

Defense Committee's Approval

University of Bahrain
College of Engineering
Department of Electrical and Electronics Engineering
Master of Science in Artificial Intelligent Systems

Student **Abbas Ali Amrallah** ID number **19961416** has defended his thesis by which he has fulfilled the requirement for the degree of Master of Science in Artificial Intelligent Systems (*Applying Signal Processing and AI Tools for the Early Detection of Heart Anomalies*) on **Sunday April 30th, 2023**.

The committee recommends awarding of M.Sc. Degree with the grade of: Pass.

(Defense Committee)

Name		Signature
<u>Dr. Zouhir Bahri</u>	Supervisor	
<u>Professor Kamran Arshad</u>	External examiner	<u>Kamran Arshad</u>
<u>Professor Mohab A. Mangoud</u>	Internal examiner	

Academic department's stamp





University of Bahrain
College of Engineering
Department of Electrical and Electronics Engineering

APPLYING SIGNAL PROCESSING AND AI TOOLS FOR THE EARLY DETECTION OF HEART ANOMALIES

A Thesis Submitted in Partial Fulfillment of the Requirements for the degree of
Master of Science in Artificial Intelligence System

Written by
Abbas Ali Hasan

Supervised by
Dr. Zouhir Bahri
University of Bahrain

April 30, 2023

Defense Committee's Approval

Abstract

This thesis provides a comparative study related to various signal processing and Artificial Intelligence (AI) techniques applied to Phonocardiography signals (PCG) to automatically differentiate normal heart sounds from five types of common murmurs. This should help to provide a simple-to-use and cost-effective PCG-based system for the early detection of cardiovascular anomalies. The average detection rate for heart anomalies in our study was 92.14%, while the classification rate was 71.02%. This is achieved by using Deep Neural Network (DNN) classification with Hyperbolic Tangent (tanh) activation function, a 5-layer, and 100 neurons per layer configuration. This thesis shows that the Discrete Wavelet Transform (DWT) was found to be the best denoising algorithm and the Heart Sound Envelopogram (HSE) was the best segmentation method for the PCG signal. Mel Frequency Cepstral Coefficients (MFCC) Features conjugated with DNN classification outperformed the Time and Frequency-Domain counterparts. In the framework of smart and preventative health care, the proposed system serves as a convenient home-care tool by providing potentially-ill individuals with vital early warnings and guiding them to cardiologists for more precise diagnoses.

Acknowledgments

Special thanks to my supervisor Dr. Zouhir Bahri and my advisor Dr. Ebrahim Mattar for their continuous support and guidance. I also appreciate the encouragement of all academic and faculty staff of the Electrical and Electronics Engineering Department at the University of Bahrain.

Table of Contents

Abstract.....	ii
Acknowledgments	iii
Table of Contents	iv
List of Tables	vii
List of Figures	viii
Abbreviations	x
Chapter 1: Introduction.....	1
1.1 Motivation and objective	1
1.2 Contributions of this work.....	1
1.3 Literature review	2
Chapter 2: Background.....	5
2.1 PCG signals overview	5
2.2 PCG signal processing.....	6
2.2.1 Infinite Impulse Response filters.....	6
2.2.2 Wavelet transform	7
2.2.3 Empirical Mode Decomposition.....	8
2.2.4 Butterworth and Chebychev filters.....	9
2.3 Signal segmentation	11
2.3.1 Wavelet decomposition using shannon energy	11
2.3.2 Template correlation.....	11
2.3.3 Heart sound envelogram	12
2.4 Feature extraction.....	13
2.4.1 Time-Domain Features	13
2.4.2 Frequency-Domain Features	15
2.4.3 Mel-Frequency Cepstral Coefficients features.....	17
2.5 Classification	19
2.5.1 K-Nearest Neighbors classification	19
2.5.2 Support Vector Machines classification	19
2.5.3 Decision Tree classification	19

2.5.4 Deep Neural Network classification	19
Chapter 3: Thesis details.....	23
3.1 Process overview.....	23
3.2 Dataset	25
3.3 PCG signal processing.....	26
3.3.1 Infinite Impulse Response filter	27
3.3.2 Discrete Wavelet Transform filter	28
3.3.3 Empirical Mode Decomposition filter	31
3.3.4 Butterworth filter	33
3.3.5 Chebyshev filter for PCG signal denoising.....	34
3.4 Signal segmentation	35
3.5 Feature extraction:.....	38
3.5.1 Time-domain features	38
3.5.2 Frequency-Domain features	39
3.5.3 Mel-frequency cepstral coefficients (MFCC) features	39
3.6 Classification experiments.....	39
3.6.1 K-nearest neighbors	40
3.6.2 Support vector machines	41
3.6.3 Decision tree.....	43
3.6.4 Deep neural network.....	44
3.7 Deep learning configuration	46
Chapter 4: Results and recommendation	47
4.1.1 Filters performance comparison	47
4.1.2 Segmentation algorithms comparison.....	47
4.2 Classification methods comparisonresults	48
4.3 Deep neural network recommended configuration	48
Chapter 5: Conclusion	52
5.1 Summary.....	52
5.2 Future points of expansion	53

Appendix A: Matlab Code..... 54

1. Filters Comparison..... 54

2. Segmentation Comparison..... 60

3. Feature Extraction..... 62

4. Classification Code 64

5. Recommended Methodology..... 66

6. Testing code..... 68

References..... 71

List of Tables

Table 1: MSE results for IIR filter	28
Table 2: Subband selection for DWT frequencies	29
Table 3: MSE results for DWT filter.....	31
Table 4: MSE results for EMD filter	33
Table 5: MSE results for BTW filter	34
Table 6: MSE results for CHV filter	35
Table 7: Segmentation evaluation criteria for segmentation methods	37
Table 8: Segmentation rating results for normal class	37
Table 9: Segmentation rating results for S3 class	37
Table 10: Segmentation rating results for AS class	37
Table 11: Segmentation rating results for MR class	38
Table 12: Segmentation rating results for MS class.....	38
Table 13: Segmentation rating results for MVP class	38
Table 14: Mean Square Error comparison results for all filters.....	47
Table 15: Segmentation rating results for all classes	47
Table 16: Classification results summary.....	48
Table 17: DNN Parameters selection test results.....	50
Table 18: Multiple testing using best performed DNN parameters	51

List of Figures

Figure 1: Miscellaneous heart sounds [21].	5
Figure 2: PCG Data waveform for differentc	6
Figure 3: Process of Discrete Wavelet Decomposition [26].	8
Figure 4: Process of Empirical Mode Decomposition [8]	9
Figure 5: Butterworth filter frequency response [27].	10
Figure 6: Chebyshev filter frequency response [27]	11
Figure 7: Template used for template matching	12
Figure 8: Mel-Frequency Cepstral Coefficient (MFCC) process.	18
Figure 9: Deep Neural Network architecture	20
Figure 10: Sigmoid activation function [42].	21
Figure 11: Tanh activation function [42].	21
Figure 12: ReLU activation function [42]	22
Figure 13: Overall system diagram	23
Figure 14: Thesis overview architecture.	24
Figure 15: Stethoscope modified for data recording	25
Figure 16: Denoised PCG using IIR filter	27
Figure 17: Daubechies 6 Wavelet function [44]	28
Figure 18: DWT decomposition for the PCG signal	29
Figure 19: 10 level DWT decomposition for the PCG signal.	30
Figure 20: DWT filter for the PCG signal	30
Figure 21: Intrinsic Mode Functions of noisy PCG signal	32
Figure 22: Denoised PCG using EMD filter	32
Figure 23: Denoised PCG using BTW filter.	33
Figure 24: Denoised PCG using Chebyshev.	34
Figure 25: Segmentation method selection plan	36
Figure 26: Stacked graph of PCG segmentation	36
Figure 27: Confusion matrix.	40
Figure 28: KNN classification with Time-Domain features.	40
Figure 29: KNN classification with Frequency-Domain features.	41
Figure 30: KNN classification with Cepstral-Domain features	41
Figure 31: SVM classification with Time-Domain features.	42
Figure 32: SVM classification with Frequency-Domain features.	42

Figure 33: SVM classification with Cepstral-Domain features 43

Figure 34: Decision Tree classification with Time-Domain features 43

Figure 35: Decision Tree classification with Frequency-Domain features 44

Figure 36: Decision Tree classification with Cepstral-Domain features..... 44

Figure 37: ANN classification with Time-Domain features..... 45

Figure 38: ANN classification with Frequency-Domain features..... 45

Figure 39: ANN classification with Cepstral-Domain features 46

Figure 40: Final proposed methodology 51

Abbreviations

(AI)	Artificial Intelligence
(AS)	Aortic Stenosis
(AC)	Autocorrelation
(BWT)	Butterworth
(CC-TC)	Template Correlation using Correlation Coefficient
(CHV)	Chebyshev
(CWT)	Continuous Wavelet Transform
(COVID)	Coronavirus disease
(DNN)	Deep Neural Network
(DWT)	Discrete Wavelet Transform
(DT)	Decision Tree
(db)	Daubechies Wavelet
(DCT)	Discrete Cosine Transform
(ECG)	Electro-Cardiography
(EMD)	Empirical Mode Decomposition
(FDF)	Frequency-Domain Features
(FFT)	Fourier transform-based
(FP)	False Positive
(FN)	False Negative
(HSE)	Heart Sound Envelopogram
(IIR)	Infinite Impulse Response
(IMFs)	Intrinsic Mode Function
(KNN)	K-Nearest Neighbor
(LDA)	Linear Discriminant Analysis
(MFCC)	Mel Frequency Cepstral Coefficients
(MLP)	Multilayer Perceptron
(MR)	Mitral Regurgitation
(MS)	Mitral Stenosis
(MSE)	Mean Square Error
(MVP)	Mitral Valve Prolapse
(PCA)	Principal Component Analysis
(PCG)	Phonocardiography
(ReLU)	Rectified Linear Units
(S1)	First Heart Sounds
(S2)	Second Heart Sounds
(S3)	Third Heart Sounds
(S4)	Fourth Heart Sounds
(SNR)	Signal-to-Noise Ratio
(SVM)	Support Vector Machine
(SE-TC)	Shannon Energy with Template Correlation
(SSD)	Sum of Squared Difference
(tanh)	Hyperbolic Tangent
(TDF)	Time-Domain Features
(TP)	True Positive
(TN)	True Negative
(WGN)	White Gaussian Noise

CHAPTER 1: INTRODUCTION

1.1 Motivation and objective

Cardiovascular disease is among the leading causes of death worldwide, which has a significant impact on the expense of the global healthcare system [1]. The early identification of heart issues should significantly lower the risks of problems and exorbitant treatment expenses before they worsen [2].

Even though the Electro-Cardiography (ECG) is more effective and accurate in the analysis of heart issues compared to the Phono-Cardiography (PCG) [3], the former requires special equipment and trained medical staff, necessitating the patient to be admitted to the hospital most of the time. On the other hand, PCG is most dominant among medical staff due to its efficacy and simplicity. Hence, this thesis aims at an efficient and cost-effective tool powered by Machine Learning in the context of a smart and proactive public health system, making it handy to use by untrained users, providing them with vital early detection of potential risks, and directing them to seek more precise diagnoses through cardiologists. The remainder of this thesis is structured as follows: Chapter 2 summarizes the main tools used to process PCG signals, Chapter 3 provides the details of the the thesis, Chapter 4 presents the results and recommendations of this study, and finally, Chapter 5 summarizes this work and opens directions for future expansion.

1.2 Contributions of this work

To our knowledge, Extensive comparisons of PCG signal processing and classification techniques are not frequently found in the free literature. As such, the main contributions of this work may be summarized as follows:

- A comparative study of the main PCG denoising tools highlighting the Wavelet transform as the method of choice.
- An investigation into the main tools of segmentation applied to the PCG signals showing the merits of the envelopogram.
- A demonstration through the simulation results that the CEPSTRAL technique produced the best feature extraction method when compared to other commonly used approaches.

- A comparative study of the main classification techniques illustrating the superiority of Deep Neural Networks (DNN).
- An optimization of the DNN by finetuning its various parameters, namely the number of hidden layers, number of neurons per layer, and the activation function.
- Propose a simple-to-use and cost effective early warning system that is useful in the context of home-care and smart health systems.

1.3 Literature review

Several researchs have been done in the previous few decades to automatically distinguish between normal PCG and cardiac murmurs. Lately, Ismail et. al. [4] presented a thorough survey report on this topic listing the various signal processing, segmentation, feature extraction, and machine learning techniques applied to the PCG signals to detect and classify heart anomalies. They separated the algorithms that were accessible into time, frequency, and hybrid methods. Although time-domain segmentation techniques provide accurate and direct localization of heart sounds, they need a significant amount of signal preprocessing due to their heightened sensitivity to noise. Therefore, the majority of common techniques convert the PCG signal from the temporal domain into another domain, where heart sounds may be distinguished more readily.

Various signal processing methods have been applied to PCG signals in the context of denoising them. This includes classical digital filters, namely IIR [5], Butterworth [6] Chebyshev (CHV) [7], Empirical Mode Decomposition (EMD) [8], in addition to the Discrete Wavelet Transform (DWT) [9].

Of the main segmentation techniques reported in the open literature, N. Kouras [10] applied segmentation to the PCG signal using Daubechies db10 DWT and Shannon entropy, with successful segmentation results. On the other hand, the Template Matching autocorrelation approach, developed by F. Agrafioti and D. Hatzinakos [3], significantly enhances the classification performance of the ECG signals while simultaneously serving as an intruder detector, making it a reliable and accurate biometric tool. Through the application of a Hilbert transform to the Shannon entropy envelope, Varghees et al [11] used new automated robust heart sound activity detection method and reported an average sensitivity of 99.43% and positive predictivity of 93.56% over clean and noisy pathological and non-pathological PCG signals with signal-to-noise ratio of 5 dB.

With regards to the feature extraction different methods, Fasil O and Rajesh R [12] classified EEG signals using Time-Domain features and suggested exponential energy feature, and they found that these approaches achieved classification accuracy of 89% and 99.5%, respectively, for the Ralph Andrzejak EEG dataset. In order to identify certain correlations between the time and Frequency-Domain representations of PCG, Singh J. and Anand R. [13] conducted experiments in both time and Frequency-Domains. The results showed that the Time-Domain analysis provides details on various time intervals related to first and second heart sounds. The major frequency components associated with various heart sounds are revealed by the Frequency-Domain analysis. The research also reveals the link between time and Frequency-Domain properties, showing that the dominating frequencies in S1 and S2 are substantially bigger for longer durations of S1 and S2. Iqtidar K. and Qamar U. [8] suggested 1D-Adaptive Local Ternary Patterns fused them into the MFCC features to create a robust feature extraction model, and they obtained classification accuracy of 98.3% and 97.2% mean through SVM classification.

According to the literature, numerous methods were employed by those tasked with categorizing PCG signals. Basak H, Roy A, [14] applied SVM and ANN classification using PCA and LDA features and achieved the best classification performance of 96.67% with ANN classification using PCA features. In order to test the applicability and usefulness of various decision tree designs, Pavlopoulos S [15] employed a limited sample of PCG signals. The results showed high levels of success for differentiating between different classes. Arslan and Karhan M [16] extracted MFCC features and looked at how the modes obtained by the Hilbert-Huang transform affected the classification. As a classifier, they applied the KNN, MLP, SVM, and DNN. When classifying PCG signals with the DNN classifier, their suggested Hilbert-Huang transform-based feature extraction method performed better than other methods with overall accuracy of 98.9%.

To assess the discriminating potential of the produced characteristics, Safara et al. [17] proposed utilizing Hankel matrix-based classification using singular value decomposition to distinguish between normal cardiac sounds and murmurs. In their work, segmentation was performed using the discrete wavelet transform (DWT), and a classifier powered by the adaptive-neuro fuzzy inference system (ANFIS) employed Shannon entropy as a feature. Other than mitral and pulmonary stenosis, this approach does not provide information about the classification accuracy of murmurs.

Gupta et al. [18] provided a new method for grouping Heart Sounds into a single cardiac cycle using K-means clustering and homomorphic filtering. To classify PCG signals as either normal or belonging to the murmurs (systolic and diastolic) category, the features set consisting of systole and diastole energies, spectral and cepstral coefficients were employed. They used the Daubechies-2 wavelet decomposition to extract the coefficient eigenvectors of heart sounds and then used them as the input for neural networks. Through the use of homomorphic filtering and the K-means classification approach, the accurate identification rate was 90.29%.

Choi [19] introduced a method for classifying valvular heart sounds as problematic using a wavelet packet with 8-level decomposition and a support vector machine with a fifth-order polynomial kernel function and retrieved two energy characteristics via SLBS from chosen nodes to identify the normal heart sounds. On the other hand, Reed et al. [20] offered a computer-aided diagnostic technique for five different clinical situations based on a Coifman fourth-order wavelet kernel and seven-level wavelet decomposition.

CHAPTER 2: BACKGROUND

2.1 PCG signals overview

The PCG signal may contain a variety of heartbeat phases as shown in Figure 1. Systole (S1) and diastole (S2) are reference heartbeats that are taken into consideration while assessing all other atypical signals, such as the third and fourth heart sounds, clicks, knocks, ejection, opening, and snaps. Timing (systolic or diastolic), duration (such as long or short), intensity (such as loud or soft), and pitch are the major characteristics that separate murmurs from regular heart sounds (low or high frequency). These unusual noises are occasionally described by words like "musical," "whoop," or "honk," all of which have complex higher frequencies that only qualified medical professionals can detect with a stethoscope [21].

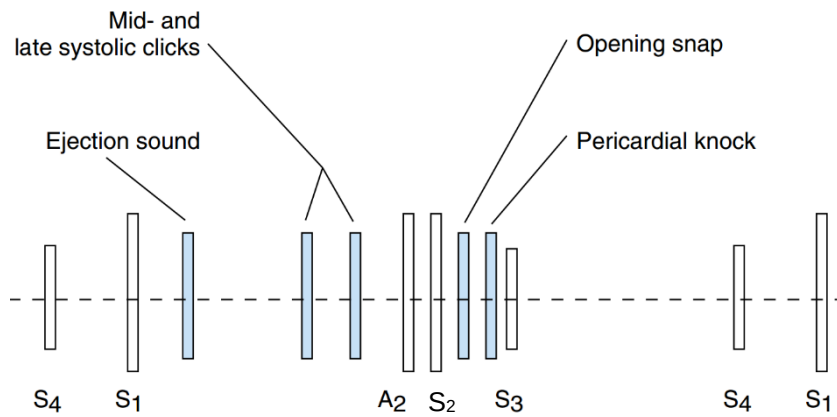


Figure 1: Miscellaneous heart sounds [21].

Frequencies associated with heart murmurs range from 20 to 500 Hz. Mitral Stenosis (MS), Aortic Stenosis (AS), Mitral Regurgitation (MR), the third, as well as fourth heart, sounds, and other murmurs, can occur at low frequencies of less than 100 Hz (S3 and S4). However, some other murmurs like mitral valve prolapse (MVP) have higher pitch noises with prominent frequencies at around 400 Hz [22]. Figure 2 depicts in time-domain a normal PCG signal and the five common heart murmurs dealt with in this work.

