MACHINE LEARNING TECHNIQUES FOR AUTOMATED DETECTION OF CARDIAC ARRHYTHMIAS

by

Vignesh Kalidas

APPROVED BY SUPERVISORY COMMITTEE:

Lakshman S. Tamil, Chair

Mehrdad Nourani

Gopal Gupta

Vincent Ng

This dissertation is dedicated to my
parents, for without their unwavering support,
countless sacrifices and endless prayers,
none of this would have been
even remotely possible.

MACHINE LEARNING TECHNIQUES FOR AUTOMATED DETECTION OF CARDIAC ARRHYTHMIAS

by

VIGNESH KALIDAS, B.TECH, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

May 2020

ACKNOWLEDGMENTS

First and foremost, I would like to extend my sincere and deepest gratitude to my advisor Dr. Lakshman S. Tamil for his guidance and support throughout the course of my doctoral studies. His visionary ideas and passion for conducting high quality research with integrity have been inspirational and the lessons learnt from him are invaluable. I am grateful for his patience and understanding that have been instrumental in helping me complete my doctoral research in a timely and efficient manner. I would also like to thank the other members of my supervisory committee - Dr. Mehrdad Nourani, Dr. Gopal Gupta, Dr. Vincent Ng and the External Chair, Dr. Mario Rotea for their patience, support and constructive feedback about my dissertation and research as a whole. I am also thankful to Timothy Cogan, PhD student in the Department of Electrical Engineering at UTD for his assistance in setting up the AutoECG website. I would also like to thank Dr. Christopher Simmons in the Department of Computer Science for providing access to cloud services required to host the AutoECG website. Finally, I would like to extend my sincere gratitude to the Erik Jonsson School of Engineering and Computer Science for supporting my research with the Jonsson Graduate Study Scholarship during the early years of my research along with providing me with the Graduate Teaching Assistantship throughout the course of my doctoral studies at UTD.

March 2020

MACHINE LEARNING TECHNIQUES FOR AUTOMATED

DETECTION OF CARDIAC ARRHYTHMIAS

Vignesh Kalidas, PhD
The University of Texas at Dallas, 2020

Supervising Professor: Lakshman S. Tamil, Chair

Cardiac Arrhythmias are cardiac abnormalities that arise as a consequence of irregularities in the electrical conduction system of the heart. In this dissertation, a comprehensive set of machine learning techniques, complemented by logical analysis, are presented for accurate detection of fifteen different cardiac arrhythmias - both ventricular and supraventricular. This includes, along with normal sinus rhythm, (1) ventricular fibrillation (VF), (2) ventricular tachycardia (VT), (3) premature ventricular complexes (PVC), (4-6) ventricular bigeminy/trigeminy/quadrigeminy, (7) ventricular couplets, (8) atrial fibrillation, (9) supraventricular ectopic beats (SVEB), (10-12) supraventricular bigeminy/trigeminy/quadrigeminy, (13) supraventricular couplets, (14) supraventricular tachycardia and (15) bradycardia.

In this dissertation, information from single-lead electrocardiogram (ECG) signals is utilized to create a rich set of arrhythmia-specific features to aid in the development of highly accurate arrhythmia detection models. ECG is a waveform representation of the heart's electrical activity and cardiac arrhythmias often manifest as morphological variations on the ECG. Prior to performing any arrhythmia analysis, the incoming ECG signal is pre-processed to remove low frequency and high frequency artifacts using Stationary Wavelet

Transforms and Denoising Convolutional Autoencoders. This is complemented by signal quality assessment using Convolutional Neural Networks where ECG segments corrupted by high grade motion artifacts are identified and suppressed from further arrhythmia analysis. Following this, detection of Ventricular Fibrillation and Sustained Ventricular Tachycardia is implemented using a Random Forests classifier. Next, beat detection using a combination of Convolutional Autoencoders and adaptive thresholding is carried out to accurately detect R-peak locations which is key to performing robust arrhythmia analysis. Subsequently, algorithms for detection of PVC-beat-based ventricular arrhythmias are implemented using Semisupervised Autoencoders combined with Random Forests and logical analysis. This is followed by atrial fibrillation detection using Markov models in conjunction with Random Forests. Finally, logical sequence analysis techniques are applied to detect additional SVEB-based supraventricular arrhythmias.

The algorithms presented in this dissertation achieve a sensitivity of 98.85%, positive predictive value (PPV) of 95.77% and F-Score of 96.82% in detecting ventricular fibrillation/sustained ventricular tachycardia episodes on records from MIT-BIH Malignant Ventricular Ectopy Database and American Heart Association Database. In terms of R-peak detection, 99.63% sensitivity, 99.88% PPV and 99.75% F-Score is achieved on the MIT-BIH Arrhythmia Database (MITDB) records. Following this, the PVC detection algorithm achieves sensitivity, PPV and F-Score values of 93.17%, 94.41% and 93.78% on the MITDB records. Similarly, the SVEB detection algorithm achieves sensitivity, PPV and F-Score values of 92.11%, 83.77% and 87.74% on the MITDB records. In the context of atrial fibrillation detection, a sensitivity of 96.88%, PPV of 98.87% and F-Score of 97.86% is obtained on the MIT-BIH Atrial Fibrillation records.

The working of afore-mentioned algorithms is demonstrated by deploying them in a cloud platform, **AutoECG** - a web service that facilitates online arrhythmia detection

by analyzing ECGs uploaded by authorized users. **AutoECG** is device-agnostic and can process ECG data of varying duration (30s to 24 hours). Following ECG analysis, the **AutoECG** software generates an arrhythmia summary report for further review by qualified medical practitioners. This affirms the translational nature of the research presented in this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The cardiovascular system consists of the heart and a group of blood vessels, that comprises the arteries, veins and capillaries. The cardiovascular system is responsible for regulating blood circulation throughout the human body which is key to carrying out several bodily functions. Efficient blood circulation ensures proper transportation of oxygen as well as nutrients to different parts of the human body along with timely removal of metabolic wastes. Hence proper functioning of the cardiovascular system is essential for survival and conditions that result in abnormal functioning of the cardiovascular system can have serious short-term and long-term consequences on an individual's health and life. The dissertation presents a comprehensive description of research work that focuses on development of accurate algorithms for detection of a particular class of cardiac abnormalities, termed ***Cardiac Arrhythmias***.

Cardiac Arrhythmias are cardiac abnormalities that arise as a consequence of irregularities in the electrical conduction system of the heart. The Sino-Atrial (SA) node, also known as heart's natural pacemaker, initiates electrical impulses that traverse through the cardiac chambers in an ordered manner, resulting in completion of one cardiac cycle i.e., one heartbeat. Any deviation from normal conduction sequence of these electrical impulses, in terms of conduction rate, regularity or excitation by sources other than the SA node, results in cardiac arrhythmias. Cardiac arrhythmias can be short term or persistent, symptomatic or asymptomatic and benign or life-threatening, depending on the type and cause of dysrhythm. More than four million Americans are affected by some form of arrhythmia with increased prevalence among the elderly population. Prevalence of Premature Ventricular Complexes (PVCs) is about 69% in adults over 75 years while atrial Fibrillation is prevalent in about 2.7 million Americans and is expected to reach 12.1

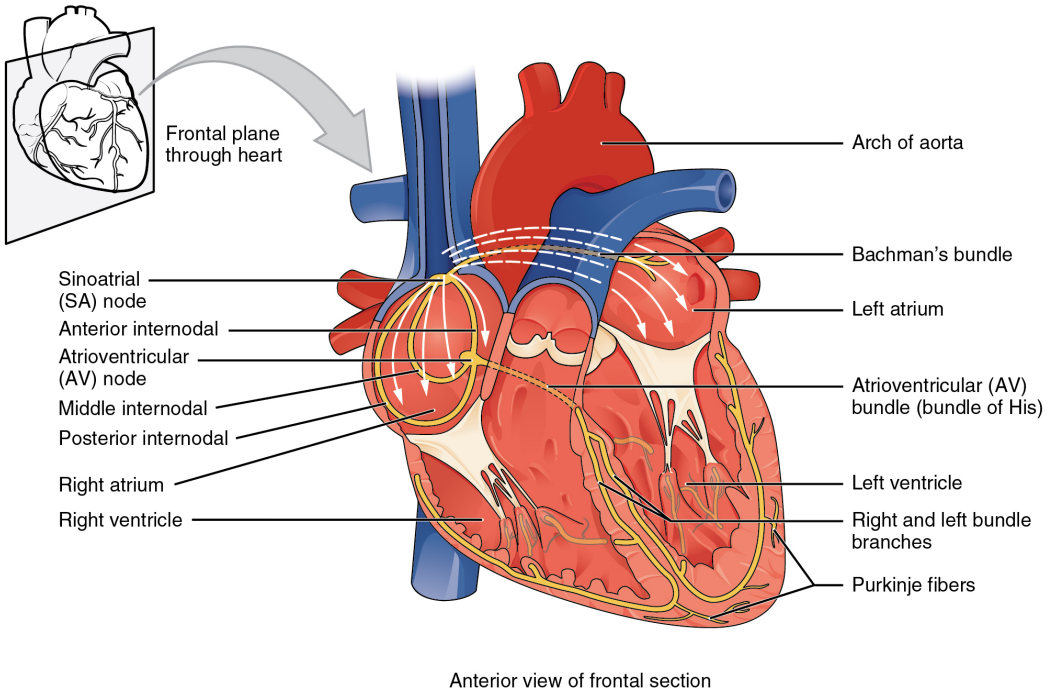

Figure 1.1. Structure of the human heart. (Source: (Betts et al., 2013).)

million by 2030. These arrhythmias are highly correlated with congestive heart failure, stroke, high BP and possible sleep apnea. Oftentimes, delayed diagnosis can increase severity of these arrhythmias and therefore, timely and accurate detection of arrhythmias is vital for patient care and well-being in the long term. The research work explained in this dissertation is a contribution in this direction where a comprehensive set of machine learning techniques, primarily Deep Learning and Random Forests based and complemented by logical analysis techniques, have been developed for accurate detection of fifteen different cardiac arrhythmias in a fully automated manner. Implementing a fully automated system minimizes burden on physicians and helps them prioritize their patients thus allowing them to attend to more patients as well as speeding up diagnosis and treatment. Such systems also have the advantage that they can be easily integrated into telemedicine platforms for remote delivery of healthcare services.

This chapter is organized as follows: Section 1.1 provides a formal description of cardiac functionality. This is followed by Section 1.2 that provides useful information about the arrhythmias that are the primary focus of this dissertation. Section 1.3 describes the generation, components and characteristics of ECG signal while a brief description of the fundamental concepts of machine learning and its use in arrhythmia classification is given in Section 1.4.

1.1 Cardiac Functionality

The human heart is a hollow muscular organ located in the region between the lungs, approximately in the middle of the chest (Aehlert, 2018). The human heart consists of four main chambers: the right atrium, the left atrium, the right ventricle and the left ventricle. The atria (singular, atrium) correspond to the two upper chambers of the heart. De-oxygenated blood from different parts of the body reach the heart via the right atrium while oxygen-rich blood is received by the left atrium from the lungs. Blood flows into the ventricles through the Atrio-Ventricular (AV) valve. The ventricles refer to the two lower chambers of the heart. Pumping blood is the primary responsibility of the ventricles. De-oxygenated blood from the right atrium is pumped back to the lungs via the right ventricle while the oxygenated blood from the left atrium is pumped out to the rest of the body. Figure 1.1 shows the structure of the human heart*.

The heart can be visualized as a pump that is responsible for circulating blood throughout the body. An internal wall, *septum*, acts as a divider between the right and left atria (atrial septum) and similarly between the right and left ventricles (ventricular septum). This enables the heart to function as two pumps, each with its own purpose. Specifically, the right side of the heart (right atrium and right ventricle) takes care of

*Image reproduced/adapted under Creative Commons Attribution License 4.0 license.

pumping de-oxygenated blood (received from rest of the body) to the lungs and carry oxygenated blood from the lungs to the left side of the heart. This action is termed *pulmonary circulation*. The left side of the heart (left atrium and left ventricle) is responsible for pumping oxygenated blood (received from the lungs) to the rest of the body. This is referred to as *systemic circulation*.

The above pumping process, when repeated continuously, results in a *cardiac cycle*. The cardiac cycles consists of both the pulmonary circulation and the systemic circulation. There are two phases for each chamber as part of the cardiac cycle, namely *Systole* and *Diastole*. During systole, the chambers are contracting and blood is pumped out whereas diastole is associated with blood filling in the chambers, indicating a period of relaxation for the chambers. In particular, during ventricular diastole, the right ventricle gets filled with de-oxygenated blood while the left ventricle receives oxygen-rich blood, from right atrium and left atrium respectively. And during ventricular systole, the right ventricle contracts which expels oxygen-deficient blood to the lungs via the pulmonary arteries while the left ventricle's contractile action pumps oxygen-rich blood to the rest of the body through the aorta and its branches. Irregularity of the the afore-mentioned systolic and diastolic processes, as a consequence of abnormalities in cardiac muscles, valves or the conduction system results in cardiac dysfunctions and can have an adverse effect on the efficacy of the heart as a pump. In particular, abnormalities that arise as a consequence of irregularities in the heart's *electrical conduction system* result in ***Cardiac Arrhythmias***, which is the focus of this dissertation.

1.1.1 Electrical Conduction System

Usually, an electrical impulse in a human cell is generated as a consequence of stimulation by a nerve. The heart's electrical conduction system, on the other hand, has specialized

cells called *pacemaker cells* that are capable of generating impulses without requiring nerve stimulation. This property of pacemaker cells to generate electrical impulses without requiring external stimulation is termed *automaticity*. Proper automaticity requires concentrations of Sodium (NA^+), Potassium (K^+) and Calcium (Ca^{2+}) to be maintained at the right levels.

Movement of these ion particles across cell membranes result in changes in voltage, also known as *membrane potential* which is responsible for regulating the heart's functionality as a pump. At rest, smaller potassium ions leak out of a cardiac cell, while large negatively charged proteins remain inside the cell along with phosphates. Hence the inside of the cell is more negatively charged with respect to the outside. This negatively charged state (inside the cell) is known as *Polarization*. Electrical impulse transmission occurs as a consequence of pacemaker cells "firing". This is achieved by flow of electrolytes across the cell membrane. During this phase, positively charged sodium and potassium ions flow inside the cardiac cell resulting in higher positive charge inside the cells with respect to the outside. This flow of ions across the cell membrane is termed *Depolarization*. Depolarization is a pre-requisite for the cardiac muscles (atria and ventricles) to contract which results in expulsion of blood. (It must be noted that muscle contraction is a mechanical event that is dependent on the occurrence of depolarization which is an electrical activity.) On the other hand, *Repolarization* refers to the cardiac cell returning back to its normal/initial state i.e., *polarized state*. This phase involves outflow of positively charged sodium and potassium cells from the cardiac cell, thus making the inside more negative once again.

1.1.2 The Heartbeat

The ***Sino-Atrial (SA) node***, also known as the heart's natural pacemaker, is located in he right atrium just beneath the superior vena cava and is responsible for initiating

a heartbeat. The SA node consists of specialized pacemaker cells and initiates a normal heartbeat by generating an electrical impulse. The SA node has the highest firing rate compared to other pacemaker sites of the heart and thus its status as the heart's natural pacemaker. On average, the SA node has a firing rate of 60 to 100 beats-per-min (bpm). This rate is also termed the average *heart rate*. It must be noted that the average heart rate varies from individual to individual and what may constitute as normal heart rate for one individual may not necessarily be normal for others.

The electrical impulses that are generated by the SA node traverse along the left and right atrial muscles cell by cell in a waveform like pattern that results in atrial depolarization. This makes the atria contract and consequently blood starts getting pumped into the ventricles. In the meantime, the electrical impulses continue traversing the cardiac cells and reach another set of specialized conducting cells, called the ***Atrioventricular (AV) node*** which is responsible for slowing down the conduction rate. This allows the atria to completely empty blood into ventricles, thus ensuring that atrial contraction is completed before ventricular contraction can begin. Following this, the electrical impulses reach the His-Purkinje system and continue downwards to the heart's apex and integrate with ventricular muscle cells resulting in ventricular depolarization. Consequently ventricular contraction occurs and blood is pumped out of the ventricles. Following this, ventricles undergo repolarization which brings them back to normal state and they are ready for the next wave of electrical impulses (originating in the SA node ideally). This whole process constitutes a single heartbeat and keeps repeating forever. The normal periodic occurrence of the heartbeat cycle is known as ***Sinus Rhythm*** and any deviation from sinus rhythm patterns would lead to occurrence of ***arrhythmias***. Figure 1.2 shows the cardiac cycle and corresponding manifestation of electrical activity on an electrocardiogram[†].

[†]Image reproduced/adapted under Creative Commons Attribution License 4.0 license.

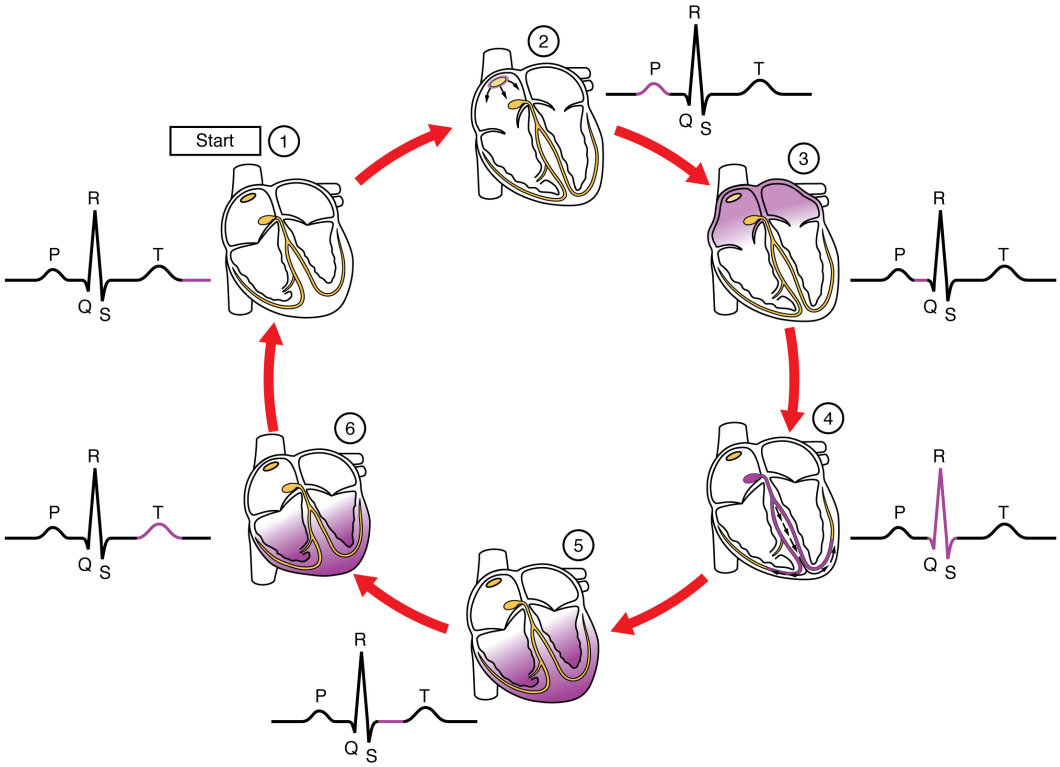

Figure 1.2. Cardiac cycle. (Source: (Betts et al., 2013).)

1.2 Arrhythmias

Any deviation from the afore-mentioned sequence of electrical impulse generation/conduction results in an arrhythmia. This deviation can be due to either (i) abnormalities in impulse conduction and/or (ii) abnormalities in impulse formation. The former category includes blocks of conduction, re-entry of impulses to cells that are already excited in the current cycle, etc. Conduction blocks include fully blocked pathways as well as partial blocks. In partial blocks, the electrical impulses are conducted at a much slower rate than usual. Myocardial infarction and ischemia are often the consequence of blocks. Arrhythmias related to conduction blocks include AV blocks, bradycardia, etc. On the other hand, disorders of impulse formation refer to irregularities that occur as a consequence of irregular/unusual electrical impulse formation. Since the SA node is

the designated (natural) pacemaker, any impulse formation by cells other than the SA pacemaker cells can result in these type of abnormalities. These include ectopic beats such as ectopic ventricular beats, premature atrial complexes, etc.

Arrhythmias that originate in the ventricles are termed **Ventricular Arrhythmias** whereas those that originate above the ventricles are termed **Supraventricular Arrhythmias**. The latter includes both atrial (originating in the atria) and junctional (originating in the AV junction) arrhythmias. A third category of arrhythmias is known as ***Sinus Arrhythmias*** and occurs as a consequence of irregularities in the SA node. In this dissertation, algorithms are developed for accurate detection of fifteen different types of arrhythmias and include all the above three types of dysrhythms. They are as follows:

1. Ventricular Fibrillation and Sustained Ventricular Tachycardia
2. Premature Ventricular Complexes
3. Ventricular Bigeminy
4. Ventricular Trigeminy
5. Ventricular Quadrigeminy
6. Ventricular Couplets
7. Ventricular Runs
8. Atrial Fibrillation
9. Supraventricular Ectopy
10. Supraventricular Bigeminy
11. Supraventricular Trigeminy

12. Supraventricular Quadrigeminy

13. Supraventricular Couplets

14. Supraventricular Runs and Sinus Tachycardia

15. Sinus Bradycardia

The different phases of cardiac electrical activity such as depolarization, repolarization, etc., during normal sinus rhythm as well as in the presence of arrhythmias, can be captured on an ***Electrocardiogram*** (ECG). In fact, ECGs are routinely used for monitoring an individual's cardiac activity either through visual inspection or using computer algorithms and arrhythmias frequently manifest as morphological variations on an ECG.

1.3 Electrocardiogram

The electrocardiogram (ECG) is a graphical recording of the heart's electrical activity and is the most commonly used tool for arrhythmia analysis. The different phases of the heart's electrical activity are represented as different types of waveforms on the ECG and presence of arrhythmias results in morphological changes of these waveforms. Thus ECG signal data contains rich information pertaining to the heart's functionality and provides useful insights about presence of arrhythmias. Apart from helping understand arrhythmias, the ECG also serves various other purposes such as analyzing effects of medications, evaluating cardiac injuries, monitoring heart rate, detecting ischemic damages, etc., among other things.

1.3.1 ECG electrodes

ECGs are obtained using electrodes that are placed in various positions and the tracing of electrical activity between a pair of electrodes is termed a ***lead***. Different lead orientations

help view heart from different perspectives and hence provide information about different cardiac regions. Specifically, the leads help analyze the cardiac electrical activity in two planes, namely, ***frontal*** and ***horizontal*** planes.

As the name suggests, the frontal planar leads look at the heart from the front of the body. there are two types of frontal leads: ***Standard limb leads*** and ***Augmented limb leads***. The standard limb leads include Leads I, II and III while the augmented limb leads include Leads aVR, aVL and aVF. On the other hand, horizontal planar leads view the heart from the top i.e., if the human body were sliced horizontally and hence the name. This includes six chest leads, also called precordial leads, viz. V1, V2, V3, V4, V5 and V6. Each of the twelve lead orientations help capture the heart's electrical activity from a certain specific perspective. In this research, ECG information from ***lead II*** orientation is used as lead II is most ideal for capturing the QRS complex which is the most characteristic and important fiducial marker in an ECG. The QRS complex is the predominantly used ECG waveform feature for heart rate computations and the arrhythmia algorithms developed in this research rely heavily on heart rate metrics, thus rendering lead II as the most logical choice.

1.3.2 Components of an ECG

As mentioned earlier in 1.1.2, different phases of cardiac activity manifest as different waveforms on the ECG. These waveform components primarily comprise:

1. Isoelectric Line
2. P-wave
3. QRS complex
4. T wave

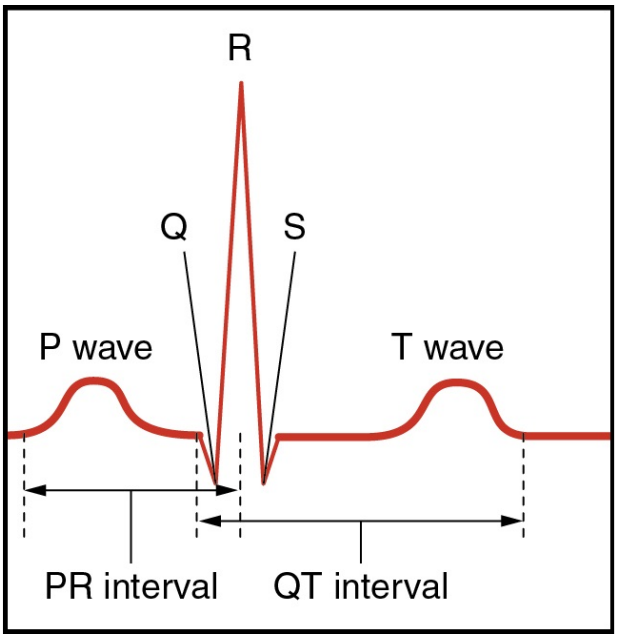

Figure 1.3. Components of an electrocardiogram. (Source: (Betts et al., 2013)).

Figure 1.3 shows the primary components of an electrocardiogram[‡]. The following subsections provide a brief description of the above four waveform types:

Isoelectric line

The isoelectric line or the baseline represents the condition where there is no electrical activity. Deflections above and below the baseline correspond to depolarization activity depending on electrical activity towards or away from the positive electrode. The isoelectric line is used as a reference to determine the amplitude of each wave deflection and can be used to identify unusual amplitude changes such as abnormally-peaked P-waves, inverted T-waves, etc. Oftentimes, the isoelectric line is superimposed with a low frequency component which results in what is known as *baseline wander*. Baseline wander is a low frequency artifact and is explained in more detail in Chapter 2.

[‡]Image reproduced/adapted under Creative Commons Attribution License 4.0 license.

P-wave

The P-wave represents atrial depolarization of the heart. This is the first waveform manifestation of the electrical impulse generated from the SA node. On lead II, this is seen as a positively deflected inverted 'v'. The up-slope (first half) refers to right atrial depolarization and the second half (down-slope) corresponds to left atrial depolarization. The P-wave usually has a smoothened peak and has the lowest amplitude among the ECG fiducial points. Abnormal P-waves can often be a pointer to enlarged atria and unusually tall or pointed P-waves could indicate heart failure, atrial fibrillation, supraventricular ectopy, etc. Inverted P-waves on lead II often correspond to junctional ectopy.

QRS complex

The QRS complex is the most characteristic feature in an ECG and corresponds to the ventricular depolarization phase of the heartbeat cycle. Since ventricles have a much larger muscle area and mass compared to the atria, the QRS complex has a significantly higher magnitude and appears much larger than P-waves on the ECG. Thus, although atrial repolarization and ventricular depolarization occur almost simultaneously, atrial repolarization activity gets hidden under the larger QRS complex and is usually not visible on the ECG. As the name suggests, the QRS complex is a complex that is made up of three waveforms, viz. Q-wave, R-wave and S-wave. The Q-wave and S-wave are negative deflections from the isoelectric line The R-wave is usually the first positive deviation, after the P-wave, from the isoelectric line and the peak location of the R-wave is termed as ***R-peak***. Under normal conditions, the QRS complex is expected to have a duration of 0.11s or less. In reality, the QRS complex has different morphologies depending on the lead from which it is measured. In order to glean useful arrhythmia information using QRS complex information, the lead information must also be provided. For instance, often

times the QRS complex may be negative in certain lead orientations such as leads V, V2, etc., owing to their orientation w.r.t to the net ventricular depolarization vector, but this is not indicative of any arrhythmia on its own. In this research, we use lead II and in lead II, under normal conditions, the QRS complex has a positive R-wave and negative Q- and S-waves. Thus a negative QRS complex in lead II is indicative of a cardiac abnormality and would necessitate further medical investigation.

