cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 58 | 52 | 48 | 51 | 51 | **68** |
| Precision | 58 | 52 | 48 | 51 | 51 | **68** |
| Recall | 59 | 52.5 | 48 | 51 | 51 | **68.5** |
| f1-score | 58.4 | 51.5 | 48 | 51 | 51 | **67.5** |

**Table 3.4**: Results obtained for SVM classifier using 10-fold cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 60 | 48 | 49 | 50 | 48 | **69** |
| Precision | 69.1 | 54.4 | 55.4 | 55.4 | 49.1 | **74.3** |
| Recall | 46.7 | 45.3 | 45.3 | 44.1 | 47.3 | **62** |
| f1-score | 52.7 | 43.8 | 43.8 | 43.4 | 45.1 | **65.9** |

For SVM, from Tables 3.3 and 3.4 we can observe that both validations have achieved almost the same metrics. Also, the performance of the algorithms is improving. In case of raw signals, the accuracy is 58% and 60%, while after using the filter of critical frequency 50Hz the accuracy increased to 68% and 69%.

**Table 3.5:** Results obtained for Logistic Regression classifier using leave one out cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 52 | 55 | 56 | 57 | **60** | 54 |
| Precision | 52 | 55 | 56 | 57 | **60** | 54 |
| Recall | 52 | 55 | 56.5 | 57 | **60** | 54 |
| f1-score | 52 | 55 | 56.2 | 57 | **60** | 54 |

**Table 3.6**: Results obtained for Logistic Regression classifier using 10-fold cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 54 | 53 | 54 | 57 | **61** | 56 |
| Precision | 69.1 | 52.8 | 55.5 | 56.3 | **64.3** | 58.9 |
| Recall | 46.7 | 47.2 | 49.3 | 50.3 | **60.5** | 61.6 |
| f1-score | 52.7 | 47.5 | 50.2 | 50.8 | **60.1** | 55.7 |

For Logistic Regression, from Tables 3.5 and 3.6 we can observe that the classification performance is only improving till the use of a filter with critical frequency 60Hz. After that, the performance is not improving but decreasing. The classification accuracy with raw signals is 52% and 54%, while after using a filter with critical frequency 60Hz is 60% and 61%. This can be seen as an improvement in the performance of the classifier.

**Table 3.7**: Results obtained for DecisionTree classifier using leave one out cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 68 | 50 | 57 | 62 | 64 | **75** |
| Precision | 68 | 50 | 57 | 62 | 64 | **75** |
| Recall | 68 | 50 | 57 | 62 | 64 | **75.5** |
| f1-score | 68 | 50 | 57 | 62 | 64 | **75.2** |

**Table 3.8**: Results obtained for DecisionTree classifier using 10-fold cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 65 | 55 | 53 | 47 | 52 | **70** |
| Precision | 56.6 | 58.6 | 47.2 | 48.3 | 47.6 | **70.4** |
| Recall | 52 | 49.9 | 47.7 | 35.3 | 50.2 | **62.7** |
| f1-score | 50.2 | 49.5 | 47 | 38.4 | 47.2 | **64.4** |

For Decision tree, from Tables 3.7 and 3.8, we have achieved the best classification performance compared with the other architectures used. While using the raw signals the accuracy is 68% and 65%, but after using the filter with a critical frequency of 50Hz the accuracy improved to 75% and 70%. Also, the Decision Tree classifier outperformed all other classifiers in terms of accuracy, precision, recall and f1-score.

**Table 3.9**: Results obtained for AdaBoost classifier using leave one out cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 57 | 57 | 57 | 56 | 57 | **61** |
| Precision | 57 | 57 | 57 | 56 | 57 | **61** |
| Recall | 57 | 57 | 57 | 56.5 | 57 | **61.5** |
| f1-score | 57 | 57 | 57 | 56.2 | 57 | **61.2** |

**Table 3.10**: Results obtained for AdaBoost classifier using 10-fold cross-validation

| | Raw Signal | Critical frequency 116Hz | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz |
|---|---|---|---|---|---|---|
| Accuracy | 59 | 55 | 53 | 50 | 50 | **64** |
| Precision | 58.5 | 55.8 | 46.8 | 53.5 | 47.5 | **68** |
| Recall | 50.4 | 54.9 | 49.9 | 39.4 | 50.2 | **59.4** |
| f1-score | 53 | 53.1 | 46.8 | 41.2 | 46.4 | **58.6** |

For Adaboost, from Tables 3.9 and 3.10, we can see that there is not much improvement in the classification performance. In case of the raw signals, the accuracy achieved is 57% and 59%, while after using the filter with a critical frequency of 50Hz we can see that the accuracy is 61% and 64%. This is not much of an improvement in terms of performance, so we can say that the Adaboost classifier is not good for classifying this current dataset.