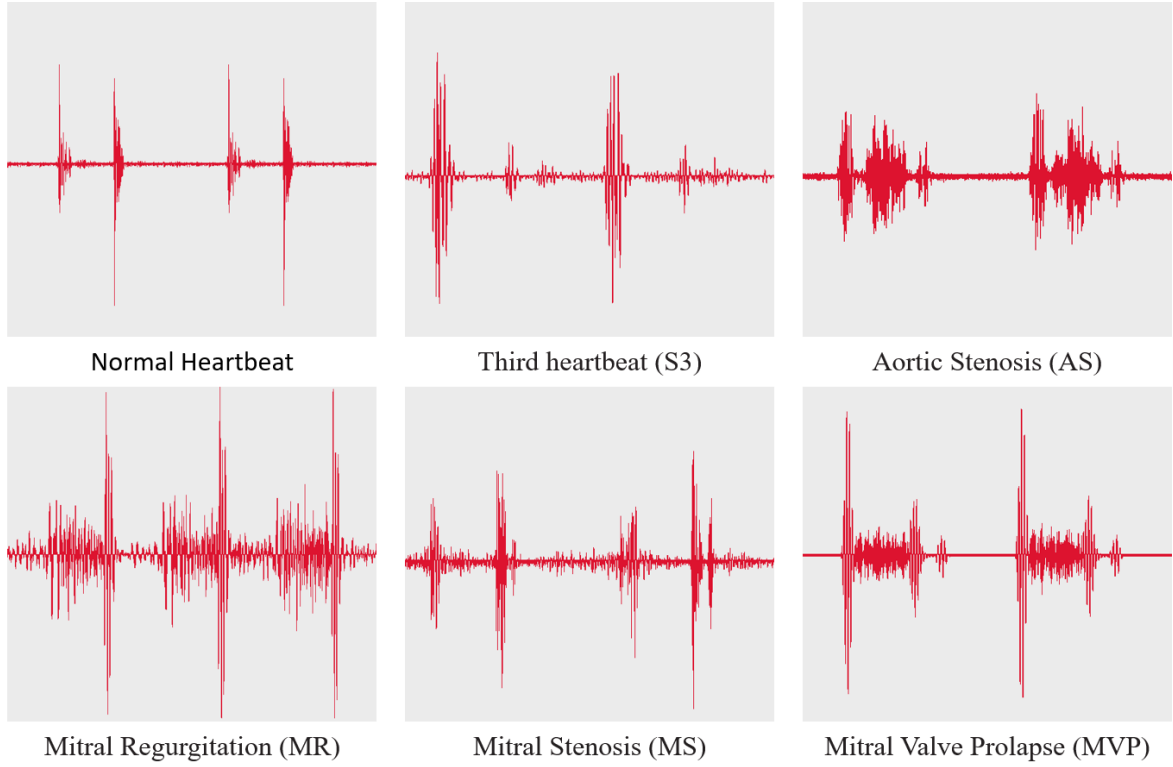


Figure 2: PCG Data waveform for differentc

2.2 PCG signal processing

2.2.1 Infinite Impulse Response filters

Despite the fact that they do not possess a linear phase profile, Infinite Impulse Response (IIR) filters are often used as classical digital filters owing to their compact and efficient implementation. IIR filters compute the current output samples utilizing the current and previous input and output samples, hence the name feedback and recursive. The general form is shown in Eq. (1) [5].

$$y(n) = - \sum_{k=1}^N a(k) \cdot y(n-k) + \sum_{k=0}^M b(k) \cdot x(n-k) \quad (1)$$

Where $a(k)$ elements are the feedback coefficients and the $y(n-k)$ elements are the output. The $b(k)$ elements are the feed-forward elements where the $x(n-k)$ element represents the input data stream. In many IIR filters, the number of feedback coefficients (N) and the number of feed-forward elements (M) are the same ($N=M$)

2.2.2 Wavelet transform

The most popular technique for converting from the Time-Domain to the Frequency-Domain is Fourier transform-based (FFT) spectral analysis. The Fourier theory states that a signal can be broken down into a variety of sines and cosines. Its disadvantage, though, is that it doesn't offer any time-related information. The wavelet transform, on the other hand, offers an analysis that is comparable to the FFT but uses a different merit function. The primary distinction is that the wavelet transform function, as opposed to the sines and cosines decomposition, is localized in both the time and Frequency-Domains. [23]

The essential idea behind the wavelet transform is to describe any function as a collection of wavelets that create the wavelet transform's basis function. Wavelets, which can be extended and moved to capture features that are confined in time and frequency, are tiny waves that are placed at various moments. The wavelets, also known as daughter wavelets, are scaled and graphically reproduced versions of a finite-length oscillating waveform. The basis for the best wavelet selection is based on the features of the original [9] and the required analysis goal.

The wavelet transform is regarded as a multi-resolution analysis since it offers a collection of time-frequency properties of the fundamental signal with various resolutions. Wavelet transform methods come in two varieties: Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT). The DWT creates a mutually orthogonal set of wavelets that are scaled and shifted version of the mother wavelet as given by Eq. (2) [24]

$$\psi_{m,k}(t) = \frac{1}{\sqrt{S_0^m}} \psi\left(\frac{t - k\tau_0 S_0^m}{S_0^m}\right) \quad (2)$$

where s_0 is a fixed scaling parameter set for more than 1, m controls wavelet dilation, k controls wavelet translation, τ_0 is the translation value, which must be greater than 0, and ψ is the mother wavelet.

Mallat [25] created a DWT algorithm in 1989 that uses a method akin to high-pass and low-pass filtering to extract the detail and approximation coefficients from the original signal, respectively. A level of decomposition is the name given to each repetition, which produces an approximation coefficient with numerous detail coefficients.

Figure 3 shows the structure of the DWT decomposition process which illustrates the approximation and detail coefficients for a decomposition level 2 where $x(n)$ is the original input signal and the $h(n)$ and $g(n)$ are the high-pass and low-pass filters. The DWT provides a

multilevel decomposition with an approximation of the decomposed signal and detail coefficients at each level of decomposition.

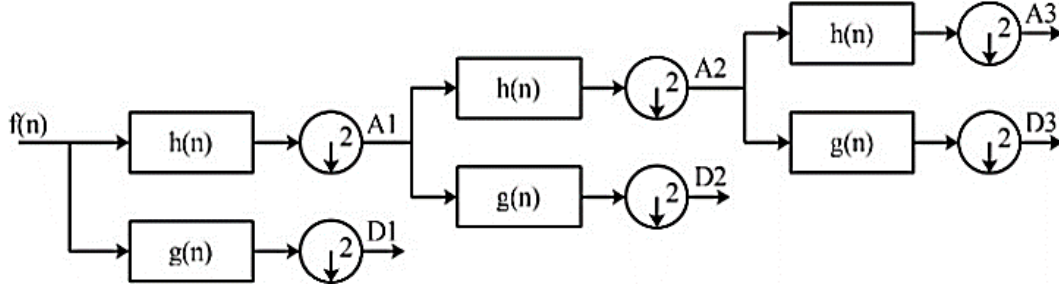


Figure 3: Process of Discrete Wavelet Decomposition [26]

2.2.3 Empirical Mode Decomposition

A signal is divided into several intrinsic mode functions (IMF) using the empirical mode decomposition (EMD) technique, coupled with a residue that represents the trend. The frequency information is derived using EMD from non-stationary and non-linear data sets. The nonlinear signal characteristics are extracted using EMD. An IMF has a mean value of zero and only one extreme between zero-crossings. The original PCG signal $x(t)$ is decomposed as [8]:

$$x(t) = \sum_{i=1}^n (c_i) + r_n \quad (3)$$

where c_i represents the extracted IMFs, r_n represents a residual signal, and several functions n in the set depend on the input signal. The complete pictorial process of EMD is shown in Figure 4 [8].

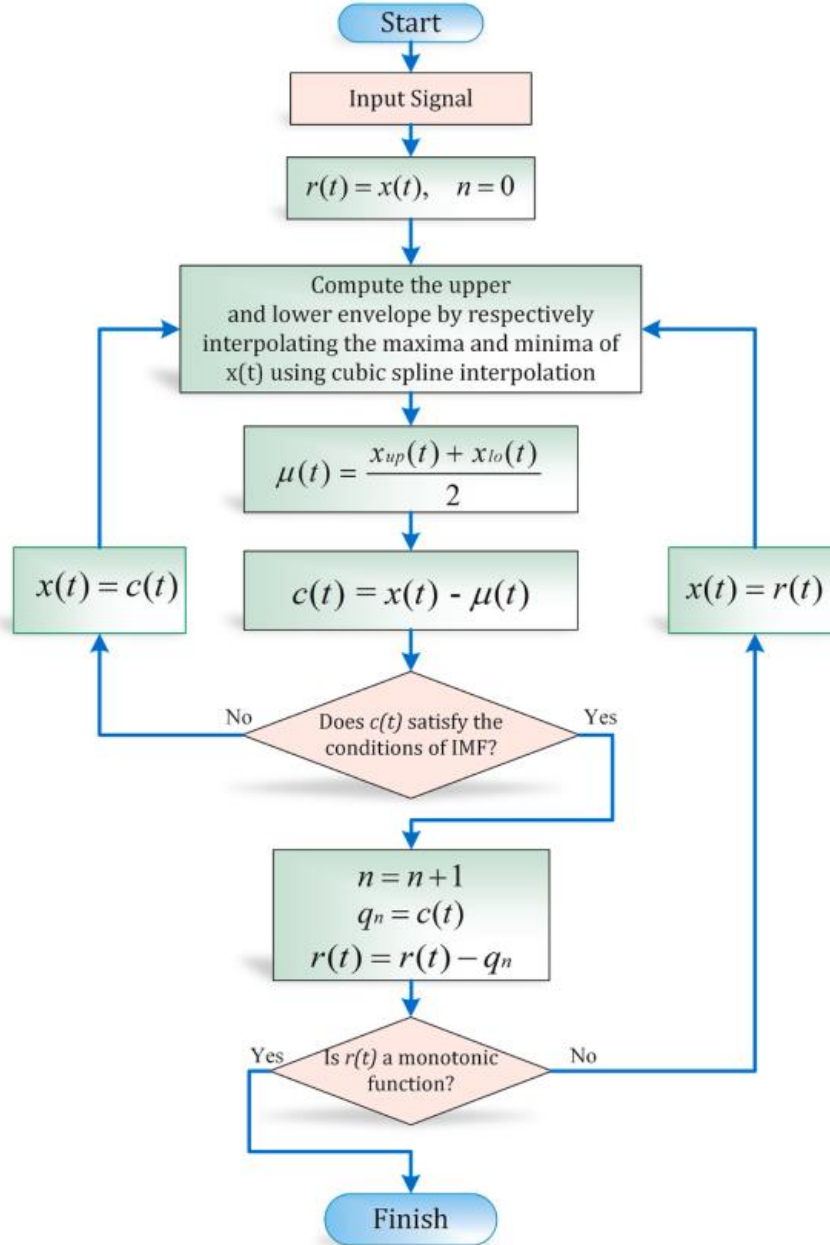


Figure 4: Process of Empirical Mode Decomposition [8]

2.2.4 Butterworth and Chebychev filters

The Butterworth filter BTW, also known as a maximally flat magnitude filter, is made to have a passband frequency response that is as flat as feasible (Figure 5) [27]. Even though it does not produce a very crisp output, the BTW filter is widely employed in many applications that require signal processing to obtain an effectively flat response within the specified frequency limits. The BTW filter seeks to attain its roll-off rate more gradually when compared to other filters (such as Chebyshev), contending that behavior below cut-off frequency rates is more crucial than at any other frequency [6], [28].

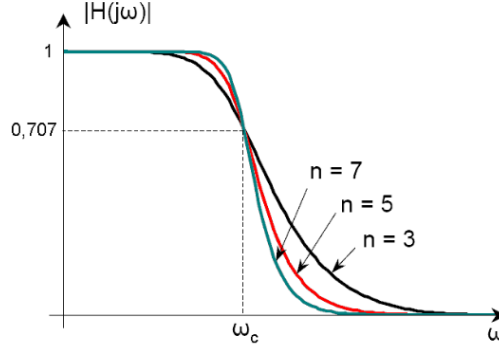


Figure 5: Butterworth filter frequency response [27]

The higher order of the Butterworth filter provides a higher number of cascaded stages; however, Butterworth's ideal frequency response is unattainable as it produces an excessive passband ripple. The general form representing an "nth" Order Butterworth filter, the frequency response is given in Eq. (4) [29]:

$$H(j\omega) = \frac{1}{\sqrt{1 + \varepsilon^2 \left(\frac{\omega}{\omega_c}\right)^{2n}}} \quad (4)$$

Where: n is the filter order, ω is equal to $2\pi f$ and ε is the maximum passband gain and ω_c is the cutoff frequency Chebyshev filter.

Chebyshev filters (CHV) are characterized by ripples in their pass and stop bands but offer lower transition bands than their BTW counterparts (Figure 6) [27]. Chebyshev minimizes the error between the actual filter with ripples and the idealized in the passband [7].

The gain response $G_n(\omega)$ in CHV Filter is calculated as described in Eq. (5) [30]:

$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 C_n^2(\omega/\omega_0)}} \quad (5)$$

Where ω is the angular frequency of the n th-order low-pass filter is equal to the absolute value of the transfer function $H_n(s)$ evaluated at $s=j\omega$, ε is the ripple factor, ω_0 is the cutoff frequency and C_n is a Chebyshev polynomial of the n th order.

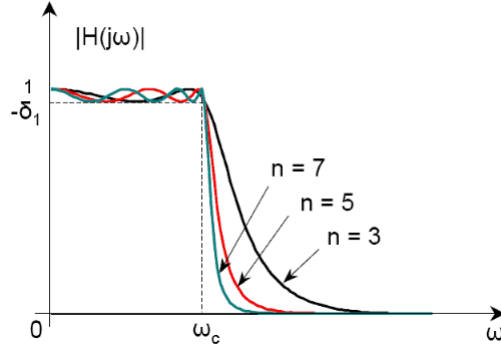


Figure 6: Chebyshev filter frequency response [27]

2.3 Signal segmentation

2.3.1 Wavelet decomposition using shannon energy

This method uses the Discrete Wavelet Transform (DWT) in conjunction with Shannon entropy. The entropy of the detailed coefficients will be calculated at each level [10]. The algorithm applied initially decomposes the PCG signal into six wavelet filter levels (Daubechies (db)), where the best level of decomposition is set based on the Shannon entropy calculated values at each level of decomposition. Then the mean value of the Shannon entropy determines the boundary between the main components and the pathological murmur and divides the main signal into several segments. finally, all segments are matched with a template heartbeat signal (Figure 7) using the Sum of Squared Difference (SSD) algorithm using Eq. (6) [31]:

$$ssd = \frac{1}{N} \sum_{k=1}^N (x_k - y_k)^2 \quad (6)$$

Where x_k is the feature of the registered template and y_k is from the authentication template, N represents the total number of features. After calculating similarity, a threshold is used to decide whether either given template is matched or not. SSD is a simple method and gives better results as compared to the correlation used in template matching.

2.3.2 Template correlation

This method uses feature selection based on Autocorrelation (AC) of PCG segments matched with the template signal shown in Figure 7. The method consists of four stages [3] [31]:

1. *Windowing*, where the preprocessed PCG signal is subjected to segmentation into nonoverlapping windows.
2. *Normalized autocorrelation* computation for every window.
3. *Dimensionality reduction* with the Discrete Cosine Transform (DCT) or Linear Discriminant Analysis (LDA).
4. *Classification* based on features obtained from the DCT or LDA.

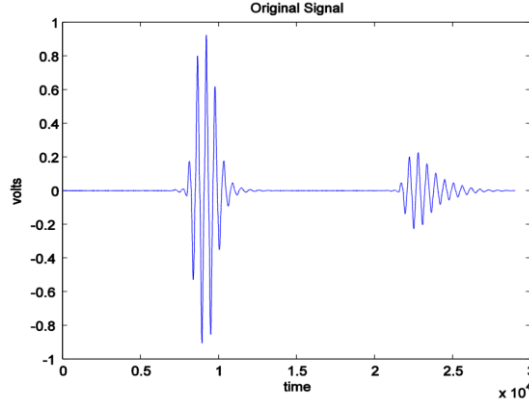


Figure 7: Template used for template matching

The PCG is a nonperiodic but highly repetitive signal. The motivation behind AC is that it embeds information about representative and highly distinctive characteristics of signals recorded from different individuals.

Autocorrelation is applied to windowed PCG signals. Windowing can be performed if only the signal is longer than the average heartbeat length to include more than one pulse. On such PCG windows, autocorrelation provides a shift-invariant perspective of similarity features over multiple heartbeat cycles. The normalized autocorrelation coefficients $R_{xx}[m]$ can be computed by Eq. (7) [3]:

$$\hat{R}_{xx}[m] = \frac{\sum_{i=0}^{N-|m|-1} x[i]x[i+m]}{\hat{R}_{xx}[0]} \quad (7)$$

Where $x[i]$ is the windowed PCG for $i=0, 1, \dots, (N-|m|-1)$, $x[i+m]$ is the time-shifted version of the windowed PCG with a time lag of $m = 0, 1, \dots, (M-1)$; $M \ll N$, and N is the length of the windowed signal. The dimensionality of autocorrelated signals is relatively high and DCT or LDA are applied for reduction.

2.3.3 Heart sound envelopogram

This segmentation approach is based on the envelope generated using the normalized average Shannon energy, which minimizes the impact of low value noise and facilitates the

detection of low intensity sounds. In this method, cardiac sounds will be divided into four parts: the cardiac sound (S1), systolic period, cardiac sound (S2), and diastolic period. The pattern of energy distribution is crucial for the diagnosis of valvular and septal defects, which cause murmurs of characteristic envelope shapes in particular locations of the cycle [32]. The energy function used is time-varying and is calculated as detailed in Eq. (8) [33]:

$$E(n) = \sum_{k=1}^M x^2(n-k) w(k) \quad (8)$$

where $x(n)$ is the signal and the weighting sequence $w(k) = M+1-k$ with $M=32$. This function is computed for the systolic and diastolic segments separately. The presence of murmurs is indicated when significant energy appears after S1 in the systolic segment or after S2 in the diastolic segment.

2.4 Feature extraction

2.4.1 Time-Domain Features

The preprocessed signal's Time-Domain properties, including mean, standard deviation, skewness, kurtosis, peak amplitude, root mean square, shape factor, impulse factor, and clearance factor, are computed to assess their capacity to identify the existence of murmur.

The standard deviation (SD) of the normalized PCG signal denoted by Xp_t is given as [34]:

$$SD = \sqrt{\frac{1}{N} \sum_{t=1}^N (Xp_t - \mu(Xp_t))^2} \quad (9)$$

Where $\mu(Xp_t)$ represents the mean value of Xp_t given by [34]:

$$\mu(Xp_t) = \frac{1}{N} \sum_{t=1}^N Xp_t \quad (10)$$

The Peak Amplitude (X_{peak}) is calculated by [34]:

$$X_{Peak} = \text{Max}(|Xp_t|) \quad (11)$$

The Root Mean Square (RMS) value is calculated as described in Eq. (12) [34]:

$$\text{RMS} = \sqrt{\frac{1}{N} \left(\sum_{t=1}^N (X_{p_t})^2 \right)} \quad (12)$$

The ratio of peak values to the effective value, which indicates how severe the peaks are in a waveform, is shown by the Crest Factor which is computed as follows: Eq. (13) [34]:

$$\text{CRF} = \frac{Y_{\text{Peak}}}{\text{RMS}} \quad (13)$$

The Impulse Factor (IF) which compares the height of a peak to the mean level of the signal is calculated as described in Eq. (14) [34]:

$$\text{IF} = \frac{Y_{\text{Peak}}}{\mu(X_{p_t})} \quad (14)$$

The Shape Factor (SHF), which is unaffected by signal size but is reliant on signal form is calculated as described in Eq. (15) [34]:

$$\text{SHF} = \frac{\text{RMS}}{\mu(X_{p_t})} \quad (15)$$

The ratio of the peak value of a signal to the square root of the absolute value of the PCG signal is known as the clearance factor (CF). The Energy (E) need to be calculated in order to calculate the Clearance Factor (CLF) of the signal which are shown in Eq. (16) and Eq. (17) [34]:

$$E = \frac{\left[\sum_{t=1}^N \sqrt{|X_{p_t}|} \right]^2}{N} \quad (16)$$

$$\text{CLF} = \frac{X_{\text{Peak}}}{E} \quad (17)$$

Higher-order statistics that are generated using third order equation provide insight into system behavior due to the asymmetry of the signal distribution, known as Skewness (Sk) which is calculated as described in Eq. (18) [34]:

$$\text{Sk} = \frac{\frac{1}{N} \sum_{t=1}^N (X_{p_t} - \mu(X_{p_t}))^3}{\left[\frac{1}{N} \sum_{t=1}^N (X_{p_t} - \mu(X_{p_t}))^2 \right]^{3/2}} \quad (18)$$

The fourth moment, known as the Kurtosis (Kurt), can be determined using higher-order statistical insight that indicates the length of a signal distribution's tails or, alternatively, the possibility that the signal is an outlier. Kurt can be calculated as described in Eq. (19) [34]:

$$\text{Kurt} = \frac{\frac{1}{N} \sum_{t=1}^N (X_{p_t} - \mu(X_{p_t}))^4}{\left[\frac{1}{N} \sum_{t=1}^N (X_{p_t} - \mu(X_{p_t}))^2 \right]^2} - 3 \quad (19)$$

2.4.2 Frequency-Domain Features

Frequency-Domain (spectral-domain) features are used to extract the frequency-related relationship of the signals of various categories. These features are extracted by first transforming the Time-Domain signal to the Frequency-Domain using Fast Fourier Transform (FFT) algorithm. From the FFT spectrum, 10 features are extracted, such as Spectral Flux, Spectral Crest, Spectral Flatness, Spectral Skewness, Spectral Centroid, Spectral Kurtosis, Spectral Spread, Spectral Roll Off, Spectral Slope, and Spectral Decrease [13], [35].