The R-peak is the most significant ECG fiducial point and is often considered to be representative of a QRS complex as a whole. The distance between consecutive R-peaks, termed as the ***RR-interval*** provides useful insights into underlying heart rate dynamics and is frequently used for evaluating an individual's cardiac health. In this dissertation, the RR-interval information serves as the primary source of ECG information for performing arrhythmia analysis and hence accurate detection of QRS complexes is a pre-requisite for developing robust arrhythmia detection techniques.

T-wave

The T-wave represents ventricular repolarization. It usually has the same deflection as that of the preceding QRS complex and hence on lead II, it appears as a positively deflected wave under normal cardiac activity. The T-wave is not perfectly symmetric with a shallower slope for the first half and a steeper slope for the second half. The offset (end) of T-wave is identified by return to the baseline and indicates end of one cardiac cycle i.e., a single heart beat. In the presence of abnormalities such as premature ventricular complexes, etc., the T-wave has an opposite polarity to that of the QRS complex. Oftentimes, unusually tall T-waves might be indicative of hyperkalemia, which refers to excessive concentration of potassium (K^+) in the blood. Taller T-waves can mimic QRS complexes and can affect the performance of a QRS detector. Hence care must be taken to avoid misclassifying T-waves as QRS complexes.

1.4 Machine learning for ECG analysis

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that involve decision making based by learning from data (Goodfellow et al., 2016). A long-standing definition for what constitutes learning is given in (Mitchell, 1997) as follows: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with E." The term "experience" can be associated with domain-specific data and "Tasks" could refer to performing some decision making such as classification or regression. Finally, the "performance measure" can be any statistical metric such as sensitivity, specificity, accuracy, etc. which evaluate how well the algorithm has learnt to perform its intended task. Machine learning problems can be broadly divided into three categories, namely *Supervised learning*, *Unsupervised learning* and *Reinforcement learning* (Bishop, 2006). Supervised learning is the process of learning a function that maps an input to its target labels. Depending on the type of target labels, the task can be either regression (continuous valued targets) or classification (discrete/categorical valued targets). Some examples of supervised learning algorithms are Logistic Regression, Linear Regression, Support Vector Machines, Decision Trees, Random Forests, Convolutional Neural Network classifiers, etc. (Goodfellow et al., 2016; James et al., 2013; Mitchell, 1997). Unsupervised learning involves learning patterns without the aid of any target labels (Goodfellow et al., 2016). This category includes machine learning techniques clustering, autoencoder-based representation learning (Goodfellow et al., 2016), dimensionality reduction using principal components analysis (Joliffe and Morgan, 1992), visualization techniques such as t-SNE (Maaten and Hinton, 2008), etc. Reinforcement learning consists of learning to make decisions so as to maximize rewards (Goodfellow et al., 2016).

In this dissertation, a combination of supervised learning and unsupervised learning techniques are implemented to perform robust arrhythmia analysis. Autoencoder based

deep learning models (Goodfellow et al., 2016) are used in Chapter 2 through Chapter 4. Section 2.3.1 gives a brief description of the types of autoencoders used in this research. Another supervised learning technique used prominently in this research is Random Forests, which is an ensemble learning approach. Random Forests based arrhythmia detection models are used in the detection of several ventricular (Chapter 4) arrhythmias as well as atrial fibrillation (Chapter 5). A brief introduction about Random Forests is presented in Section 4.3.1. In terms of previously published work in the field of ECG analysis, context-specific literature review is presented in Chapter 3 through Chapter 5. A more generic review of ML techniques employed in the analysis of cardiovascular systems can be found in (Sevakula et al., 2020).

The remainder of this dissertation is organized as follows. Chapter 2 discusses algorithms for signal pre-processing which includes ECG denoising and signal quality analysis. Next, Chapter 3 discusses techniques for performing efficient beat detection. This is followed by description of ventricular arrhythmia detection algorithms in Chapter 4. Subsequently, supraventricular arrhythmia detection algorithms are presented in Chapter 5. Chapter 6 provides a brief discussion regarding a cloud-based arrhythmia detection platform termed **AutoECG**. Finally, the dissertation concludes with Chapter 7 that provides a summary of the topics discussed in this dissertation along with exploring avenues for future research.

References

Aehlert, B. J. (2018). *ECGs Made Easy* (6 ed.). Elsevier.

Betts, G. J., K. A. Young, J. A. Wise, E. Johnson, B. Poe, D. H. Kruse, O. Korol, J. E. Johnson, M. Womble, and P. DeSaix (2013). *Anatomy and Physiology*. OpenStax.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An introduction to statistical learning*, Volume 112. Springer.

Joliffe, I. and B. Morgan (1992). Principal component analysis and exploratory factor analysis. *Statistical methods in medical research 1*(1), 69–95.

Maaten, L. v. d. and G. Hinton (2008). Visualizing data using t-sne. *Journal of machine learning research 9*(Nov), 2579–2605.

Mitchell, T. M. (1997). *Machine Learning* (1 ed.). USA: McGraw-Hill, Inc.

Sevakula, R. K., W.-T. M. Au-Yeung, J. P. Singh, E. K. Heist, E. M. Isselbacher, and A. A. Armoundas (2020). State-of-the-art machine learning techniques aiming to improve patient outcomes pertaining to the cardiovascular system. *Journal of the American Heart Association 9*(4), e013924.

CHAPTER 2

ECG SIGNAL PRE-PROCESSING

2.1 Introduction

The electrocardiogram (ECG) signal acquisition process is subject to interference due to several factors that can often compromise the integrity of the measured signal. This includes patient movements, electrode disconnects, powerline interference, muscle noise, etc. Hence, it is imperative that the ECG signal be pre-processed for noise removal (denoising) and signal quality analysis prior to performing any arrhythmia analysis. This chapter focuses on the techniques developed in this work for this purpose.

There are primarily three types of artifacts that affect the quality of arrhythmia classification or any kind of ECG analysis for that matter. They are:

1. Low frequency noise
2. High frequency noise
3. Electrode Motion(EM) noise

Low frequency and high frequency noise generally occupy frequency bands that do not overlap with QRS complex frequencies and hence can be suppressed without compromising useful QRS information and fall under the category of ECG denoising while EM noise cannot be suppressed without losing valuable ECG information and can only be managed. ECG segments deemed to be too noisy i.e., corrupted to a large extent by EM noise, are omitted from further arrhythmia analysis to mitigate occurrence of false positives. The approaches required to handle the above artifacts vary for each type. The following section provides a brief review of techniques used for handling the afore-mentioned classes of artifacts. This is followed by a comprehensive description of the ECG denoising and noise classification methodologies implemented in this work.

2.2 Low frequency noise suppression

Low frequency noise refers to superposition of low frequency components (usually less than 1Hz) with the ECG signal. This results in wandering/drifting of the isoelectric line. Hence this artifact is termed as baseline wander or baseline drift. Factors causing baseline wander artifacts include patient movement, respiratory factors such as breathing, changes in electrode impedance, etc. Since the frequency range of baseline wander is usually less than 1Hz, suppressing this artifact does not result in loss of any useful information required for analysis of the arrhythmias pertaining to this work. For ensuring reliable baseline wander removal, Stationary Wavelet Transform is used. The technique is explained below.

2.2.1 Stationary Wavelet Transforms

The most commonly used tool for spectral analysis of signals to aid in filtering operations is the Fourier transform. But the Fourier transform assumes the input signal is stationary i.e., all frequency components occur at all time instants. This, therefore, limits the application of Fourier transform to non-stationary or transient signals such as the ECG. Transient signals have different frequency components dominating different sections of the signal and hence require more sophisticated tools to perform efficient spectral analysis. The solution to this is the Discrete Wavelet Transform (DWT). DWT is a signal processing tool that provides efficient time-frequency representation of a signal. The term wavelet refers to a small wave or a template (mother wavelet) which is matched with the input signal. The template is stretched and compressed by a finite factor (usually 2) to analyze the input signal at multiple levels of resolution. Stationary Wavelet Transform (SWT) (Nason and Silverman, 1995) is a variant of DWT where there is dyadic compression in the frequency domain without any downsampling in the time domain. The wavelet coefficients therefore have the same length (duration) as that of the input at each scale which helps reduce

resolution errors at higher scales (lower frequencies). The SWT step can be viewed as a means to compute the effective band-pass for the signal at each scale.

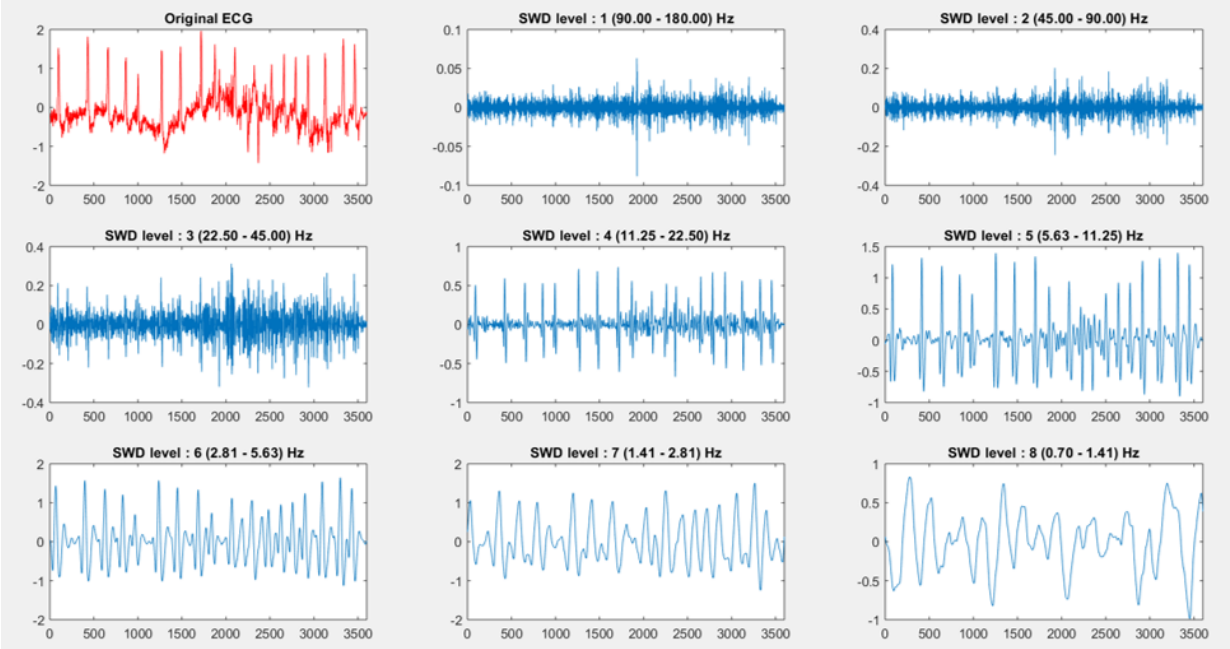

Figure 2.1. SWT decomposition of ECG signal. (SWD - Stationary Wavelet Detail)

2.2.2 Methodology

All the models described in this work have been trained and validated using signals sampled at 360Hz. Therefore the incoming ECG signal is first resampled at 360Hz before performing any task. Following signal resampling, SWT is applied to the signal. In this work, the Daubechies3 (Db3) (Daubechies, 1992) is used as the mother wavelet for computing SWT detail coefficients. Since the signals are resampled at 360Hz, frequency components upto 180Hz can be re-constructed from the SWT detail coefficients, as per Nyquist-Shannon theorem for sampling signals. Nine-level SWT is applied to the resampled signal. Owing to dyadic compression in the frequency domain, this results in the first eight levels corresponding to frequency components in the [0.7Hz to 180Hz] (see Figure 2.1).

Therefore, reconstructing the signal using coefficients from only the first eight levels leads to suppression of all frequencies below 0.7Hz. This results in removal of the major low frequency components that correspond to baseline wander artifacts. Although this does not ensure filtering out of frequencies upto 1Hz, it must be noted that reconstructing the signal using only the first seven detail coefficients leads to suppression of frequencies upto 1.4Hz (as a consequence of dyadic compression) and this could compromise useful ECG information. Hence the first eight levels are retained to perform efficient low frequency noise removal in this work.

2.3 High frequency noise suppression

High frequency (HF) noise refers to corruption of ECG signal by spectral components generally higher than 30Hz. This makes the isoelectric line appear extremely chaotic and can often mask useful ECG fiducial markers such as the P-wave, T-wave, etc. Although the bandwidth of HF noise is outside the useful QRS complex range, it can still result in false beat detection, which can result in unreliable arrhythmia analysis. This becomes especially significant for beat detection methods that adaptively update their thresholds, as the one used in this work (see Chapter 3). Factors causing HF artifacts include powerline interference (50Hz/60Hz), thermal noise, muscle (EMG) artifacts, etc.

Similar to the baseline wander removal approach described in Section 2.2.2, HF noise can, in theory, be removed by suppressing SWT detail coefficients that correspond to frequencies 30Hz and above. This would be equivalent to retaining only the detail coefficients from level-4 to level-8 to reconstruct the HF-noise suppressed signal. But there is an inherent disadvantage to this approach. SWT can be viewed as a sequence of band-pass filtering operations and since a band-pass filter is essentially a cascaded version of high-pass and low-pass filters, the low-pass filtering process (removing level-1 to level-3

SWT detail coefficients), results in mitigation of QRS amplitudes and makes the QRS complexes shallower and wider. In fact, any low-pass filtering technique would give rise to this type of undesired effect on the QRS complex. Although this is not a deterrent to the beat detection process itself, this can often result in a large number of false positives while performing PVC classification (see Chapter 4), as PVC beats are primarily characterized by shallower slopes and wider QRS complexes. Therefore, it is necessary to adopt an adaptive denoising technique that removes HF noise while retaining QRS complex amplitudes and its sharpness. To achieve that, a deep-learning based approach, using Denoising Convolutional Autoencoders, is developed in this work. Section 2.3.1 gives a brief overview of autoencoders including denoising autoencoders (Vincent et al., 2008; Bengio et al., 2013; Goodfellow et al., 2016). Section 2.3.2 then describes the technique developed in this work for HF-noise removal.

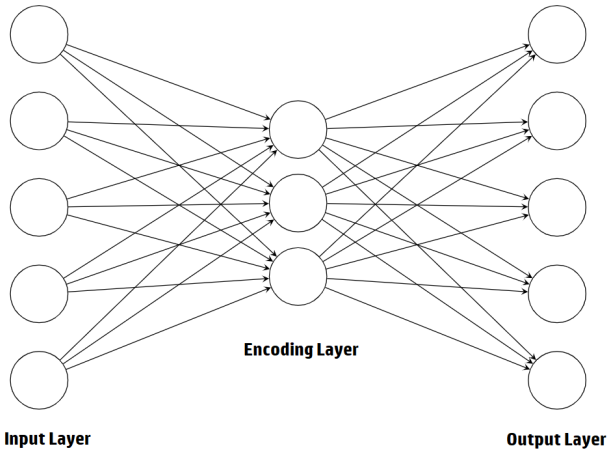

Figure 2.2. Simple Undercomplete Autoencoder Architecture

2.3.1 Autoencoders

Autoencoders (AE) are a class of neural networks that fall under the category of unsupervised learning (Goodfellow et al., 2016). Typically, an autoencoder attempts to

replicate its input at its output and hence the data required to train autoencoders is characterized by lack of separate output labels. This is an indication of the unsupervised nature of these networks. An autoencoder learns an encoding function $h = f(x)$ (encoder) and a decoding function, g, which uses the learnt encodings to generate an output $y = g(h)$. Then the loss between y and x is minimized so that only the most useful encodings required for generating useful reconstructions are learnt. A very simple autoencoder architecture is shown in Figure 2.2. As can be seen from this figure, the AE architecture has an encoder component and a decoder component. Typical implementations often have multiple layers, for both the encoder and decoder components, and usually the decoder architecture is a mirror reflection of the encoder architecture. The encoder attempts to learn an effective low-dimensional representation while the decoder tries to recreate the input data from these low-dimensional embeddings. Oftentimes, it is of significance to train an autoencoder network to learn only the most useful features of the input data and hence the encoding layer usually has a lower dimension than the input dimension. Such autoencoders are referred to as **Undercomplete Autoencoders** and they are trained such that they are forced to learn only the most informative low-dimensional encodings of the input data required to reconstruct the input. All autoencoders used in this work are undercomplete autoencoders and henceforth, the term "autoencoder" implies an undercomplete autoencoder in this work, unless specified otherwise.

The autoencoder network can be seen a specialized version of feed-forward networks and hence can be trained using backpropagation (Rumelhart et al., 1986) with a suitable optimizer (Goodfellow et al., 2016). The data fed to the output layer is usually the same as that fed to the input layer, so that the network learns to reproduce this input. It is important to ensure that this reconstruction is not identical but a very close approximation to the input data. Otherwise the network might just be copying input to output without learning anything of value about the underlying input distribution.

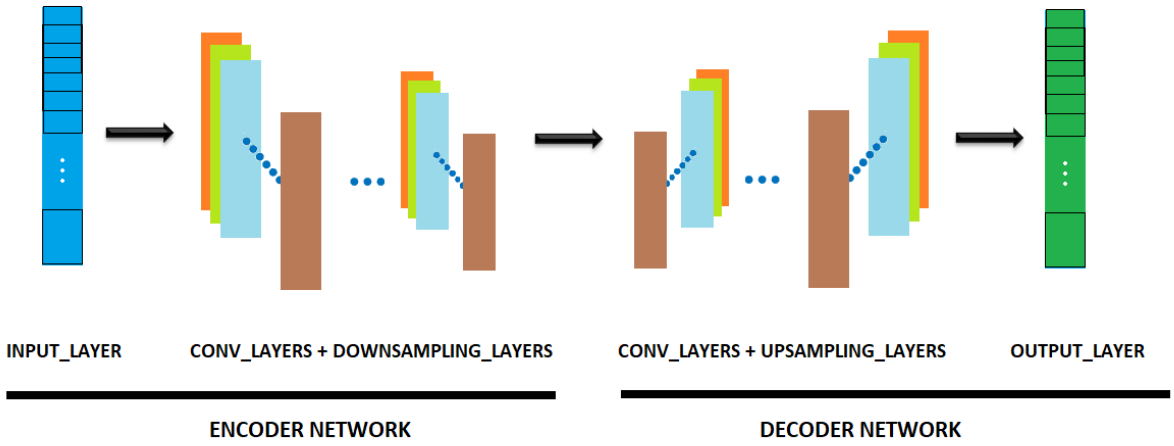

Figure 2.3. Convolutional Autoencoder architecture

A variant of vanilla (regular) autoencoders is the **denoising autoencoder (DAE)**. In this, data is first corrupted by adding random Gaussian noise to it. This noise-corrupted data acts as the input and the original clean data acts as the output to the DAE network. This ensures that the DAE network learns to extract useful information from the noisy input data. Another variant of autoencoders is the **Convolutional Autoencoder** (CAE) (Masci et al., 2011). The CAE network can be viewed as a combination of a convolutional neural network (CNN) and a vanilla autoencoder. The encoder is characterized by convolutional and pooling layers with decreased spatial dimensionality and increasing depth (channels per layer) corresponding to increased feature maps. The decoder network is the opposite of the encoder network i.e., the decoder has increased spatial dimensionality (by upsampling) and decreased depth. to be more precise the last layer of the decoder has the same depth as that of the input data (see Figure 2.3). The advantage of the CAE architecture over vanilla autoencoders is that neurons (nodes) of a feature map in a convolutional layer share the same parameters which makes the network translation-invariant and hence can capture similar patterns at any location in the input. This helps overcome the dimension hopping

problem that occurs with regular vanilla autoencoder (or any regular feed-forward neural network) as well reducing the model size.

2.3.2 Methodology

In this work, a One-Dimensional Denoising Convolutional Autoencoder (1D-DCAE), is implemented to achieve robust HF noise suppression. The data for training and validating the 1D-DCAE HF-noise model is obtained from the MIT-BIH Arrhythmia Database (MITDB) (Goldberger et al., 2000; Moody and Mark, 2001). There are 48 records in total in the MITDB database and each of these records contain two-lead ECG signals. These signals are sampled at 360Hz and are thirty minutes long in duration. ECG signals from twenty-two records belonging to the MITDB database were used to form the training data and ECG signals from another twenty-two records in the same database were used to form the validation data (De Chazal et al., 2004). Throughout this work, when using the MITDB records, ECG signals from lead MLII (Modified Limb Lead II) alone are used for analysis. The records used for the training and validation data are shown in Table 2.1.

Table 2.1. MITDB Dataset split for Training and Validation

Dataset	**Records**
MITDB-DS1 (training)	101,106,108,109,112,114,115,116,118,119,122, 124,201,203,205,207,208,209,215,220,223,230
MITDB-DS2 (validation)	100,103,105,111,113,117,121,123,200,202,210, 212,213,214,219,221,222,228,231,232,233,234

Each of the ECG signals from the records in Table 2.1 are used to prepare the training and validation data for the 1D-DCAE network as follows:

1. Each ECG signal is first subjected to baseline wander suppression as described in Section 2.2.2.

2. The BW-suppressed ECG signal is then divided into non-overlapping segments, ECG_{bw}, each 650ms in duration.

3. Random Gaussian noise with zero mean and unit variance is added to each of these segments. This constitutes the noisy input data ECG_{bw-ns} for the 1D-DCAE network

4. The output for the 1D-DCAE network is the original, noise-free BW-suppressed data, ECG_{bw}.

This network is trained end-to-end using the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.001. Mini-batch gradient descent is used to train the network with a mini-batch size of 64 and an epoch size of 200. Optimal model weights are chosen based on the accuracy obtained on the validation dataset and the weights that give the best performance on the validation dataset are used in the final model. This ensures that the 1D-DCAE network does not overfit on the training data and instead learns to extract useful ECG components from HF noise. The 1D-DCAE network with these best weights is the HF-noise suppression model, $HFnoise_{model}$.

2.3.3 Denoising new incoming ECG signals

An incoming ECG signal first undergoes resampling at 360 Hz. This is followed by baseline wander removal as described in Section 2.2.2. Then this signal is divided into non-overlapping segments, each 650ms, in duration and passed as input to $HFnoise_{model}$. The

output of this model are the HF-noise suppressed segments. The non-overlapping denoised segments are concatenated in the same order as they were divided and the resultant one-dimensional signal is the final denoised ECG signal.

This concludes the ECG denoising section. Figure 2.4 shows an example of a denoised ECG segment from record 203 in the MITDB database. The next section describes this work's approach to tackling the third type of artifact, i.e., EM noise, which determines the quality or utility of the denoised signal for arrhythmia analysis. Henceforth in this dissertation, the terms "signal" or "input signal" refer to the denoised ECG signal unless specified otherwise.

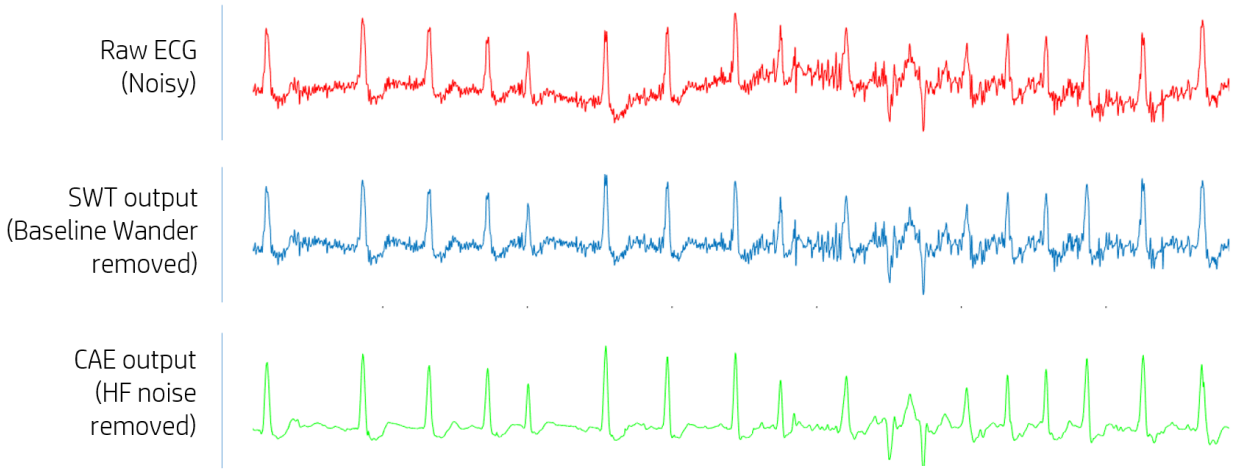

Figure 2.4. Denoised ECG segment using SWT and 1D-DCAE

2.4 Signal Quality Analysis

After ECG denoising, signal quality analysis is performed to reduce arrhythmia misclassifications. This refers to electrode motion (EM) noise detection in particular. EM artifacts need more attention as exaggerated presence of EM artifacts could potentially hinder accurate beat detection. EM artifacts are usually caused by severe patient movement

and/or electrode displacements and often mimic QRS complexes in their morphology. Figure 2.5 shows an example of an ECG signal corrupted by EM noise. These artifacts usually occupy the same spectral bandwidth as that of QRS complexes. Hence it is not a good idea to attempt to filter out EM noise as this may result in loss of actual QRS complex information as well. Instead, it is more efficient to identify (classify) ECG segments exhibiting significant EM interference and suppress further ECG analysis in these segments. The task of identifying extremely noisy segments is an important precursor to performing reliable arrhythmia analysis and in this work, a One-Dimensional Convolutional Neural Network (1D-CNN) model (Goodfellow et al., 2016; LeCun et al., 1995) is developed for detection of ECG segments that are characterized by a high degree of EM noise.