From Tables 3 to 10 we could interpret that with decreasing the critical frequencies of the filter, i.e. with removing the higher frequency components the classification performance of the classifiers is improving.

As seen in the performance results of all the classifiers, using the different filters with different critical frequencies we have observed that the true positives for the normal class are increasing by removing the higher frequency components of the signals. This is an indication that the dataset taken here has high-frequency noise components which were hindering the classification performance as seen in [1]. Thus, we can say that our approach of using a Chebyshev filter combined with a Chebyshev function is useful in detecting cardiac diseases in case of less number of samples.

# Chapter 4

# Classification of Multiclass Noisy ECG Signals using Chebfun

## 4.1 Dataset Description

The dataset signals are taken from the MIT-BIH Arrhythmia Database. From the dataset, 50 signals from each class is taken to get the small dataset. For the large dataset the all signals from the 4 classes will be taken. Table 4.1 shows the details of the dataset. The dataset was downloaded from [16], an open source site. Here only 4 classes were considered for this experiment because the remaining classes in the dataset had very few samples, so in order to avoid class imbalance the below mentioned 4 classes were taken.

**Table 4.1.** The class wise details of the dataset

| **Classes in the Dataset Chosen** | **Number of records of ECG signal** |
|---|---|
| Normal Sinus Rhythm (NSR) | 283 |
| Premature Ventricular Complex (PVC) | 133 |
| Left bundle branch block (LBBBB) | 103 |
| Atrial fibrillation (AFIB) | 135 |

## 4.2 Methodology

As this work is an extension of [14], the work done was that the noisy signals are filtered using Chebyshev type-II filter and feature extraction was done using chebfun. These features were given to different machine learning algorithms for classification. This work has been replicated successfully to check the validity of it for multiclass ECG classification of a small dataset. The additional work here is that a large noisy dataset is also taken and classification is done to that using a CNN.

For the large and noisy dataset, initially the classification was tried with a 1D-CNN without any preprocessing. This gave an accuracy of 64%. So, the dataset is converted to 2D. The noisy signals are filtered using Chebyshev type-II filter and feature extraction was done using chebfun. These feature are then converted to spectrogram images, which is a conversion into the frequency domain from the time domain through a visual portrayal of the spectrum of frequencies of a signal as it varies with time. A spectrogram is generated by Fourier transform using the time and frequency as x-axis and y-axis respectively. Also, the different colours show the magnitude of the spectrum. The spectrogram images are fed as input to a 2D-CNN architecture where training and classification are done accordingly for the different datasets used.

## 4.3 Deep Learning Architecture

The noisy signals were filtered with Chebyshev type-II filter and feature extraction was done using chebfun. Then these features were converted to spectrographic images. So, the problem was converted to 2D and now a 2D-CNN was used to classify the spectrographs. This gave better results where accuracy was close to 85%. The 2D-CNN used is shown in Fig.4.1. It also has same number of convolution layers and 3 dense layers.