The fluctuation of the spectrum over time is measured by spectral flux which is calculated as described in Eq. (20) [36]

$$\text{flux}(t) = \left(\sum_{k=b_1}^{b_2} |S_k(t) - S_k(t-1)|^P \right)^{1/P} \quad (20)$$

Where S_k is the spectral value at bin k of the PCG signal, b_1 & b_2 are the band edges, over which to calculate the spectral flux, and P is the metric norm type.

The peakiness of the spectrum may be determined by the spectrum's crest. More tonality is indicated by a higher spectral crest, and more noise is indicated by a lower spectral crest. The spectral crest is calculated as described in Eq. (21) [36]

$$\text{crest} = \frac{\max(S_k \in [b_1, b_2])}{\frac{1}{b_2 - b_1} \sum_{k=b_1}^{b_2} S_k} \quad (21)$$

The peakiness of the spectrum may be determined by its spectral flatness. Whereas tonality is indicated by a lower spectral flatness, noise is indicated by a larger spectral flatness. The spectral flatness is calculated as described in Eq. (22) [36]

$$\text{flatness} = \frac{\left(\prod_{k=b_1}^{b_2} S_k\right)^{\frac{1}{b_2-b_1}}}{\frac{1}{b_2-b_1} \sum_{k=b_1}^{b_2} S_k} \quad (22)$$

The centroid's symmetry is measured by the spectral skewness which is calculated as described in Eq. (23) [36]

$$\text{skewness} = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_1)^3 S_k}{(\mu_2)^3 \sum_{k=b_1}^{b_2} S_k} \quad (23)$$

Where f_k is the frequency in Hz corresponding to bin k . μ_1 is the spectral centroid, calculated as described by the spectral Centroid function. μ_2 is the spectral spread, calculated as described by the spectral Spread function.

The spectral centroid which is the frequency-weighted sum normalized by the unweighted sum is calculated as described in Eq. (24) [36]

$$\text{centroid} = \frac{\sum_{k=b_1}^{b_2} f_k S_k}{\sum_{k=b_1}^{b_2} S_k} \quad (24)$$

The flatness or non-Gaussianity of the spectrum around its centroid is measured by spectral kurtosis and calculated as described in Eq. (25) [36]

$$\text{kurtosis} = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_1)^4 S_k}{(\mu_2)^4 \sum_{k=b_1}^{b_2} S_k} \quad (25)$$

The "instantaneous bandwidth" of the spectrum is represented by the spectral spread, which is used to indicate a tone's dominance. The spectral spread is calculated as described in Eq. (26) [36]

$$\text{Spread} = \sqrt{\frac{\sum_{k=b_1}^{b_2} (f_k - \mu_1)^2 S_k}{\sum_{k=b_1}^{b_2} S_k}} \quad (26)$$

The frequency bin under which a given rate of the total energy exists is identified by the spectral rolloff point (i), which measures the bandwidth of the audio signal and is calculated as in Eq. (27) [36]

$$\sum_{k=b_1}^i S_k = \kappa \sum_{k=b_1}^{b_2} S_k \quad (27)$$

Where κ is the percentage of the total energy contained between b_1 and i .

The spectral slope which measures the amount of decrease of the spectrum is calculated as described in Eq. (28) [36]

$$\text{slope} = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_f)(S_k - \mu_s)}{\sum_{k=b_1}^{b_2} (f_k - \mu_f)^2} \quad (28)$$

Where μ_f is the mean frequency.

Spectral decrease shows how much the spectrum has decreased while emphasizing the slopes of the lower frequencies and is calculated as described in Eq. (29) [36]

$$\text{decrease} = \frac{\sum_{k=b_1+1}^{b_2} \frac{S_k - S_{b_1}}{k-1}}{\sum_{k=b_1+1}^{b_2} S_k} \quad (29)$$

2.4.3 Mel-Frequency Cepstral Coefficients features

A representation of a sound's short-term power spectrum used in sound processing is called a Mel-Frequency Cepstrum (MFC), which is based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency.

An MFC is made up of a number of coefficients known as Mel-frequency cepstral coefficients (MFCC) [37]. They were created using an audio clip's cepstral representation (a nonlinear "spectrum-of-a-spectrum"). When used in audio compression, for instance, this frequency warping can improve the representation of sound and potentially lower the transmission bandwidth and storage needs of audio signals.

MFCC is used extensively recently for feature extraction and classification of heart sounds [8], [38]. MEL frequency is related to linear frequency in Eq. (30) [38]

$$\text{Mel}(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (30)$$

As shown in Figure 8 [38], the signal-to-noise ratio will be improved as the PCG signal is pre-weighted. The segmented PCG signals will be divided into frames using an overlapping window of around 30 ms considering 20 ms for the frame-blocking stage. To avoid parasitic

spectral leakage, a hamming window of the length of 1323 samples was chosen by setting the sampling frequency to 44.1 kHz.

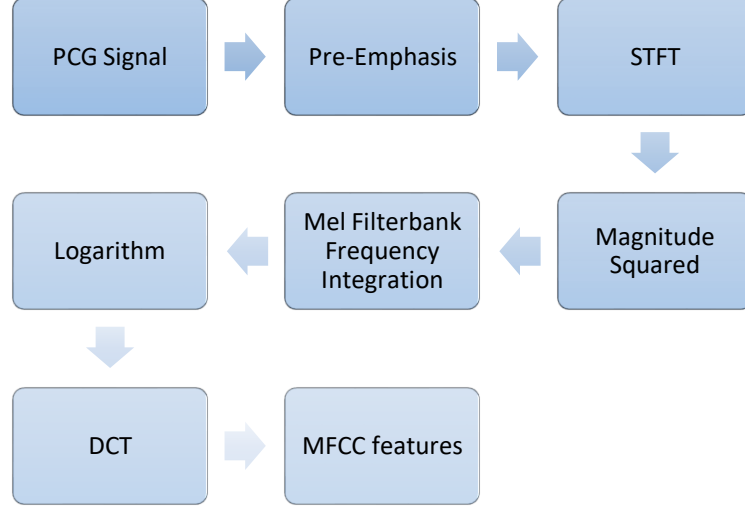


Figure 8: Mel-Frequency Cepstral Coefficient (MFCC) process.

Fast Fourier transform (FFT) is applied to the data to transform each segment to its Frequency-Domain version. After the PCG signal is segmented, the selected best segment is transformed into the Mel inverse spectrum domain after being filtered by a set of band-pass Mel triangular filters. The higher band of the PCG signal is compressed then by the algorithm of the Mel spectrum coefficient extracted from each Mel triangular filter, then DCT transforms the logarithmic Mel spectrum coefficients as illustrated in Eq. (31) [38].

$$c[n] = \sum_{m=0}^{N-1} S[m] \cos\left(\frac{\tau n}{M}\left(m - \frac{1}{2}\right)\right), n = 1, 2, \dots, M \quad (31)$$

Where $c[n]$ is the number of cepstral coefficients, and M is the filter bank. Typically, 8 to 13 cepstral coefficients are used in conventional MFCC systems. Since the zeroth coefficient indicates the average log energy of the input signal, which conveys very little speaker-specific information, it is frequently removed.

2.5 Classification

2.5.1 K-Nearest Neighbors classification

The idea of the nearest neighbor algorithm is to trace the nearest predetermined parameter of the training samples to a new point and estimate its label. A constant (k) can be defined in K-nearest neighbor learning KNN, or it can be related to its local density of points.

KNN is a non-generalized machine learning which is calculated by a simple majority vote of the other nearest neighbors and points, where a query point is assigned the class with the most representative of the nearest neighbors. [39]. If the feature representation values go on vastly different scales, then the normalization of the training data improves its accuracy dramatically [16].

2.5.2 Support Vector Machines classification

Support Vector Machine (SVM) is considered one of the main methods used in myoelectric signals. The function of the SVM is to categorize an n -dimensional hyperplane and categorize a set of input features into classes. SVM works by data mapping to a high-dimensional feature space so that it categorizes the data points, even if the data is not linearly separable. If a separator between the categories is detected, then the data will be modified so that the separator could be drawn as a hyperplane [14]. A linear kernel function was used in SVM to distinguish between multiple classes of beats extracted from PCG signals.

2.5.3 Decision Tree classification

Decision Tree (DT) is a method for supervised machine learning that is used for both regression and classification. While the associated decision tree is incrementally developed, the dataset is divided into smaller subsets. The features extracted from the training dataset, whose feature attributes (independent variables) and classifying attribute (dependent variable) are known, are used in the construction of the decision tree algorithm. The best attribute and condition for splitting the dataset are determined, taking into consideration the classifying attribute to get the highest classification accuracy [15].

2.5.4 Deep Neural Network classification

Deep Neural Network (DNN) is an advanced neural network that learns by communicating with other nodes (neurons) in multi-layered structures. Thanks to DNN, data can be categorized, trends can be spotted, and upcoming events can be predicted. DNN separates the

input data into layers of abstraction with several levels and varying weights for each element (Figure 9) [40]. These weights can be automatically changed throughout training until the DNN successfully completes the required aim [14].

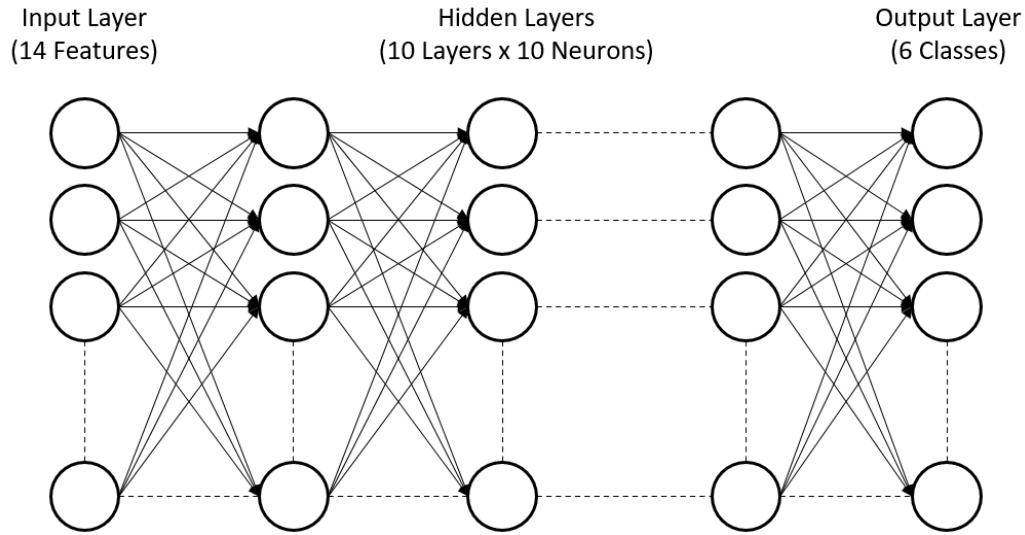


Figure 9: Deep Neural Network architecture

A nonlinear mapping is carried out by the neurons in the neural network's layers one after the other. With the use of backpropagation algorithms and a weighted sum of neurons, this mapping enables network learning [41].

Activation functions are used to learn non-linear and complex functional mappings between the inputs and the response variable in an artificial neural network. activation function should be differentiable so that backpropagation optimization can be performed in the network while computing gradients of error (loss) concerning weights, to optimize weights and reduce errors [42]. Three of the most popular non-linear activation functions, namely Sigmoid, Tangent Hyperbolic (tanh), and Rectified Linear Units (ReLU), are considered in this work among the many types of activation functions available in DNN.

Sigmoid:

A sigmoid function (Figure 10) ranges between 0 and 1. It is mostly used for the output layer of a binary classification. Sigmoid provides a bound output in the range of (0,1) which is better than linear activation functions that provide $(-\infty, \infty)$.

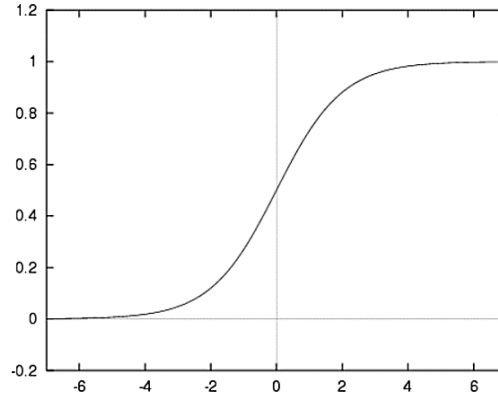


Figure 10: Sigmoid activation function [42]

Sigmoid scales down large negative numbers toward 0 and large positive numbers toward 1. Its output is not zero-centered, which makes gradient updates go too far in different directions and makes optimization harder. It has a vanishing gradient problem and slow convergence.

The sigmoid function is defined as follows:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (32)$$

Tangent Hyperbolic (tanh):

The tanh function (Figure 11) scales the values between -1 and 1. The gradient for tanh is steeper than it is for sigmoid. Unlike sigmoid, it is centered around zero, which makes optimization easier. It is usually used in hidden layers. It has a vanishing gradient problem.

The tanh function is defined as follows:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (33)$$

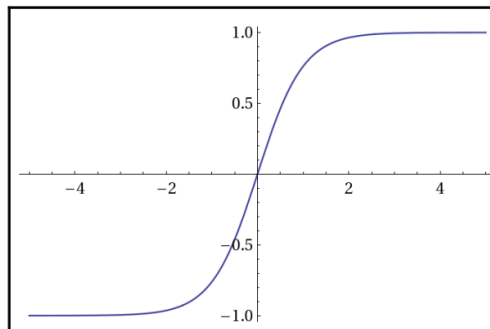


Figure 11: Tanh activation function [42]

Rectified linear units (ReLU):

ReLU is a non-linear function that ranges from 0 to infinity (Figure 12). Its convergence is faster than sigmoid and tanh. It is used mostly in hidden layers. It does not have a vanishing gradient problem.

The ReLU function is defined as follows:

$$f(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (34)$$

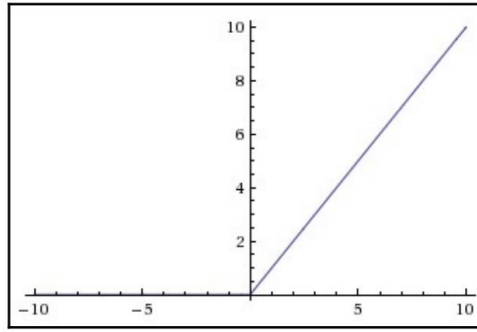


Figure 12: ReLU activation function [42]

This chapter provided an outline of the techniques and methods employed in this thesis. The upcoming chapters will conduct a comparison study that employs these techniques and equations to filter, extract a single heartbeat segment, extract features and classify the PCG signals.

CHAPTER 3: THESIS DETAILS

3.1 Process overview

In order to classify the PCG signal, it must go through a number of processing stages as shown in Figure 13. For each stage, Various approaches were examined and the most effective one was chosen. Figure 14 depicts the various ramifications in this thesis. For the signal denoising phase, 5 methods of filtering methods were performed: 3 Band Pass Filters configured with bandwidth 20-500 Hz and order 4: Custom-IIR, Butterworth (BWT), and Chebyshev (CHV), as well as Empirical Mode Decomposition (EMD) (with 10 IMFs (Intrinsic Mode Function) with relative energy greater than 25%), in addition to the Discrete Wavelet Transform (DWT) with 20 and 500 Hz selected bands and Daubechies 6 (db6) as a mother wavelet. The Mean Square Error (MSE) attribute was used as a performance measure of the filters and was evaluated by calculating the difference between the original clean data and after adding White Gaussian Noise (WGN) and artifacts across the PCG signals.

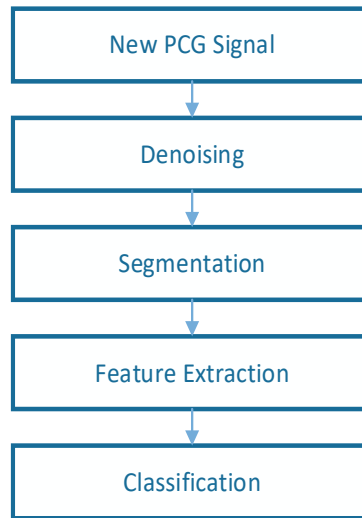


Figure 13: Overall system diagram

In the segmentation stage, 3 techniques were performed: Segmentation by Wavelet Decomposition using Shannon Energy with Template Correlation (SE-TC), Segmentation based on Heart Sound Envelopogram (HSE), and Segmentation by Template Correlation using Correlation Coefficient (CC-TC). To evaluate these techniques' use and compare their effectiveness, a visual comparison was conducted using a stacked comparison graph that was created for all available data to visualize the segmented portion with its reference to the original data, then carried out a visual assessment of all the data.

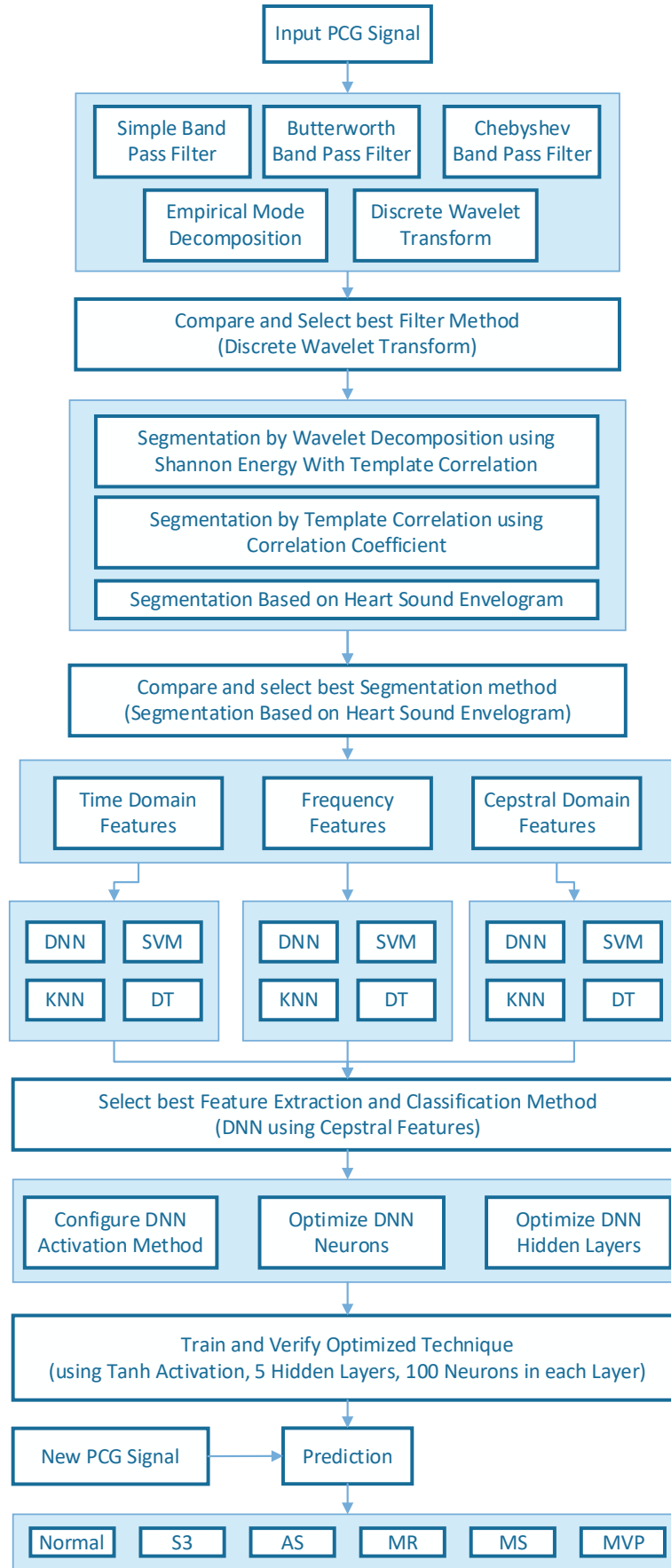


Figure 14: Thesis overview architecture

Next, in the feature extraction phase, 3 different methods were applied to the segmented portions of the PCG signal: Time-Domain Features (TDF), Frequency-Domain Features (FDF), and Mel Frequency Cepstral Coefficients (MFCC). These features were injected into the training and verification process using 70% of the data for training and 30 % for testing conducted by 4 classification techniques: Support Vector Machine (SVM), Deep Neural Network (DNN), Decision Tree (DT), and K-Nearest Neighbor (KNN).