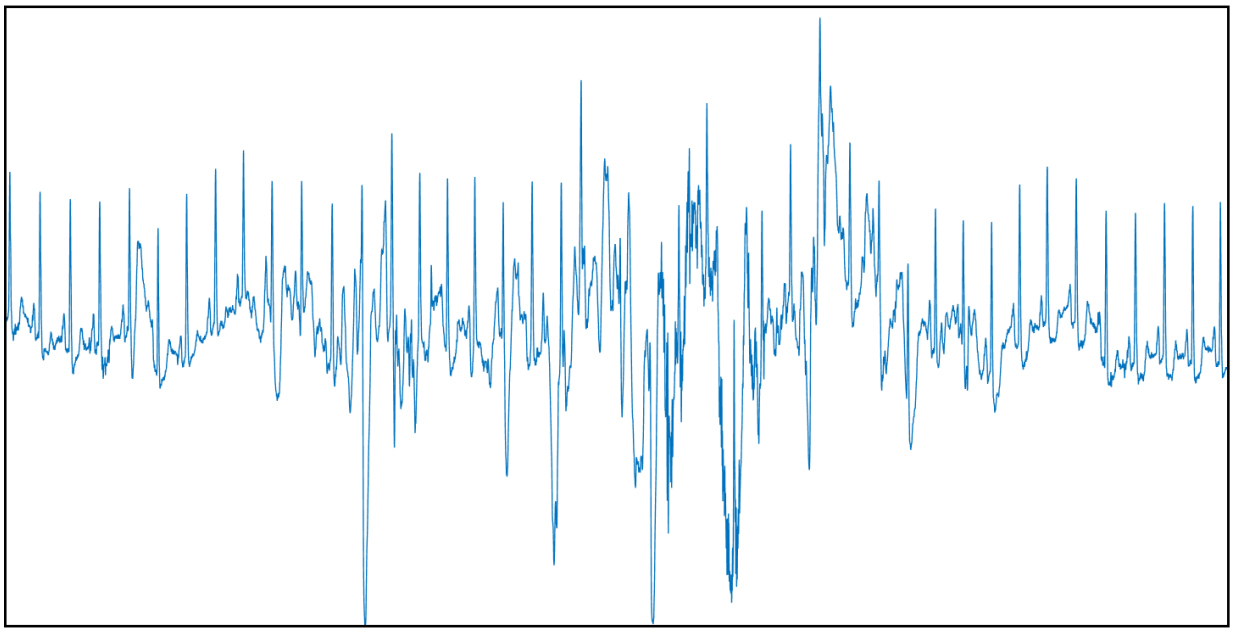

Figure 2.5. Example of EM noise

2.4.1 Data preparation

The data for training the CNN model for EM noise detection is obtained from the MITDB database and follows the training/ validation split shown in Table 2.1. Since the signals

in the MITDB database are relatively clean and lack any significant EM noise presence, synthetic EM noise data, available in the MIT-BIH Noise Stress Test Database (NSRTDB) (Goldberger et al., 2000; Moody et al., 1984), is added to the signals in the MITDB database to create the noisy input data. This is accomplished using the ***nst*** function (physionet.org/physiotools/wag/nst-1.htm), available as part of Physionet's WFDB software package. The *nst()* function takes as input a clean ECG signal and a purely noise (EM) signal, along with their respective sampling frequencies, and outputs a noisy ECG signal, as per (2.1). The *nst()* function also takes as input the desired signal-to-noise ratio (SNR), in decibels (dB), for the output signal.

$$noisy_{ECG} = clean_{ECG} + A * noise_{EM} + B \tag{2.1}$$

where, $noisy_{ECG}$ refers to the synthetically created noisy ECG signal, $clean_{ECG}$ refers to the original EM-noise-free ECG signal, $noise_{EM}$ refers to the pure EM noise signal, A refers to the gain applied to he $noise_{EM}$ signal and B refers to the offset (DC shift).

The *nst()*, by default, uses the first five minutes of the (clean) ECG signal to learn the QRS complex characteristics necessary to automatically compute parameters A and B in (2.1) so as to create an output noisy signal with the desired SNR value. Specifically, the *nst()* function outputs a synthetic noisy signal characterized by alternating noisy and clean segments - each two minutes long in duration - starting from 5:00 (the fifth minute). As an example, for a signal in the MITDB database, the *nst()* function would result in an output signal with noisy data from 5:00 to 7:00, 9:00 to 11:00, 13:00 to 15:00 and so on. Thus, creating synthetic EM noise with several different SNR values, results in a rich set of alternating noisy and clean data. More information about the *nst()* function can be found at (physionet.org/physiotools/wag/nst-1.htm)

The 1D-CNN network developed in this work for EM noise classification uses information from both the noisy ECG signal and its SWT level-4 detail coefficients, (see

Section 2.2.2). The level-4 coefficients correspond to the frequency range [11.25Hz to 22.5Hz], which is where QRS energy is highly concentrated. Thus providing this information to the 1D-CNN network, along with the ECG signal, helps the network learn to distinguish between EM noise artifacts and true QRS complexes. Figure 2.6 shows the architecture of the CNN model used for EM nosie detection.

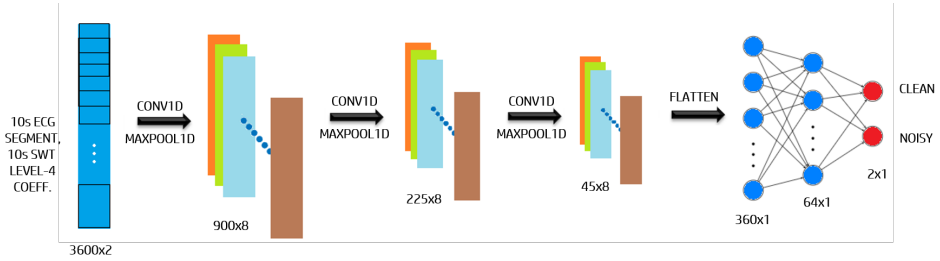

Figure 2.6. Convolutional Neural Network architecture for detection of electrode motion artifacts.

The process governing the generation of training and validation data is as follows.

1. Denoise the raw ECG signal using the techniques described in Sections 2.2.2 and 2.3.2.

2. Create synthetic noisy ECG signals using the earlier described process for SNR values of -4dB.

3. Compute the SWT level-4 detail coefficients (see Section 2.2.2) for this synthetic signal.

4. Divide the synthetic signal into M non-overlapping segments, each 10 seconds in duration (3600 samples at 360Hz). Here M is the number of 10-second ECG segments present in the synthetic signal.

5. Similarly, divide the SWT level-4 detail coefficient signal into M non-overlapping segments, each 10 seconds in duration.

6. Pair up each 10-second noisy segment with its corresponding SWT-level-4 10-second segment to form a [M x 3600 x 2] input.

7. Repeat steps 1 to 6 for SNR values in the set [-2dB, 0dB, 2dB, 4dB]

8. Repeat steps 1 to 7 for all signals in the MITDB database.

2.4.2 Training the 1D-CNN network

The above data preparation process results in a data matrix D_{em} of size Kx3600x2, where K is the total number of 10-segments obtained as a result of the synthetic data generation process. In this work the value of K is 19800 with 9900 examples belonging the training set 9900 examples belonging to the validation set. The data examples in D_{em} that belong to the training set are used to train the CNN model while examples belonging to the validation set are used to determine optimal network weights that aid in robust EM noise classification. The network is trained using backpropagation and Adam optimizer with initial learning rate of 0.005. Rectified Linear Units (ReLU) activation function (Goodfellow et al., 2016) is applied to the hidden layers while softmax activation function is applied to the output layer. Mini-batch gradient descent is used for updating network weights with a mini-natch size of 64 and L2-regularization with a regularization coefficient of 0.01 is applied to the hidden layers (except MaxPooling layers) to reduce overfitting.

2.4.3 Results

Table 2.2 shows the performance of the 1D-CNN EM noise classification model on the training and validation datasets in terms of Sensitivity (Se), Positive Predictive Value (PPV) and F-Score (Fsc). The equations for these metrics are given in (2.2) through (2.4).

$$Se = \frac{TP}{TP + FN} * 100 \tag{2.2}$$

$$PPV = \frac{TP}{TP + FP} * 100 \tag{2.3}$$

$$F1 = \frac{2 * TP}{(2 * TP) + FP + FN} * 100 \tag{2.4}$$

where, TP refers to the True Positives i.e., correctly classified noisy segments, FP refers to False Positives i.e., clean segments misclassified as noise and FN refers to False Negatives i.e., noisy segments that are misclassified as being clean.

Table 2.2. Evaluation on MITDB and INCARTDB Datasets

Dataset	**Se**	**PPV**	**F1**
MITDB-DS1 (Training dataset)	99.69%	99.94%	99.82%
MITDB-DS2 (Validation dataset)	99.61%	99.59%	96.60%

Along with class labels, the 1D-CNN classifier also outputs prediction probability for each 10s segment. This can be interpreted as the confidence with which 1D-CNN network makes a prediction (noisy or clean). Usually segments with score greater than or equal to 0.5 are classified as noise and those with scores less than 0.5 are classified as clean. In this work, segments with scores exceeding 0.9 (high probability i.e., extremely noisy) are considered to be of poor quality and are hence omitted from further analysis. Figure 2.7 provides the confusion matrix information for EM classification performance on the

Validation Dataset Confusion Matrix			
	noisy	clean	
NOISY	5128 (TP)	20 (FN)	5148
CLEAN	21 (FP)	4731 (TN)	4752
	5149	4751	9900

Figure 2.7. Confusion matrix for EM noise classification on the validation dataset

validation dataset. In this figure, the rows(upper case `NOISY` and `CLEAN`) represent the true labels and the columns represent the predicted labels (lower case `noisy` and `clean`).

This concludes the signal quality analysis section and the signal pre-processing section of this dissertation overall. The subsequent three chapters describe QRS complex detection (Chapter 3), Ventricular Arrhythmia detection (Chapter 4) and Supraventricular Arrhythmia detection (Chapter 5).

References

Bengio, Y., A. Courville, and P. Vincent (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence 35*(8), 1798–1828.

Daubechies, I. (1992). *Ten Lectures on Wavelets*. USA: Society for Industrial and Applied Mathematics.

De Chazal, P., M. O'Dwyer, and R. B. Reilly (2004). Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE transactions on biomedical engineering 51*(7), 1196–1206.

Goldberger, A. L., L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley (2000). Physiobank, physiotoolkit, and physionet. *Circulation 101*(23), e215–e220.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y., Y. Bengio, et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks 3361*(10), 1995.

Masci, J., U. Meier, D. Cireşan, and J. Schmidhuber (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, pp. 52–59. Springer.

Moody, G. B. and R. G. Mark (2001, May). The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine 20*(3), 45–50.

Moody, G. B., W. Muldrow, and R. G. Mark (1984). A noise stress test for arrhythmia detectors. *Computers in cardiology 11*(3), 381–384.

Nason, G. P. and B. W. Silverman (1995). *The Stationary Wavelet Transform and some Statistical Applications*, pp. 281–299. New York, NY: Springer New York.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature*, 533–536.

Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103.

CHAPTER 3

ECG BEAT DETECTION

3.1 Introduction

Beat detection* is vital to the field of automated cardiac monitoring and acquires further importance in the context of cardiac arrhythmia detection. Accurate beat detection algorithms using QRS complex information from electrocardiogram (ECG) signals result in useful heart rate variability analysis that subsequently leads to accurate detection of cardiac arrhythmias and other abnormalities. This gains special significance in today's world dominated by non-invasive wearable ECG sensors for real-time cardiac monitoring outside of hospitals and other emergency care centers. These sensors are worn by individuals on a continual basis while performing day-to-day activities. Therefore, there is a high probability that the signals obtained from these sensors are corrupted by external noise, thereby rendering the beat detection process error-prone and thus cumbersome. This external noise can be attributed to, but not limited to, artifacts due to movements necessitated by an individual's routine activities, sensor disconnects, wireless signal transmission interference, baseline wander, powerline interference, muscle movements, etc., resulting in corruption of vital ECG information as described in Chapter 2. Hence it is necessary to develop algorithms that perform well in such scenarios with minimal false detections. Secondly, it is extremely vital that the algorithm can adapt to varying heart rates exhibited by various cardiac arrhythmias when present. In the presence of arrhythmia, the heart rate is not constant and keeps fluctuating depending upon the type, severity and the number

*Several portions of this chapter have been reproduced/adapted with permission from material published in (Kalidas and Tamil, 2017): *V. Kalidas and L. Tamil, "Real-time QRS detector using Stationary Wavelet Transform for Automated ECG Analysis," 2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE), Washington, DC, 2017, pp. 457-461, doi: 10.1109/BIBE.2017.00-12.*

of concurrently occurring arrhythmias. Given these factors, it is imperative that beat detection algorithms are extremely robust to noise without compromising on detection accuracy, especially under arrhythmic conditions.

As mentioned in Section 1.3, the QRS complex is the most significant fiduciary point in an ECG and the R-peak (peak of R-wave) is often used as a representative marker for the QRS complex as a whole. Hence the QRS detection algorithm explained in this chapter focuses on detecting these R-peak locations accurately. To achieve this, a combination of Convolutional Autoencoders (CAEs) and adaptive thresholding is used in a two-stage approach. Combining CAEs with a simple number of adaptive parameters enables the algorithm to robustly detect QRS complexes in the presence of noise as well as in the presence of arrhythmias. he performance of the algorithm was evaluated on the MIT-BIH Arrhythmia database (MITDB) and St. Petersburg 12-lead INCART database (INCARTDB) (Goldberger et al., 2000; Moody and Mark, 2001). It must be noted that the terms *QRS complex*, *R-peak* and *beat* will be used interchangeably throughout this dissertation and they all refer to the R-peak location in general.

This chapter is organized as follows. Section 3.2 provides a brief overview of previously published methods in literature for beat detection. Section 3.3 explains the beat detection algorithm in detail with the necessary equations and threshold settings along with brief explanation about the datasets used. The performance evaluation results of the algorithm are reported in section 3.4. The chapter finally concludes with a brief discussion on the scope and future improvements of our algorithm in section 3.5.

3.2 Literature Review

In literature, various approaches have been proposed for beat detection that generally incorporate a combination of one or more of techniques such as adaptive thresholding (Pan

and Tompkins, 1985; Christov, 2004; Lai et al., 2015; Hamilton and Tompkins, 1986), Hilbert transform (Benitez et al., 2000), multi-scale morphological transformation (Sun et al., 2005), wavelet analysis (Merah et al., 2015; Martínez et al., 2004; Li et al., 1995; Bahoura et al., 1997), etc. Most the algorithms cited in literature are offline algorithms. Offline algorithms (Benitez et al., 2000; Merah et al., 2015; Martínez et al., 2004), have the inherent disadvantage in real-time analysis. Also, most of the offline algorithms work well on longer durations of ECG signal (Li et al., 1995). Longer segments are more stable to pre-processing steps such as band-pass filtering and normalization and as such, are less susceptible to transient noise and taller T-waves. At the same time, they are not always sensitive to transient changes in ECG morphologies that are characteristic of arrhythmic episodes. This could often result in skipping of arrhythmic or low-amplitude beats. On the other hand, online real-time algorithms are highly suitable for evolving ECG morphologies and thus can detect arrhythmic beats more accurately. They have adaptive parameters that usually work well with most signals. The processing and response time is relatively very short as well, of the order of three to ten seconds. There is a growing interest in the field of real-time beat detection and hence the need to constantly improve existing methods to continuously improve performance. The methods of (Pan and Tompkins, 1985) and (Hamilton and Tompkins, 1986) were the earliest published in the field of real-time QRS detection that demonstrated very high detection accuracy. Later, (Christov, 2004) proposed a real-time algorithm using combined adaptive thresholds that included back-search for missed beats, thus achieving improved detection performance. Also, wavelet-based techniques (Merah et al., 2015; Martínez et al., 2004; Li et al., 1995; Bahoura et al., 1997) started gaining attention for ECG delineation purposes. Wavelet transforms (WT) allow simultaneous time-frequency analysis of a non-stationary signal at different resolutions that makes them suitable for ECG signal analysis. Their ability to capture transient changes effectively makes them a good choice for arrhythmia detection. Most

of the wavelet based techniques in literature such as (Martínez et al., 2004; Li et al., 1995; Bahoura et al., 1997) use Discrete Wavelet Transforms (DWT) due to their faster computation and decreased redundancy, compared to Continuous Wavelet Transforms (Merah et al., 2015). A variant of DWT, namely Stationary Wavelet Transform (SWT) (Pesquet et al., 1996; Nason and Silverman, 1995) has recently gained popularity in the field of ECG analysis (Merah et al., 2015; Li and Lin, 2009). SWT is similar to Discrete Wavelet Transform (DWT) with the exception that there is no decimation in the time domain. Only a dyadic subsampling of scales (frequency domain) is performed. Hence there is translation invariance and lack of resolution loss at lower frequencies, which are major bottlenecks of DWT (Laguna et al., 1997; Nason and Silverman, 1995). This increases redundancy in coefficients but the advantage is that the additional artifacts induced due to time-domain downsampling at higher scales are mitigated.

3.3 Methodology

Prior to performing any beat detection, signal pre-processing as described in Chapter 2 and ventricular fibrillation detection as described in Chapter 4 are implemented to minimize false beat detections. Following this, detection of QRS complexes is carried out in a two-stage process. The first stage involves use of a One-dimensional Convolutional Autoencoder (described in Section 2.3.1) to extract potential QRS complex locations. These locations may, at times, include ECG components that are not QRS complexes but mimic them closely such as unusually tall T-waves, voltage spikes, etc. To avoid mislabeling these components as R-peaks, an adaptive thresholding approach is adopted which forms the second stage. These two stages are briefly explained in the following subsections.

3.3.1 Stage I: Initial beat segmentation using Convolutional Autoencoders

In this stage the denoised ECG signal is first divided into non-overlapping ten-second segments. Each of these segment is scaled in the [-1, 1] range. This results in the creation of a ECG data matrix X of size Nx3600 where N is the number of non-overlapping ten-second segments in the ECG signal and 3600 represents ten seconds of ECG data at 360 Hz sampling frequency. Thus the i^{th} row in this matrix corresponds to the i^{th} (non-overlapping) ten-second ECG segment in the input signal. This matrix is then fed as input to a One-dimensional Convolutional Autoencoder (1D-CAE) network, termed the *beat extraction network*, whose architecture is shown in Figure 3.1.

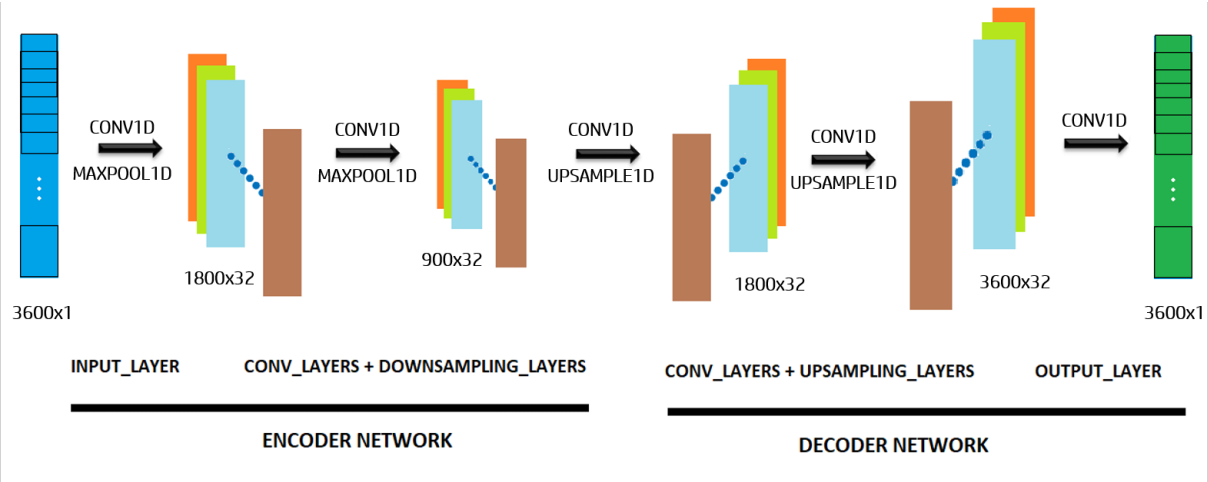

Figure 3.1. Convolutional Autoencoder network architecture for initial beat extraction.

Training the 1D-CAE beat extraction model

The 1D-CAE beat extraction network is trained in a manner similar to a Denoising Convolutional Autoencoder described in Section 2.3.1. ECG signals from the MIT-BIH Arrhythmia Database (MITDB) are used to the train and validate the learnt model. The MITDB database contains 48 ECG records in total. Each record consists of two-lead ECG

signal data sampled at 360 Hz and signal from each lead is thirty minutes long in duration. ECG data from MLII lead is used for training and validating the 1D-CAE beat extraction model and the training/validation split is identical to that followed in Section 2.3.2. For convenience, this split is reproduced here in Table 3.1.

Table 3.1. MITDB Dataset split for Training and Validation

Dataset	**Records**
MITDB-DS1 (training)	101,106,108,109,112,114,115,116,118,119,122, 124,201,203,205,207,208,209,215,220,223,230
MITDB-DS2 (validation)	100,103,105,111,113,117,121,123,200,202,210, 212,213,214,219,221,222,228,231,232,233,234

As can be seen in Figure 3.1, the input and output layers of the 1D-CAE network each have 3600 nodes, corresponding to 3600 samples of ECG data. Hence for an input data matrix X of size Nx3600, this network outputs an output data matrix Y of size Nx3600. Rectified Linear Units (ReLU) activation is applied to each node in each of the four hidden layers (two hidden layers in the encoder network and two hidden layers in the decoder network). On the other hand, hyperbolic tangent function is applied to each node in the output layer so that the output node values are constrained in the range [-1, 1]. Mean squared error (mse) is used as the loss function to be minimized and gradient-descent based optimization is carried out using the Adam optimization function with an initial learning rate of 0.001. The network is trained to read an input denoised ten-second ECG segment and learn to copy potential QRS complex samples (in the input) to the

output while suppressing (zeroing out) the amplitudes of other ECG components. While the training set examples are used for updating the network parameters, the validation set examples are used to evaluate the model after every parameter update and the network parameters that give the least loss on the validation data set are chosen as the optimal network parameters. The 1D-CAE model with these optimal parameters constitutes the optimal beat extraction model and is labeled $beat_{model}$. the output from $beat_{model}$ is used in stage two for implementing adaptive thresholding in order to minimize false beats and missed beats. Figure 3.2 shows an example depicting the input and output of the 1D-CAE beat extraction network.

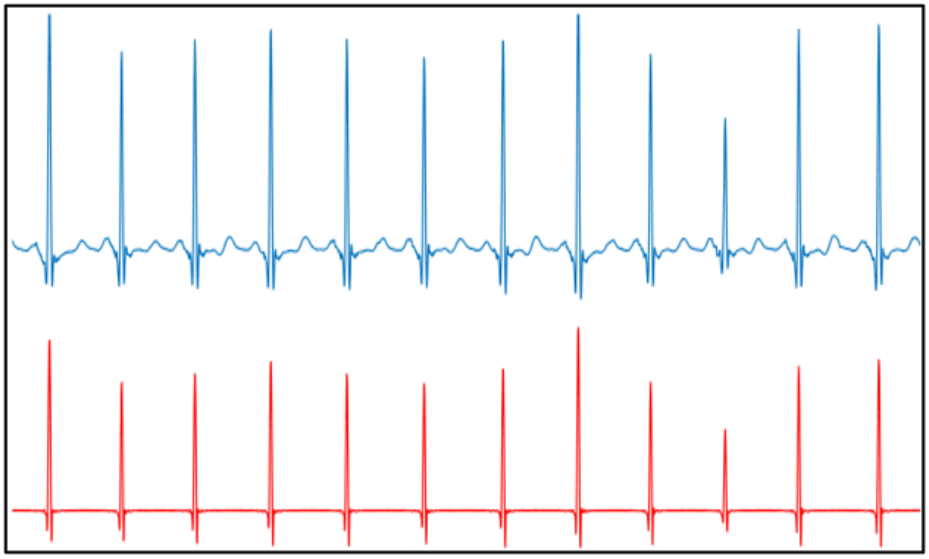

Figure 3.2. Input and output for the 1D-CAE beat extraction model. *Top*. Input ten-second denoised ECG segment. *Bottom*. Output of the model. It can be seen that only QRS complexes (tall peaks) are copied to the output while other ECG components such as P-waves and T-waves are essentially nullified in the output.

3.3.2 Stage II: Adaptive Thresholding

Following beat segmentation in the previous section, adaptive thresholding is implemented to accurately identify beat locations (indices) and minimize occurrence of false beats and

missed beats. The output matrix Y from $beat_{model}$ consists of N ten-second beat vectors with potential QRS complexes enhanced and other ECG components suppressed. From each for these vectors, the corresponding R-peak locations must be detected while being robust to noise and simultaneously being able to identify low amplitude true beats. The adaptive thresholding stage is implemented to achieve these two goals. The adaptive thresholding stages consists of five major phases:

1. Initial peak detection
2. Missed beat detection
3. False beat removal
4. Threshold update
5. Final R-peak location update

Before delving into details about the five phases, some useful initializations and definitions are presented below:

Initialization and Definitions

1. *Fs*: Sampling frequency of the denoised ECG signal i.e., 360 Hz (since all signals are resampled at 360 Hz prior to signal pre-processing - see Section 2.2.2).
2. *pk_thr*: R-peak threshold - Minimum R-peak amplitude threshold for identifying potential R-peaks. Initialized to 0.25 for learning stage.
3. *rr_thr*: RR-interval threshold - Minimum RR-interval threshold between two consecutive R-peaks. Initialized to (0.25*Fs) samples i.e., 250 milliseconds for learning stage since physiological constraints require two heartbeats to be spaced at least 250 milliseconds from each other temporally.

4. *missed_thr*: Minimum separation threshold between consecutive peaks for identifying missed beats.

5. *temp_locs*: Vector storing temporary peak locations (indices) for the current beat vector

6. *pk_vec*: Vector storing peak amplitudes for the current beat vector

7. *ppi_vec*: Vector storing peak-to-peak intervals for the current beat vector

8. ***ecg_locs***: Initial empty vector for storing actual R-peak locations.

The five phases stated earlier are explained next for the i^{th} beat vector in Y.

Phase 1: Initial peak detection

1. Scan the ten-second beat vector to identify peak locations with minimum amplitude of *pk_thr* units and separated by at least *rr_thr* samples.

2. Update *temp_locs* with locations of above found peaks.

3. Update *pk_vec* with amplitudes of these peaks.

4. Update *ppi_vec* with peak-to-peak interval (PPI) values computed from these peaks. PPI values are simply the successive difference values between peak locations (indices).

Phase 2: Missed Beat Detection

1. Determine interval values from *ppi_vec* which exceed a predefined threshold, *missed_thr*, computed as follows:

if (i = 1):

$$missed\_thr = (1.5 * Fs) \tag{3.1}$$

else:

$$missed\_thr = (1.5 * ppi\_thr) \tag{3.2}$$

Equation (3.1) corresponds to the learning stage i.e., the first ten-second beat vector (hence, $i = 1$). Equation (3.2) corresponds to the remaining beat vectors (i.e., $i > 1$).

2. Scan each interval found in step 1 for peaks with a minimum amplitude of 0.05 units.

3. Update *temp_locs*, *pk_vec* and *ppi_vec* vectors appropriately with the new peaks found in the previous step

Phase 3: False beat removal

Along with detecting actual QRS complexes, the above described *missed beat detection* phase might have detected peaks that may not correspond to true QRS complexes. These include tall T-waves, voltage spikes, trivial EM noise peaks, etc.. and are collectively termed *false beats*. To minimize such false beat detections, the following steps are implemented:

1. Determine interval values from *ppi_vec* which are smaller than a predefined threshold, *falsebeat_thr*, computed as follows:

if (i = 1):

$$falsebeat\_thr = (0.33 * Fs) \tag{3.3}$$

else:

$$falsebeat\_thr = (0.35 * ppi\_thr) \tag{3.4}$$

Equation (3.3) corresponds to the learning stage i.e., the first ten-second beat vector (hence, $i = 1$). Equation (3.4) corresponds to the remaining beat vectors (i.e., $i > 1$).

2. For the intervals found in the previous step, obtain the corresponding pair of peak indices and peak amplitudes (since each peak-to-peak interval is computed from a pair of peaks) for each interval from *temp_locs* and *pk_vec* respectively.

3. For each pair of peak indices and peak amplitudes obtained in the previous step, retain the peak index with the higher peak amplitude and delete the other one.