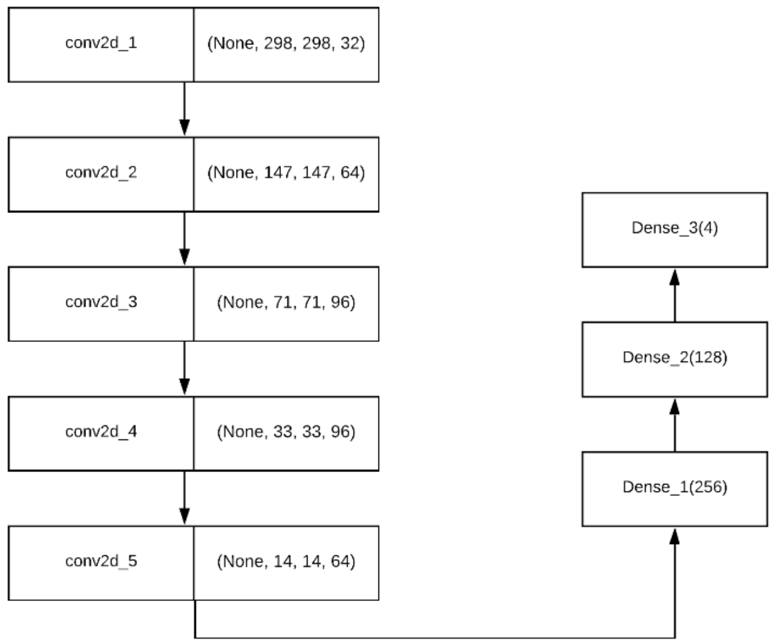



**Fig. 4.1**. Architecture of 2D-CNN used for classification

## 4.4 Experiments and Results

The small dataset is first given to Chebyshev type-II the filter and was initially tried with different critical frequencies and stop band attenuation. The final filter with 50Hz critical frequency and stop band attenuation of 3db was found to be giving the best results. Then the signal is given to a Chebyshev function for feature extraction.

The Chebfun coefficients that are specified here are set to the minimum needed for reconstructing the signal as seen in fig.4.2. The coefficients are set to 2500 for this study. Next the coefficients are classified by SVM, Logistic Regression and Decision Tree.

Decision Tree was found to give the best result of 74% accuracy.

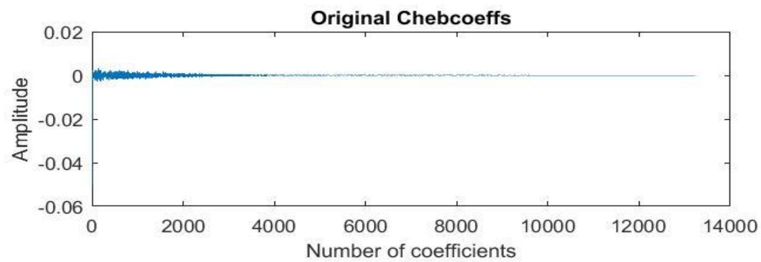



**Fig. 4.2.** The magnitude present in the chebfun coefficients of one signal

**Table 4.2**: Metrics achieved for SVM classifier with 10-fold cross validation

| | Raw Signal | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz | Critical frequency 40Hz |
|---|---|---|---|---|---|---|
| Accuracy | 0.43 | 0.41 | 0.40 | 0.40 | **0.47** | 0.41 |
| Precision | 0.52 | 0.31 | 0.29 | 0.29 | **0.50** | 0.31 |
| Recall | 0.43 | 0.41 | 0.40 | 0.40 | **0.47** | 0.41 |
| f1-score | 0.41 | 0.29 | 0.27 | 0.27 | **0.37** | 0.28 |

**Table 4.3** Metrics achieved for Logistic Regression classifier using 10-fold cross validation

| | Raw Signal | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz | Critical frequency 40Hz |
|---|---|---|---|---|---|---|
| Accuracy | 0.36 | 0.41 | 0.45 | 0.45 | **0.49** | 0.48 |
| Precision | 0.45 | 0.45 | 0.52 | 0.55 | **0.59** | 0.53 |
| Recall | 0.36 | 0.41 | 0.45 | 0.45 | **0.49** | 0.48 |
| f1-score | 0.36 | 0.39 | 0.43 | 0.43 | **0.48** | 0.46 |

**Table 4.4**: Metrics achieved for DecisionTree classifier using 10-fold cross validation

| | Raw Signal | Critical frequency 100Hz | Critical frequency 80Hz | Critical frequency 60Hz | Critical frequency 50Hz | Critical frequency 40Hz |
|---|---|---|---|---|---|---|
| Accuracy | 0.59 | 0.60 | 0.62 | 0.65 | **0.74** | 0.62 |
| Precision | 0.61 | 0.63 | 0.66 | 0.68 | **0.76** | 0.66 |
| Recall | 0.59 | 0.60 | 0.62 | 0.65 | **0.74** | 0.62 |
| f1-score | 0.57 | 0.59 | 0.60 | 0.64 | **0.72** | 0.62 |