As MFCC and DNN create the best results in detection and classification rates, an additional process of finetuning was conducted for the DNN parameters: the activation function (tanh, sigmoid, and ReLU), the number of hidden layers (3, 5, and 10), as well as the number of neurons in each hidden layer (10, 50, and 100).

3.2 Dataset

The classical stethoscope used in this study was modified to enable heartbeats to be measured using a computer. This involved connecting a microphone plug to the stethoscope. Figure 15 illustrates the modified stethoscope with the microphone. The modified stethoscope was used to measure the heartbeats precisely in the research presented in this thesis.



Figure 15: Stethoscope modified for data recording

While a more expensive option, such as high-quality cellphones or digital electronics stethoscopes, could be utilized to provide higher-quality PCG signals, a low-cost but noisy modified conventional stethoscope was employed as the input device in this study.

The PCG signal is tainted with the noise surrounding and artifacts all the time due to ambient sounds and stethoscope jitters, therefore, denoising the signal is essential before proceeding to the next phase (segmentation) where a single beat should be separated before extracting its features for the classification process.

The first goal was to collect heart sounds from people and those with various heart murmurs using a modified stethoscope at hospitals across Bahrain with the help of specialized medical professionals. However, access to patients in hospitals has been severely constrained during the Covid-19 outbreak.

A total of 70 PCG records were obtained from individuals with a confirmed healthy status, along with approximately 120 recorded data mainly sourced from recordings made by past students. The remaining data was obtained from available online sources [43] [41]. All of these recordings were used in the study. The dataset includes 6 classes as follows:

- Normal heartbeat (601 files)
- Third heartbeat (S3) (76 files)
- Aortic Stenosis (AS) (227 files)
- Mitral Regurgitation (MR) (225 files)
- Mitral Stenosis (MS) (222 files)
- Mitral Valve Prolapse (MVP) (200 files)

It is significant to highlight that the data collected were extremely diverse concerning the signal-to-noise ratio (SNR) and sampling rate. While some were of acceptable quality, others (such as those made with the modified stethoscope) were very noisy. This has made the detector/classifier more challenging.

3.3 PCG signal processing

All applications that use noisy data must include signal processing. The PCG signal is recorded using a standard stethoscope that is connected to a computer, and the recorded PCG signal includes lots of noise. Five different types of filtering methods were used in the noise removal phase: In addition to Custom-IIR, Butterworth (BWT), and Chebyshev (CHV) bandpass filters, algorithms such as Discrete Wavelet Transform (DWT) and Empirical Mode Decomposition (EMD) were employed. To assess the effectiveness of the filters, the Mean

Square Error (MSE) attribute was calculated between the clean original data and its noisy version following the addition of the White Gaussian Noise (WGN) and artifacts to all signals.

3.3.1 Infinite Impulse Response filter

The PCG signals were denoised using a custom Infinite Impulse Response IIR filter with a bandwidth of 20-500 Hz and order 4. Figure 16 depicts the PCG signal before and after it has been denoised (filtered). The signal was greatly attenuated as shown in the figure below, and the noise in the PCG signal was successfully filtered.

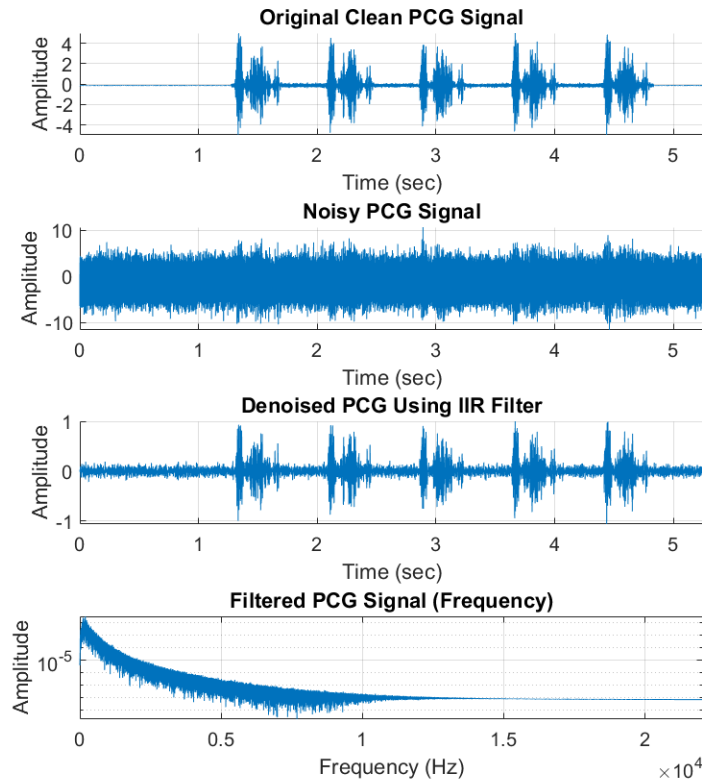


Figure 16: Denoised PCG using IIR filter

From the clean data for each class, 10 samples were randomly chosen. The average MSE attribute was calculated for the IIR Filter comparing the base clean data to its data after adding White Gaussian Noise (WGN) and artifacts for various classes. The MSE results of the IIR filter are given in Table 1.

IIR Filter						
Filter	Normal	S3	AS	MR	MS	MVP
Test 1	0.033	0.0411	0.0807	0.0563	0.0522	0.068
Test 2	0.0447	0.0282	0.0739	0.0695	0.1122	0.0422
Test 3	0.0388	0.1087	0.0603	0.1122	0.081	0.0926
Test 4	0.0201	0.1169	0.1038	0.077	0.034	0.031
Test 5	0.0326	0.0699	0.0806	0.08	0.038	0.082
Test 6	0.0923	0.0312	0.1055	0.1096	0.0095	0.0189
Test 7	0.0518	0.1123	0.1332	0.037	0.1123	0.0859
Test 8	0.112	0.0428	0.0429	0.0403	0.0527	0.0407
Test 9	0.0251	0.1165	0.1055	0.0774	0.0699	0.0686
Test 10	0.1125	0.112	0.0367	0.0903	0.0558	0.0714
Average	0.05629	0.078	0.08231	0.075	0.0618	0.0601

Table 1: MSE results for IIR filter

3.3.2 Discrete Wavelet Transform filter

The input PCG signal was decomposed into 10 levels using Discrete Wavelet Transform DWT filter. The Daubechies 6 (db6) wavelet template Figure 17 is selected as the mother wavelet because it morphologically resembles the PCG signal [44].

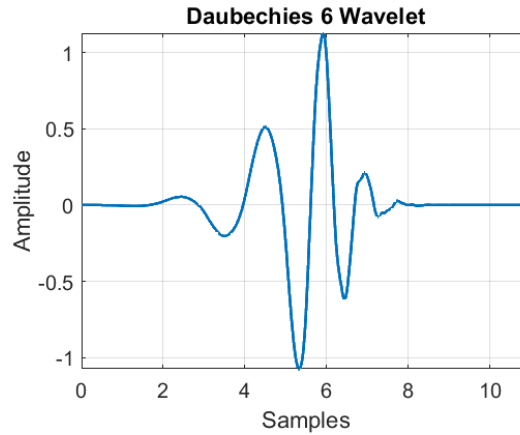


Figure 17: Daubechies 6 Wavelet function [44]

After the decomposition of PCG up to 10 levels, 11 signals/subbands (D1, D2, D3, D4, D5, D6, D7, D9, D10 A10) were obtained, each containing different frequency ranges of 20-500 Hz. Figure 18 illustrates the decomposition of a signal into various levels given in Table 2. Among these subbands, only the detail coefficients D6, D7, D8, D9, and D10 were chosen since they contain the frequency range of a heart sound, and the necessary frequencies for abnormal heartbeats fall between 20 and 500 Hz. The selected bands were added to perform the wavelet reconstruction. The PCG signal was decomposed into 10 levels of Daubechies wavelet (Figure 19).

Frequency (Hz)	Subband	Selection
44100	-	-
22050	-	-
11025	D1	FALSE
5513	D2	FALSE
2756	D3	FALSE
1378	D4	FALSE
689	D5	FALSE
345	D6	TRUE
172	D7	TRUE
86	D8	TRUE
43	D9	TRUE
22	D10	TRUE
11	D11	FALSE

Table 2: Subband selection for DWT frequencies

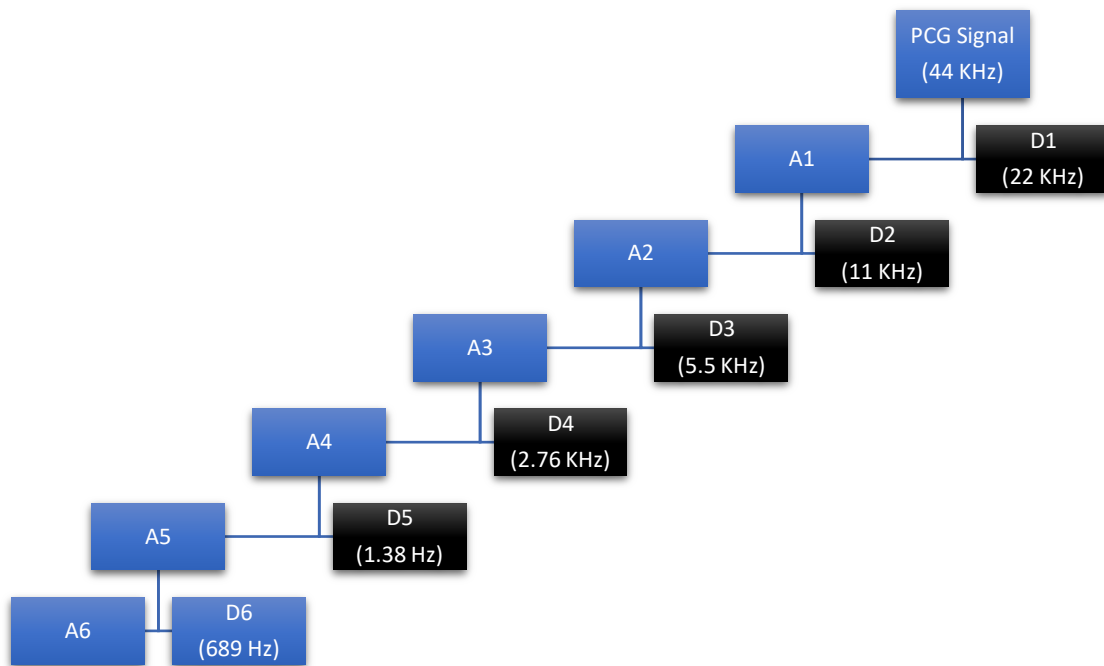


Figure 18: DWT decomposition for the PCG signal

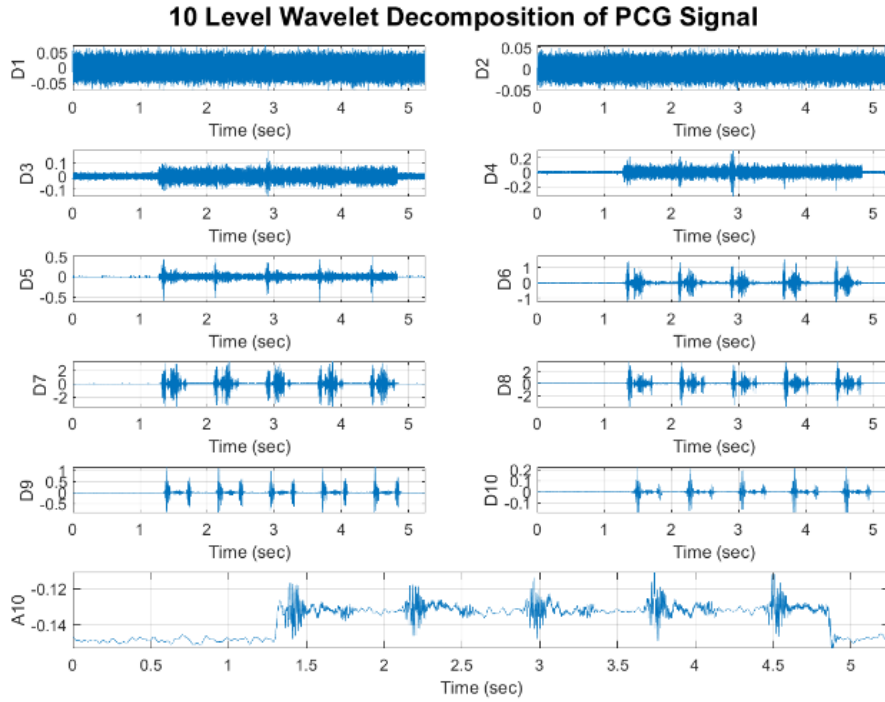


Figure 19: 10 level DWT decomposition for the PCG signal

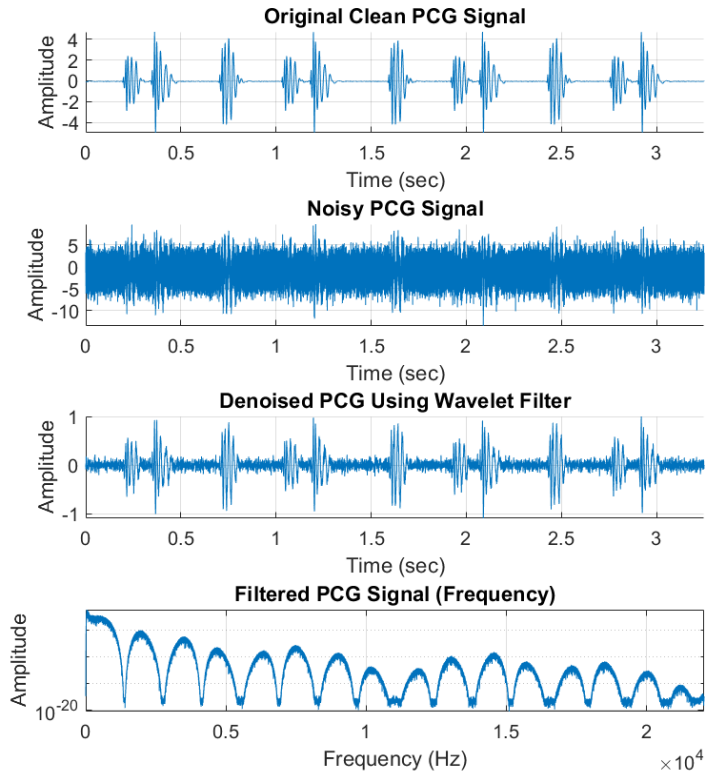


Figure 20: DWT filter for the PCG signal

DWT Filter with 20 and 500 Hz selected bands and Daubechies 6 (db6) as a mother wavelet is designed to denoise the PCG signals. Although the ripples can be seen in the Frequency-

Domain version of the signal, their magnitude is negligible. The signal was greatly attenuated as shown in Figure 20, and the noise in the PCG signal was successfully filtered. From the clean data for each class, 10 samples were randomly chosen. The average MSE attribute was calculated for the DWT Filter comparing the base clean data to its data after adding White Gaussian Noise (WGN) and artifacts for various classes. The MSE results of the DWT filter are given in Table 3.

Wavelet Denoising						
Filter	Normal	S3	AS	MR	MS	MVP
Test 1	0.0129	0.0091	0.0165	0.0108	0.02	0.0153
Test 2	0.0144	0.0132	0.0278	0.0519	0.0259	0.0135
Test 3	0.0162	0.0162	0.0136	0.0137	0.0373	0.0579
Test 4	0.0123	0.014	0.0297	0.0147	0.025	0.0163
Test 5	0.0089	0.0602	0.0744	0.0756	0.038	0.0445
Test 6	0.0137	0.0173	0.0137	0.0149	0.0383	0.0091
Test 7	0.0407	0.0331	0.0959	0.0156	0.0266	0.0257
Test 8	0.1151	0.0281	0.0179	0.0224	0.0183	0.0184
Test 9	0.0109	0.0323	0.0138	0.0147	0.0575	0.0189
Test 10	0.0824	0.0338	0.0169	0.0465	0.0378	0.0208
Average	0.03275	0.0257	0.03202	0.0281	0.0325	0.024

Table 3: MSE results for DWT filter

3.3.3 Empirical Mode Decomposition filter

10 IMFs with relative energy greater than 25% were selected for reconstructing the denoised version of the PCG signal were chosen for the EMD filter's configuration. Based on numerous tests on the PCG signal that demonstrate the best outcomes of signal denoising, the ratio of 25% was chosen for the EMD filter. The decomposed harmonics disturbance signal is shown in Figure 21. Although Figure 22 below shows that the EMD Filter performed badly with high noise data, it was shown to work effectively with low noise data. From the clean data for each class, 10 samples were randomly chosen. The average MSE attribute was calculated for the EMD Filter comparing the base clean data to its data after adding White Gaussian Noise (WGN) and artifacts for various classes. The MSE results of the EMD filter are given in Table 4.