4. Update *temp_locs*, *pk_vec* and *ppi_vec* vectors accordingly.

Phase 4: Threshold update

1. Update *rr_thr* and *pk_thr* thresholds to be used for the next successive beat vector as follows:

   RR-interval threshold update

$$rr\_thr = max((0.25 * Fs), 0.5 * (rr\_thr) + 0.5 * (median(ppi\_vec))) \tag{3.5}$$

   R-peak threshold update

$$pk\_thr = 0.5 * (median(pk\_vec)) \tag{3.6}$$

Phase 5: Final R-peak location update

The peak index values in *temp_locs*, after undergoing missed beat detection and false beat removal, are determined to be the final R-peak locations and are appended to the *ecg_locs* vector.

Each of the above five phases are successively applied to each ten-second beat vector in Y (output from $beat_{model}$), using thresholds from the preceding cycle. The first ten-second beat vector is considered to be the *learning stage* that aids in learning the average RR-interval and R-peak thresholds for the ECG signal (since different individuals have different

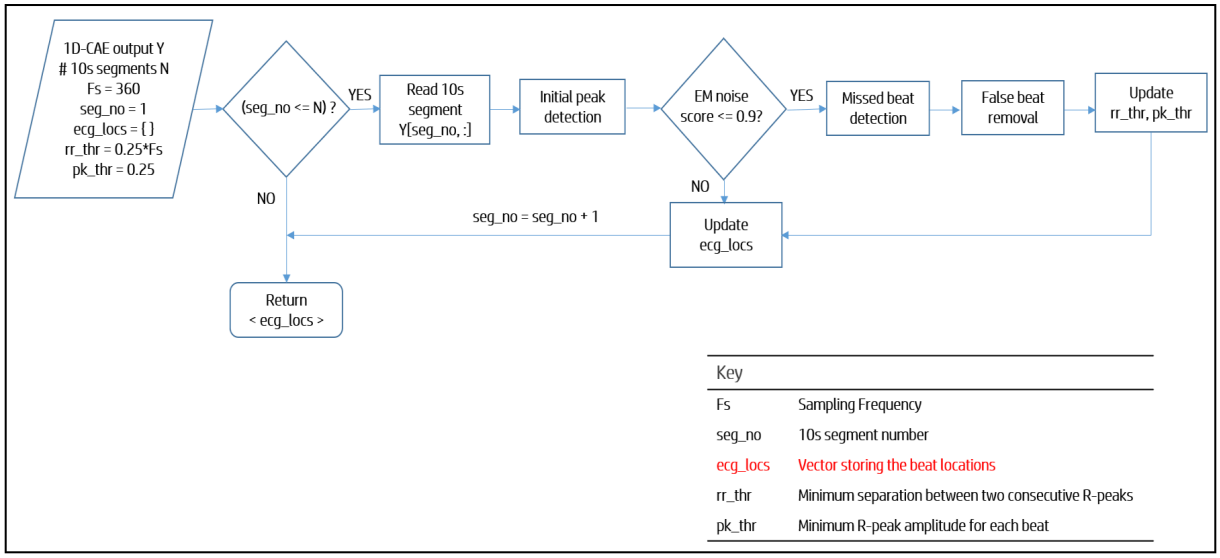

Figure 3.3. Flow diagram depicting the complete adaptive thresholding stage.

heart rates). It is important to note that if a ten-second beat vector's corresponding ten-second input ECG segment had been classified as too noisy i.e., had been assigned an EM noise classification score greater than 0.9 (see Section 2.4, then missed beat detection, false beat removal and threshold update phases are omitted for that particular beat vector. A flow diagram concisely depicting the adaptive thresholding stage is shown in Figure 3.3.

3.4 Results

The above described QRS complex detection algorithm was evaluated on the MIT-BIH Arrhythmia database (MITDB), the St.Petersburg's 12-lead INCART Database (INCARTDB) and the MIT-BIH Atrial Fibrillation Database (AFDB). There are 48 two-lead ECG records in the MITDB database, with the ECG signals sampled at 360 Hz. Similarly the INCARTDB database contains 75 twelve-lead ECG records, with the ECG signals sampled at 257 Hz. THE AFDB database on the other hand is a long-term monitoring database and contains 23 two-lead ECG records, sampled at 250 Hz and signal in each lead is approximately 10 hours long in duration. Since this dissertation is focused on

Table 3.2. QRS detection performance on MITDB, INCARTDB and AFDB records

Database (Sampling frequency)	**Total beats**	**Se**	**PPV**	**Mean error (# of samples)**
MITDB (360 Hz)	109494	99.63%	99.88%	5.03 ms (<2 samples)
INCARTDB (257 Hz)	175906	99.53%	99.61%	8.00 ms (˜2 samples)
AFDB (250 Hz)	1090874	99.48%	97.35%	12.70 ms (<4 samples)

developing arrhythmia detection algorithms using information from only single-lead ECGs, signals from MLII lead for the MITDB database, Lead V for the INCARTDB database and lead ECGI for the AFDB database were used resulting in 109494 beats for the MITDB records, 175906 beats for the INCARTDB records and 1090874 beats for the AFDB records. The algorithm achieves a sensitivity of 99.63% and a positive predictive value (PPV) of 99.88% on ECG records in the MITDB database, while achieving a sensitivity of 99.53% and PPV of 99.61% on the INCARTDB ECG records. For ECG records in the AFDB database, the beat detection algorithm achieves a sensitivity of 99.48% and a PPV of 97.35%. Although the AFDB database is not a widely used database to report QRS detection performance, it plays a significant role in the context of the work presented in this dissertation. ECG signals in the AFDB database contain a large number of atrial fibrillation (described in Chapter 5 episodes which are most prominently characterized by irregular heart rate changes i.e., randomly changing RR-interval values. Hence techniques for detection of QRS complexes (or R-peaks)in the presence of atrial fibrillation must be highly adaptive to these random RR-interval transitions. The beat detection algorithm

presented in this chapter satisfies this requirement very well as evidenced by the accuracy values shown in Table 3.2, thus guaranteeing efficient performance under rapidly changing heart rate conditions which is often indicative of arrhythmias. Figure 3.4 shows an example demonstrating beat detection by the proposed approach. It can be seen that the initial peak detection step misses four beats (in orange ellipses). These are identified during the missed beat detection stage, thus ensuring that all the beats are detected correctly. In fact, the beats missed in the initial peak detection are premature ventricular complexes (PVCs), an abnormal beat whose detection is vital. PVCs are discussed in detail in Section 4.4.

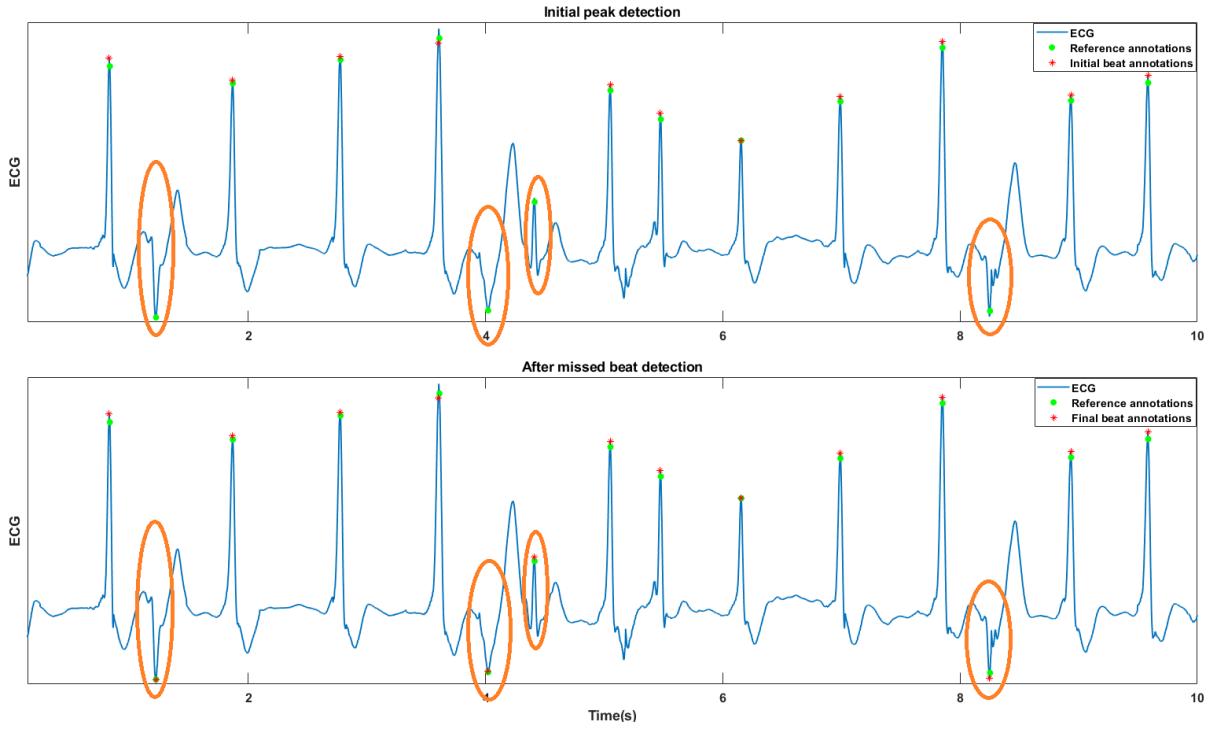

Figure 3.4. Example demonstrating beat detection. *Top.* Initial peak detection. *Bottom.* Final beat locations after missed beat detection.

Figure 3.5 shows an example of the algorithm effectively adapting to varying RR-interval values. The top plot shows a five-minute ECG segment and the bottom plot shows the true RR-interval series and the algorithm's adaptively learnt RR-interval thresholds. The

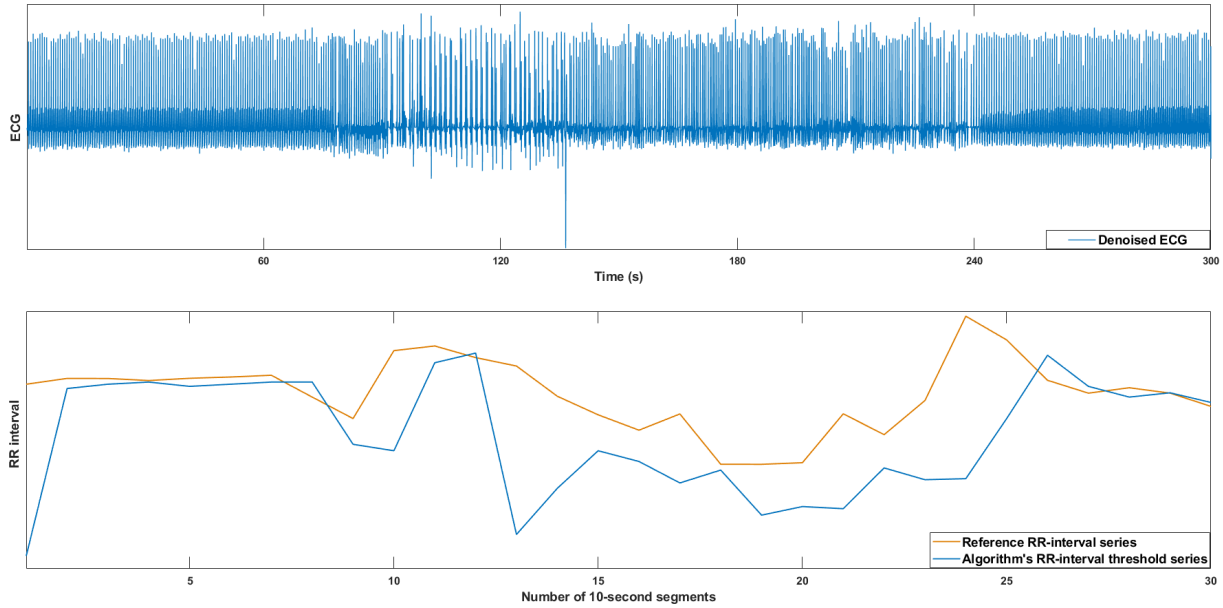

Figure 3.5. Example demonstrating adaptive learning of RR-interval changes.

ECG segment has an atrial fibrillation episode from around the 90s mark to about 240s. Correspondingly the RR-interval values vary irregularly in the bottom plot (labeled as *Reference RR-interval series*). It can be seen that the algorithm's learnt RR-thresholds (labeled as *Algorithm's RR-interval threshold series*) track the true RR-interval values and are almost always lower than the true RR-intervals. This ensures that the algorithms maximizes detection of true R-peaks while missing only very few, if any, of them. The last column in Table 3.2 shows the mean absolute error between the true R-peak locations and algorithm detected R-peak locations. It can be seen that this error is about two to four samples for all the three databases. Figure 3.6 shows the histograms depicting the absolute error for beats belonging to the three databases. Each of these databases contains a combination of ventricular and supraventricular dysrhtyhms and the beat detection algorithm continues to function accurately under these conditions, thus demonstrating its ability to adapt to different types of arrhythmias.

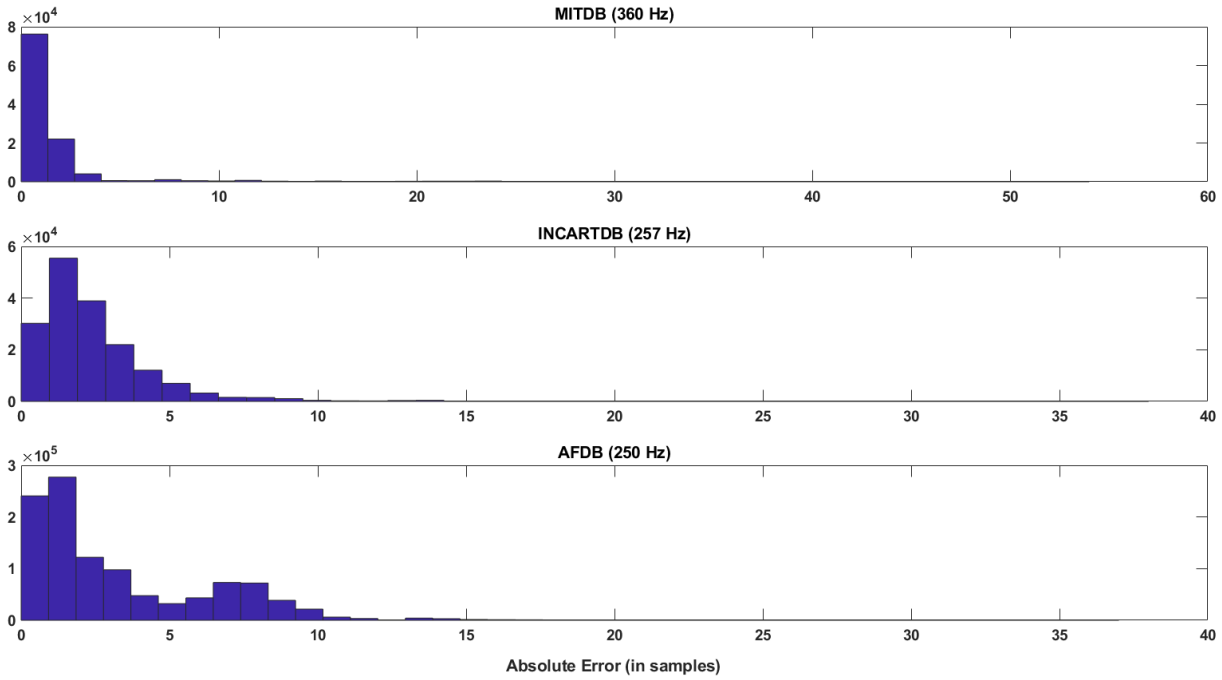

Figure 3.6. Histograms depicting absolute error in detection of beats for each database.

3.5 Conclusion

In this chapter, an algorithm for automated detection of QRS complexes in an ECG signal is presented. The algorithm incorporates a combination of one-dimensional convolutional autoencoders and adaptive thresholding to achieve accurate results in the presence of noise as well as under different arrhythmic conditions. Three different databases containing records with multiple classes of arrhythmias are evaluated upon to demonstrate the performance of the beat detection approach. The algorithm requires the first ten seconds of the incoming ECG signal to be fairly clean as it is used as the learning template. Otherwise, there could be a slight degradation in the algorithm's performance.

This concludes the description of the beat detection stage. The upcoming two chapters delve into the arrhythmia detection component of this research. In that regard,

ventricular arrhythmia detection techniques are presented in Chapter 4, followed by that of supraventricular arrhythmias in Chapter 5.

References

Bahoura, M., M. Hassani, and M. Hubin (1997). Dsp implementation of wavelet transform for real time ecg wave forms detection and heart rate analysis. *Computer methods and programs in biomedicine 52*(1), 35–44.

Benitez, D. S., P. Gaydecki, A. Zaidi, and A. Fitzpatrick (2000). A new qrs detection algorithm based on the hilbert transform. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, pp. 379–382. IEEE.

Christov, I. I. (2004). Real time electrocardiogram qrs detection using combined adaptive threshold. *Biomedical engineering online 3*(1), 28.

Goldberger, A. L., L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation 101*(23), e215–e220.

Hamilton, P. S. and W. J. Tompkins (1986). Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database. *IEEE transactions on biomedical engineering* (12), 1157–1165.

Kalidas, V. and L. Tamil (2017, Oct). Real-time qrs detector using stationary wavelet transform for automated ecg analysis. In *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 457–461.

Laguna, P., R. G. Mark, A. Goldberg, and G. B. Moody (1997). A database for evaluation of algorithms for measurement of qt and other waveform intervals in the ecg. In *Computers in cardiology 1997*, pp. 673–676. IEEE.

Lai, D., F. Zhang, and C. Wang (2015). A real-time qrs complex detection algorithm based on differential threshold method. In *2015 IEEE international conference on digital signal processing (DSP)*, pp. 129–133. IEEE.

Li, C., C. Zheng, and C. Tai (1995). Detection of ecg characteristic points using wavelet transforms. *IEEE Transactions on biomedical Engineering 42*(1), 21–28.

Li, S. and J. Lin (2009). The optimal de-noising algorithm for ecg using stationary wavelet transform. In *2009 WRI World Congress on computer science and information engineering*, Volume 6, pp. 469–473. IEEE.

Martínez, J. P., R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna (2004). A wavelet-based ecg delineator: evaluation on standard databases. *IEEE Transactions on biomedical engineering 51*(4), 570–581.

Merah, M., T. Abdelmalik, and B. Larbi (2015). R-peaks detection based on stationary wavelet transform. *Computer methods and programs in biomedicine 121*(3), 149–160.

Moody, G. B. and R. G. Mark (2001). The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine 20*(3), 45–50.

Nason, G. P. and B. W. Silverman (1995). The stationary wavelet transform and some statistical applications. In *Wavelets and statistics*, pp. 281–299. Springer.

Pan, J. and W. J. Tompkins (1985). A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering* (3), 230–236.

Pesquet, J.-C., H. Krim, and H. Carfantan (1996). Time-invariant orthonormal wavelet representations. *IEEE transactions on signal processing 44*(8), 1964–1970.

Sun, Y., K. L. Chan, and S. M. Krishnan (2005). Characteristic wave detection in ecg signal using morphological transform. *BMC cardiovascular disorders 5*(1), 28.

CHAPTER 4

VENTRICULAR ARRHYTHMIAS

4.1 Introduction

Ventricular arrhythmias refer to cardiac rhythms that originate in the ventricles. The ventricles may generate electrical impulses either (i) as a consequence of irritable sites that can result in premature beats or due to Sino-atrial (SA) node failing to discharge impulses or impulses from the SA node not getting conducted properly/getting blocked completely. The former category of beats is labeled Premature Ventricular Complexes (PVCs) and the latter is termed Ventricular Escape beats. This chapter is focused on detecting arrhythmias that fall in the former category i.e., those that occur due to presence of PVCs.

Premature Ventricular Complexes (PVCs) are ectopic heart beats that occur as a consequence of electrical impulses originating from ventricles rather than the Sino-Atrial (SA) node (see Chapter 1. They can be viewed as premature depolarization of the cardiac cells (Ahn, 2013). Under normal conditions, electrical impulses originate from the SA node and are propagated through the heart via the atria, atrio-ventricular node, Purkinjee fibres and the ventricles. This represents one cardiac cycle or one heart beat (see Chapter 1). In the presence of PVCs, these impulses abnormally originate from irritable sites in the ventricles instead. This results in premature and abnormal heart beats that disrupt normal sinus rhythm. PVCs can be distinguished from other beats based on their QRS morphology on electrocardiogram (ECG) signals. In terms of their manifestation on ECGs, PVCs are characterized by wider QRS complexes (exceeding 0.12s) (Aehlert, 2018) with T-waves having an opposite polarity to that of the QRS complex along with absence of P-waves. PVCs are impulses that originate in the ventricles and therefore atrial depolarization does not take place which explains the absence of P-waves. PVCs can be uniform or multiform (see Figure 4.1) in their appearance, depending on the origin of electrical impulse discharge

in the ventricles. Uniform PVCs refer to PVCs that originate from the same irritable site in the ventricles. On the other hand, multiform PVCs occur due to impulses being discharged from several irritable sites in the ventricles. PVCs are often followed by a compensatory pause but this is not a strict condition. PVCs can occur in healthy individuals as well as in individuals with structural heart disease (Ahn, 2013; Aehlert, 2018). PVC prevalence is estimated to be around 40% to 75% according to (Ng, 2006; Kennedy et al., 1985) with the elderly population being relatively more susceptible (Aehlert, 2018). PVCs can often devolve into life-threatening conditions such as ventricular tachycardia (VT) and ventricular fibrillation (VF) and can potentially lead to heart failure if left untreated. The American Heart Associate Heart Disease and Stroke Statistics - Update 2015 (Mozaffarian et al., 2015) reports that about 23% of out-of-hospital cardiac arrests exhibit an initial rhythm of VF or VT. Therefore timely and accurate detection of PVCs assumes vital importance in the context of cardiac arrhythmia analysis.

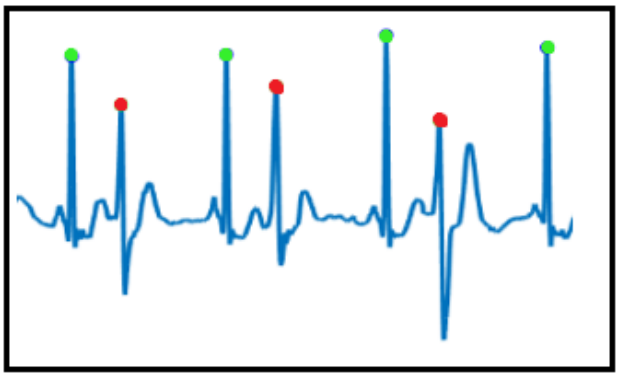

Figure 4.1. Example of Premature Ventricular Complexes. The red markers indicate PVCs. It can be seen that the first two PVCs are of similar morphology (uniform PVCs) and the third one has a different morphology compared to the first two (multiform PVCs).

The ventricular arrhythmias discussed in this chapter can be classified into two groups, namely, *beat-independent ventricular arrhythmias* and *beat-dependent ventricular arrhythmias*. The former group refers to arrhythmia conditions which do not require beat

detection while the latter group includes arrhythmias that have beat detection as a pre-requisite step. They are as follows:

1. Beat-independent Ventricular Arrhythmias

    (i) Ventricular Fibrillation and Sustained Ventricular Tachycardia

2. Beat-dependent Ventricular Arrhythmias

    (i) Premature Ventricular Complexes

    (ii) Ventricular Couplets

    (iii) Ventricular Runs

    (iv) Ventricular Bigeminy

    (v) Ventricular Trigeminy

    (vi) Ventricular Quadrigeminy

This chapter* is organized as follows: Section 4.2 presents a survey of methods that have been proposed in literature in the field of ventricular arrhythmia detection. This is followed by a detailed description of a ventricular fibrillation/sustained ventricular tachycardia detection algorithm in Section 4.3. Subsequently, an in-depth explanation of algorithms focused on detection of premature ventricular complexes and associated beat-dependent arrhythmias is provided in Section 4.4. Finally, concluding remarks are presented in Section 4.5.

*Several portions of this chapter have been reproduced/adapted with permission from material accepted for publication at the 42nd IEEE Engineering in Medicine and Biology Conference 2020.

4.2 Literature Review

Many methods have been proposed in literature for detection of ventricular fibrillation and sustained ventricular tachycardia. This includes techniques that employ time-domain and frequency-domain analysis along with wavelet analysis. The authors of (Thakor et al., 1990) propose a threshold-based approach where a binary sequence is generated for the ECG signal and a sequential hypothesis testing is applied to the sequence's time-interval based probability distribution values. another approach (Chen et al., 1987) use a two-step process based on autocorrelation analysis and regression test for detection of VF. Complexity measure based techniques based on comparison and accumulation are presented in (Zhang et al., 1999). Similarly, the methods of (Moraes et al., 2002) employ a two-step process for VF detection. The first step is consists of a VF Filter Leakage method (Kuo, 1978). This is complemented by a complexity measure analysis to distinguish between VF and VT. The authors of (Barro et al., 1989) present a simple frequency-domain based approach using a simple set of four parameters for effective VF detection. Wavelet-based techniques are proposed in the works of (Addison et al., 2000; Watson et al., 2000; Balasundaram et al., 2011). In terms of methods that combine time and frequency domain approaches, the authors in (Krasteva and Jekova, 2005) propose a real-time VF/VT detection algorithm using a twelve-feature set combining seven features from band-pass filtering and five features from ECG peak detection steps. Another method that combines time-domain and frequency-domain processing of ECG signals for VF/VT detection is presented in (Kalidas and Tamil, 2016). This method uses Support Vector Machines (SVM) for classifying the features and was developed as part of the Annual Physionet/Computing in Cardiology Challenge 2015: Reducing False Arrhythmia Alarms in the ICU ((Clifford et al., 2015). In fact, the works of (Plesinger et al., 2015; Kalidas and Tamil, 2016; Fallet et al., 2015) were the top-ranked papers at the competition and included algorithms for

detection of five life-threatening ventricular arrhythmias. Another SVM-based approach for detection of VF episodes is proposed in (Alonso-Atienza et al., 2012). Similarly, the authors of (Li et al., 2013) present a SVM-based algorithm for detection of VF/VT episodes using a total of fourteen features. Another machine learning approach for VF detection using neural networks is the basis approach described in (Clayton et al., 1994). Finally, a comprehensive comparison and review of ventricular fibrillation detection techniques can be found in (Clayton et al., 1993; Jekova, 2000; Amann et al., 2005).

In terms of detection of premature ventricular complexes, several methods have been proposed in literature. The authors of (De Chazal et al., 2004) compute eight different feature sets derived from analysis of ECG morphology, RR-intervals and other heartbeat dynamics and use Linear Discriminant classifiers on these feature sets for performing final PVC classification. A combination of wavelet analysis and RR-interval analysis is described in (Llamedo and Martínez, 2011) for automated PVC detection. Another technique that applies wavelet analysis and Linear Discriminant Analysis (LDA) forms the basis of PVC detection in (Senhadji et al., 1995). An approach combining clustering with Hermite functions and self-organizing maps is proposed in (Lagerholm et al., 2000). A K-Nearest Neighbor based PVC detection algorithm that uses twenty six features is presented in (Christov et al., 2005). In the method proposed by (Sayadi et al., 2009), Bayesian filtering techniques are used to classify PVC bets using beat morphology information. Techniques that rely on implementation of neural network architectures can be found in the methods adopted by (Yeap et al., 1990; Hu et al., 1993; Osowski and Linh, 2001). In the recent past deep learning has gained significance in the context of classifying ventricular beats. Convolutional neural network (CNN) based algorithm that uses information from raw ECG signals for accurate PVC detection can be found in (Zubair et al., 2016; Kiranyaz et al., 2016; Ince et al., 2009). These three methods use the first five minutes of ECG data as a learning template and hence this enables their methods to achieve very good performance

in terms of developing patient-adaptive PVC detection algorithms. The authors of (de Chazal and Reilly, 2006) propose another patient adaptive PVC detection approach using linear discriminant analysis. In this the Another patient adaptive system that also allows for expert assistance is described in (Llamedo and Martinez, 2012). In this approach an initial classification is performed using the methods developed in (Llamedo and Martínez, 2011) that uses RR-interval and beat morphology features along with a linear discriminant classifier. Following this Expectation Maximization Clustering is performed to group the beats into different clusters each with a signature beat label. After this, an expert re-assigns a subset of beats in each cluster that are misclassified. The accuracy of their approach increases with increase in the number of beats that get re-annotated by the expert, which is expected. Another such approach that allows for expert assistance using Kalman filters is described in (Oster et al., 2015).