**Table 4.5**: Confusion Matrices obtained for Decision Tree classifier using 10-fold cross validation

| | | Raw Signal | | | | Critical frequency 80Hz | | | | Critical frequency 60Hz | | | | Critical frequency 50Hz | | | | Critical frequency 40Hz | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PREDICTED VALUES | **0** | **24** | 11 | 7 | 8 | **20** | 0 | 30 | 0 | **27** | 0 | 23 | 0 | **35** | 0 | 15 | 0 | **23** | 0 | 27 | 0 |
| | **1** | 4 | **33** | 7 | 6 | 0 | **43** | 0 | 7 | 0 | **39** | 0 | 11 | 0 | **45** | 0 | 5 | 0 | **39** | 0 | 11 |
| | **2** | 12 | 6 | **25** | 7 | 26 | 1 | **23** | 0 | 26 | 0 | **24** | 0 | 21 | 1 | **28** | 0 | 31 | 0 | **19** | 0 |
| | **3** | 6 | 5 | 4 | **35** | 0 | 10 | 0 | **40** | 0 | 10 | 0 | **40** | 0 | 11 | 0 | **39** | 0 | 7 | 0 | **43** |
| | | **0** | **1** | **2** | **3** | **0** | **1** | **2** | **3** | **0** | **1** | **2** | **3** | **0** | **1** | **2** | **3** | **0** | **1** | **2** | **3** |
| | | Actual Values | | | | | | | | | | | | | | | | | | | |

From tables 4.2, 4.3 and 4.4, SVM gave a maximum accuracy of 47% and Logistic Regression gave a maximum accuracy of 49%. While, Decision Tree gave a maximum accuracy of 74%. Hence, it is clear that Decision tree can perform better than SVM and Logistic Regression for classifying the chebfun features. Also, in table 4.5 the confusion matrices for Decision tree is given.

For the large dataset, the signals were filtered and feature extraction was done using chebfun. Then these features were given to a 1D-CNN for classification. This did not give good result.

Hence, each signal features were converted to spectrogram and given to a 2D-CNN. The results of the classification done by both CNNs can be seen in table 4.6. The confusion matrix obtained for the 2D-CNN can be seen in table 4.7.

**Table 4.6**: Weighted average of the Metrics Obtained for each class for 1D-CNN and 2D-CNN

| | 1D-CNN | 2D-CNN |
|---|---|---|
| Accuracy | 0.64 | **0.85** |
| Precision | 0.41 | **0.86** |
| Recall | 0.64 | **0.85** |
| f1-score | 0.50 | **0.85** |

**Table 4.7**: Confusion matrix for the 2D-CNN

<table>
<tr><td></td><td></td><td>Class 0</td><td>Class 1</td><td>Class 2</td><td>Class 3</td></tr>
<tr><td rowspan="5">PREDICTED</td><td>Class 0</td><td>23</td><td>2</td><td>2</td><td>0</td></tr>
<tr><td>Class 1</td><td>0</td><td>21</td><td>0</td><td>0</td></tr>
<tr><td>Class 2</td><td>8</td><td>2</td><td>45</td><td>2</td></tr>
<tr><td>Class 3</td><td>0</td><td>3</td><td>1</td><td>23</td></tr>
<tr><td colspan="5">ACTUAL</td></tr>
</table>

# Chapter 5

# Conclusion and Future Work

In this work, the Effectiveness of Chebfun in ECG classification is shown using two different applications.

- **Noise Reduction of ECG using Chebyshev filter and Classification using Machine Learning Algorithms:** In the proposed work, we have analyzed the possibilities of the Chebyshev type II filter for the noise reduction in ECG signals and chebfun feature extraction for cardiac disease detection. From the experiments and the result, we could conclude that the Chebyshev filter when used with chebfun features extracted from the ECG signal using the DecisionTree classifier has outperformed all other classifiers used. From the experiments, it's also possible to infer that the True Positives of each class is increasing, hence giving better performance metrics. Also, this work can be taken forward by using Chebyshev Type I filter and Butterworth filter for noise reduction. The other future task would be to explore the possibilities of Chebyshev function for the interpretation of different classes in cardiac disease.

- **Classification of Multiclass Noisy ECG Signals using Chebfun:** In this work, it was found that chebfun feature extraction and Chebyshev type-II filter noise removal can be used to classify different cardiac diseases. Also we were able to deduce from the experiments and the results that, chebfun extracted features can be used with Decision tree for classifying small datasets with accuracy of 74%. For the larger datasets, the noise filtered and chebfun extracted features when converted to spectrogram can be effectively classified using CNN with 85% accuracy. Also, from the work done it was observed that CNN gave a better result than Machine learning algorithms although the pre-processing technique was same for both the methods.

# Bibliography


1. Sanjana K, B Ganga Gowri, Sowmya V, Gopalakrishnan E A, and Soman K P, “Detection of Cardiac Disease for Less Number of ECG samples using Chebyshev Coefficients”, Book chapter, Efficient Data Handling for Massive Internet of Medical Things- Healthcare Data Analytics, Book series: Springer- Internet of Things- Technology, Communications and Computing (2020).