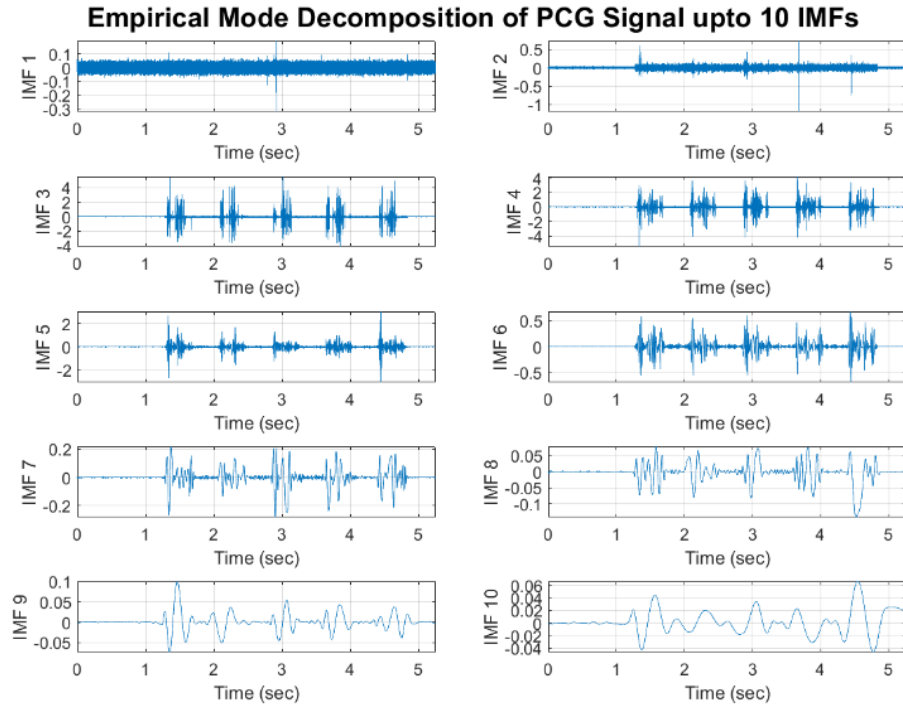


Figure 21: Intrinsic Mode Functions of noisy PCG signal

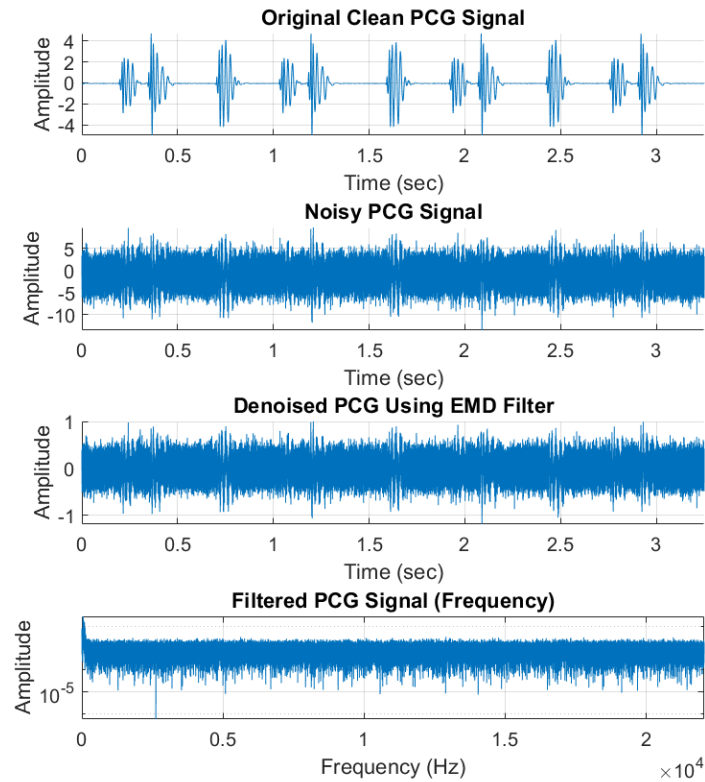


Figure 22: Denoised PCG using EMD filter

EMD Denoising						
Filter	Normal	S3	AS	MR	MS	MVP
Test 1	0.0253	0.0839	0.0735	0.0639	0.0673	0.1821
Test 2	0.0385	0.0756	0.0747	0.0519	0.1276	0.0776
Test 3	0.0426	0.0491	0.0796	0.0794	0.1551	0.1566
Test 4	0.0271	0.0499	0.1045	0.0778	0.0837	0.0419
Test 5	0.0813	0.1299	0.101	0.0838	0.0287	0.1378
Test 6	0.0132	0.0479	0.1277	0.1089	0.0855	0.0193
Test 7	0.0794	0.0689	0.1019	0.0728	0.1181	0.0693
Test 8	0.1521	0.1007	0.1067	0.0383	0.1393	0.0943
Test 9	0.0738	0.0848	0.0396	0.0785	0.111	0.093
Test 10	0.0646	0.0673	0.0602	0.0868	0.0937	0.061
Average	0.05979	0.0758	0.08694	0.0742	0.101	0.0933

Table 4: MSE results for EMD filter

3.3.4 Butterworth filter

Butterworth filter BTW with a bandwidth of 20-500 Hz and order 4 was designed to denoise the PCG signals. The denoised (filtered) version of the PCG signal is shown in Figure 23. The figure below shows the signal attenuates significantly and the noise in the PCG signal was filtered effectively. From the clean data for each class, 10 samples were randomly chosen. The average MSE attribute was calculated for the BTW Filter comparing the base clean data to its data after adding White Gaussian Noise (WGN) and artifacts for various classes. The MSE results of the BTW filter are given in Table 5.

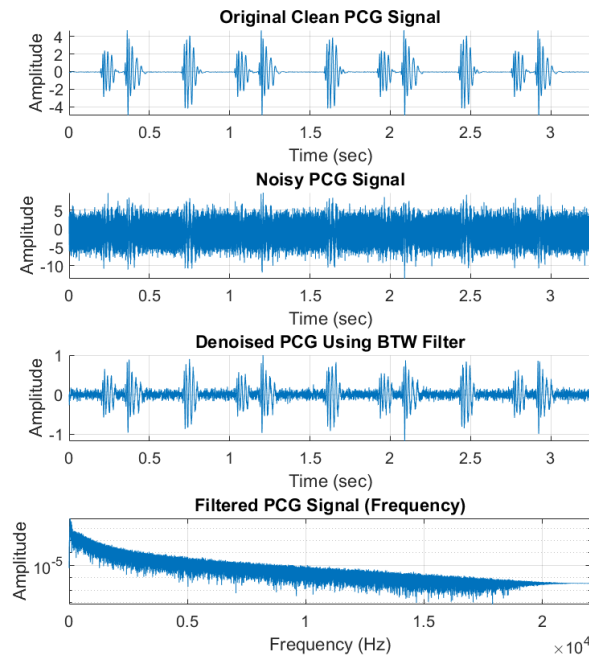


Figure 23: Denoised PCG using BTW filter.

Butterworth Filter						
Filter	Normal	S3	AS	MR	MS	MVP
Test 1	0.033	0.0411	0.0807	0.0563	0.0522	0.068
Test 2	0.0447	0.0756	0.0739	0.0695	0.1122	0.0422
Test 3	0.0388	0.1087	0.0603	0.1122	0.081	0.0926
Test 4	0.0201	0.1169	0.1038	0.0774	0.034	0.031
Test 5	0.0326	0.0699	0.086	0.08	0.0625	0.082
Test 6	0.0923	0.0312	0.1055	0.1096	0.1101	0.0189
Test 7	0.0518	0.1123	0.1332	0.037	0.1123	0.0859
Test 8	0.112	0.0428	0.0429	0.0403	0.0527	0.0407
Test 9	0.0251	0.1165	0.1055	0.0774	0.0699	0.0686
Test 10	0.1125	0.112	0.0367	0.0903	0.0558	0.0714
Average	0.05629	0.0827	0.08285	0.075	0.0743	0.0601

Table 5: MSE results for BTW filter

3.3.5 Chebyshev filter for PCG signal denoising

CHV with a bandwidth of 20-500 Hz and order 4 is used to denoise the PCG signals. The denoised (filtered) version of the PCG signal is shown in Figure 24. The figure below shows the signal attenuates significantly and the noise in the PCG signal was filtered effectively. From the clean data for each class, 10 samples were randomly chosen. The average MSE attribute was calculated for the CHV Filter comparing the base clean data to its data after adding White Gaussian Noise (WGN) and artifacts for various classes. The MSE results of the CHV filter are given in Table 6.

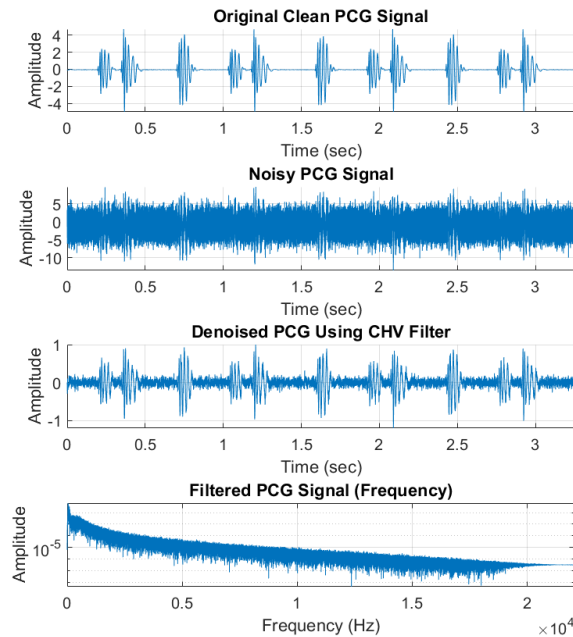


Figure 24: Denoised PCG using Chebyshev

Chebyshev Filter						
Filter	Normal	S3	AS	MR	MS	MVP
Test 1	0.026	0.0313	0.0614	0.0458	0.0431	0.0521
Test 2	0.0337	0.0228	0.0612	0.0568	0.899	0.0384
Test 3	0.0309	0.0935	0.0546	0.1103	0.0724	0.0854
Test 4	0.017	0.995	0.0816	0.0632	0.031	0.0292
Test 5	0.0259	0.0658	0.0779	0.0831	0.0601	0.0715
Test 6	0.0691	0.0296	0.0825	0.0889	0.0966	0.0155
Test 7	0.0488	0.096	0.1302	0.0343	0.09	0.0742
Test 8	0.1312	0.0408	0.0391	0.0389	0.0426	0.0371
Test 9	0.02	0.0958	0.0824	0.0632	0.0654	0.0633
Test 10	0.1081	0.0959	0.0296	0.0808	0.0507	0.0623
Average	0.05107	0.1567	0.07005	0.0665	0.1451	0.0529

Table 6: MSE results for CHV filter

In the upcoming chapter, the process of comparing the results to determine the optimal filter will be elaborated upon, building upon the findings presented in this section.

3.4 Signal segmentation

Signal segmentation is a very crucial step in the signal processing performed on signals that include patterns. The objective of segmentation is to remove the unwanted segments of the signal and to extract only relevant portions from the PCG waveform. 3 techniques were applied in order to extract a single heartbeat from the PCG signal: Segmentation by Wavelet Decomposition using Shannon Energy with Template Correlation (SE-TC), Segmentation based on Heart Sound Envelopogram (HSE), and Segmentation by Template Correlation using Correlation Coefficient (CC-TC). This is shown in Figure 25.

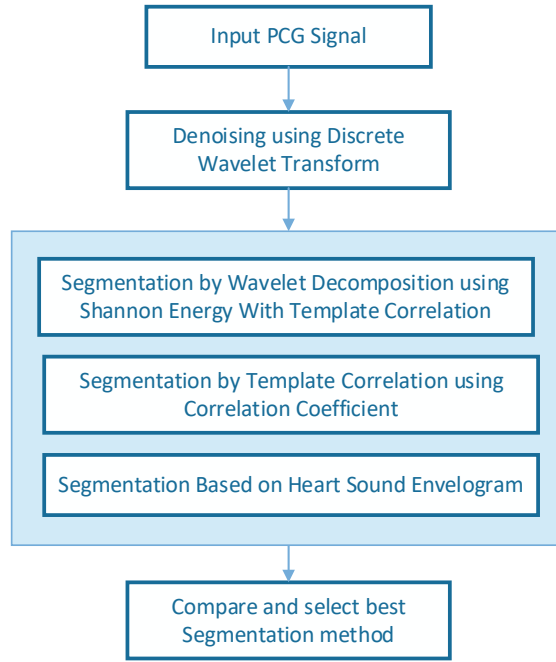


Figure 25: Segmentation method selection plan

A stacked comparison graph is generated for all accessible data to show the segmented section referred to the original data, and then a visual of all data is undertaken to test the operation of these methods and evaluate their effectiveness.

A stacked comparison graph (Figure 26) is generated for all accessible data to display the segmented section referred to the original data, and then a visual rating for all data is undertaken to assess the operation of these algorithms and compare their performance.

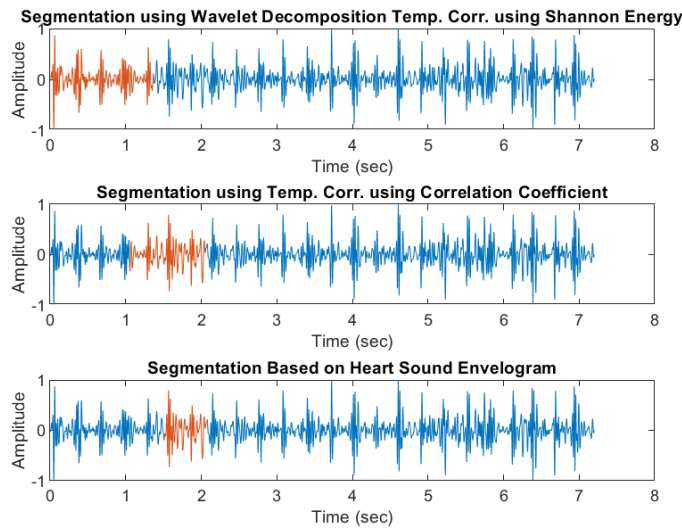


Figure 26: Stacked graph of PCG segmentation

The criteria for measuring the segmentation's effectiveness are listed in Table 7. Table 8 to Table 13 respectively summarize the segmentation findings all 6 PCG classes (Normal, S3, AS, MR, MS, and MPV).

Rating	Grade	Criteria
3	Excellent	detected S1 then S2
2	Good	detected S2 then S1 or long portion
1	Poor	detected short portion or very long portion
0	Not detected	

Table 7: Segmentation evaluation criteria for segmentation methods

	Shannon	Corr.Coeff	Envelopgram
Total	601	601	601
3 scored	60.4%	14.5%	76.9%
2 scored	24.0%	83.0%	20.0%
1 scored	15.1%	2.5%	2.2%
0 scored	0.5%	0.0%	1.0%
Rate	1468	1274	1639

Table 8: Segmentation rating results for normal class

	Shannon	Corr.Coeff	Envelopgram
Total	73	73	73
3 scored	61.6%	34.2%	74.0%
2 scored	32.9%	50.7%	24.7%
1 scored	4.1%	15.1%	1.4%
0 scored	1.4%	0.0%	0.0%
Rate	186	160	199

Table 9: Segmentation rating results for S3 class

	Shannon	Corr.Coeff	Envelopgram
Total	227	227	227
3 scored	80.6%	31.7%	50.2%
2 scored	15.0%	59.5%	47.1%
1 scored	4.4%	8.4%	2.6%
0 scored	0.0%	0.4%	0.0%
Rate	627	505	562

Table 10: Segmentation rating results for AS class

	Shannon	Corr.Coeff	Envelogram
Total	225	225	225
3 scored	72.0%	12.0%	68.4%
2 scored	20.9%	81.8%	28.9%
1 scored	7.1%	6.2%	2.7%
0 scored	0.0%	0.0%	0.0%
Rate	596	463	598

Table 11: Segmentation rating results for MR class

	Shannon	Corr.Coeff	Envelogram
Total	222	222	222
3 scored	31.1%	39.6%	84.7%
2 scored	38.7%	55.4%	8.6%
1 scored	30.2%	5.0%	3.6%
0 scored	0.0%	0.0%	3.2%
Rate	446	521	610

Table 12: Segmentation rating results for MS class

	Shannon	Corr.Coeff	Envelogram
Total	200	200	200
3 scored	60.5%	33.5%	93.5%
2 scored	24.5%	66.5%	5.5%
1 scored	15.0%	0.0%	0.5%
0 scored	0.0%	0.0%	0.5%
Rate	491	467	584

Table 13: Segmentation rating results for MVP class

3.5 Feature extraction:

Feature extraction is an essential part of every machine-learning process. One type of feature is often insufficient to comprehensively represent the PCG signal of all categories, thus it makes the training more challenging. For the feature extraction phase, 3 methods were applied to the PCG signal: Time-Domain Features (TDF), Frequency-Domain Features (FDF), and Mel Frequency Cepstral Coefficients (MFCC) to evaluate each method and select the best-performing feature extraction method for the training process.

3.5.1 Time-domain features

The preprocessed signal's Time-Domain properties, including mean, standard deviation, skewness, kurtosis, peak amplitude, root mean square, shape factor, impulse factor, and

clearance factor, are computed to assess their capacity to identify the existence of murmur. Additionally, the relevance of their statistical significance is assessed [34].

3.5.2 Frequency-Domain features

Frequency-Domain (spectral-domain) features are used to extract the frequency-related relationship of the signals of various categories. These features are extracted by first transforming the Time-Domain signal to the Frequency-Domain using Fast Fourier Transform (FFT) algorithm. From the FFT spectrum, 10 features are extracted, such as Spectral Flux, Spectral Crest, Spectral Flatness, Spectral Skewness, Spectral Centroid, Spectral Kurtosis, Spectral Spread, Spectral Roll Off, Spectral Slope, and Spectral Decrease [13], [35].

3.5.3 Mel-frequency cepstral coefficients (MFCC) features

Preprocessed PCG is first pre-weighted to improve the signal-to-noise ratio before computing MFCC characteristics from it. Using a Hamming window with a 30 ms length and 20 ms overlap, the PCG signals are separated into frames. Each frame's data is transformed using the Fast Fourier Transform (FFT) into its Frequency-Domain representation. Frames that are in the Frequency-Domain are transformed into Mel inverse spectrum domain by passing them through several Mel triangular band-pass filters. The logarithmic Mel spectrum coefficients are then transformed using the discrete cosine transform (DCT). 13 MFCC characteristics were calculated from each frame in this study. A simple and condensed feature representation was obtained by averaging the MFCC feature values over all frames. The MFCC feature vector's total dimensions were 1 x 13.

3.6 Classification experiments

The last stage of the process is the classification phase which categorizes the features extracted from the PCG signal into 6 classes (Normal, Third heart sound S3, Aortic valve stenosis AS, Mitral valve regurgitation MR, Mitral valve stenosis MS, Mitral valve prolapse MVP) through the training and verification process which are set to 70% and 30% respectively. 3 kinds of features were extracted from the PCG signal (Time-Domain Features (TDF), Frequency-Domain Features (FDF), and Mel Frequency Cepstral Coefficients (MFCC)). These were incorporated into the training and verification phase using the following classification techniques: Deep Neural Network (DNN), Support Vector Machine (SVM), K-Nearest

Neighbor (KNN), and Decision Tree (DT) to measure their accuracy and select the best performing method.

The accuracy of the classification is presented as confusion matrixes which is a performance measurement technique for machine learning classification. These measures are identified from the classifier's true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values as shown in Figure 27.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 27: Confusion matrix

3.6.1 K-nearest neighbors

For optimum performance K value of 1 was selected. All other features extracted from beats were standardized. The result confusion matrices for time, frequency, and Cepstral-Domains are shown in Figure 28, Figure 29, and Figure 30 respectively.

		KNN (Time Features)							
True Class	Normal	148	6		12	8	4	83.1%	16.9%
	S3	5	16				1	72.7%	27.3%
	AS	3		50	7	6	2	73.5%	26.5%
	MR	13		10	42	3		61.8%	38.2%
	MS	3		4	10	35	12	54.7%	45.3%
	MVP	7		8	2	9	34	56.7%	43.3%
		82.7%	72.7%	69.4%	57.5%	57.4%	64.2%		
		17.3%	27.3%	30.6%	42.5%	42.6%	35.8%		
		Normal	S3	AS	MR	MS	MVP		
		Predicted Class							

Figure 28: KNN classification with Time-Domain features

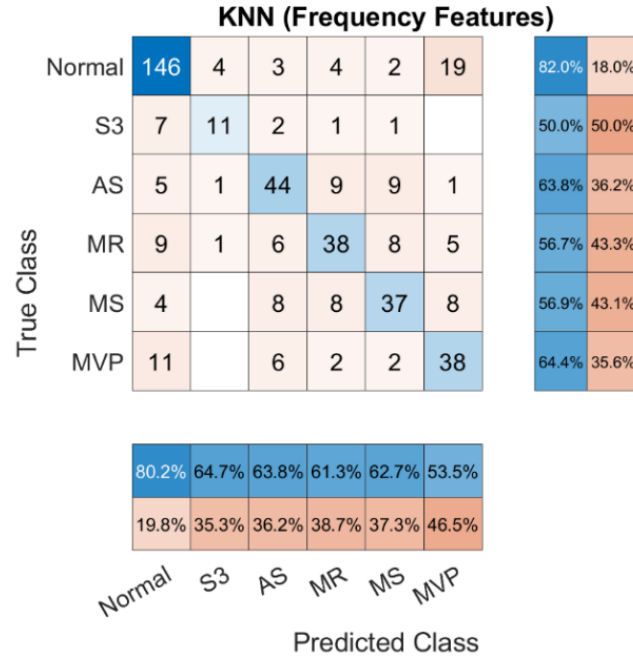


Figure 29: KNN classification with Frequency-Domain features

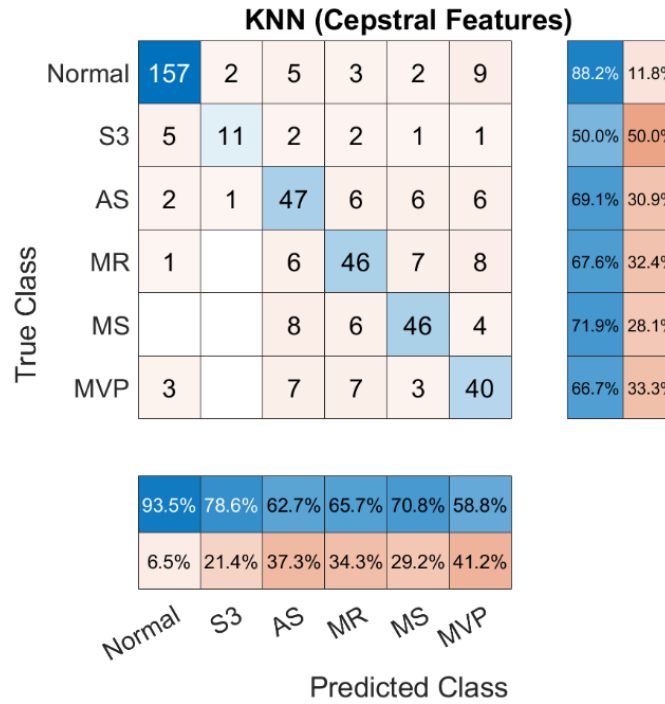


Figure 30: KNN classification with Cepstral-Domain features

3.6.2 Support vector machines

One vs One technique for multiclass SVM was employed. The result confusion matrices for time, frequency, and Cepstral-Domains are shown in Figure 31, Figure 32, and Figure 33 respectively.