4.3 Ventricular Fibrillation and Sustained Ventricular Tachycardia

Ventricular Fibrillation (VF) is a subclass of ventricular arrhythmias that is characterized by chaotic or fibrillatory cardiac activity. The heart is essentially quivering in the presence of VF and is unable to pump blood effectively. There is an absence of pulse and no discernible ECG features such as P-wave, QRS complexes, T-waves, etc. are present. The lack of useful of features is the reason VF detection has been categorized as a beat-independent approach here. On the ECG, they appear as oscillatory waveforms with frequency of about 5 Hz (i.e., nearly 300 bpm - too high to effectively pump blood). Several factors can cause VF and this includes acute coronary syndromes, hypertrophy, severe cardiac failure, adverse effects of medications, external factors such as electrocution, etc. On the other hand, Sustained Ventricular Tachycardia (VT) refers to rapid occurrence of PVCs, at a rate between 150 bpm to 250 bpm, usually for more than 30s. Sustained VT

is often a precursor to VF and at high heart rates, QRS complexes and other waveforms are often not clearly distinguishable on the ECG. Depending on the morphology of PVCs, VT episodes may be monomorphic (consecutive PVCs with same shape and amplitude) or polymorphic (consecutive PVCs with varying shape and amplitude). Sustained VT can often indicate presence of myocardial ischemia (Aehlert, 2018). Both sustained VT and VF are shockable rhythms and hence *defibrillation* is the most commonly adopted approach for treating individuals with sustained VT and VF. Both sustained ventricular tachycardia and ventricular fibrillation, when left undetected or untreated, can devolve into asystole and pulseless electrical activity (Aehlert, 2018) which are a non-shockable rhythms and even a defibrillator may not be helpful in these circumstances (Aehlert, 2018; Jekova, 2007). Hence early detection of sustained VT and VF is very much necessary to increase the individual's chance of survival. Figure 4.2 and Figure 4.3 show electrical activity and ECG manifestation during sustained VT and VF[†].

In this section, a machine learning model using Random Forests (RF) for accurate detection of sustained ventricular tachycardia and ventricular fibrillation is described. The features that act as input to the RF classifier include both time-domain and feature-domain features. The following subsections explain the approach in detail.

4.3.1 Random Forests

Random Forests is a machine learning algorithm that consists of growing an ensemble of binary decision trees for performing regression/ classification. Introduced in (Breiman, 2001a), Random Forests algorithm is described by the authors as a modification of bagging which creates a forest of de-correlated trees and averages their predictions. Decision tree models generally suffer from high-variance and thus averaging out predictions from multiple

[†]Images reprinted with permission from Mayo Foundation for Medical Education and Research.

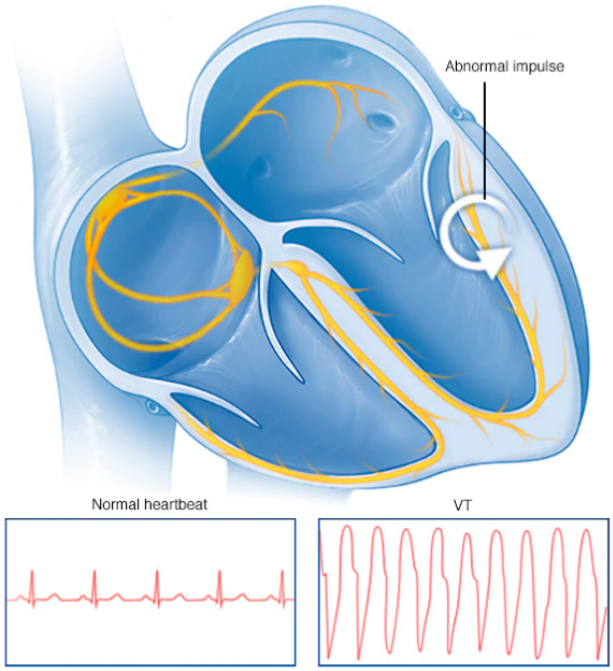

Figure 4.2. Electrical activity and ECG manifestation during Ventricular Tachycardia. (Source: `mayoclinic.org`. ©Mayo Foundation for Medical Education and Research.)

decision trees, each with the same variance, leads to a decrease in overall variance (Hastie et al., 2009a). The process of growing these decision trees is based on bagging (bootstrap aggregation), which is a technique for reducing the variance of an estimated prediction function. Here, bagging primarily involves creating several data subsets from the training data by random subsampling with replacement. Each of these data subsets is then used to train a decision tree model separately. Additionally, for each tree, only a random subset of predictors/features is used for creating the decision model. For each tree thus grown, a predictor is chosen to make a decision split at a particular node. This is usually done based on either the information gain/entropy or the Gini impurity that selects the best predictor for that particular node (James et al., 2013). This process is repeated at each node in a tree and subsequently multiple trees are trained. Final decisions are made based on averaging (for regression) or majority voting (for classification) of individual tree predictions. Since

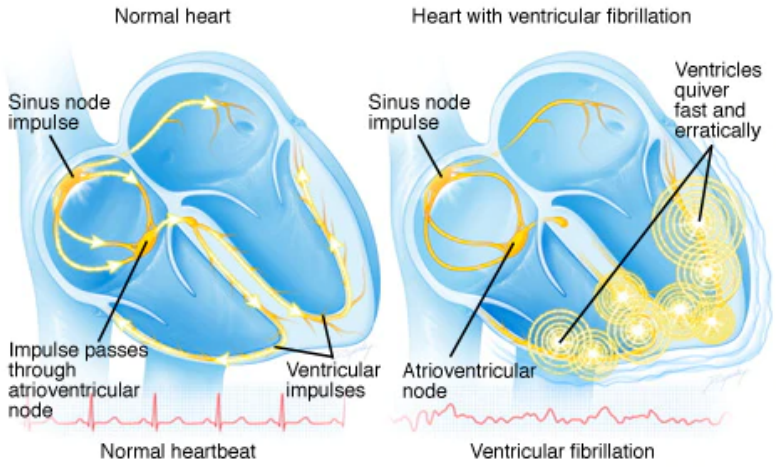

Figure 4.3. Electrical activity and ECG manifestation during Ventricular Fibrillation. (Source: `mayoclinic.org`. ©Mayo Foundation for Medical Education and Research.)

data subsampling and predictor subsampling for each tree is carried out randomly on a forest of trees, this algorithm is termed Random Forests. The inherent randomness in the training data (data subsampling and predictor subsets) results in reduced overfitting (Hastie et al., 2009a) and places a limit on the generalization error (Breiman, 2001a). Random Forest models are often viewed as a competitive alternative to boosting techniques such as Adaboost while being significantly more robust to noise and easier to train. More information about Random Forests can be found in (Breiman, 2001a; Hastie et al., 2009a; James et al., 2013).

4.3.2 Datasets

The datasets for training and validating the Random Forests based classifier are obtained from ECG records present in the MIT-BIH Malignant Ventricular Ectopy Database (VFDB) and the American Heart Association Database (AHADB). The VFDB dataset contains rhythm-annotated 22 two-lead ECG recordings, each signal 30 minutes long and sampled at 250 Hz. The rhythms present in these records primarily include ventricular

Table 4.1. Datasets for training and validating sustained VT and VF detection model

Type	Dataset	Records	VT/VF examples	Non-VT/VF examples
Training	VFDB	Odd-numbered	2100	5422
	AHADB		1024	2536
			3124	**7958**
Validation	VFDB	Even-numbered	1256	6356
	AHADB		1172	2368
			2428	**8724**

fibrillation, sustained ventricular tachycardia, ventricular flutter, atrial fibrillation, asystole, high grade ventricular activity and noise. Similarly, the AHADB dataset consists of ECG recordings representing ventricular arrhythmias. From this database, records 8201 through 8210 are used in this dissertation as contain significant number of sustained VT and VF episodes. Each of these ten records contains annotated two-lead ECG recordings, each signal 35 minutes in duration and sampled at 250 Hz. For the purpose of model development, distinction is not made between sustained ventricular tachycardia, ventricular fibrillation and ventricular flutter and these episodes are collectively labeled as *VT/VF* examples, while the remaining rhythms are labeled as *Non-VT/VF* examples. Table 4.1 provides a concise summary of these datasets.

4.3.3 Feature Extraction

Prior to feature extraction for VT/VF detection, the signal is subject to ECG denoising and EM noise classification as described in Chapter 2. Following that, the denoised signal is

segmented into non-overlapping five-second epochs. From each five-second epoch, a total of seven features are derived. This feature derivation includes time-domain, frequency-domain and stationary wavelet transform (SWT) analysis. They are described below:

1. *max_freq*: Dominant frequency from Fourier analysis computed using Fast Fourier Transform (FFT). This feature is the primary frequency component (in Hz) present in the normalized FFT for the 5s epoch. In the presence of sustained VT/VF, this will take a value in the range 2.5Hz to 8Hz, corresponding to approximately 150 bpm to 500 bpm which is the clinical heart rate range for sustained VT/VF episodes.

2. *xft_ratio*: Ratio of FFT ranges. This feature refers to the ratio between (i) FFT values corresponding to the range [2.5, 8] Hz and (ii) FFT values corresponding to the range [0, 20] Hz. This feature essentially indicates the extent to which clinically significant VT/VF frequency components dominate the frequency content of the five-second epoch.

3. *ccf6*: Pearson's Correlation Coefficient between the five-second epoch and its level-6 SWT detail coefficients, computed using Daubechies3 ('Db3') wavelet as the mother wavelet. At 360 Hz sampling frequency and using dyadic compression in the frequency domain, level 6 detail coefficients correspond to the frequency range [2.8, 5.6] Hz. Hence this feature measures how correlated the ECG epoch is with respect to the SWT level-6 coefficients. In the presence of VT/VF, this value is expected to be relatively large.

   For one-dimensional vectors X and Y; this feature is computed as below (Johnson and Wichern, 2002):

$$Cov(X,Y) = \frac{\sum_{k=1}^{m}(X_k - \overline{X})(Y_k - \overline{Y})}{m-1} \tag{4.1}$$

$$R_{XY} = \frac{Cov(X,Y)}{\sqrt{Cov(X,X)Cov(Y,Y)}} \tag{4.2}$$

where,

- m in (4) is the number of samples in X (or Y),

- X refers to the five-second ECG epoch and Y refers to its SWT level-6 detail coefficients (both have the same length, m)

- $\overline{X}$ refers to the mean (i.e., average) of X

- $\overline{Y}$ refers to the mean of Y.

It must be noted that the term $Cov(X,X)$ in the denominator of 4.2 is simply the variance of X. Similarly $Cov(Y,Y)$ refers to the variance of Y.

4. *ccf7*: Pearson's Correlation Coefficient between the five-second epoch and its level-7 SWT detail coefficients, computed using Daubechies3 ('Db3') wavelet as the mother wavelet. At 360 Hz sampling frequency and using dyadic compression in the frequency domain, level-7 detail coefficients correspond to the frequency range [1.4, 2.8] Hz. Hence this feature measures how correlated the ECG epoch is with respect to the SWT-level7 coefficients. In the presence of VT/VF, this value is expected to be relatively large. This feature is computed similar to that of *ccf6*, except that Y in (4.1) and (4.2) refers to SWT level-7 detail coefficients.

5. *ccf67*: Pearson's Correlation Coefficient between the five-second epoch and the signal reconstructed using both SWT level-6 and level-7 coefficients. This feature is computed in a similar manner as the preceding two features, except that that Y in (4.1) and (4.2) refers to the reconstructed signal.

6. *peak_corr*: Peak auto-correlation value. This is the maximum amplitude of auto-correlation sequence computed for the five-second ECG epoch. The auto-correlation

sequence is computed as follows:

$$Corr_X[l] = \sum_k X[k]X[k+l] \tag{4.3}$$

where,

- l is the l^{th} sample in $Corr_X$. The length of $Corr_X$ is $(2N_X - 1)$ where N_X is length of X i.e., the length of the ECG epoch. Since the ECG epoch is five seconds long, N_X is equal to 1800 (at 360 Hz sampling frequency).

7. *auc_corr*: This refers to the sum of the auto-correlation (computed using (4.3)) coefficients . This is computed as follows:

$$sum\_corr = \sum_{k=1}^{2N_X-1} Corr_X[k] \tag{4.4}$$

In the presence of sustained VT/VF, both *peak_corr* and *auc_corr* are expected to be relatively large. This can be attributed to the fact that the ECG is almost sinusoidal (owing to oscillatory waveforms) in the presence of sustained VT/VF and are therefore expected to be show a high degree of auto-correlation.

4.3.4 Sustained VT and VF classification

The seven features computed in Section 4.3.3 are input to a Random Forest classifier. This classifier was trained using the examples present in the training dataset (See Table 4.1). Hyperparameter tuning using Grid search was performed to identify the best parameters for the Random Forest model. The hyperparameter tuning search space for Random Forests included four primary hyperparameters is given in Table 4.2. The last column in the table shows the best combination of parameters chosen i.e., the model trained with the combination of these particular parameter values was determined to achieve the best performance on the validation dataset (see Table 4.1. This model is labeled $VTVF_{model}$.

Table 4.2. Hyperparameter tuning search space for VT/VF Random Forests model

Hyperparameter	**Range of values**	**Best value**
Number of trees	[10,15,20,25,...,100]	**50**
Maximum tree depth	[5,10,15,..., 50,`None`]	**19**
Maximum features	[`auto`, 7] (`auto` : square root of number of features)	**7**
Class weight	['balanced', None]	**'balanced'**

The *Number of trees* parameter refers to the number of trees in the Random Forests model. The *Maximum tree depth* parameter refers to the maximum depth of each tree. This is an important hyperparameter to select as deeper trees generally result in overfitting. The third hyperparameter, *Maximum features* refers to the number of features to be randomly selected for performing splits at each node (Hastie et al., 2009b). The last hyperparameter is the *Class weight*. This refers to the weight assigned to each class and plays a significant role in removing biases when dealing with imbalanced classes. By default, the 'balanced' option assigns weights inversely proportional to the class frequencies. It can be seen from Table 4.2 that the grid search selected the 'balanced' option for the *Class weight* parameter. This is because the training dataset is inherently imbalanced with an imbalance ratio of about 1:2.5.

4.3.5 Results

The performance of $VTVF_{model}$ was evaluated on the training and validation datasets and the corresponding results are shown in . As shown in Table 4.1, the training dataset consists of a total of 3124 VT/VF examples and 7958 Non-VT/VF examples. Similarly,

the validation dataset consists of 2428 VT/VF examples and 8724 Non-VT/VF examples. ECG signals from both the leads were used for creating the datasets but they were used independent of each other so as to increase the data count for training and validation purposes.

Table 4.3. Performance of VT/VF model on VFDB and AHADB datasets

Dataset	**Se**	**PPV**	**FSc**	**Sp**	**Remarks**
Training	98.85	96.17	97.49	98.45	Odd-numbered records from VFDB and AHADB
Validation	97.90	95.77	96.82	98.80	Even-numbered records from VFDB and AHADB

The results are presented in terms of Sensitivity (Se), Positive predictive value (PPV), F-Score (FSc) and Specificity (Sp). The equations for these metrics are given in (4.5) through (4.8).

$$Se = \frac{TP}{TP + FN} * 100 \tag{4.5}$$

$$PPV = \frac{TP}{TP + FP} * 100 \tag{4.6}$$

$$FSc = \frac{2 * TP}{(2 * TP) + FP + FN} * 100 \tag{4.7}$$

$$Sp = \frac{TN}{TN + FP} * 100 \tag{4.8}$$

where, TP refers to the True Positives i.e., correctly classified VT/VF examples, FP refers to False Positives i.e., Non-VT/VF examples misclassified as VT/VF, TN refers

Training dataset confusion matrix			
	vt/vf	non vt/vf	
VT/VF	3088 (TP)	36 (FN)	3124
NON VT/VF	123 (FP)	7835 (TN)	7958
	3211	7871	11082

Figure 4.4. Confusion matrix depicting model performance on training dataset

Validation dataset confusion matrix			
	vt/vf	non vt/vf	
VT/VF	2377 (TP)	51 (FN)	2428
NON VT/VF	105 (FP)	8619 (TN)	8724
	2482	8670	11152

Figure 4.5. Confusion matrix depicting model performance on validation dataset

to True Negatives i.e., correctly classified Non-VT/VF examples and FN refers to False Negatives i.e., VT/VF examples that are misclassified as Non-VT/VF.

Figures 4.4 and 4.5 show the confusion matrices depicting the performance of $VTVF_{model}$ on the training and validation datasets respectively. In these figures, the rows(upper case `VT/VF` and `NON VT/VF`) represent the true labels and the columns represent the predicted labels (lower case `vt/vf` and `non vt/vf`).

Figure 4.17 shows a two-dimensional mapping of the seven PVC features. This 2D-representation is computed using the popular visualization technique called *t-SNE*, which short for t-Distributed Stochastic Neighbor Embedding ((Maaten and Hinton, 2008)). From this plot it can be seen that the PVC and non-PVC representations form (almost) disjoint class-specific clusters thus indicating the usefulness of the seven features for distinguishing PVC beats from non-PVC beats.

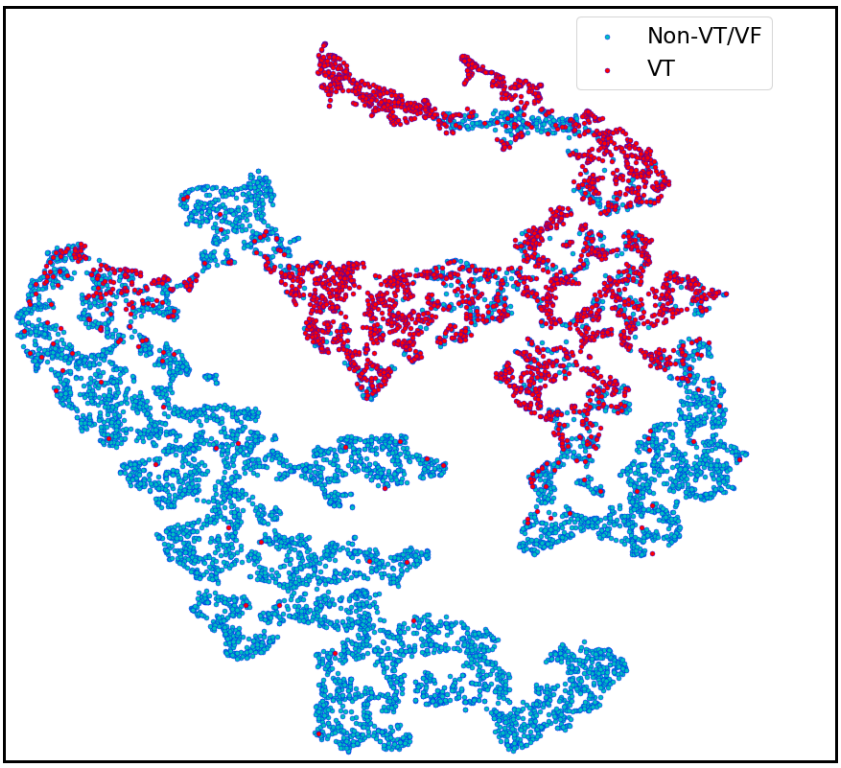

Figure 4.6. Scatterplot showing 2D-representation of the seven VT/VF features using t-SNE.

This concludes the section on algorithm description for ventricular fibrillation and sustained ventricular tachycardia detection. The subsequent sections present detailed description of algorithms for detection of beat-dependent arrhythmias that are inherently dependent on accurate detection of premature ventricular complexes.

4.4 Premature Ventricular Complexes

In this section, a novel metadata-augmented deep-learning based method is presented to accurately detect PVCs from single lead ECG signals using a combination of a Semisupervised Autoencoder (SSAE) (Goodfellow et al., 2016) and Random Forest models (Breiman, 2001b). Raw ECG beat segments along with manual features derived from

three consecutive beats are used to train the SSAE network so as to efficiently learn three-dimensional class-specific encodings (stage 1). The learnt encodings, along with the manually computed beat-triplet features, are then input to a Random Forests classifier for final PVC classification (stage 2). Figure 4.7 shows the block diagram summarizing the approach. The following sections describe the approach in detail.

4.4.1 Datasets

The PVC classifier is trained, validated and tested using ECG signals from records in the MIT-BIH Arrhythmia Database (MITDB) (Moody and Mark, 2001) and the St.Petersburg INCART 12-lead Arrhythmia Database (INCARTDB) (Goldberger et al., 2000), available from the Physionet database (Goldberger et al., 2000). The MITDB database consists of 48 two-lead ECG records, sampled at 360Hz and 30 minutes in duration. The INCARTDB database consists of 75 twelve-lead ECG records, sampled at 257Hz and 30min in duration. AAMI (ANSI/AAMI EC57:2012, 2012) standards are followed for beat labelling. Forty-four records from MITDB are split into training (MITDB-DS1) and validation (MITDB-DS2) datasets as described in (Llamedo and Martínez, 2011). The remaining four records i.e., 102, 104, 107 and 217, have been omitted from analysis owing to presence of paced beats as per AAMI recommendations. The training/validation split is as per table 2.1 described in Chapter 2. For convenience, the information is reproduced in Table 4.4. The INCARTDB records, on the other hand, form the test dataset. In this work, Fusion and Supraventricular beats are considered as non-PVC beats while unclassified (Q) and aberrated beats are omitted from analysis. ECG signals from leads MLII and lead-II are used for MITDB and INCARTDB databases respectively.

Table 4.4. MITDB Dataset split for Training and Validation

Dataset	Records
MITDB-DS1 (training)	101,106,108,109,112,114,115,116,118,119,122, 124,201,203,205,207,208,209,215,220,223,230
MITDB-DS2 (validation)	100,103,105,111,113,117,121,123,200,202,210, 212,213,214,219,221,222,228,231,232,233,234

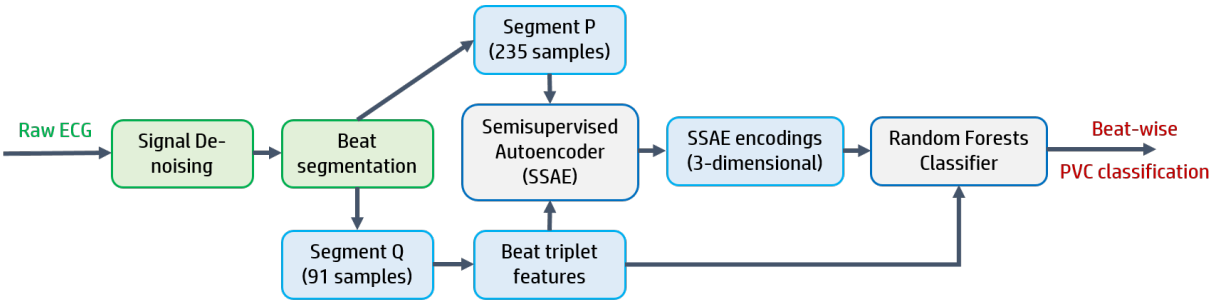

Figure 4.7. Block diagram for PVC classification

4.4.2 Signal Pre-processing

The techniques described in Chapter 2 (Sections 2.2.2 and 2.3.2) are used to denoise the ECG signal (baseline wander removal and high frequency noise suppression). Following this, electrode motion (EM) noise classification is performed as per Section 2.4 in Chapter 2. Subsequently beat detection (see Chapter 3) is performed and these beat locations are then used to create the features required for accurate PVC classification.

4.4.3 Feature extraction

The algorithm developed in this work requires atleast three consecutive beats for performing PVC classification, owing to the manner in which the features are computed. For training, validating and testing the models used in PVC classification, R-peak locations are used from the corresponding manual annotations available at Physionet for records in each dataset. For each R-peak, two ECG beat segments, namely P and Q, are derived from the denoised ECG signal. Beat segment P is obtained using a window of 250ms prior to the R-peak location and 400ms afterward (a total of 235 samples at 360Hz sampling frequency). Similarly, beat segment Q is obtained using a window of 125ms prior to the R-peak location and 125ms afterward (91 samples). These two segments are used to create inputs for training a semisupervised autoencoder.

Beat-triplet features

As it can be observed, the autoencoder has two input layers. The first set of input is simply the 235 samples of beat segment P. The second set of inputs to the autoencoder comprises four features computed from three consecutive R-peaks (r_i, r_{i-1}, r_{i-2}):

(i) RR-interval in terms of heart rate (HR). RR-interval is simply the duration between the two consecutive R-peaks given by (4.9). For each R-peak r_i; this feature is computed as follows:

$$rr_i = r_i - r_{i-1} \tag{4.9}$$

$$HR_i = (Fs * 60)/rr_i \tag{4.10}$$

where, Fs is the sampling frequency i.e., 360Hz.

(ii) Successive Difference of RR-intervals in percentage (%) ($SDRR$). This is the percentage change in duration between two consecutive RR intervals:

$$SDRR_i = \frac{rr_i - rr_{i-1}}{rr_{i-1}} * 100 \tag{4.11}$$

(iii) Correlation Coefficient ($R_{i,i-1}$). This feature is computed using beat segments Q_i and Q_{i-1} that correspond to R-peaks r_i and r_{i-1}. For one-dimensional vectors Q_i and Q_{i-1}; this feature is computed as below (Johnson and Wichern, 2002):

$$Cov(X,Y) = \frac{\sum_{k=1}^{m}(X_k - \overline{X})(Y_k - \overline{Y})}{m-1} \tag{4.12}$$

$$R_{i,i-1} = \frac{Cov(Q_i, Q_{i-1})}{\sqrt{Cov(Q_i, Q_i)Cov(Q_{i-1}, Q_{i-1})}} \tag{4.13}$$

where m in (4.12) is the number of samples in Q_i (or Q_{i-1}).

(iv) The last feature is the Skewness of Cross-correlation (SCC). As the name implies, this is skewness (Zwillinger and S.Kokoska, 2000) of the cross-correlation sequence computed between segments Q_i and Q_{i-1}. Cross-correlation ($Corr_{X,Y}$) of two sequences, X and Y, is computed as:

$$Corr_{X,Y}[l] = \sum_k X[k]Y[t+l] \tag{4.14}$$

where l is the l^{th} sample in $Corr_{X,Y}$. The length of $Corr_{X,Y}$ is $m_X + m_Y - 1$, where m_X and m_Y are the number of samples in X and Y respectively. SCC_i is then computed as follows:

$$\mu_3(X) = \frac{1}{M}\sum_{k=1}^{M}(X_k - \overline{X})^3 \tag{4.15}$$

$$Skew(X) = \frac{\mu_3(X)}{(Cov(X,X))^{3/2}} \tag{4.16}$$

$$SCC_i = Skew(Corr_{Q_i,Q_{i-1}}) \tag{4.17}$$

where M is the number of samples in X, $\mu_3(X)$ is the third central moment of X (Zwillinger and S.Kokoska, 2000). Additionally, the sequence $Corr_{Q_i,Q_{i-1}}$ is scaled in the range [-1,1] before using it in (4.17).