2. B Ganga Gowri, K P Soman, and D Govind, "Improved Epoch Extraction from Telephonic Speech Using Chebfun and Zero Frequency Filtering", Interspeech (2018), pp.2152-2156.

3. Mahesh S Chavan, Ra Agarwala, M D Uplane, “Comparative Study of Chebyshev I and Chebyshev II Filter used For Noise Reduction in ECG Signal”, International Journal Of Circuits (2008), Systems and Signal Processing Issue 1, Volume 2.

4. Sonal Jagtap, Mahadev Uplane, “A Real Time Approach: ECG Noise Reduction in Chebyshev Type II Digital Filter”, International Journal of Computer Applications (2012), 49.10.5120/7659-0763.

5. https://physionet.org/content/challenge-2017/1.0.0/

6. Mahesh Chavan, Ra Agarwala, Mahadev Dattatreya Uplane, “Application of the Chebyshev type II digital filter for noise reduction in ECG signal”, Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision (2015), pp1-8.

7. Steven W Smith, CHAPTER 20 - Chebyshev Filters, Editor(s): Steven W Smith, Digital Signal Processing, Newnes (2003), Pages 333-342, ISBN 9780750674447.

8. https://www.chebfun.org/docs/guide/

9. Neethu Mohan and Soman K P, “Power System Frequency and Amplitude Estimation Using Variational Mode Decomposition and Chebfun Approximation System”, Twenty Fourth National Conference on Communications (NCC) (2018).

10. J Rajendran and Soman K P, “Design and Optimization of Band Pass Filter for SoftwareDefined Radio Telescope”, International Journal of Information and Electronics Engineering (2012), vol. 2.

11. J Joseph, J Gini and K I Ramachandran, “Removal of BW and Respiration Noise in abdECG for fECG Extraction”, Advances in Signal Processing and Intelligent Recognition Systems (2018), vol. 678. Springer International Publishing, Cham, pp. 3-14.

12. K S Kumar, B Yazdanpanah and P R Kumar, "Removal of noise from electrocardiogram

using digital FIR and IIR filters with various methods", International Conference on Communications and Signal Processing (ICCSP) (2015), pp. 0157-0162, doi: 10.1109/ICCSP. 2015.7322780.

13. Om Yadav, Shashwati Ray, “Efficient ECG Approximation Using Chebyshev Polynomials”, International Conference on Inventive Research in Computing Applications (ICIRCA) (2018), 1110-1115. 10.1109/ICIRCA.2018.8597372.

14. M Bhanu Prakash, Sowmya V, Gopalakrishna E A and Soman K P, "Noise Reduction of ECG using Chebyshev filter and Classification using Machine Learning Algorithms", International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (2021), pp. 434-441, doi: 10.1109/ICCCIS51004.2021.9397163.

15. V Sujadevi, Soman K P, R Vinayakumar, and A P Sankar, “Anomaly detection in phonocardio- grams employing deep learning”, Computational Intelligence in Data Mining. Springer (2019), pp. 525–534.

16. https://physionet.org/content/mitdb/1.0.0/

17. V G Sujadevi, Soman K P, R Vinayakumar, and A P Sankar, A U Prem, “Anomaly Detection in Phonocardiogram Employing Deep Learning", Advances in Intelligent Systems and Computing (2019), vol. 711, pp. 525-534.

18. Gopika P, Sowmya V, Gopalakrishnan E A, Soman K P, “Performance Improvement of Deep Learning Architectures for Phonocardiogram Signal Classification using Fast Fourier Transform”, International Conference on Intelligent Computing and Communication Technologies (ICICCT) (2019), Springer.

# List of Publications based on this research work

1. M. Bhanu Prakash, Sowmya V, Gopalakrishnan E A, and Soman K P, "Noise Reduction of ECG using Chebyshev filter and Classification using Machine Learning Algorithms", 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2021. **(Scopus Indexed)**

2. M. Bhanu Prakash, Sanjana. K, B. Ganga Gowri, Sowmya V, Gopalakrishnan E A, and Soman K P, “Detection of Cardiac Disease with Less Number of Electrocardiogram Sensor Samples using Chebyshev”, International Conference on Communication and Intelligent Systems, ICCIS 2020. **(Accepted and Presented)**

3. M Bhanu Prakash, V Sowmya, Gopalakrishna E A, K P Soman, “Classification of Multiclass Noisy ECG Signals using Chebfun”, International Conference on Computational Intelligence & Data Engineering, ICCIDE 2021. **(Submitted)**