		SVM (Time Features)							
True Class	Normal	150	1	4	9	4	11	83.8%	16.2%
	S3	9	8	1		3		38.1%	61.9%
	AS	24		24	8	6	7	34.8%	65.2%
	MR	20		6	25	11	5	37.3%	62.7%
	MS	12		7	6	24	15	37.5%	62.5%
	MVP	18		3	8	8	23	38.3%	61.7%
		64.4%	88.9%	53.3%	44.6%	42.9%	37.7%		
		35.6%	11.1%	46.7%	55.4%	57.1%	62.3%		
		Normal	S3	AS	MR	MS	MVP		
		Predicted Class							

Figure 31: SVM classification with Time-Domain features

		SVM (Frequency Features)							
True Class	Normal	169	7	1	1			94.9%	5.1%
	S3	7	12	2				57.1%	42.9%
	AS	7	1	41	11	1	8	59.4%	40.6%
	MR	5	4	6	31	9	13	45.6%	54.4%
	MS	7	3	6	9	26	13	40.6%	59.4%
	MVP	27		1	12	8	12	20.0%	80.0%
		76.1%	44.4%	71.9%	48.4%	59.1%	26.1%		
		23.9%	55.6%	28.1%	51.6%	40.9%	73.9%		
		Normal	S3	AS	MR	MS	MVP		
		Predicted Class							

Figure 32: SVM classification with Frequency-Domain features

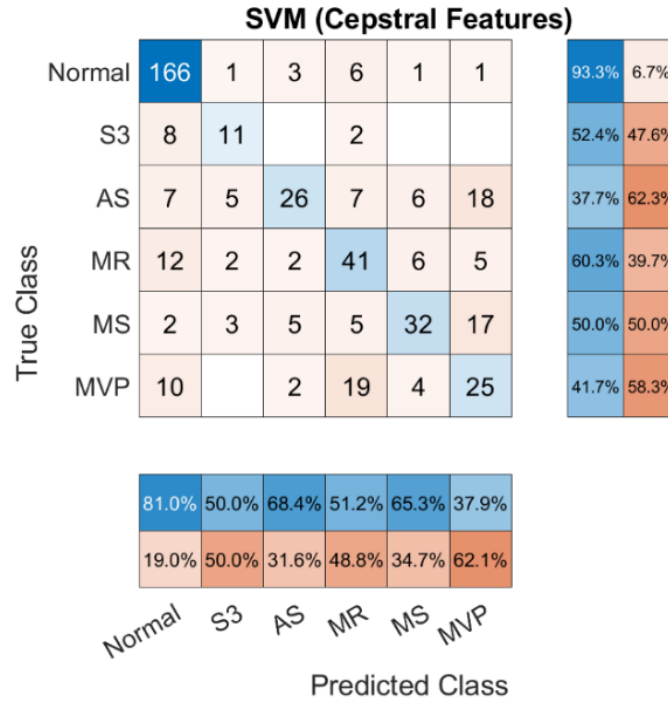


Figure 33: SVM classification with Cepstral-Domain features

3.6.3 Decision tree

The maximum number of splits was set to 100 while developing the decision trees. The result confusion matrices for time, frequency, and Cepstral-Domains are shown in Figure 34, Figure 35, and Figure 36 respectively.

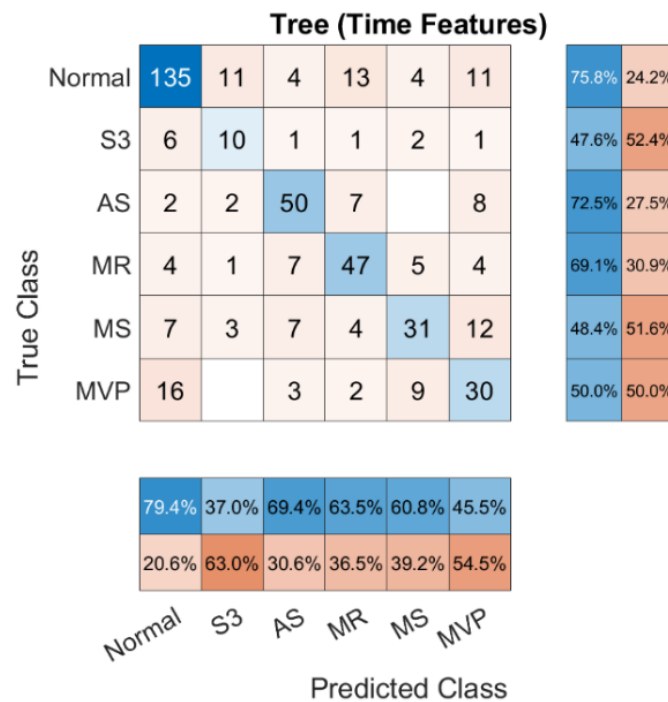


Figure 34: Decision Tree classification with Time-Domain features

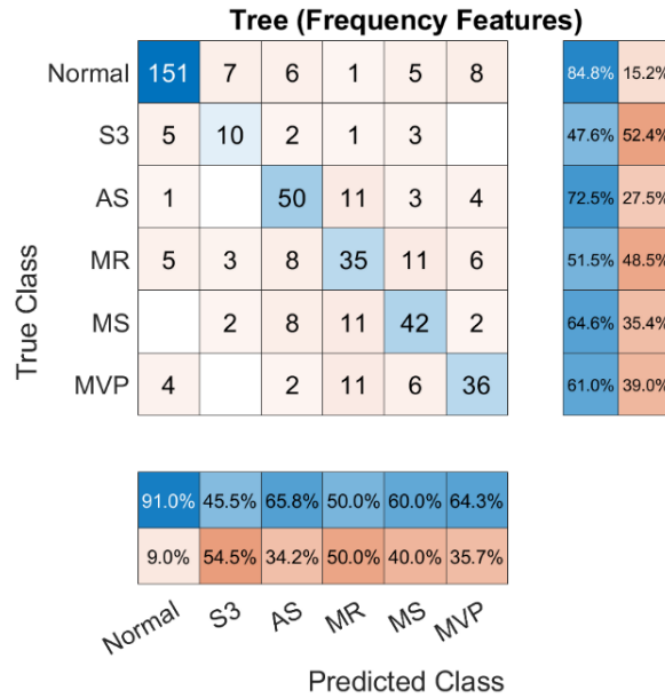


Figure 35: Decision Tree classification with Frequency-Domain features

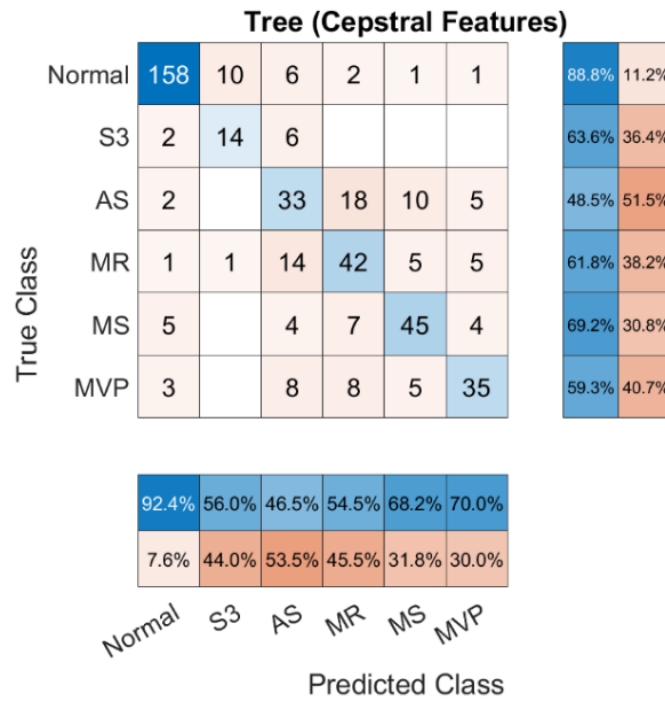


Figure 36: Decision Tree classification with Cepstral-Domain features

3.6.4 Deep neural network

The DNN was configured to 5 layers considering 100 Neurons within each layer, and the activation function of Tanh was employed. The total number of iterations limit is set to 1000, and features were standardized before processing. The result confusion matrices for time,

frequency, and Cepstral-Domains are shown in Figure 37, Figure 38, and Figure 39 respectively.

		DNN (Time Features)							
		Normal	S3	AS	MR	MS	MVP		
True Class	Normal	158	5	4	5	4	2	88.8%	11.2%
	S3	4	16		1			76.2%	23.8%
	AS	5	3	46	4	7	3	67.6%	32.4%
	MR	4	1	6	51	4	2	75.0%	25.0%
	MS	3		2	8	48	4	73.8%	26.2%
	MVP	2		5	4	6	43	71.7%	28.3%
		89.8%	64.0%	73.0%	69.9%	69.6%	79.6%	10.2%	36.0%
		10.2%	36.0%	27.0%	30.1%	30.4%	20.4%		
		Normal	S3	AS	MR	MS	MVP		
		Predicted Class							

Figure 37: ANN classification with Time-Domain features

DNN (Frequency Features)									
True Class	Normal	160	2	2	2	1	12	89.4%	10.6%
	S3	5	12	2	1	2		54.5%	45.5%
	AS		1	47	7	8	5	69.1%	30.9%
	MR	2		11	47	4	3	70.1%	29.9%
	MS	1		12	8	41	2	64.1%	35.9%
	MVP	9		1	3	4	43	71.7%	28.3%
		90.4%	80.0%	62.7%	69.1%	68.3%	66.2%		
		9.6%	20.0%	37.3%	30.9%	31.7%	33.8%		
		Normal	S3	AS	MR	MS	MVP		
Predicted Class									

Figure 38: ANN classification with Frequency-Domain features

		DNN (Cepstral Features)							
True Class	Normal	170	1	1	1	1	5	95.0%	5.0%
	S3	6	15		1			68.2%	31.8%
	AS	1	4	48	9	2	5	69.6%	30.4%
	MR	2	1	10	44	2	8	65.7%	34.3%
	MS	4	2	2	7	46	3	71.9%	28.1%
	MVP	3		5	3	3	45	76.3%	23.7%
		91.4%	65.2%	72.7%	67.7%	85.2%	68.2%		
		8.6%	34.8%	27.3%	32.3%	14.8%	31.8%		
		Normal	S3	AS	MR	MS	MVP		
		Predicted Class							

Figure 39: ANN classification with Cepstral-Domain features

3.7 Deep learning configuration

The findings indicate that, when compared to alternative classification and feature methods, DNN employing Cepstral features provides the highest accuracy for anomaly detection and anomaly classification. The DNN's following parameters were adjusted further by changing them: The activation function, the number of hidden layers (3, 5, and 10), the number of neurons in each hidden layer (10, 50, and 100), and (tanh, sigmoid, and ReLU).

To evaluate the system's effectiveness, extensive system testing and training were carried out, using 70% of the data for training and 30% for testing. The DNN was modified as indicated below:

Stage 1: Select the ideal activation function from (Sigmoid, Tanh, and ReLU) using preset parameters for the number of DNN hidden layers and neurons in each layer (5 layers and 100 neurons within each layer)

Stage 2: Select the number of neurons with fixed activation functions that work well (Tanh performed best in stage 1).

In stage 3, pick the number of hidden layers using the stage 1 and stage 2 parameters that perform best. (100 Neurons and Tanh's activation function)

CHAPTER 4: RESULTS AND RECOMMENDATION

The outcomes of the various approaches used for denoising, segmentation, feature extraction, and classification are now being compared to choose the most effective methodology for each stage, as a result of the experiments conducted in chapter 3.

4.1.1 Filters performance comparison

Using the identical White Gaussian Noise (WGN) and artifacts added to all of the signals, the mean square error (MSE) between the noisy data and its original clean data was calculated to evaluate the performance of the filters. Table 14 lists the MSE for each of the used filters and demonstrates that the DWT produces the least MSE.

	Normal	S3	AS	MR	MS	MVP
IIR	5.63%	7.80%	8.23%	7.50%	6.18%	6.01%
DWT	3.28%	2.57%	3.20%	2.81%	3.25%	2.40%
EMD	5.98%	7.58%	8.69%	7.42%	10.10%	9.33%
BTW	5.63%	8.27%	8.29%	7.50%	7.43%	6.01%
CHV	5.11%	15.67%	7.01%	6.65%	14.51%	5.29%

Table 14: Mean Square Error comparison results for all filters

4.1.2 Segmentation algorithms comparison

The outcomes of the various segmentation methods are contrasted in Table 15 below as each stacked comparison graph of the segmentation procedure shown in section 3.4 was assessed using the criteria listed in Table 7. As can be shown, segmentation based on the heart sound envelopogram algorithm yields the best outcomes when compared to other methodologies.

	Shannon	Corr.Coeff	Envelopogram
Total	1548	1548	1548
3 scored (Excellent)	61.0%	27.6%	74.6%
2 scored (Good)	26.0%	66.1%	22.5%
1 scored (Bad)	12.7%	6.2%	2.2%
0 scored (Error)	0.3%	0.1%	0.8%
Rate	3814	3390	4192

Table 15: Segmentation rating results for all classes

4.2 Classification methods comparison results

There are two objectives for this work:

1. To detect anomalies (differentiate normal from abnormal heartbeat)
2. To classify the anomalies

Considering these objectives, on one hand, the accuracy of anomaly detection can be calculated by averaging the true positive and false negative results for the normal class. On the other hand, anomaly classification accuracy can be calculated by averaging the true positive and false negative for other classes.

The summary table (Table 16) shows that the anomaly detection accuracy and the anomaly classification accuracy for the DNN using Cepstral features are the highest, with 93.2% accuracy for anomaly detection and 71.07% accuracy for anomaly classification.

	True Positive						True Negative						Anomaly Detection Accuracy	Anomaly Classification Accuracy
	Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP		
KNN Time	83.1	72.7	73.5	61.8	54.7	56.7	82.7	72.7	69.4	57.5	57.4	64.2	82.90	64.06
KNN Frequency	82.0	50.0	63.8	56.7	56.9	64.4	80.2	64.7	63.8	61.3	62.7	53.5	81.10	59.78
KNN Cepstral	88.2	50.0	69.1	67.6	71.9	66.7	93.5	78.6	62.7	65.7	70.8	58.8	90.85	66.19
SVM Time	83.8	38.1	34.8	37.3	37.5	38.3	64.4	88.9	53.3	44.6	42.9	37.7	74.10	45.34
SVM Frequency	94.9	57.1	59.4	45.6	40.6	20.0	76.1	44.4	71.9	48.4	59.1	26.1	85.50	47.26
SVM Cepstral	93.3	52.4	37.7	60.3	50.0	41.7	81.0	50.0	68.4	51.2	65.3	37.9	87.15	51.49
DT Time	75.8	47.6	72.5	69.1	48.4	50.0	79.4	37.0	69.4	63.5	60.8	45.5	77.60	56.38
DT Frequency	84.8	47.6	72.5	51.5	64.6	61.0	91.0	45.5	65.8	50.0	60.0	64.3	87.90	58.28
DT Cepstral	88.8	63.6	48.5	61.8	69.2	59.3	92.4	56.0	46.5	54.5	68.2	70.0	90.60	59.76
DNN Time	88.8	76.2	67.6	75.0	73.8	71.7	89.8	64.0	73.0	69.9	69.6	79.6	89.30	72.04
DNN Frequency	89.4	54.5	69.1	70.1	64.1	71.7	90.4	80.0	62.7	69.1	68.3	66.2	89.90	67.58
DNN Cepstral	95.0	68.2	69.6	65.7	71.9	76.3	91.4	65.2	72.7	67.7	85.2	68.2	93.20	71.07

Table 16: Classification results summary

4.3 Deep neural network recommended configuration

As the findings indicate that DNN employing Cepstral features provides the highest accuracy for anomaly detection and anomaly classification. The DNN's following parameters were adjusted further by changing them: The activation function, the number of hidden layers (3, 5, and 10), the number of neurons in each hidden layer (10, 50, and 100), and (tanh, sigmoid, and ReLU). The results show that the Tanh activation function, 5 hidden layers, and 100 Neurons within each layer are the DNN settings that gave the greatest performance for anomaly

detection and classification as given in

Testing Different Activation Methods															
Layers	5														
Neurons	100														
	Training Time (s)	True Positive						True Negative							
		Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP	Anomaly Detection Accuracy	Anomaly Classification Accuracy
Relu	112	92.2%	63.6%	58.8%	65.7%	71.9%	60.0%	92.7%	56.0%	58.0%	62.9%	69.7%	69.2%	92.45%	63.58%
Sigmoid	155	90.5%	66.7%	60.9%	70.1%	75.0%	70.0%	89.5%	60.9%	72.4%	58.0%	77.4%	76.4%	90.00%	68.78%
Tanh	83	93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%	93.30%	72.14%

Testing Different Number of Neurons															
Layers	5														
Activation	Tanh														
	Training Time (s)	True Positive						True Negative							
		Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP	Anomaly Detection Accuracy	Anomaly Classification Accuracy
10 Neurons	35	92.1%	63.6%	73.5%	73.5%	75.4%	66.1%	94.8%	60.9%	60.2%	68.5%	86.0%	76.5%	93.45%	70.42%
50 Neurons	52	90.5%	54.5%	72.1%	68.7%	68.8%	58.3%	92.0%	52.2%	61.3%	63.9%	72.1%	72.9%	91.25%	64.48%
100 Neurons	83	93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%	93.30%	72.14%

Testing Different Number of Hidden Layers with 100 Neurons															
Neurons		100													
Activation		Tanh													
	Training Time (s)	True Positive						True Negative						Anomaly Detection Accuracy	Anomaly Classification Accuracy
		Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP		
		91.0%	50.0%	69.6%	54.4%	73.4%	69.5%	87.6%	52.4%	58.5%	71.2%	75.8%	70.7%		
		93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%		
		89.3%	61.9%	63.2%	72.1%	81.5%	65.0%	88.8%	81.2%	68.3%	64.5%	71.6%	75.0%		
3 Layers	43	91.0%	50.0%	69.6%	54.4%	73.4%	69.5%	87.6%	52.4%	58.5%	71.2%	75.8%	70.7%	89.30%	64.55%
5 Layers	83	93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%	93.30%	72.14%
10 Layers	244	89.3%	61.9%	63.2%	72.1%	81.5%	65.0%	88.8%	81.2%	68.3%	64.5%	71.6%	75.0%	89.05%	70.43%

Table 17. Additionally, 10 simulations were run with this arrangement, and the results of these tests were averaged for better accuracy as given in Table 18. In terms of anomaly detection and classification rates, the results showed average performances of over 92.14% and 71.02%, respectively.