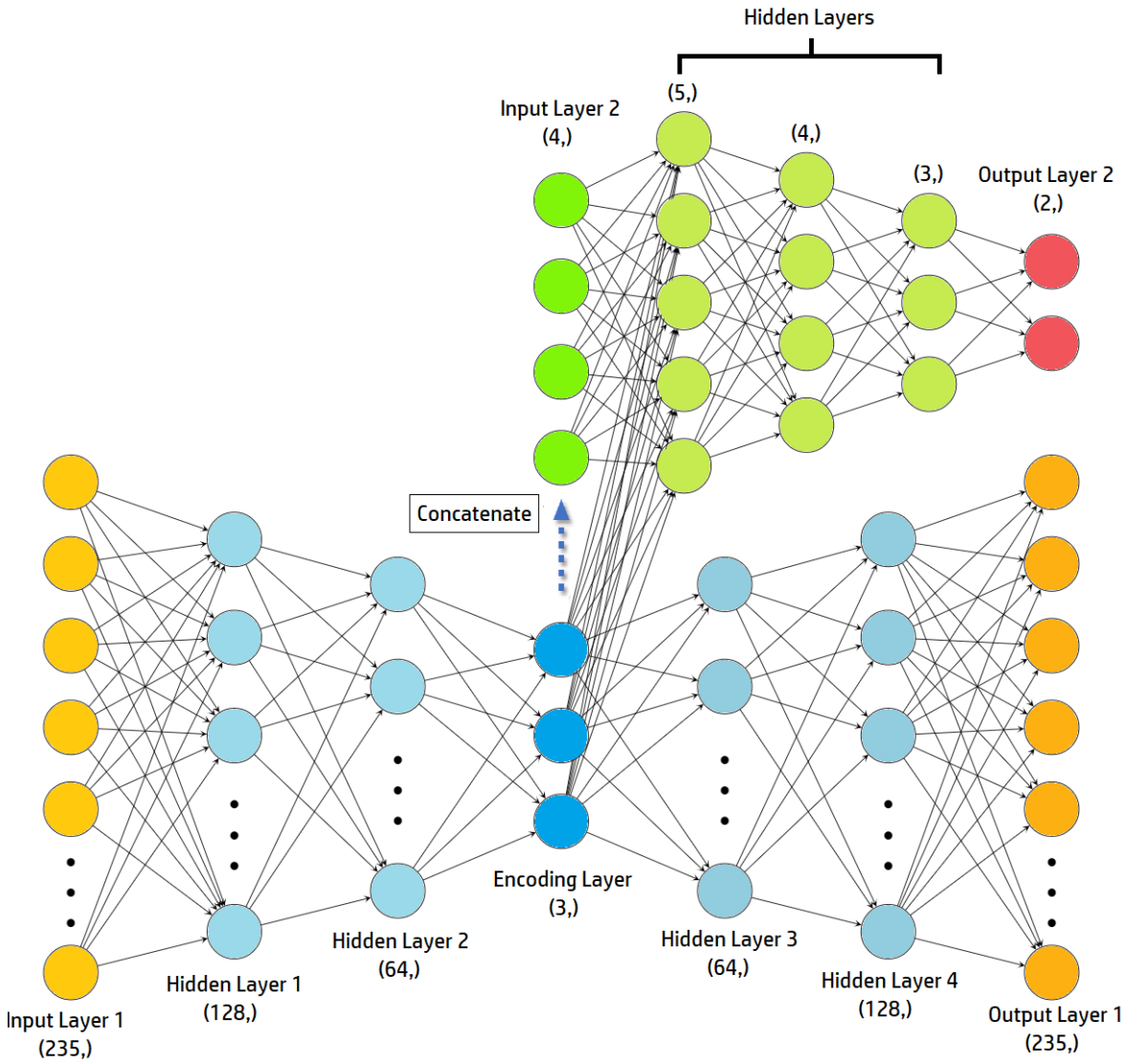

Figure 4.8. Architecture of the Semisupervised Autoencoder network

Semi-supervised Autoencoder (SSAE) network

In this work, a neural-network based semisupervised autoencoder is implemented to efficiently learn three-dimensional encodings of individual ECG beat morphologies that will be used as features for final PVC classification. The network is semisupervised in that the architecture consists of an autoencoder component (unsupervised learning) (Goodfellow et al., 2016) and a binary classifier neural network (supervised learning) (Goodfellow et al., 2016) embedded together. Unlike traditional autoencoders, the architecture here actually uses manually computed features (*Input Layer 2* in Figure 4.8) along with raw ECG data (*Input Layer 1* in Figure 4.8). This helps the network to not only learn low-dimensional encodings of beat morphologies but also ensures that these encodings are suitable to be used as features for distinguishing PVC from non-PVC beats. It should be noted that the binary classifier component used here is for generating class-specific encodings only and not for final PVC classification. The overall architecture of the SSAE network is shown in Figure 4.8. The input to *Input Layer 1* is the raw ECG samples of segment P (scaled in the range [-1,1]), described in Section 4.4.3. The input to *Input Layer 2* consists of the beat-triplet features (Section 4.4.3). Input Layer 2 is concatenated with the output of Encoding Layer to form the overall input to the binary classifier component (see Figure 4.8). The target data for *Output Layer 1* (autoencoder's decoder output) is identical to the input of *Input Layer 1*. *Output Layer 2* is the binary classifier component's output and its target data corresponds to the output class (label) of each beat i.e., PVC or non-PVC. Ideally, the output of this layer should be $[0, 1]^T$ for PVC beats and $[1, 0]^T$ otherwise.

SSAE-derived features

The records in the MITDB-DS1 dataset are used for training the SSAE network. This hybrid network is trained end-to-end using backpropagation (Rumelhart et al., 1986).

ReLU (Rectified Linear Units) (Goodfellow et al., 2016) activation function is used for the hidden layers while "tanh" and "softmax" activations (Goodfellow et al., 2016) are applied to *Output Layer 1* and *Output Layer 2* respectively. Layer weights that gave the best performance, in terms of F-score, on the validation dataset (MITDB-DS2) are selected and the SSAE network with these layer weights is labeled $SSAE_{best}$. The three-dimensional encodings (output of Encoding Layer) from the $SSAE_{best}$ model constitute the SSAE-derived features.

4.4.4 PVC Classification

The SSAE-derived features (see Section 4.4.3) along with beat-triplet features (see Section 4.4.3), all computed from records in MITDB-DS1 dataset, are used for training a binary Random Forests model for detecting PVC beats. Hyper-parameter tuning using Grid Search is employed for selecting the best Random Forests model for PVC classification. The hyperparameter tuning search space for Random Forests included four primary parameters as given in Table 4.5. Explanations for the hyperparameters are the same as provided in Section 4.3.4.

Table 4.5. Hyperparameter tuning search space for PVC Random Forest model

Hyperparameter	**Range of values**	**Best value**
Number of trees	[10,11,12,...,250]	**162**
Maximum tree depth	[5,10,15,..., 50, `None`]	**None**
Maximum features	[`auto`, 7] (`auto` : square root of number of features)	**auto**
Class weight	['balanced', None]	**'balanced'**

This concludes the section on PVC detection algorithm description. The occurrence of PVCs in certain specific patterns gives rise to different ventricular arrhythmias. Since these patterns are clearly defined in medical literature (Aehlert, 2018), simple logical analysis that searches for these patterns is implemented to detect the presence of these arrhythmias. It must be remembered that this logical analysis step is based on beat label information obtained in the PVC classification stage and hence the detection accuracy for these arrhythmias inherently depends on the accuracy of PVC detection, which is provided later in Section 4.4.10. These arrhythmias and the logic required to detect them are briefly discussed below.

4.4.5 Ventricular Bigeminy

Ventricular Bigeminy (VBI) refers to the occurrence of alternating PVC and non-PVC beats. To confirm ventricular bigeminy, presence of atleast three consecutive pairs of Non-PVC/PVC beats is required. If PVC beats are denoted as `V` and non-PVC beats are denoted as `N`, then the algorithm searches for the pattern `[NVNVNV...]`. Hence atleast six beats are required to detect presence of VBI episodes. The occurrence of two consecutive `N` beats marks the end of a ventricular bigeminy episode. To detect another episode, the algorithm once again searches for the sequence `[NVNVNV...]` in the remaining beats. Once the sequence is found, it marks the onset of another VBI episode and continues scanning until it encounters two consecutive `N` beats which marks the offset of this new episode. This process is repeated for all the remaining beats in the input ECG signal. Figure 4.9 shows an example of a ventricular bigeminy episode.

4.4.6 Ventricular Trigeminy

Ventricular Trigeminy (VTRI) refers to the occurrence of a PVC beat after every two consecutive non-PVC beats. To confirm ventricular trigeminy, presence of atleast three

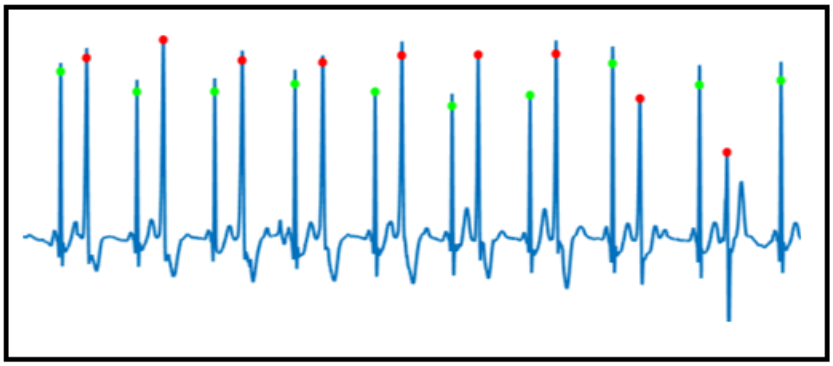

Figure 4.9. Example of Ventricular Bigeminy episode. The red markers indicate PVC beats and green markers indicate Non-PVC beats

consecutive triplets of the form `[NNV]` is required. To detect a Ventricular Trigeminy episode, the algorithm searches for the pattern `[NNVNNVNNV...]`. Hence atleast nine beats are required for detection of VTRI episodes. The occurrence of any beat-triplet other than `[NNV]` beat-triplet marks the end of a ventricular trigeminy episode. To detect another episode, the algorithm once again searches for the sequence `[NNVNNVNNV...]` in the remaining beats. Once the sequence is found, it marks the onset of another VTRI episode and continues scanning until it encounters a beat-triplet other than `[NNV]` which marks the offset of this new episode. This process is repeated for all the remaining beats in the input ECG signal. Figure 4.10 shows an example of a ventricular trigeminy episode.

4.4.7 Ventricular Quadrigeminy

Ventricular Quadrigeminy (VQUAD) refers to the occurrence of a PVC beat after every three consecutive non-PVC beats. To confirm ventricular quadrigeminy, presence of atleast three consecutive quadruplets of the form `[NNNV]` is required. To detect a ventricular trigeminy episode, the algorithm searches for the pattern `[NNNVNNNVNNNV...]`. Hence atleast twelve beats are required for detection of VQUAD episodes. The occurrence of any beat-quadruplet other than `[NNNV]` beat-triplet marks the end of a ventricular quadrigeminy

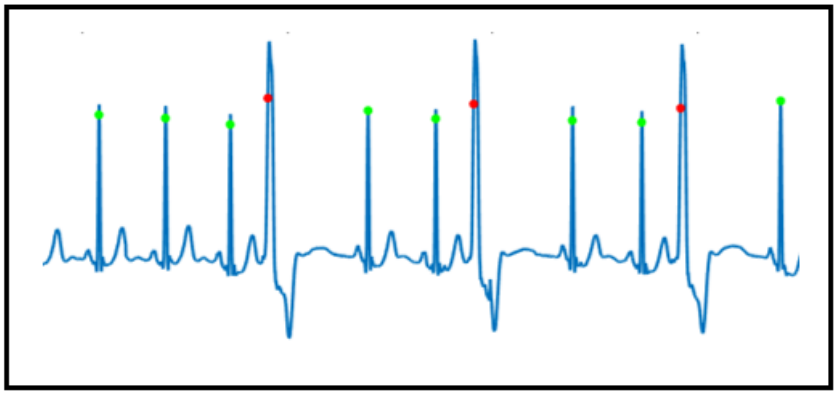

Figure 4.10. Example of Ventricular Trigeminy episode. The red markers indicate PVC beats and green markers indicate Non-PVC beats

episode. To detect another episode, the algorithm once again searches for the sequence `[NNNVNNNVNNNV...]` in the remaining beats. Once the sequence is found, it marks the onset of another VQUAD episode and continues scanning until it encounters a beat-quadruplet other than `[NNNV]` which marks the offset of this new episode. This process is repeated for all the remaining beats in the input ECG signal. Figure 4.11 shows an example of a ventricular quadrigeminy episode.

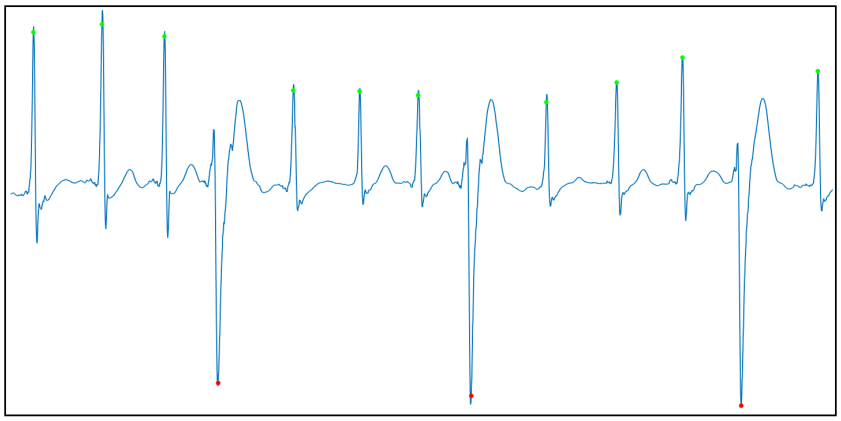

Figure 4.11. Example of Ventricular Quadrigeminy episode. The red markers indicate PVC beats and green markers indicate Non-PVC beats

4.4.8 Ventricular Runs

Ventricular runs refer to occurrence of three or more consecutive PVCs at heart rates greater than 100 bpm. At heart rates greater than 120 bpm, these rhythms are also known as Short Ventricular Tachycardia episodes. In fact, when ventricular runs persist for a duration greater than 30s at heart rates exceeding 150 bpm, it leads to sustained VT episodes.

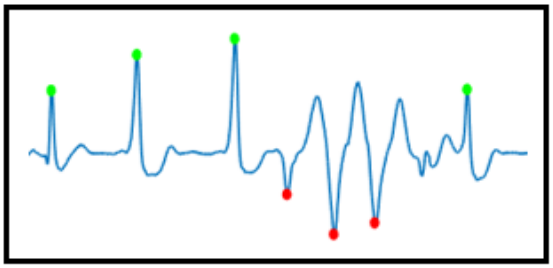

Figure 4.12. Example of Ventricular Run episode. The red markers indicate PVC beats and green markers indicate Non-PVC beats

To detect ventricular runs, the algorithms searches for groups of three or more consecutive PVC beats. For every group of three or more consecutive PVC beats, the mean heart rate is computed as follows:

$$RR_{group} = \frac{1}{n-1}\sum_{k=1}^{n-1} rr_k \tag{4.18}$$

$$HR_{group} = \frac{Fs * 60}{RR_{group}} \tag{4.19}$$

where,

- n is the number of PVCs in the beat group.
- rr_k is the k^{th} RR-interval in the group, computed using (4.9).
- Fs is the sampling frequency.

If the value of HR_{group} exceeds 100 bpm, that group of beats is labeled as a ventricular run episode. Figure 4.12 shows an example of a ventricular run episode.

4.4.9 Ventricular Couplets

Ventricular couplets or pairs refer to occurrence of exactly two consecutive PVCs at heart rates greater than 100 bpm. To detect ventricular couplets, the algorithm simply searches for the presence of `[VV]` pattern in the beats and each such pair is marked as a ventricular couplet episode. Figure 4.13 shows an example of a ventricular couplet episode.

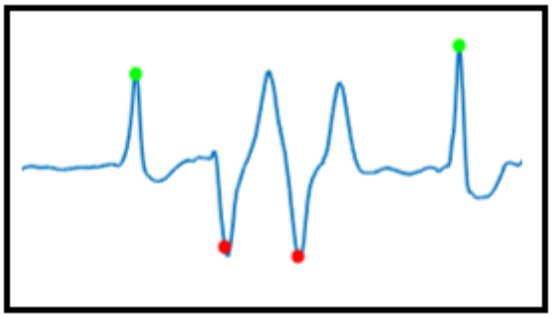

Figure 4.13. Example of Ventricular Couplet episode. The red markers indicate PVC beats and green markers indicate Non-PVC beats

Table 4.6. Dataset information for PVC classification

Type	Dataset	PVC	Non PVC	Total
Training	MITDB–DS1	3680	47055	50735
Validation	MITDB-DS2	3218	46370	49588
Test	INCARTDB	19990	155684	175674

4.4.10 Results

The performance of the algorithm was evaluated on the MITDB and INCARTDB databases. The MITDB-DS1 dataset has 3680 PVC and 47055 non-PVC examples. Similarly, the MITDB-DS2 dataset consists of 3218 PVC and 46370 non-PVC examples while the INCARTDB dataset consists of 19990 PVC and 155684 non-PVC examples. This information is summarized in Table 4.6. Table 4.7 shows the performance scores of the algorithm. The results are provided in terms of Sensitivity (Se), positive predictive value (PPV) and F-Score (FSc). These metrics are computed using (4.5) through (4.7). It can be seen that our algorithm achieves a sensitivity of 100.00%, PPV of 98.55% and FSc of 99.27% on the training (MITDB-DS1) dataset. Similarly it achieves a sensitivity of 92.67%, PPV of 95.88% and FSc of 94.10% on the validation (MITDB-DS2) dataset while achieving a sensitivity of 88.08%, PPV of 94.76% and FSc of 91.30% on the test (INCARTDB) dataset. Since the SSAE network also has a binary classifier component, initial PVC classification performance using the SSAE network was evaluated. The corresponding results are shown in Table 4.8. It can be seen from this table that using the SSAE network alone results in decreased values of precision (PPV) and subsequently decreased F-Score (FSc) values compared to the final results in Table 4.7. This was the primary motivation behind using the additional Random Forests model for final PVC classification. The accuracy statistics in Table 4.7 and Table 4.8 reflect the performance of the PVC classification on features derived using manual R-peak annotations for the MITDB and INCARTDB records. Since this dissertation is focused on developing a fully automated arrhythmia analysis system, it is imperative that the above described algorithm performs well on R-peaks obtained using an automated R-peak detection approach. In that regard, Table 4.9 shows the performance of the PVC classification algorithm using R-peaks that were determined by the beat detection algorithm presented in Chapter 3. The results in this table are shown for all the records in each database.

Figure 4.17 shows a two-dimensional mapping of the seven PVC features. This 2D-representation is computed using the popular visualization technique called *t-SNE*, which short for t-Distributed Stochastic Neighbor Embedding ((Maaten and Hinton, 2008)). From this plot it can be seen that the PVC and non-PVC representations form (almost) disjoint class-specific clusters thus indicating the usefulness of the seven features for distinguishing PVC beats from non-PVC beats.

Training dataset (MITDB – DS1) confusion matrix			
	pvc	non pvc	
PVC	3680 (TP)	0 (FN)	3680
NON PVC	54 (FP)	47001 (TN)	47055
	3734	47001	50735

Figure 4.14. Confusion matrix depicting model performance on training dataset

Validation dataset (MITDB – DS2) confusion matrix			
	pvc	non pvc	
PVC	2982 (TP)	236 (FN)	3218
NON PVC	138 (FP)	46232 (TN)	46370
	3120	46468	49588

Figure 4.15. Confusion matrix depicting model performance on validation dataset

Figures 4.14, 4.15 and 4.16 show the confusion matrices depicting the performance of the PVC classification model on the training, validation and test datasets respectively. In these figures, the rows(upper case `PVC` and `NON PVC`) represent the true labels and the columns represent the predicted labels (lower case `pvc` and `non pvc`).

Table 4.7. Performance evaluation on MITDB and INCARTDB Datasets using the SSAE-RF model

Dataset	**Se**	**PPV**	**FSc**
MITDB-DS1 (Training dataset)	100.00%	98.55%	99.27%
MITDB-DS2 (Validation dataset)	92.67%	95.58%	94.10%
MITDB (Overall)	96.58%	97.20%	96.89%
INCARTDB (Test dataset)	88.08%	94.76%	91.30%

Table 4.8. Performance Evaluation on MITDB and INCARTDB Datasets using only the SSAE network

Dataset	**Se**	**PPV**	**FSc**
MITDB-DS1 (Training dataset)	96.37%	85.93%	91.03%
MITDB-DS2 (Validation dataset)	96.02%	83.33%	91.07%
MITDB (Overall)	96.42%	84.70%	90.18%
INCARTDB (Test dataset)	92.26%	89.90%	91.07%

Test dataset (INCARTDB) confusion matrix			
	pvc	non pvc	
PVC	17607 (TP)	2383 (FN)	19990
NON PVC	973 (FP)	154711 (TN)	155684
	18580	157094	175674

Figure 4.16. Confusion matrix depicting model performance on test dataset

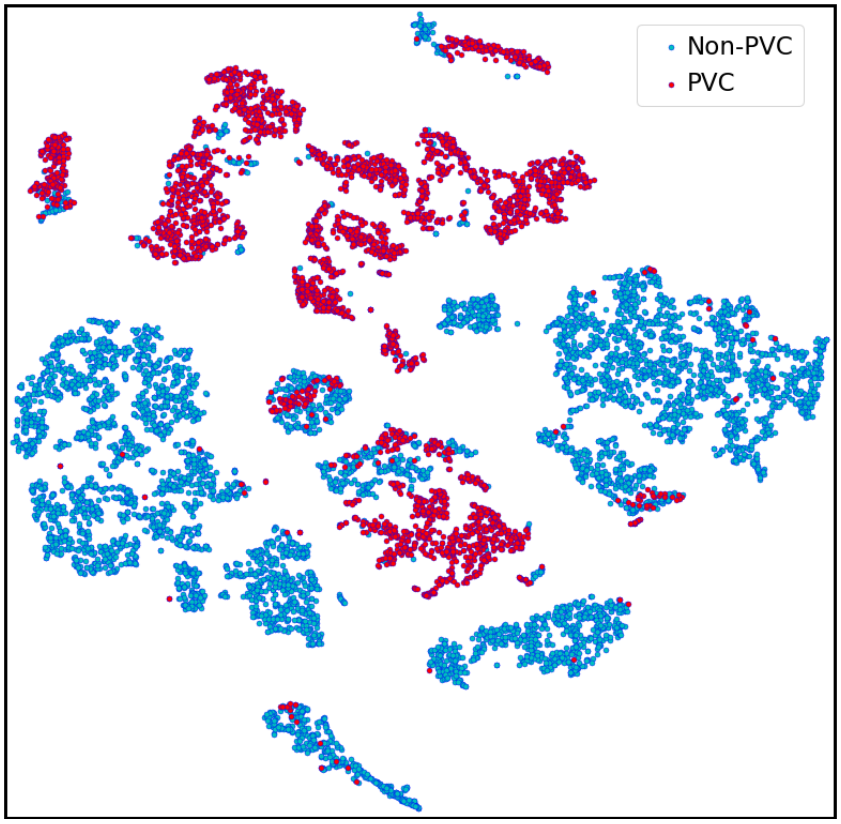

Figure 4.17. Scatterplot showing 2D-representation of the seven PVC features using t-SNE.

Figure 4.18 shows an example of PVC detection in a 5-second ECG segment. The top plot shows the denoised ECG with the algorithm's PVC and non-PVC classifications along with reference PVC labels. The bottom plot shows the classification scores of the Random

Table 4.9. Performance evaluation on MITDB and INCARTDB Datasets using algorithm-based R-peak annotations

Dataset	Se	PPV	FSc
MITDB	93.17%	94.41%	93.79%
INCARTDB	88.55%	89.06%	88.80%

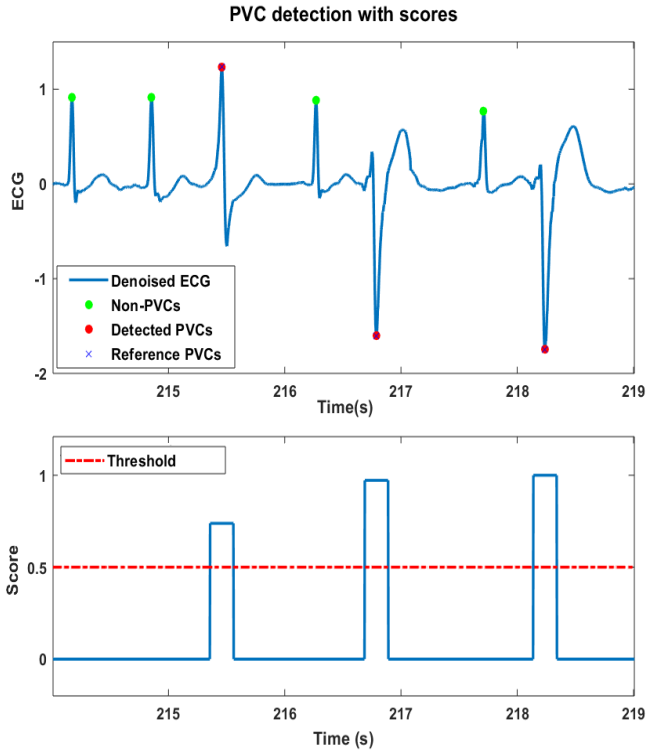

Figure 4.18. Detection of multi-form PVCs in a 5-second ECG segment extracted from record 200 in the MITDB-DS2 dataset.

Forests model for each beat. Scores above the 0.5 threshold correspond to PVC beats. It can be seen that the PVCs in this example have varied morphologies (the first one has positive QRS polarity while the remaining two PVCs exhibit negative QRS polarity), and

Table 4.10. Comparison with State-of-the-art Methods for records in the MITDB-DS2 Dataset

Method	**Se**	**PPV**	**FSc**
(Llamedo and Martínez, 2011)	83.0%	88.0%	85.4%
(Llamedo and Martinez, 2012)	89.0%	87.0%	88.0%
(de Chazal and Reilly, 2006)[a]	94.3%	94.3%	94.3%
(Oster et al., 2015)[b]	92.7%	96.2%	94.5%
(Kiranyaz et al., 2016)[c,d]	93.9%	90.6%	92.2%
(Ortín et al., 2019)[d]	95.4%	88.8%	92.0%
SSAE-RF classifier (this dissertation)	**92.7%**	**95.6%**	**94.1%**

[a]500 beats per record were used for training
[b]Semi-supervised approach with expert assistance
[c]First five minutes per record were used for training
[d]Results are for entire MITDB database (MITDB-DS1 and MITDB-DS2)

the algorithm detects all of them. This demonstrates the algorithm's ability to robustly detect multiform PVCs.

Comparison with other methods

We compared the performance of the algorithm developed in this work with other state-of-the-art techniques in literature for PVC classification. Table 4.10 and Table 4.11 show the comparison statistics for MITDB-DS2 and INCARTDB databases respectively.It must be noted that for the purpose of comparisons, results provided in Table 4.7 are used as the other works report their results based on manual R-peak annotations. It can be seen that the PVC classification algorithm described in this chapter performs comparably or better

Table 4.11. Comparison with State-of-the-art Methods for records in the INCARTDB Dataset

Method	**Se**	**PPV**	**F1**
(Llamedo and Martínez, 2011)[a,b]	82.0%	88.4%	85.1%
(Llamedo and Martinez, 2012)[a,b]	88.0%	96.0%	91.8%
(Oster et al., 2015)[b,c]	95.4%	99.3%	97.3%
SSAE-RF classifier (this dissertation)	**88.1%**	**94.8%**	**91.3%**

[a]Fusion beats are considered as PVCs
[b]Multi-lead ECG information used (leads II and V1)
[c]Semi-supervised approach with expert assistance

than the other methods. It has to be noted the algorithms of (Oster et al., 2015) are not automatic (they involve expert assistance) while the authors of (Llamedo and Martinez, 2012) use multi-lead (leads II and V1) rather than single-lead ECG information from the INCARTDB records in their approach. Also the authors of (de Chazal and Reilly, 2006) and (Kiranyaz et al., 2016) include a subset of beats from the MITDB-DS2 dataset as part of their training data (see footnotes for Table 4.10 and Table 4.11).

4.5 Conclusion

In summary, this chapter presents algorithms for automated detection of ventricular arrhythmias. Beat-independent arrhythmias such as ventricular fibrillation and sustained ventricular tachycardia are detected using time-domain and feature-domain features along with SWT analysis. These features are input to a Random Forests classifier to achieve accurate VT/VF detection performance on VFDB and AHADB ECG records. Following description of VT/VF rhythms, a novel technique for automatic detection of beat-

dependent arrhythmias based on PVC beats is presented. A combination of semisupervised autoencoders and random forests is used for achieving accurate PVC detection results on the MITDB and INCARTDB ECG records. The PVC detection approach described in this chapter demonstrates superior performance over current-state-of-the art techniques that utilize only single-lead ECG data and that do not require expert assistance. A limitation of the proposed PVC detection approach is the requirement that the ECG recording belong to lead-II. The SSAE network is trained with ECGs recorded using lead-II configuration and since ECG beat morphologies could vary with different lead orientations, the accuracy may vary accordingly (negative-polarity QRS complexes such as in lead V1 might result in false positives, for instance). Since lead-II configuration is the primarily used setup for gaining insights into QRS complex (beat) morphology in single-lead ECG systems, the algorithm's preference for using data from lead-II configuration is justified. Using the beat labels from the PVC classification stage, logical analysis-based techniques for detection of additional five ventricular arrhythmias viz., ventricular bigeminy, ventricular trigeminy, ventricular quadrigeminy, ventricular runs and ventricular couplets, is presented. This concludes the description of algorithms for detecting ventricular arrhythmias. The next chapter provides in-depth analysis of algorithms that focus on detecting supraventricular arrhythmias.

References

Addison, P. S., J. N. Watson, G. R. Clegg, M. Holzer, F. Sterz, and C. E. Robertson (2000). Evaluating arrhythmias in ecg signals using wavelet transforms. *IEEE Engineering in Medicine and Biology Magazine 19*(5), 104–109.

Aehlert, B. J. (2018). *ECGs Made Easy* (6 ed.). Elsevier.

Ahn, M.-S. (2013). Current concepts of premature ventricular contractions. *Journal of lifestyle medicine 3*(1), 26.

Alonso-Atienza, F., E. Morgado, L. Fernández-Martínez, A. Garcia-Alberola, and J. Rojo-Alvarez (2012). Combination of ecg parameters with support vector machines for the detection of life-threatening arrhythmias. In *2012 Computing in Cardiology*, pp. 385–388. IEEE.

Amann, A., R. Tratnig, and K. Unterkofler (2005). Reliability of old and new ventricular fibrillation detection algorithms for automated external defibrillators. *Biomedical engineering online 4*(1), 60.

ANSI/AAMI EC57:2012 (2012). Testing and reporting performance results of cardiac rhythm and st segment measurement algorithms.

Balasundaram, K., S. Masse, K. Nair, T. Farid, K. Nanthakumar, and K. Umapathy (2011). Wavelet-based features for characterizing ventricular arrhythmias in optimizing treatment options. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 969–972. IEEE.

Barro, S., R. Ruiz, D. Cabello, and J. Mira (1989). Algorithmic sequential decision-making in the frequency domain for life threatening ventricular arrhythmias and imitative artefacts: a diagnostic system. *Journal of biomedical engineering 11*(4), 320–328.

Breiman, L. (2001a). Random forests. *Machine learning 45*(1), 5–32.

Breiman, L. (2001b). Random forests. *Machine Learning 45*(1), 5–32.

Chen, S., N. V. Thakor, and M. M. Mower (1987). Ventricular fibrillation detection by a regression test on the autocorrelation function. *Medical and Biological Engineering and Computing 25*(3), 241–249.

Christov, I., I. Jekova, and G. Bortolan (2005). Premature ventricular contraction classification by the kth nearest-neighbours rule. *Physiological measurement 26*(1), 123.

Clayton, R., A. Murray, and R. Campbell (1994). Recognition of ventricular fibrillation using neural networks. *Medical and Biological Engineering and Computing 32*(2), 217–220.

Clayton, R. H., A. Murray, and R. W. Campbell (1993). Comparison of four techniques for recognition of ventricular fibrillation from the surface ecg. *Medical and Biological Engineering and Computing 31*(2), 111–117.

Clifford, G. D., I. Silva, B. Moody, Q. Li, D. Kella, A. Shahin, T. Kooistra, D. Perry, and R. G. Mark (2015). The physionet/computing in cardiology challenge 2015: reducing false arrhythmia alarms in the icu. In *2015 Computing in Cardiology Conference (CinC)*, pp. 273–276. IEEE.

De Chazal, P., M. O'Dwyer, and R. B. Reilly (2004). Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE transactions on biomedical engineering 51*(7), 1196–1206.

de Chazal, P. and R. B. Reilly (2006, Dec). A patient-adapting heartbeat classifier using ecg morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering 53*(12), 2535–2543.

Fallet, S., S. Yazdani, and J.-M. Vesin (2015). A multimodal approach to reduce false arrhythmia alarms in the intensive care unit. In *2015 Computing in Cardiology Conference (CinC)*, pp. 277–280. IEEE.

Goldberger, A. L., L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley (2000). Physiobank, physiotoolkit, and physionet. *Circulation 101*(23), e215–e220.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hastie, T., R. Tibshirani, and J. Friedman (2009a). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hastie, T., R. Tibshirani, and J. Friedman (2009b). *Random Forests*, pp. 587–604. New York, NY: Springer New York.

Hu, Y. H., W. J. Tompkins, J. L. Urrusti, and V. X. Afonso (1993). Applications of artificial neural networks for ecg signal detection and classification. *Journal of electrocardiology 26*, 66–73.

Ince, T., S. Kiranyaz, and M. Gabbouj (2009). A generic and robust system for automated patient-specific classification of ecg signals. *IEEE Transactions on Biomedical Engineering 56*(5), 1415–1426.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An introduction to statistical learning*, Volume 112. Springer.

Jekova, I. (2000). Comparison of five algorithms for the detection of ventricular fibrillation from the surface ecg. *Physiological measurement 21*(4), 429.

Jekova, I. (2007). Shock advisory tool: Detection of life-threatening cardiac arrhythmias and shock success prediction by means of a common parameter set. *Biomedical Signal Processing and Control 2*(1), 25–33.

Johnson, R. A. and D. W. Wichern (2002). *Matrix Algebra and Random Vectors*, pp. 50–83. Upper Saddle River, New Jersey: Prentice Hall.

Kalidas, V. and L. Tamil (2016). Cardiac arrhythmia classification using multi-modal signal analysis. *Physiological measurement 37*(8), 1253.

Kennedy, H. L., J. A. Whitlock, M. K. Sprague, L. J. Kennedy, T. A. Buckingham, and R. J. Goldberg (1985). Long-term follow-up of asymptomatic healthy subjects with frequent and complex ventricular ectopy. *New England Journal of Medicine 312*(4), 193–197.

Kiranyaz, S., T. Ince, and M. Gabbouj (2016, March). Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering 63*(3), 664–675.

Krasteva, V. and I. Jekova (2005). Assessment of ecg frequency and morphology parameters for automatic classification of life-threatening cardiac arrhythmias. *Physiological measurement 26*(5), 707.

Kuo, S. (1978). Computer detection of ventricular fibrillation. *Proc. of Computers in Cardiology, IEEE Comupter Society*, 347–349.

Lagerholm, M., C. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo (2000). Clustering ecg complexes using hermite functions and self-organizing maps. *IEEE Transactions on Biomedical Engineering 47*(7), 838–848.

Li, Q., C. Rajagopalan, and G. D. Clifford (2013). Ventricular fibrillation and tachycardia classification using a machine learning approach. *IEEE Transactions on Biomedical Engineering 61*(6), 1607–1613.

Llamedo, M. and J. P. Martinez (2012, Aug). An automatic patient-adapted ecg heartbeat classifier allowing expert assistance. *IEEE Transactions on Biomedical Engineering 59*(8), 2312–2320.

Llamedo, M. and J. P. Martínez (2011, March). Heartbeat classification using feature selection driven by database generalization criteria. *IEEE Transactions on Biomedical Engineering 58*(3), 616–625.

Maaten, L. v. d. and G. Hinton (2008). Visualizing data using t-sne. *Journal of machine learning research 9*(Nov), 2579–2605.

Moody, G. B. and R. G. Mark (2001, May). The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine 20*(3), 45–50.

Moraes, J. C. T. d. B., M. Blechner, F. N. Vilani, and E. Costa (2002). Ventricular fibrillation detection using a leakage/complexity measure method. In *Computers in Cardiology*, pp. 213–216. IEEE.

Mozaffarian, D., E. J. Benjamin, A. S. Go, D. K. Arnett, M. J. Blaha, M. Cushman, S. de Ferranti, J.-P. Després, H. J. Fullerton, V. J. Howard, M. D. Huffman, S. E. Judd, B. M. Kissela, D. T. Lackland, J. H. Lichtman, L. D. Lisabeth, S. Liu, R. H. Mackey, D. B. Matchar, D. K. McGuire, E. R. Mohler, C. S. Moy, P. Muntner, M. E. Mussolino, K. Nasir, R. W. Neumar, G. Nichol, L. Palaniappan, D. K. Pandey, M. J. Reeves, C. J. Rodriguez, P. D. Sorlie, J. Stein, A. Towfighi, T. N. Turan, S. S. Virani, J. Z. Willey, D. Woo, R. W. Yeh, and M. B. Turner (2015). Heart disease and stroke statistics 2014 - 2015 update. *Circulation 131*(4), e29–e322.

Ng, G. A. (2006). Treating patients with ventricular ectopic beats. *Heart 92*(11), 1707–1712.

Ortín, S., M. C. Soriano, M. Alfaras, and C. R. Mirasso (2019). Automated real-time method for ventricular heartbeat classification. *Computer methods and programs in biomedicine 169*, 1–8.

Osowski, S. and T. H. Linh (2001). Ecg beat recognition using fuzzy hybrid neural network. *IEEE Transactions on Biomedical Engineering 48*(11), 1265–1271.

Oster, J., J. Behar, O. Sayadi, S. Nemati, A. E. W. Johnson, and G. D. Clifford (2015, Sep.). Semisupervised ecg ventricular beat classification with novelty detection based on switching kalman filters. *IEEE Transactions on Biomedical Engineering 62*(9), 2125–2134.

Plesinger, F., P. Klimes, J. Halamek, and P. Jurak (2015). False alarms in intensive care unit monitors: detection of life-threatening arrhythmias using elementary algebra, descriptive statistics and fuzzy logic. In *2015 Computing in Cardiology Conference (CinC)*, pp. 281–284. IEEE.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature*, 533–536.

Sayadi, O., M. B. Shamsollahi, and G. D. Clifford (2009). Robust detection of premature ventricular contractions using a wave-based bayesian framework. *IEEE Transactions on Biomedical Engineering 57*(2), 353–362.

Senhadji, L., G. Carrault, J. Bellanger, and G. Passariello (1995). Comparing wavelet transforms for recognizing cardiac patterns. *IEEE Engineering in Medicine and Biology Magazine 14*(2), 167–173.

Thakor, N. V., Y.-S. Zhu, and K.-Y. Pan (1990). Ventricular tachycardia and fibrillation detection by a sequential hypothesis testing algorithm. *IEEE Transactions on Biomedical Engineering 37*(9), 837–843.

Watson, J. N., P. S. Addison, G. R. Clegg, M. Holzer, F. Sterz, and C. E. Robertson (2000). A novel wavelet transform based analysis reveals hidden structure in ventricular fibrillation. *Resuscitation 43*(2), 121–127.

Yeap, T., F. Johnson, and M. Rachniowski (1990). Ecg beat classification by a neural network. In *[1990] Proceedings of the Twelfth Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1457–1458. IEEE.

Zhang, X.-S., Y.-S. Zhu, N. V. Thakor, and Z.-Z. Wang (1999). Detecting ventricular tachycardia and fibrillation by complexity measure. *IEEE Transactions on biomedical engineering 46*(5), 548–555.

Zubair, M., J. Kim, and C. Yoon (2016). An automated ecg beat classification system using convolutional neural networks. In *2016 6th international conference on IT convergence and security (ICITCS)*, pp. 1–5. IEEE.

Zwillinger, D. and S.Kokoska (2000). *CRC Standard Probability and Statitics Tables and Formulae*. New York: Chapman & Hill.

CHAPTER 5

SUPRAVENTRICULAR ARRHYTHMIAS

5.1 Introduction

Supraventricular arrhythmias refer to cardiac rhythms that originate in the cardiac cells above the ventricles. This includes the atrio-ventricular (AV) node, atria and the Sino-Atrial (SA) node. The electrical impulses may be abnormally conducted from any of these sites. When the atria initiate cardiac cycles by abnormally firing electrical impulses, it results in occurrence of Premature Atrial Complexes (PACs). On the other hand, when these ectopic beats are initiated in the AV node, this results in occurrence of Premature Junctional Complexes (PJCs). Although the SA node is the heart's natural pacemaker, still there may be irregularities associated with its impulse discharge. For example, the SA node may fire slower than usual due to conduction blocks which could result in Sinus Bradycardia. Similarly, abnormally rapid electrical discharges from the SA node could result in presence of Sinus Tachycardia. In this dissertation, no distinction is made between PACs and PJCs in terms of detection and are together termed **Supraventricular Beats (SVEBs)**. The supraventricular arrhythmias discussed in this chapter are as follows:

1. Atrial Fibrillation
2. Supraventricular Ectopic Beats
3. Supraventricular Bigeminy
4. Supraventricular Trigeminy
5. Supraventricular Quadrigeminy
6. Supraventricular Couplets

7. Supraventricular Runs

8. Sinus Bradycardia

Although atrial fibrillation (AF) is theoretically a sequence of supraventricular ectopic beats and hence a supraventricular arrhythmia, in this dissertation, AF is detected using fixed-length ECG segments (60 epochs) whereas SVEBs and remaining arrhythmias are detected on a beat-by-beat basis. Hence AF detection is carried out first followed by detection of SVEBs and other arrhythmias*.

5.2 Atrial fibrillation

The American Heart Association defines atrial fibrillation (AF) as a "supraventricular tachyarrhythmia that is characterized by uncoordinated atrial activation with consequent deterioration of mechanical function" (January et al., 2014; Kirchhof et al., 2016). Multiple theories have been put forth to explain the trigger mechanisms for atrial fibrillation. The common thread to all these theories is the existence of multiple focal points of atrial excitation, instead of just the Sino-Atrial node, which is the heart's natural pacemaker. These multiple focal points are a result of impulse re-entry, which is viewed as a disorder of impulse propagation, rather than impulse formation (Veenhuyzen et al., 2004). This leads to disorganized atrial activity resulting in stochastic changes in heart rates. Figure 5.1) shows the electrical activity and corresponding ECG manifestation during AF†. In the presence of AF, an individual's heart rate fluctuates in a highly irregular manner

*Several portions of this chapter have been reproduced/adapted with permission from material published in (Kalidas and Tamil, 2019): *V. Kalidas and L. Tamil, "Detection of Atrial Fibrillation Using Discrete-State Markov Models and Random Forests," Computers in Biology and Medicine, Oct 2019, vol. 113, doi:10.1016/j.compbiomed.2019.103386.*

†Image reprinted with permission from Mayo Foundation for Medical Education and Research.

and the atrial rates can vary anywhere between 240 and 300 beats per minute (bpm). Figure 5.2 shows an example of ECG segment with AF episode. It can be seen that the RR-interval behavior is random during AF presence. AF episodes can be sustained or self-terminating and are usually grouped into various categories depending on episode duration (January et al., 2014). Occurrence of two or more episodes of AF is termed *Recurrent AF* whereas episodic occurrences that terminate spontaneously within seven days are termed as *Paroxysmal AF*. When the arrhythmia lasts for more than seven days, it is categorized as *Persistent AF* and episodes lasting more than a year are labeled as *Permanent AF*. Although there does exist a significant number of instances where AF is not associated with any other detectable cardiac dysfunction, for about two-thirds of patients, AF is usually associated with other cardiac abnormalities (Zoni-Berisso et al., 2014). AF is also known to contribute to a five-fold increase in strokes and can turn life-threatening if not properly diagnosed and treated. This is especially the case in the presence of asymptomatic AF, also known as *silent AF*, where no visible symptoms are present. AF episodes often lead to other disorders apart from stroke such as heart failure, hemodynamic impairment and thromboembolic events (January et al., 2014). AF also impacts cognitive functioning, leading to decreased quality of life and increased healthcare costs (Zoni-Berisso et al., 2014). AF is one of the most commonly occurring arrhythmias, affecting about 1–2% of the general population and its occurrence varies with age and gender. The incidence proportion of AF is between 0.12% and 0.16% among individuals younger than 49 years, approximately 4% in individuals in the age group of 60–70 years and about 10%–17% among those older than 80 years (Kirchhof et al., 2016; Zoni-Berisso et al., 2014). Also, the prevalence of AF is slightly higher in males compared to females (a ratio of 1.2:1) (Kirchhof et al., 2016), and paroxysmal AF is more prevalent in younger adults. In 2010, 20.9 million men and 12.6 million women were estimated to suffer from AF, with increased prevalence in developed countries. It is expected that 25% of middle–aged adults

in the U.S. and Europe will develop AF and by the year 2030, 14–17 million individuals in U.S. are estimated to be affected by AF. The rate of prevalence of AF in U.S. is expected to increase with 120,000 to 215,000 patients being additionally diagnosed every year. This increase in AF prevalence can be attributed to multiple factors including aging population, improved diagnosis techniques and advanced patient monitoring systems that help detect silent and paroxysmal AF episodes better, along with significant strides made in understanding the science behind atrial fibrillation (Zoni-Berisso et al., 2014). Automated methods for classification of AF episodes generally rely on the information extracted from electrocardiogram (ECG) signals. The absence of periodically occurring P-waves or presence of fibrillatory f-waves in ECG (seen as undulations of the isoelectric baseline) coupled with irregular heart rate fluctuations are primary indicators of AF. Despite the significant progress made in understanding the factors contributing to occurrence of atrial fibrillation episodes, development of automated techniques to detect AF episodes remains far from achieving satisfactory results due to several factors. First, there are several other arrhythmias that typically mimic AF in terms of their manifestation on the ECG as well as possessing similarities in terms of spectral content, extent of heart rate variability, etc. Secondly, the presence of external noise, especially owing to electrode/patient movements, hinders the performance of AF classifiers severely, giving rise to increased misclassification rates, and this gains further significance in the context of today's wearable sensors. Currently, there is a growing interest in the use of wearable sensors for ECG recording and subsequent arrhythmia monitoring (Zhao et al., 2018). The modern-day wearable sensors offer cost-effective and hassle-free solutions for vital monitoring purposes, but are extremely susceptible to noise interference and are highly sensitive to motion artifacts. These aforementioned factors necessitate the need to develop an AF classifier that is highly robust to noise while being capable of accurately identifying AF rhythms, especially in the presence of other similar arrhythmias. These challenges serve as the primary motivation for the work

presented here. In this chapter, an RR-interval analysis technique to detect AF episodes from single-lead ECG signals is presented. The discrete-state transitions in the RR-interval time series are modeled as an eight-state Markov process. The use of Markov models offers the advantage that sequential pattern changes in heart rates can be effectively captured, thus aiding in better distinction between AF and other arrhythmias with prominent heart rate variations. These Markov probabilities, along with other statistical parameters that help quantify randomness in RR-interval transitions, are input to a Random Forests based AF classifier for initial AF classification.

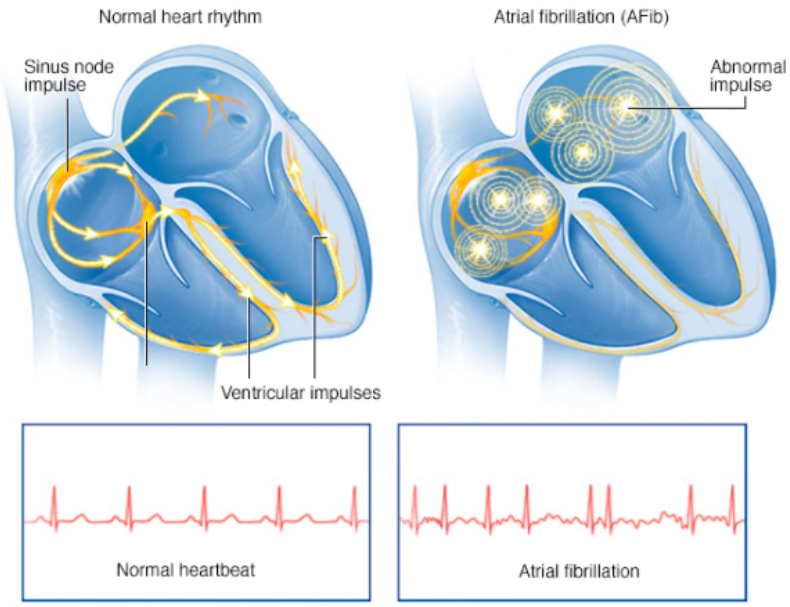

Figure 5.1. Electrical activity and ECG manifestation during Atrial Fibrillation. (Source: `mayoclinic.org`. ©Mayo Foundation for Medical Education and Research.)

5.2.1 Literature Review

Several algorithms have been proposed in literature for detection of atrial fibrillation episodes. Automated AF detection techniques usually involve analyzing ECG signals to

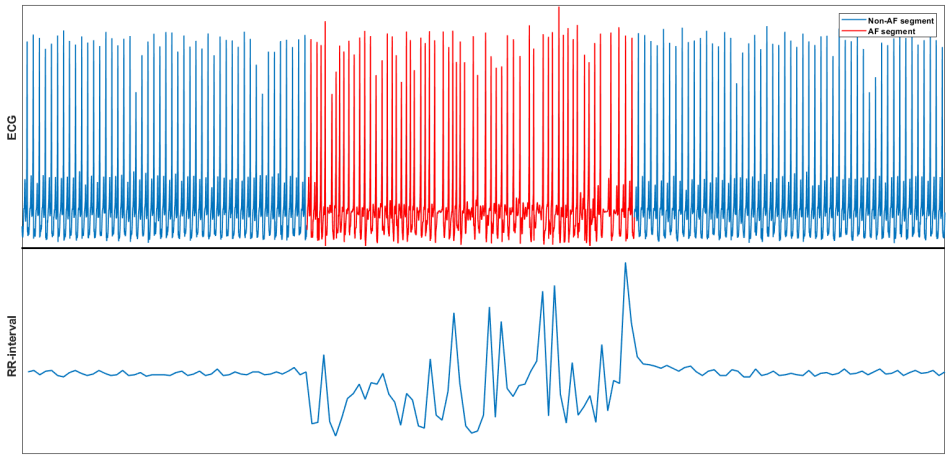

Figure 5.2. RR-interval behavior during AF presence. *Top.* ECG segment with AF episode. *Bottom.* Corresponding RR-interval series.

estimate the degree of irregularity in the RR-interval sequence, which is a characteristic feature of AF (other than in the presence of paced heart rates). In that regard, the authors of (Carrara et al., 2015) propose use of linear and non-linear heart rate dynamics to distinguish AF rhythms from normal sinus rhythms and sinus rhythms with frequent ectopic beats. In another RR-interval based approach (Moody, 1983), the sequence of RR intervals during AF is modeled as a three-state Markov process, namely short, regular and long. The RR-intervals were assigned to one of these states based on the relation of the current RR-interval with the exponentially averaged mean RR-interval. Histogram analysis on distributions of RR intervals and first difference of RR-intervals are the basis of the approach adopted in (Tateno and Glass, 2001), where reference distributions were created for AF and non-AF classes and at test time, the RR-interval distributions were matched with these reference distributions using Kolmogorov-Smirnov test for goodness-of-fit (Stephens, 1974). Other ventricular rate-based methods include studying Lorenz plots (Sarkar et al., 2008) of RR intervals, where successive RR-intervals are scatter-plotted against each other. The scatter-plots are more spread out in the presence of AF whereas they stay in one or more

tight clusters in its absence. AF classification methods based purely on evaluation of RR-interval entropy have also been found to perform reasonably well in the past (Lake and Moorman, 2011; Liu et al., 2018; Zhao et al., 2018). These entropy-based approaches rely on the template-matching (vector similarity) property of RR-segments in an ECG epoch, where lack of template similarity under a given tolerance indicates presence of AF rhythms (Lake and Moorman, 2011; Liu et al., 2018; Zhao et al., 2018). The advantage of using RR-interval based techniques is that the determination of R-peak locations and subsequent computation of heart rate metrics are robust to most noise sources except electrode motion artifacts. On the other end of the spectrum, heart-rate independent approaches do not include any beat detection procedures but they utilize information from atrial activity and such techniques include analyzing P-wave absence (Ladavich and Ghoraani, 2015), measuring average F-wave activity (Du et al., 2014), analyzing wavelet features (Ródenas et al., 2015; Asgari et al., 2015) and implementing echo state neural networks for QRST cancellation (Petrenas et al., 2012). Use of P-wave and f-wave information can help prevent false detections in the presence of arrhythmias with continuously varying heart rates. Both heart-rate based methods and heart-rate-independent techniques have their limitations. The ECG lead orientation often results in mitigated amplitudes of regular and normal P-waves. The distinction between f-waves and isoelectric baseline noise is rendered difficult by frequent overlap of their spectral bandwidths, resulting in poor atrial activity analysis. On the other hand, examination of irregularities in the ventricular heart rate is complicated in the presence of other arrhythmias with significant heart rate variations such as frequently occurring premature ventricular contractions (PVCs) and non-arrhythmic factors such as electrode motion artifacts that often mimic QRS complex morphologies. These factors can render QRS detection algorithms inaccurate at times. Over the past decade or so, with rapid advances in data collection, storage and GPU capabilities, multiple techniques using deep learning as well as ensemble learning methods have been proposed for AF classification.