Testing Different Activation Methods															
Layers	5														
Neurons	100														
	Training Time (s)	True Positive						True Negative						Anomaly Detection Accuracy	Anomaly Classification Accuracy
		Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP		
Relu	112	92.2%	63.6%	58.8%	65.7%	71.9%	60.0%	92.7%	56.0%	58.0%	62.9%	69.7%	69.2%	92.45%	63.58%
Sigmoid	155	90.5%	66.7%	60.9%	70.1%	75.0%	70.0%	89.5%	60.9%	72.4%	58.0%	77.4%	76.4%	90.00%	68.78%
Tanh	83	93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%	93.30%	72.14%

Testing Different Number of Neurons															
Layers	5														
Activation	Tanh														
	Training Time (s)	True Positive						True Negative						Anomaly Detection Accuracy	Anomaly Classification Accuracy
		Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP		
10 Neurons	35	92.1%	63.6%	73.5%	73.5%	75.4%	66.1%	94.8%	60.9%	60.2%	68.5%	86.0%	76.5%	93.45%	70.42%
50 Neurons	52	90.5%	54.5%	72.1%	68.7%	68.8%	58.3%	92.0%	52.2%	61.3%	63.9%	72.1%	72.9%	91.25%	64.48%
100 Neurons	83	93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%	93.30%	72.14%

Testing Different Number of Hidden Layers with 100 Neurons															
Neurons	100														
Activation	Tanh														
	Training Time (s)	True Positive						True Negative						Anomaly Detection Accuracy	Anomaly Classification Accuracy
		Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP		
3 Layers	43	91.0%	50.0%	69.6%	54.4%	73.4%	69.5%	87.6%	52.4%	58.5%	71.2%	75.8%	70.7%	89.30%	64.55%
5 Layers	83	93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%	93.30%	72.14%
10 Layers	244	89.3%	61.9%	63.2%	72.1%	81.5%	65.0%	88.8%	81.2%	68.3%	64.5%	71.6%	75.0%	89.05%	70.43%

Table 17: DNN Parameters selection test results

					True Positive						True Negative					
					Normal	S3	AS	MR	MS	MVP	Normal	S3	AS	MR	MS	MVP
Layers	5	Test 01	92.30%	72.32%	93.3%	52.4%	76.5%	63.2%	84.6%	64.4%	91.3%	100.0%	68.4%	70.5%	75.3%	67.9%
Neurons	100	Test 02	92.30%	70.95%	91.0%	77.3%	70.6%	59.7%	80.0%	73.3%	93.6%	65.4%	69.6%	70.2%	67.5%	75.9%
Activation	Tanh	Test 03	93.30%	72.14%	93.3%	59.1%	69.6%	70.1%	83.1%	72.9%	93.3%	76.5%	71.6%	69.1%	74.0%	75.4%
Features	Cepstral	Test 04	91.75%	71.46%	90.4%	81.8%	72.1%	61.2%	83.1%	66.7%	93.1%	62.1%	68.1%	82.0%	74.0%	63.5%
		Test 05	93.20%	71.07%	95.0%	68.2%	69.6%	65.7%	71.9%	76.3%	91.4%	65.2%	72.7%	67.7%	85.2%	68.2%
		Test 06	91.15%	70.61%	93.9%	68.2%	65.2%	67.2%	78.1%	62.7%	88.4%	83.3%	66.2%	72.6%	71.4%	71.2%
		Test 07	91.80%	69.36%	90.5%	52.4%	72.1%	67.6%	81.5%	74.6%	93.1%	57.9%	71.0%	68.7%	76.8%	71.0%
		Test 08	93.25%	71.15%	92.7%	76.2%	60.3%	63.2%	84.6%	76.7%	93.8%	66.7%	68.3%	69.4%	76.4%	69.7%
		Test 09	91.20%	70.51%	92.7%	63.6%	66.7%	76.1%	73.4%	67.8%	89.7%	66.7%	67.6%	70.8%	79.7%	72.7%
		Test 10	91.10%	70.62%	92.1%	63.6%	62.3%	74.6%	73.8%	71.2%	90.1%	77.8%	61.4%	73.5%	71.6%	76.4%
		Average	92.14%	71.02%	92.5%	66.3%	68.5%	66.9%	79.4%	70.7%	91.8%	72.2%	68.5%	71.5%	75.2%	71.2%

Table 18: Multiple testing using best performed DNN parameters

In summary of this comparative study, Figure 40 illustrates the recommended method that best detected and classified the PCG signals for the data set used in this project.

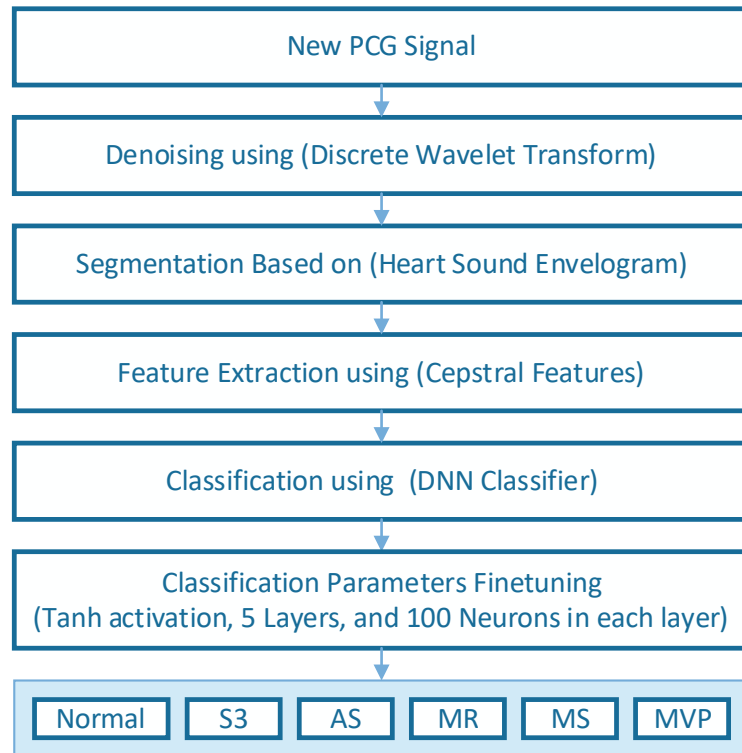


Figure 40: Final proposed methodology

CHAPTER 5: CONCLUSION

5.1 Summary

A comparative study related to various signal processing and AI techniques applied to PCG signals, for the purpose of automatically differentiating normal heart sounds from five types of common murmurs, was provided in this thesis. Five common heart murmurs, the third heart sound (S3), aortic stenosis (AS), mitral regurgitation (MR), mitral stenosis (MS), and mitral valve prolapse (MVP), were the focus of this study that should help to provide a simple-to-use and cost-effective (PCG)-based system for the early detection of cardiovascular anomalies.

The preliminary findings point to average rates for detection and classification of heart anomalies of approximately 92% and 71%, respectively. This was accomplished by utilizing a variety of filtering, segmentation, feature extraction, and AI approaches applied to the PCG signals. It was found that better outcomes were obtained utilizing a 5-layer Deep Neural Network (DNN) with 100 neurons per layer and the Hyperbolic Tangent (tanh) as the activation function.

The best methods for segmenting and denoising the PCG signal, respectively, were the Discrete Wavelet Transform (DWT) and Heart Sound Envelopogram (HSE). The Mel Frequency Cepstral Coefficients (MFCC) Features were utilized to train and execute the DNN since they performed better than their Frequency and Time-Domain equivalents.

While a more expensive option, such as high-quality cellphones or digital electronics stethoscopes, can be utilized to provide higher-quality PCG signals, a low-cost but noisy modified conventional stethoscope was employed as the input device in this study. As the COVID limits are altered, more data about a larger variety of heart murmurs should be collected thanks to easier access to heart patients in hospitals. The suggested system is anticipated to perform better with extra data and further modification of the DNN parameters, to classify more heart murmurs with higher accuracy.

The proposed system's success in creating early warnings and preliminary indications to potentially ill people and directing them to specialized cardiologists for more precise diagnostics—although it was not intended to be a primary tool for the diagnosis—was demonstrated by its relatively high detection rate. In the context of smart and preventive public health systems, this should be a useful home-care tool.

5.2 Future points of expansion

This research was carried out between 2021 and 2022 during the Covid-9 epidemic, which prevented us from speaking with heart patients directly in the hospitals and forced us to rely primarily on internet data, along with some information supplied by our institution from earlier initiatives. This application will have a chance to be more accurate in its training and validation as well as its prediction in future works by having the opportunity to obtain a larger dataset through direct measurement of the patients in hospitals and receiving the confirmation from the specialized doctors for their classification. Also, as mentioned earlier, there were only 6 classes in the dataset that was available. This program can grow and produce more accurate findings by adding new classes of cardiac abnormalities in the future. Furthermore, a lot of other helpful features, such as IOT and cloud computing, may be included to improve the application's usability and process efficiency. With the increasing affordability of higher-quality smartphones, the system is expected to be transformed into a portable device that is easily accessible to everyone.

APENDIX A: MATLAB CODE

The purpose of this appendix is to present the coding efforts in this thesis. Due to the very large size of the custom-written Matlab code, the main structure is highlighted without providing the fine details, for the sake of conciseness.

1. Filters Comparison

```
clear all
close all
clc

%% Read PCG File
disp('Pick an PCG Data File from the Database please.....')
[x,fs] = ReadWave(); % Import PCG data from the loaded file

%% Raw PCG Acquisition
x = 5*x(:,1)/max(abs(x(:,1))); % Normalization
t = (0:length(x)-1)/fs; % Defining the Time Axis in seconds
X = abs(2/length(x)*fft(x)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr1 = linspace(0,fs/2,length(x)/2); % Defining the Frequency
Axis in Hz

%% Adding Noise (Artifact and White noise to PCG data) in
Raw PCG
nx = AddNoise(x,fs); % Adding Noise (Artifact and White
noise to PCG data)in Raw PCG
NX = abs(2/length(nx)*fft(nx)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr2 = linspace(0,fs/2,length(nx)/2); % Defining the
Frequency Axis in Hz

%% Denoising using Simple Bandpass Filter
BFx = BandpassFilter(nx,fs); % Denoising using Simple
Bandpass Filter
BFX = abs(2/length(BFx)*fft(BFx)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr3 = linspace(0,fs/2,length(BFx)/2); % Defining the
Frequency Axis in Hz

figure
subplot(411)
hold on
plot(t,x)
xlabel('Time (sec)')
```

```

ylabel('Amplitude')
title('Original Clean PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(412)
hold on
plot(t,nx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Noisy PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(413)
hold on
plot(t,BFx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Denoised PCG Using IIR Filter')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(414)
semilogy(fr1,BFX(1:length(BFx)/2))
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Filtered PCG Signal (Frequency)')
grid on
axis tight
set(gca,'FontSize',15)

%% Denoising using Discrete Wavelet Transform
levelForReconstruction = [false, false, false, false, false,
true, true, true, true, true, false];
WLx = WaveletTransform(nx,fs); % Denoising using Discrete
Wavelet Transform
WLX = abs(2/length(WLx)*fft(WLx)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr4 = linspace(0,fs/2,length(WLx)/2); % Defining the
Frequency Axis in Hz

```

```

figure
subplot(411)
hold on
plot(t,x)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Original Clean PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(412)
hold on
plot(t,nx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Noisy PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(413)
hold on
plot(t,WLx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Denoised PCG Using Wavelet Filter')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(414)
semilogy(frl,WLX(1:length(WLx)/2))
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Filtered PCG Signal (Frequency)')
grid on
axis tight
set(gca,'FontSize',15)

%% Denoising using Empirical Mode Decomposition
[EMx,REnergy] = EmpiricalDecomposition (nx); % Denoising
using Empirical Mode Decomposition

```

```

EMX = abs(2/length(EMx)*fft(EMx)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr5 = linspace(0,fs/2,length(EMx)/2); % Defining the
Frequency Axis in Hz

figure
subplot(411)
hold on
plot(t,x)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Original Clean PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(412)
hold on
plot(t,nx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Noisy PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(413)
hold on
plot(t,EMx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Denoised PCG Using EMD Filter')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(414)
semilogy(fr1,EMX(1:length(EMx)/2))
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Filtered PCG Signal (Frequency)')
grid on
axis tight
set(gca,'FontSize',15)

```

```

%% Denoising using Butterworth Filter
BPFx = ButterworthFilter(nx,fs); % Denoising using
Butterworth Filter
BPFx = abs(2/length(BPFx)*fft(BPFx)); % Fast Fourier
Transform (Magnitude Response Coefficients)
fr6 = linspace(0,fs/2,length(BPFx)/2); % Defining the
Frequency Axis in Hz

figure
subplot(411)
hold on
plot(t,x)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Original Clean PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(412)
hold on
plot(t,nx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Noisy PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(413)
hold on
plot(t,BPFx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Denoised PCG Using BTW Filter')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(414)
semilogy(fr1,BPFx(1:length(BPFx)/2))
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Filtered PCG Signal (Frequency)')
grid on
axis tight
set(gca,'FontSize',15)

```

```

%% Denoising using Chebyshev-I Filter
CFx = ChebyshevFilter(nx,fs); % Denoising using Chebyshev-I
Filter
CFX = abs(2/length(CFx)*fft(CFx)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr7 = linspace(0,fs/2,length(CFx)/2); % Defining the
Frequency Axis in Hz

figure
subplot(411)
hold on
plot(t,x)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Original Clean PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(412)
hold on
plot(t,nx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Noisy PCG Signal')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(413)
hold on
plot(t,CFx)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Denoised PCG Using CHV Filter')
grid on
axis tight
set(gca,'FontSize',15)
hold off

subplot(414)
semilogy(fr1,CFX(1:length(CFx)/2))
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Filtered PCG Signal (Frequency)')
grid on
axis tight
set(gca,'FontSize',15)

```

```

%% Calculating the Mean Square Errors
x=x/max(x); % Normalization for Comparison in the end
Err_BF=rms(x-BFx); % RMS Error for Filter based Denoising
Err_WL=rms(x-WLx'); % RMS Error for Wavelet based Denoising
Err_EM=rms(x-EMx); % RMS Error for EMD based Denoising
Err_BF=rms(x-BPFx); % RMS Error for Butterworth Filter based
Denoising
Err_CF=rms(x-CFx); % RMS Error for Chebyshev-I Filter based
Denoising

disp('Root Mean Square Error for Simple Band Filter based
Denoising: ')
disp(Err_BF)
disp('Root Mean Square Error for Wavelet based Denoising: ')
disp(Err_WL)
disp('Root Mean Square Error for EMD based Denoising: ')
disp(Err_EM)
disp('Root Mean Square Error for Butterworth Filter based
Denoising: ')
disp(Err_BF)
disp('Root Mean Square Error for Chebyshev-I Filter based
Denoising: ')
disp(Err_CF)

```

2. Segmentation Comparison

```

clear all
close all
clc

%% Read PCG File
disp('Pick an PCG Data File from the Database please.....')
[x,fs,filename,pathname] = ReadWave(); % Import PCG data
from the loaded file

%% Raw PCG Acquisition
x = 5*x(:,1)/max(abs(x(:,1) )); % Normalization
t = (0:length(x)-1)/fs; % Defining the Time Axis in seconds
X = abs(2/length(x)*fft(x)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr1 = linspace(0,fs/2,length(x)/2); % Defining the Frequency
Axis in Hz

```

```

%% Denoising using Discrete Wavelet Transform
levelForReconstruction = [false, false, false, false, false,
true, true, true, true, true, false];
WLx = WaveletTransform(x,fs); % Denoising using Discrete
Wavelet Transform
WLX = abs(2/length(WLx)*fft(WLx)); % Fast Fourier Transform
(Magnitude Response Coefficients)
fr4 = linspace(0,fs/2,length(WLx)/2); % Defining the
Frequency Axis in Hz

%% Segmentation using Wavelet Decomposition and
Reconstrurction
[temp,fs_t] = audioread('Template.wav'); % Load the template
and sampling frequency
t=linspace(0, length(WLx)/fs, length(WLx)); %time vector; %
Load the signal and sampling frequency; %load the signal
[beg,en]=segment_hs(WLx, fs, temp, fs_t); % Use the segment
heart sound function
% Save the segment
if beg==0 || en <1
    disp('No signal')
else
    plot(t,WLx);
    hold on;
    plot(t(beg:en),WLx(beg:en));
    title('Segmentation using Wavelet Decomposition and
Reconstrurction')
    legend('Actual signal','extracted segment')
    xlabel('Time (sec)')
    ylabel('Amplitude')
hold off
    [folder, baseFileName, extension] =
fileparts(string(filename)); %Know the name of the file
withouth the .wav
    exportgraphics(gca, strcat('./Figg/',baseFileName,
'.png')); %save the plot
    NewFilename=strcat('./Audio/',baseFileName, '.wav');
    audiowrite(NewFilename,WLx(beg:en),fs); %save the audio
end

```

3. Feature Extraction

```
clear all; close all; clc;
%% Load the template and sampling frequency
[temp,fs_t] = audioread('Template.wav'); % Load the template
and sampling frequency

%% Get names of all folders
Data_Path='Data\';
p = struct2cell(dir('Data'));
foldernames=p(1,3:end);
folders = strcat(Data_Path,foldernames);
fprintf('Get files names \n')

TF=[];
FF=[];
CF=[];
% for i=1:1
for i=1:length(folders)
    d=strcat(folders(i), '/*.wav'); %Define the extension and
load the files with the extension
    pa=strcat(folders(i)); %Define the path
    addpath(string(pa));
    data = struct2cell(dir(string(d))); %Load the names of
the .wav files
    names = data(1,:);
    figpath =
string(strcat('./Segmentation_Comparison_Figg/',foldernames(
i)));

    if not(isfolder(figpath))
        mkdir(figpath)
    end

%     for j=1:5
for j=1:length(names)
    clc
    fprintf(strcat(string(names(j)), '\n'))

%% Load the signal and sampling frequency
    [x,fs] = audioread(string(names(j))); %load the
signal
    t=linspace(0, length(x)/fs, length(x)); %time vector

    %% Raw PCG Acquisition
    x = 5*x(:,1)/max(abs(x(:,1) )); % Normalization
    t = (0:length(x)-1)/fs; % Defining the Time Axis in
seconds
```

```

        X = abs(2/length(x)*fft(x)); % Fast Fourier
Transform (Magnitude Response Coefficients)
        fr1 = linspace(0,fs/2,length(x)/2); % Defining the
Frequency Axis in Hz

        %% Denoising using Discrete Wavelet Transform
        levelForReconstruction = [false, false, false,
false, false, true, true, true, true, false];
        WLx = WaveletTransform(x,fs); % Denoising using
Discrete Wavelet Transform
        WLX = abs(2/length(WLx)*fft(WLx)); % Fast Fourier
Transform (Magnitude Response Coefficients)
        fr4 = linspace(0,fs/2,length(WLx)/2); % Defining the
Frequency Axis in Hz

        %% Segmentation Based on Heart Sound Envelopogram
try
        [srt,fns,bestbeat]=Envelopogram(WLx,fs);
        PlotEnvogram = 1;
catch
        PlotEnvogram = 0;
        disp('Sigmentation process failed.....')

end

if PlotEnvogram==1
        %% Time Feature Extraction
        TimeFeatures =extractTDFeatures(bestbeat); %
Extract Time Features
        TF=[TF; [TimeFeatures i]]; %Appending Label as
last Column

        %% Frequency Feature Extraction
        FreqFeatures =extractFDFeatures(bestbeat',fs); %
Extract Frequency Features
        FF=[FF; [FreqFeatures i]]; %Appending Label as
last Column

        %% Cepstral Feature Extraction
        CepsFeatures =extractCDFeatures(bestbeat',fs); %
Extract Ceps Features
        CF=[CF; [CepsFeatures i]]; %Appending Label as
last Column

end

```

```

        end

end

%% Saving Features
writematrix(TF,'TimeFeatures.xlsx'); % Create an Excel File
of Features
writematrix(FF,'FrequencyFeatures.xlsx'); % Create an Excel
File of Features
writematrix(CF,'CepstralFeatures.xlsx'); % Create an Excel
File of Features

```

4. Classification Code

```

close all
clear all
clc

FeatT=importdata('TimeFeatures.xlsx');
FeatF=importdata('FrequencyFeatures.xlsx');
FeatC=importdata('CepstralFeatures.xlsx');

%% KNN Classification
[trainedClassifierT_KNN, validationAccuracyT_KNN] =
trainClassifierT_KNN(FeatT);
str=['System Accuracy of KNN on Time Domain Features:
',num2str(validationAccuracyT_KNN*100),'%'];
disp(str)

[trainedClassifierF_KNN, validationAccuracyF_KNN] =
trainClassifierF_KNN(FeatF);
str=['System Accuracy of KNN on Frequency Domain Features:
',num2str(validationAccuracyF_KNN*100),'%'];
disp(str)

[trainedClassifierC_KNN, validationAccuracyC_KNN] =
trainClassifierC_KNN(FeatC);
str=['System Accuracy of KNN on Cepstral Features:
',num2str(validationAccuracyC_KNN*100),'%'];
disp(str)

```

```

%% Tree Classification
[trainedClassifierT_Tree, validationAccuracyT_Tree] =
trainClassifierT_Tree(FeatT);
str=['System Accuracy of Tree on Time Domain Features:
',num2str(validationAccuracyT_Tree*100),'%'];
disp(str)

[trainedClassifierF_Tree, validationAccuracyF_Tree] =
trainClassifierF_Tree(FeatF);
str=['System Accuracy of Tree on Frequency Domain Features:
',num2str(validationAccuracyF_Tree*100),'%'];
disp(str)

[trainedClassifierC_Tree, validationAccuracyC_Tree] =
trainClassifierC_Tree(FeatC);
str=['System Accuracy of Tree on Cepstral Features:
',num2str(validationAccuracyC_Tree*100),'%'];
disp(str)

%% ANN Classification
[trainedClassifierT_ANN, validationAccuracyT_ANN] =
trainClassifierT_WNN(FeatT);
str=['System Accuracy of ANN on Time Domain Features:
',num2str(validationAccuracyT_ANN*100),'%'];
disp(str)

[trainedClassifierF_ANN, validationAccuracyF_ANN] =
trainClassifierF_WNN(FeatF);
str=['System Accuracy of ANN on Frequency Domain Features:
',num2str(validationAccuracyF_ANN*100),'%'];
disp(str)

[trainedClassifierC_ANN, validationAccuracyC_ANN] =
trainClassifierC_WNN(FeatC);
str=['System Accuracy of ANN on Cepstral Features:
',num2str(validationAccuracyC_ANN*100),'%'];
disp(str)
save('trainedClassifier.mat','trainedClassifierC_ANN') %
Saving the Trained Classifier

%% SVM Classification
[trainedClassifierT_SVM, validationAccuracyT_SVM] =
trainClassifierT_SVM(FeatT);
str=['System Accuracy of SVM on Time Domain Features:
',num2str(validationAccuracyT_SVM*100),'%'];
disp(str)

```

```

[trainedClassifierF_SVM, validationAccuracyF_SVM] =
trainClassifierF_SVM(FeatF);
str=['System Accuracy of SVM on Frequency Domain Features:
',num2str(validationAccuracyF_SVM*100),'%'];
disp(str)

[trainedClassifierC_SVM, validationAccuracyC_SVM] =
trainClassifierC_SVM(FeatC);
str=['System Accuracy of SVM on Cepstral Features:
',num2str(validationAccuracyC_SVM*100),'%'];
disp(str)

```

5. Recommended Methodology

```

clear all; close all; clc;
%% Load the template and sampling frequency
[temp,fs_t] = audioread('Template.wav'); % Load the template
and sampling frequency

%% Get names of all folders
Data_Path='Data\';
p = struct2cell(dir('Data'));
foldernames=p(1,3:end);
folders = strcat(Data_Path,foldernames);
fprintf('Get files names \n')

CF=[];
% for i=1:1
for i=1:length(folders)
    d=strcat(folders(i),'/*.wav'); %Define the extension and
load the files with the extension
    pa=strcat(folders(i)); %Define the path
    addpath(string(pa));
    data = struct2cell(dir(string(d))); %Load the names of
the .wav files
    names = data(1,:);
    figpath =
string(strcat('./Segmentation_Comparison_Figg/',foldernames(
i)));

    if not(isfolder(figpath))
        mkdir(figpath)
    end
end

```

```

%     for j=1:5
for j=1:length(names)
    clc
    fprintf(strcat(string(names(j)),'\n'))
    %% Load the signal and sampling frequency
    [x,fs] = audioread(string(names(j))); %load the
signal
    t=linspace(0, length(x)/fs, length(x)); %time vector

    %% Raw PCG Acquisition
    x = 5*x(:,1)/max(abs(x(:,1) )); % Normalization
    t = (0:length(x)-1)/fs; % Defining the Time Axis in
seconds
    X = abs(2/length(x)*fft(x)); % Fast Fourier
Transform (Magnitude Response Coefficients)
    fr1 = linspace(0,fs/2,length(x)/2); % Defining the
Frequency Axis in Hz

    %% Denoising using Discrete Wavelet Transform
    levelForReconstruction = [false, false, false,
false, false, true, true, true, true, false];
    WLx = WaveletTransform(x,fs); % Denoising using
Discrete Wavelet Transform
    WLX = abs(2/length(WLx)*fft(WLx)); % Fast Fourier
Transform (Magnitude Response Coefficients)
    fr4 = linspace(0,fs/2,length(WLx)/2); % Defining the
Frequency Axis in Hz

    %% Segmentation Based on Heart Sound Envelopgram
    try
        [srt,fns,bestbeat]=Envelopgram(WLx,fs);
        PlotEnvogram = 1;
    catch
        PlotEnvogram = 0;
        disp('Sigmentation process failed.....')

    end

    if PlotEnvogram==1
        %% Cepstral Feature Extraction
        CepsFeatures =extractCDFeatures(bestbeat',fs); %
Extract Ceps Features
        CF=[CF; [CepsFeatures i]]; %Appending Label as
last Column
    end
end
end

```

```

%% Saving Features
writematrix(CF, 'CepstralFeatures.xlsx'); % Create an Excel
File of Features

% CF=importdata('CepstralFeatures.xlsx');
%% ANN Classification
[trainedClassifierC_ANN, validationAccuracyC_ANN] =
trainClassifierC_WNN(CF);
str=['System Accuracy of ANN on Cepstral Features:
',num2str(validationAccuracyC_ANN*100),'%'];
disp(str)
save('trainedClassifier.mat','trainedClassifierC_ANN') %
Saving the Trained Classifier

```

6. Testing code

```

clear all
close all
clc

%% Configure recording device
% daq.getVendors() %to know available drivers
% info = audiodevinfo
daq.createSession ('directsound'); % configure microphone
fs = 8000;
bit = 16;
recObj = audiorecorder;
audiorecorder(fs,bit,1,0); % 16 bits on device 0
disp('Start speaking.')
recordblocking(recObj, 10); %record from microphone for 7
seconds
disp('End of Recording. ');
disp('Processing Recording.....');
% play(recObj);
x = getaudiodata(recObj);

%% Saving File
pathname = 'Data\Recording';
filename = 'Record_00.wav';
FileName = FileAutoNumber (pathname,filename);
audiowrite(FileName,x,fs)
%% Apply prediction function
PredictFunction(x,fs)

```

```

function []=PredictFunction(x,fs)
    %% Raw PCG Acquisition
    x = 5*x(:,1)/max(abs(x(:,1) )); % Normalization
    t = (0:length(x)-1)/fs; % Defining the Time Axis in
seconds

    %% Denoising using Discrete Wavelet Transform
    levelForReconstruction = [false, false, false, false,
false, true, true, true, true, true, false];
    WLx = WaveletTransform(x,fs); % Denoising using Discrete
Wavelet Transform
    %% Segmentation Based on Heart Sound Envelopogram
    try
        [srt,fns,bestbeat]=Envelopogram(WLx,fs);
        PlotEnvogram = 1;
    catch
        PlotEnvogram = 0;
        disp('Sigmentation process failed.....')

    end

    if PlotEnvogram==1
        %%Plot Segmentation
        plot(t,WLx);
        hold on;
        plot(t(srt:fns),WLx(srt:fns));
        title('Segmentation Based on Heart Sound
Envelopogram')
        xlabel('Time (sec)')
        ylabel('Amplitude')
        hold off
        %% Cepstral Feature Extraction
        CepsFeatures =extractCDFeatures(bestbeat',fs); %
Extract Ceps Features
        %% Prediction
        TC=importdata('trainedClassifier.mat'); % Load the
Trained Classifier
        Pred = TC.predictFcn(CepsFeatures); % Predict the
Class

        if Pred==1
            disp('%%%%%%%%')
            disp('Normal');
            disp('%%%%%%%%')
        elseif Pred==2
            disp('%%%%%%%%')
            disp('S3');
            disp('%%%%%%%%')
        end
    end
end

```

```

elseif Pred==3
    disp('%%%%%%%%')
    disp('Aortic Stenosis');
    disp('%%%%%%%%')
elseif Pred==4
    disp('%%%%%%%%')
    disp('Mitral Regurgitation');
    disp('%%%%%%%%')
elseif Pred==5
    disp('%%%%%%%%')
    disp('Mitral Stenosis');
    disp('%%%%%%%%')
elseif Pred==6
    disp('%%%%%%%%')
    disp('Mitral Valve Prolapse');
    disp('%%%%%%%%')
end
end
end

```

References

- [1] P. Hult, T. Fjällbrant, K. Hildén, U. Dahlström, B. Wranne, and P. Ask, “Detection of the third heart sound using a tailored wavelet approach: Method verification,” *Med Biol Eng Comput*, vol. 43, no. 2, 2005, doi: 10.1007/BF02345957.
- [2] M. E. Tavel, “Cardiac auscultation: A glorious past - But does it have a future?,” *Circulation*, vol. 93, no. 6. 1996. doi: 10.1161/01.CIR.93.6.1250.
- [3] F. Agrafioti and D. Hatzinakos, “ECG based recognition using second order statistics,” in *Proceedings of the 6th Annual Communication Networks and Services Research Conference, CNSR 2008*, 2008. doi: 10.1109/CNSR.2008.38.
- [4] S. Ismail, I. Siddiqi, and U. Akram, “Localization and classification of heart beats in phonocardiography signals —a comprehensive review,” *Eurasip Journal on Advances in Signal Processing*, vol. 2018, no. 1. 2018. doi: 10.1186/s13634-018-0545-9.
- [5] R. Oshana, “Overview of Digital Signal Processing Algorithms,” in *DSP Software Development Techniques for Embedded and Real-Time Systems*, 2006. doi: 10.1016/b978-075067759-2/50006-5.
- [6] O. Deperlioglu, “Segmentation of Heart Sounds by Re-Sampled Signal Energy Method,” *Brain (Bacau)*, vol. 9, no. 1, 2018.
- [7] L. H. Cherif, M. Mostafi, and S. M. Debbal, “Digitals filters in heart sound analysis,” *International Journal of clinical Medicine Research*, vol. 1, no. 3, 2014.
- [8] K. Iqtidar, U. Qamar, S. Aziz, and M. U. Khan, “Phonocardiogram signal analysis for classification of Coronary Artery Diseases using MFCC and 1D adaptive local ternary patterns,” *Comput Biol Med*, vol. 138, 2021, doi: 10.1016/j.compbiomed.2021.104926.
- [9] A. W. Galli, G. T. Heydt, and P. F. Ribeiro, “Exploring the power of wavelet analysis,” *IEEE Computer Applications in Power*, vol. 9, no. 4, 1996, doi: 10.1109/67.539845.
- [10] N. Kouras, D. Boutana, and M. Benidir, “Wavelet based segmentation and time-frequency characterisation of some abnormal heart sound signals,” in *Proceedings of the International Conference on Microelectronics, ICM*, 2012. doi: 10.1109/ICM.2012.6471392.
- [11] V. N. Varghees and K. I. Ramachandran, “A novel heart sound activity detection framework for automated heart sound analysis,” *Biomed Signal Process Control*, vol. 13, no. 1, 2014, doi: 10.1016/j.bspc.2014.05.002.
- [12] O. K. Fasil and R. Rajesh, “Time-domain exponential energy for epileptic EEG signal classification,” *Neurosci Lett*, vol. 694, 2019, doi: 10.1016/j.neulet.2018.10.062.

- [13] J. Singh and R. S. Anand, "Computer aided analysis of phonocardiogram," *J Med Eng Technol*, vol. 31, no. 5, 2007, doi: 10.1080/03091900500282772.
- [14] H. Basak, A. Roy, J. B. Lahiri, S. Bose, and S. Patra, "SVM and ANN based Classification of EMG signals by using PCA and LDA," Oct. 2021, [Online]. Available: <http://arxiv.org/abs/2110.15279>
- [15] S. A. Pavlopoulos, A. C. H. Stasis, and E. N. Loukis, "A decision tree - Based method for the differential diagnosis of Aortic Stenosis from Mitral Regurgitation using heart sounds," *Biomed Eng Online*, vol. 3, 2004, doi: 10.1186/1475-925X-3-21.
- [16] Ö. Arslan and M. Karhan, "Effect of Hilbert-Huang transform on classification of PCG signals using machine learning," *Journal of King Saud University - Computer and Information Sciences*, 2022, doi: 10.1016/j.jksuci.2021.12.019.
- [17] F. Safara, S. Doraisamy, A. Azman, A. Jantan, and S. Ranga, "Wavelet packet entropy for heart murmurs classification," *Adv Bioinformatics*, vol. 2012, 2012, doi: 10.1155/2012/327269.
- [18] C. N. Gupta, R. Palaniappan, S. Swaminathan, and S. M. Krishnan, "Neural network classification of homomorphic segmented heart sounds," *Applied Soft Computing Journal*, vol. 7, no. 1, 2007, doi: 10.1016/j.asoc.2005.06.006.
- [19] S. Choi, "Detection of valvular heart disorders using wavelet packet decomposition and support vector machine," *Expert Syst Appl*, vol. 35, no. 4, 2008, doi: 10.1016/j.eswa.2007.08.078.
- [20] T. R. Reed, N. E. Reed, and P. Fritzson, "Heart sound analysis for symptom detection and computer-aided diagnosis," in *Simulation Modelling Practice and Theory*, 2004, vol. 12, no. 2. doi: 10.1016/j.simpat.2003.11.005.
- [21] R. Pechetty and L. Nemani, "Additional Heart Sounds—Part 1 (Third and Fourth Heart Sounds)," *Indian J Cardiovasc Dis Women WINCARS*, vol. 5, no. 02, 2020, doi: 10.1055/s-0040-1713828.
- [22] S. McGee, *Evidence-Based Physical Diagnosis: Fourth Edition*. 2017.
- [23] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering*. 2019. doi: 10.1017/9781108380690.
- [24] P. S. Addison, *The illustrated wavelet transform handbook*. 2017.
- [25] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans Pattern Anal Mach Intell*, vol. 11, no. 7, 1989, doi: 10.1109/34.192463.

- [26] A. Anoop Suraj, M. Francis, T. S. Kavya, and T. M. Nirmal, "Discrete wavelet transform based image fusion and de-noising in FPGA," *Journal of Electrical Systems and Information Technology*, vol. 1, no. 1, 2014, doi: 10.1016/j.jesit.2014.03.006.
- [27] M. Bogdan and M. Panu, "LabVIEW modeling and simulation, of the digital filters," in *2015 13th International Conference on Engineering of Modern Electric Systems, EMES 2015*, 2015. doi: 10.1109/EMES.2015.7158411.
- [28] P. Oktivasari, F. Haryanto, S. A. Hamidah, R. Rian-Dini, and S. Suprijadi, "A real-time heart rate signal detection using an electronic stethoscope with labview," *J Biomed Phys Eng*, vol. 10, no. 3, 2020, doi: 10.31661/jbpe.v0i0.1183.
- [29] "https://www.electronics-tutorials.ws/filter/filter_8.html."
- [30] L. Weinberg and P. Slepian, "Takahasi's Results on Tchebycheff and Butterworth Ladder Networks," *IRE Transactions on Circuit Theory*, vol. 7, no. 2, 1960, doi: 10.1109/TCT.1960.1086643.
- [31] A. Rehman, N. A. Saqib, S. M. Danial, and S. H. Ahmed, "ECG based authentication for remote patient monitoring in IoT by wavelets and template matching," in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2018, vol. 2017-November. doi: 10.1109/ICSESS.2017.8342871.
- [32] H. Liang, S. Lukkarinen, and I. Hartimo, "Heart sound segmentation algorithm based on heart sound envelogram," in *Computers in Cardiology*, 1997. doi: 10.1109/cic.1997.647841.
- [33] R. J. Lehner and R. M. Rangayyan, "A Three-Channel Microcomputer System for Segmentation and Characterization of the Phonocardiogram," *IEEE Trans Biomed Eng*, vol. BME-34, no. 6, 1987, doi: 10.1109/TBME.1987.326060.
- [34] P. Careena, M. Mary Synthuja Jain Preetha, and P. Arun, "Detection of murmur from non-stationarity of heart sounds," *Biomedical and Pharmacology Journal*, vol. 12, no. 2, 2019, doi: 10.13005/bpj/1721.
- [35] M. U. Khan, S. Aziz, K. Iqtidar, G. F. Zaher, S. Alghamdi, and M. Gull, "A two-stage classification model integrating feature fusion for coronary artery disease detection and classification," *Multimed Tools Appl*, vol. 81, no. 10, 2022, doi: 10.1007/s11042-021-10805-3.
- [36] P. Careena, M. Mary Synthuja Jain Preetha, and P. Arun, "Significance of Frequency-Domain features of pcg records for murmur detection - An investigation," *Biomedical and Pharmacology Journal*, vol. 13, no. 2, 2020, doi: 10.13005/BPJ/1918.

- [37] Y. M. R. Huang *et al.*, “Advances in Multimedia Information Processing - PCM 2008: Preface,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5353 LNCS. 2008.
- [38] S. Aziz, M. U. Khan, M. Alhaisoni, T. Akram, and M. Altaf, “Phonocardiogram signal processing for automatic diagnosis of congenital heart disorders through fusion of temporal and cepstral features,” *Sensors (Switzerland)*, vol. 20, no. 13, 2020, doi: 10.3390/s20133790.
- [39] B. Kamgar-Parsi and L. N. Kanal, “An improved branch and bound algorithm for computing k-nearest neighbors,” *Pattern Recognit Lett*, vol. 3, no. 1, 1985, doi: 10.1016/0167-8655(85)90036-4.
- [40] “<https://www.mathworks.com/discovery/deep-learning.html>.”
- [41] Y. Khan, “<https://github.com/yaseen21khan/Classification-of-Heart-Sound-Signal-Using-Multiple-Features->.”
- [42] S. Gupta, R. A. Ansari, and D. Sarkar, *Deep learning with R cookbook*.
- [43] P. Bentley, “<http://www.peterjbentley.com/heartchallenge/>.”
- [44] “<https://www.mathworks.com/help/wavelet/gs/choose-a-wavelet.html>.”

ملخص

تقدم هذه الرسالة دراسة مقارنة تتعلق بمختلف تقنيات معالجة الإشارات والذكاء الاصطناعي Artificial Intelligence التي تُطبق على إشارات القلب الصوتية Phonocardiography للتمييز التلقائي بين الأصوات القلبية الطبيعية وخمسة أنواع من أمراض القلب. يمكن لهذه الدراسة المساعدة لتوفير نظام سهل الاستخدام وعالية الفعالية من حيث الجودة والتكلفة للكشف المبكر عن أمراض القلب. كان متوسط معدل الكشف عن أمراض القلب في دراستنا 92.14٪، بينما كان معدل التصنيف 71.02٪. تم تحقيق هذه النتائج باستخدام Deep Neural Network مع خاصية دالة (tanh)، وتكوين 5 طبقات و100 عصب في الطبقة الواحدة. توضح هذه الرسالة أن تقنية Discrete Cosine Transform كانت الخوارزمية الأفضل لإزالة الضوضاء، وكانت Heart Sound Envelopogram هي أفضل طريقة لتقسيم الإشارات القلبية. تفوقت تقنية Mel Frequency Cepstral Coefficients باستخدام طريقة التصنيف Deep Neural Network على نظيريهما في المجال الزمني والتردد. في إطار الرعاية الصحية الذكية والوقائية، يعمل النظام المقترح كأداة رعاية منزلية سهلة بتوفير تحذيرات مبكرة مهمة للأفراد المحتملين المرضى وتوجيههم إلى أطباء القلب لتشخيصات دقيقة أكثر.



جامعة البحرين
كلية الهندسة
قسم الهندسة الكهربائية والإلكترونية

تطبيق معالجة الإشارات وأدوات الذكاء الاصطناعي للكشف المبكر عن أمراض القلب

أطروحة مقدمة لاستيفاء متطلبات درجة ماجستير العلوم في نظام الذكاء الاصطناعي

كتب بواسطة
عباس علي أمر الله حسن

إشراف
د. زهير بحري

30 أبريل 2023