Most notably, the annual Physionet/Computing in Cardiology Challenge for 2017 (Clifford et al., 2017) focused on identifying AF episodes in short term ECG recordings. The top-ranked results of this challenge and their respective methods are described in (Hong et al., 2017; Teijeiro et al., 2017; Zabihi et al., 2017; Datta et al., 2017). Methods described in (Hong et al., 2017; Teijeiro et al., 2017) leverage deep learning techniques to develop an AF classifier capable of performing reliably in the presence of other arrhythmias. The algorithm discussed in (Zabihi et al., 2017) describes an approach based on training a Random Forests classifier on heart rate dependent features for AF classification. Cascaded binary classifiers using the Adaboost learning algorithm(Freund and Schapire, 1995) form the basis of the work described in (Datta et al., 2017). Apart from the algorithms proposed for the challenge, several other contributions in the field of automated AF classification have also adopted deep learning-based approaches lately. Recently, the authors of (Rajpurkar et al., 2017) have proposed a convolutional neural network (CNN) based deep learning approach to distinguish different arrhythmias. Their model was trained using privately collected ECG data and its performance was evaluated on the MITDB database. In another deep learning related approach (Xia et al., 2018), stationary wavelet transforms and CNN architectures were combined to train a classifier capable of identifying AF episodes in very short ECG segments (5s long). Although deep learning techniques offer an attractive alternative to hand-computed feature extraction, limited availability of labeled AF datasets acts as a major bottleneck for training robust deep learning models for AF classification. In comparison, the approach described in this chapter towards AF classification is characterized primarily by implementing a combination of Markov models and Random Forests classifiers to perform noise assessment and RR-interval based AF classification. The inconsistency in the detection of P-waves and delineation of fibrillatory f-waves from noisy ECG recordings contributed to omitting atrial activity analysis in the approach described in this chapter. The use of an eight-state Markov matrix for AF

classification, complemented by Random-Forest based feature-learning ensures that the AF detection algorithm can effectively distinguish AF from other arrhythmias. As can be observed in Section 5.2.7, the algorithm achieves comparable or better performance than other state-of-the-art AF classification algorithms on the standard databases. The AF detection algorithm, along with results and discussion, is described in the following sections.

5.2.2 Datasets

The Random Forests model used for AF classification was trained, validated and tested using features derived from datasets available in the Physionet database (Goldberger et al., 2000). For training the AF classification model, ECG records from the MIT-BIH Atrial Fibrillation Database (AFDB) (Moody, 1983) were used. On the other hand, ECG records from the MIT-BIH Arrhythmia Database (MITDB) (Moody and Mark, 2001) were used to form the test dataset. More information regarding the databases is provided in Section 5.2.6.

5.2.3 Signal Pre-processing

The techniques described in Chapter 2 (Sections 2.2.2 and 2.3.2) are used to denoise the ECG signal (baseline wander removal and high frequency noise suppression). Following this, electrode motion (EM) noise classification is performed as per Section 2.4 in Chapter 2. Subsequently beat detection (see Chapter 3) is performed to determine R-peak locations. This is succeeded by ventricular arrhythmia detection as described in Chapter 4.

5.2.4 Features for initial AF classification

The R-peak indices i.e., beat locations, labeled r_{pk}, are used to create RR-interval analysis based features vectors to detect AF presence. The derivation of these features is described subsequently.

First, the input signal is segmented into 60s epochs with a 30s sliding window. For each 60s epoch, the corresponding R-peaks stored in r_{pk} are used to create feature vectors using the steps described below.

1. The RR-interval time-series is computed as follows:

$$rr_{n-1} = r_n - r_{n-1}, \quad 2 \leq n \leq N \tag{5.1}$$

   where,

   - N is the total number of R-peaks in the epoch

   - r_n is index of n^{th} R-peak in the epoch - rr is the RR-interval series. It must be noted that R-peaks classified as PVCs are excluded from this computation.

2. The successive difference of RR-intervals (ΔRR) in percentage values, rr_{per}, is next calculated as follows:

$$rr_{per(n-1)} = 100 * \frac{(rr_n - rr_{n-1})}{rr_{n-1}}, \qquad 2 \leq n \leq N-1 \tag{5.2}$$

3. The rr_{per} series is discretized into 8 different states as per Table 5.1. This discretized series is labeled as rr_d with the rr_{per} boundaries for each state having been determined empirically.

Table 5.1. State assignment table for ΔRR transitions.

ΔRR values (rr_{per})	**State (rr_d)**
(-∞, -50)	1
[-50, -30)	2
[-30, -15)	3
[-15, +15)	4
[+15, +45)	5
[+45, +75)	6
[+75, +100)	7
[+100, +∞)	8

AF features I - Markov matrix features

A 8-by-8 discrete-state Markov matrix **B** for the rr_d state sequence is computed as follows:

$$\forall i, j \in [1, ..., 8]$$

$$B_{ij} = P(next_{state} = j | current_{state} = i) \tag{5.3}$$

$$\sum_{j=1}^{8} B_{ij} = 1, \quad 1 \leq i \leq 8 \tag{5.4}$$

In the presence of AF, the Markov matrix **B** is more densely populated than in the presence of other arrhythmia with similar magnitude of heart rate variation, such as ventricular bigeminy/trigeminy/quadrigeminy, etc (see Chapter 4). This can be attributed to the fact that the variations between successive RR-interval values are random in AF episodes and do not follow any specific pattern. On the other hand the variations in heart

rates for ventricular bigeminy/trigeminy, etc. is more regular and deterministic in which case the matrix B is sparsely populated.

Figure 5.3 shows an ECG segment completely dominated by AF rhythm. The corresponding Markov matrix (bottom plot) has values that are spread about the center of the matrix in a largely random fashion, indicating lack of discernible patterns in the heart rate variations. On the other hand, Figure 5.4 represents a 60s long ECG epoch containing episodes of ventricular trigeminy, ventricular bigeminy and normal sinus rhythm coupled with a few ectopic beats. Although there is significant amount of heart rate variation here as well, evidenced by the rr_{per} series (in the middle subplot), the corresponding Markov matrix in the bottom plot looks vastly different. Here, most of the transition probability values are concentrated near the left-bottom and top-right corners of the matrix indicating the presence of specific sequential patterns in the RR-interval series. The elements of this 8-by-8 discrete-state Markov matrix form the first 64 features for AF classification.

The other feature obtained from the Markov matrix **B** is the total number of zero-valued elements present in it. This quantity reflects a measure of sparsity of the Markov matrix and in the presence of AF, this value is expected to be remain minimal.

AF features II - other RR-interval features

Apart from the features derived from the Markov matrix as described above, eight other RR-interval based features are also computed to help distinguish AF from other similar rhythms effectively. They are as follows:

1. Entropy of the rr_d state sequence, which indicates the extent of randomness in the RR-interval transitions. This entropy measure is computed using natural logarithm as follows:

$$e_{ss} = (-1) * \sum_{k=1}^{8} [p(s_k) * log_e p(s_k)] \tag{5.5}$$

where, e_{ss} is entropy of the rr_d state sequence and $p(s_k)$ is the probability of occurrence of k^{th} state in the rr_d state sequence.

2. The p-value of Kolmogorov-Smirnov test (Stephens, 1974) for normality, applied to the rr_{per} series. The null hypothesis (H=0) is that the rr_{per} series follows a normal distribution which is the case for AF. The test is conducted at significance level of 5% and the p-values when the AF is present is generally much greater than 0.05 (Tateno and Glass, 2001).

3. Coefficient of variation of the RR-interval series, computed as per the equation below:

$$coeff_{var} = \frac{\sigma_{RR}}{\mu_{RR}} * 100 \tag{5.6}$$

where, σ_{RR} is the standard deviation of RR-interval series without the smallest and largest intervals. Similarly, μ_{RR} is the mean of the RR series without the smallest and largest intervals (Tateno and Glass, 2001).

4. The remaining two features are based on heart-rate quartiles derived from the RR-interval series. They include the (i) difference between second and first quartiles, and (iv) the difference between third and second quartiles.

5. All the above features together form a 70-element feature vector for each epoch. Subsequently, an [M x 70] matrix is created where M is the total number of 60s epochs in the ECG signal (with 30s slide interval). This feature matrix is fed as input to a Random Forests based classifier that classifies each row of the feature matrix (i.e., each epoch) as either exhibiting AF presence or not.

5.2.5 AF Classification

The 70 features derived in Section 5.2.4 are used for training a binary Random Forests classifier for detecting PVC beats. Five-fold cross validation is performed and hyperparameter tuning using Grid Search is employed for selecting the best model parameters for AF classification. The hyperparameter tuning search space for Random Forests included four primary parameters as given in Table 5.2. Explanations for the hyperparameters are the same as provided in Section 4.3.4.

Table 5.2. Hyperparameter tuning search space for AF Random Forests model

Hyperparameter	**Range of values**	**Best value**
Number of trees	[10,11,12,...,250]	**116**
Maximum tree depth	[5,10,15,..., 50, `None`]	**None**
Maximum features	[`auto`, 5,10,15,...,70] (`auto` : square root of number of features)	**15**
Class weight	['balanced', None]	**'None'**

5.2.6 Results

The performance of the AF detection algorithm was evaluated on the MIT-BIH Atrial Fibrillation Database (AFDB) and the MIT-BIH Arrhythmia Database (MITDB). The AFDB database (Moody, 1983) has 23 two-lead ECG records, each sampled at 250 Hz and having a duration of approximately 10 h (except record 06453 which has a duration of approximately 9 h). The MITDB database (Moody and Mark, 2001) contains 48 two-lead ECG records, each sampled at 360 Hz and 30 min long. Out of these 48 records, eight

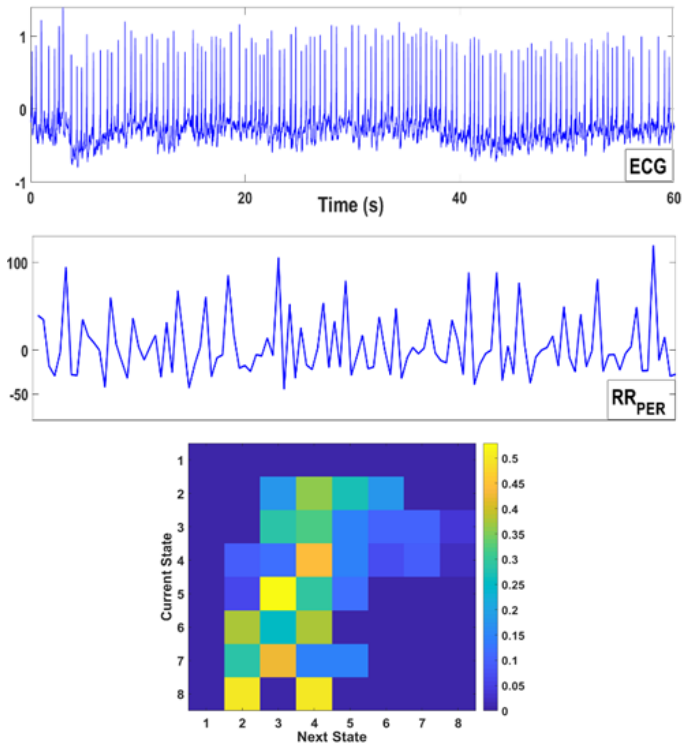

Figure 5.3. Atrial Fibrillation true positive classification

Table 5.3. Five-fold cross validation results for all Grid search combinations

Fold	**Training set F-Score**	**Validation set F-score**
Fold 1	$98.68\% \pm 0.51\%$	$98.17\% \pm 0.23\%$
Fold 2	$98.69\% \pm 0.53\%$	$98.10\% \pm 0.18\%$
Fold 3	$98.79\% \pm 0.48\%$	$97.39\% \pm 0.26\%$
Fold 4	$98.70\% \pm 0.52\%$	$98.06\% \pm 0.21\%$
Fold 5	$98.68\% \pm 0.51\%$	$98.26\% \pm 0.25\%$

records have substantial AF presence. Signals from lead ECGI for AFDB and lead MLII for MITDB databases are used respectively. evaluation. Atrial Flutter and Junctional

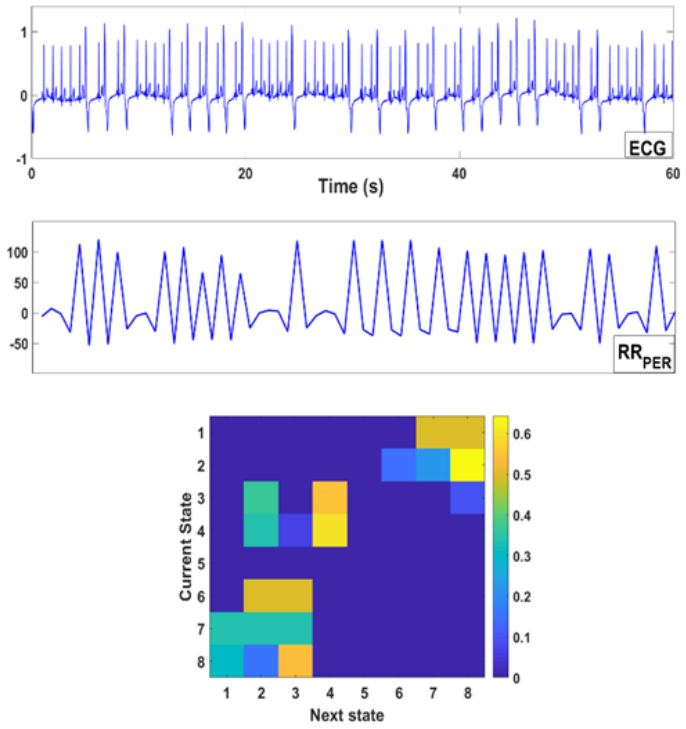

Figure 5.4. True negative classification of ventricular arrhythmias

arrhythmias in the AFDB records are grouped as non-AF in this evaluation. Since 60s epochs are used in the AF detection algorithm here, it would be unfair to compare the results with beat-to-beat annotations. Hence, instead the true beat-to-beat annotations are converted into 60s-epoch annotations. A 60s epoch was annotated as having AF only if at least 50% of the beats in that epoch were originally annotated as AF beats. Using this approach, 135 AF epochs and 1304 non-AF epochs were obtained for the MITDB database. Similarly, 5528 AF epochs and 8226 non-AF epochs were obtained for the AFDB database.

The cross-validation results for all combinations in the Grid Search are presented in Table 5.3. From these the best combination of parameters, shown in last column of Table 5.2 was chosen and an RF model with these parameters was retrained on the entire AFDB training dataset and labeled AF_{model}. The evaluation results of AF_{model} are summarized

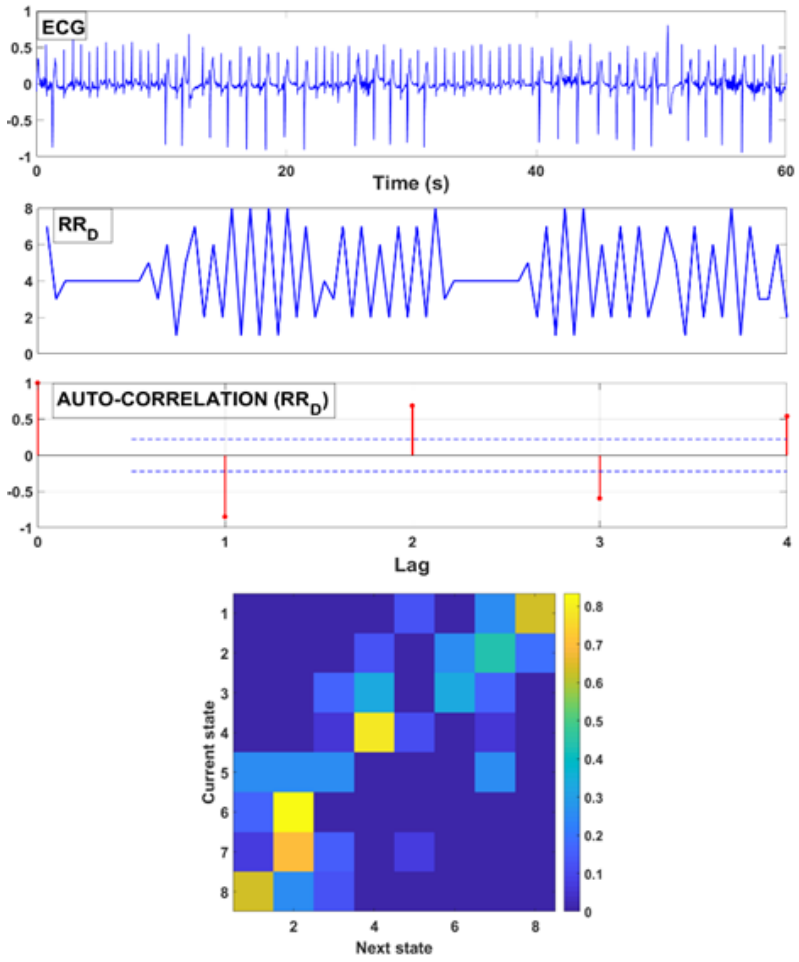

Figure 5.5. False positive suppression of ventricular arrhythmias

in terms of sensitivity (Se), Specificity (Sp), positive predicitive value and F-score (FSc) metrics in Table 5.5. The equations for computing the above three evaluation metrics are as follows:

$$Se = \frac{TP}{TP + FN} * 100 \tag{5.7}$$

$$Se = \frac{TN}{TN + FP} * 100 \tag{5.8}$$

Table 5.4. Dataset information for AF classification

Type	Dataset	AF	Non AF	Total
Training	AFDB	5528	8226	13754
Test	MITDB	135	1304	1439

$$PPV = \frac{TP}{TP + FP} * 100 \tag{5.9}$$

$$FSc = \frac{2 * TP}{(2 * TP) + FP + FN} * 100 \tag{5.10}$$

where, Se refers to sensitivity, Sp refers to specificity and FSc refers to FScore for AF detection. TP refers to the number of true positives i.e., correctly detected AF segments, FN refers to number of false negatives i.e., AF epochs misclassified as non-AF, TN refers to the number of true negatives i.e., correctly detected non-AF segments and FP refers to the number of false positives, i.e., non-AF epochs misclassified as AF.

A sensitivity of 96.88%, specificity of 99.26%, PPV of 98.87% and an F-Score of 97.87% was obtained on the AFDB database. Similarly, a sensitivity of 99.26%, a specificity of 97.10%, PPV of 77.91% and an F-Score of 87.30% was obtained on the MITDB database. The relatively lower F-scores on the MITDB database compared to the AFDB database can be attributed to the high imbalance in the proportion of non-AF and AF annotations (nearly 10:1).

5.2.7 Comparison with other AF methods

The performance of the above-described AF detection algorithm is compared with other state-of-the-art methods whose results have been reported on the AFDB and/or MITDB

Table 5.5. Performance evaluation on AFDB and AFDB Datasets

Dataset	**Se**	**Sp**	**PPV**	**FSc**
AFDB (Training dataset)	96.88%	99.26%	98.87%	99.27%
MITDB (Test Dataset)	99.26%	97.10%	77.91%	87.30%

databases. These statistics are provided in Table 5.2.7. It can be seen that the AF_{model} performs comparably or better than thee other algorithms. The first three methods listed after the first method (AF_{model}) in Table 5.2.7 are heart-rate dependent and do not utilize atrial activity information. In fact, the work presented in (Moody, 1983) is one of the earliest publications in the field of automated AF analysis. Their algorithm considers a three-state (short, regular and long) Markov model for analyzing the RR-interval sequence and the transition probabilities are compared with that of reference AF transition probabilities for rhythm identification. The authors explain that the use of a Markov model in isolation results in unacceptably high false positive rates and hence they include a filtering and interpolation stage along with an ectopic beat removal step to improve their accuracy. Although reducing data storage, the definition of only three states for RR-intervals could sometimes lead to missing crucial heart rate variation information which can consequently render it difficult to distinguish AF from other similar rhythms. In the method proposed in (Tateno and Glass, 2001), the distribution of δRR intervals for the ECG signal is tested against reference distributions of δRR intervals corresponding to AF and non-AF rhythms respectively. Kolmogorov-Smirnov goodness-of- fit tests were used for AF detection. This approach needs storage of reference distributions, thus requiring additional memory. Another potential drawback is that reference distributions as such do not capture the sequential patterns in δRR intervals that typically distinguish other

Table 5.6. Comparison with other state-of-the-art methods

Method	Remarks	Database	Se	Sp
AF_model (this dissertation)	**HRD[a]**	**AFDB**	**96.8%**	**99.2%**
		MITDB	**99.2%**	**97.1%**
(Moody, 1983)	HRD[a]	AFDB	93.5%	N/A
		MITDB	N/A	N/A
(Tateno and Glass, 2001)	HRD	AFDB	94.4%	97.2%
		MITDB	88.2%	87.0%
(Dash et al., 2009)	HRD	AFDB	94.4%	95.1%
		MITDB	90.2%	91.2%
(Asgari et al., 2015)	HRI[b]	AFDB	97.0%	97.1%
		MITDB	N/A	N/A
(Ladavich and Ghoraani, 2015)	HRI	AFDB	98.1%	91.7%
		MITDB	N/A	N/A
(Babaeizadeh et al., 2009)	HRD+AA[c]	AFDB	93.0%	98.0%
		MITDB	N/A	N/A
(Xia et al., 2018)	HRD+AA	AFDB	98.8%	97.9%
		MITDB	N/A	N/A

[a]Heart Rate Dependent methods
[b]Heart Rate Independent methods
[c]AA indicates atrial activity analysis

rhythms from AF. This could result in increased false positive rates. The authors of (Dash et al., 2009) use statistical techniques to capture the beat-to- beat variability to aid detect AF. These include Root Mean Square of Successive RR Differences (RMSSD), turning points ratio (TPR) and Shannon entropy metrics. They reason that using these statistical tools ensures that their algorithm is less reliant on the diversity of the training data. They also include an ectopic beat filtering step to remove premature beats. The corresponding results shown in Table 5.2.7 are after ectopic beat removal. Although not shown in Table 5.2.7, they report an increase of around 22% in specificity on the MITDB database (200 series) and a decrease of around 6% in sensitivity after ectopic beat removal (Dash et al., 2009). The approaches of the next two methods are purely entropy based, where AF classification is based on entropy measures corresponding to template matching of RR-segments (Liu et al., 2018; Zhao et al., 2018). The results of (Liu et al., 2018) indicate a higher sensitivity than our approach, but this comes at a significantly lower specificity for both the databases under consideration in Table 5.2.7, especially the MITDB database. The authors carried out analysis for three window types, namely, 12-beat, 30-beat and 60-beat windows. In Table 5.2.7, we have shown the results for only the 60-beat windows which provided the best accuracy amongst their three window choices. For the MITDB database, their results are provided separately for the MITDB- 100 series and MITDB-200 series in their paper. Therefore, we recomputed the specificity for the overall MITDB database using the statistics provided in Tables 1 and 2 in (Liu et al., 2018). The sensitivity on the MITDB database is used directly from Table 1 in (Liu et al., 2018) as this reflects their overall sensitivity. The algorithms in (Zhao et al., 2018) are also developed using similar entropy-based techniques. Their results are computed for 30-beat windows and it can be seen that their accuracy on the AFDB database is lower than that of the AF classification algorithm developed in this dissertation. The results on the MITDB database were not available for their work. The subsequent two methods i.e., that of (Ladavich and

Ghoraani, 2015; Asgari et al., 2015) are rate-independent and hence do not perform any beat detection. They report similar performance metrics as ours on the AFDB database. The slightly higher sensitivity (1%) reported by (Ladavich and Ghoraani, 2015) can be attributed to the use of a smaller window size of 7 beats for AF detection, compared to 60s epochs used by algorithm developed in this dissertation. But the use of this smaller window size results in decreased specificity (91.7%) and PPV (79.2%), as reported by the authors. The results of (Asgari et al., 2015) are based on 2-fold cross-validation on the AFDB database and evaluations on the MITDB database have not been reported. The next two methods listed in Table 5.2.7 i.e., that of (Babaeizadeh et al., 2009; Couceiro et al., 2008) implement both beat detection as well as atrial activity analysis in their methods. As it can be seen, our algorithm has a much higher sensitivity than theirs on the AFDB database. Additionally, the authors in (Moody, 1983) do not report a specificity for their method. Instead they report a positive predictive value (PPV) of 85.9% on the AFDB database. Similarly, PPV metrics of 96.1%, 79.2% and 98% on the AFDB database are reported by (Tateno and Glass, 2001; Ladavich and Ghoraani, 2015; Babaeizadeh et al., 2009) respectively. In comparison, AF_{model} achieves a PPV of 98.87% for the AFDB database. A deep learning-based approach listed at the end of Table 5.2.7 (Xia et al., 2018), consists of Deep Convolutional Neural Network based AF classifiers. They use 5s epochs for AF detection that can help detect very short AF episodes. They propose two kinds of inputs for their deep learning models, (i) using Short-Term Fourier Transforms and (ii) using SWT, respectively. Both the techniques provide similar accuracies. Their results are based on ten-fold cross-validation performed on the AFDB database. A sensitivity of 98.3% and a specificity of 98.2% is reported on the AFDB database, which is slightly better than our sensitivity and slightly lower than our specificity. It would be interesting to see their results on the MITDB database as well which would help in assessing their algorithm's ability to generalize across multiple databases. In fact, most algorithms in Table

5.2.7 report metrics evaluated on the AFDB database (usually cross-validated) and not on other databases such as the MITDB database, thus offering limited information regarding generalization capabilities of their respective techniques. Other recent contributions in the field of automated AF detection also use deep learning (Hong et al., 2017; Teijeiro et al., 2017; Rajpurkar et al., 2017) and ensemble approaches (Datta et al., 2017) to develop AF classifiers. In the deep-learning based approach of(Rajpurkar et al., 2017), the authors trained a CNN-based arrhythmia detector using privately collected ECG data that achieved an AF detection F-score of 66.7% on the MITDB database. It is worth noting that their classifier was trained to classify fourteen different arrhythmias and not just atrial fibrillation. The AF classification algorithms published in (Hong et al., 2017; Teijeiro et al., 2017; Zabihi et al., 2017; Datta et al., 2017) were developed and validated on the Physionet/Computing in Cardiology Challenge 2017 training and test datasets. Their algorithms were top ranked in the challenge, achieving F-scores in excess of 83% on the test dataset. They do not report accuracy statistics on the AFDB and MITDB databases, and hence they have not been included in Table 5.2.7 for comparison.

This concludes the AF detection stage. The subsequent sections describe algorithms for detection of supraventricular ectopic beats and associated arrhythmias.

5.3 Supraventricular Ectopic Beats

As mentioned at the beginning of the chapter (prior to Section 5.2), Supraventricular Ectopic Beats (SVEBs) include both PACs and PJCs. SVEB detection is performed on a beat-by-beat basis and is implemented through logical analysis techniques. They are as follows:

1. Pre-process the incoming ECG signal as described in Chapter 2.

2. Perform ventricular fibrillation/sustained ventricular tachycardia (VF/VT) detection as described in Section 4.3

3. Identify ECG segments with VF/VT and omit them for further analysis.

4. Perform R-peak (beat) detection as described in Chapter 3 and store the R-peak locations in a vector *r_peaks*. Mark each R-peak as `unvisited`.

5. Compute RR-interval between the first two R-peaks using equation (5.1) and store this value in the vector *rr_series*. Mark these two R-peaks as `non-SVEB` indicating these R-peaks are not supraventricular beats.

6. If no `unvisited` R-peaks remain in *r_peaks*, then go to step 7. Else, select the next `unvisited` R-peak r_i in *r_peaks* and implement the following steps:

   (i) Check if r_i is a PVC beat. If so, mark it as `non-SVEB` and go back to Step 6. Else, continue.

   (ii) Check if r_i is part of an AF segment. If so, mark it as `non-SVEB` and go back to Step 6. Else, continue.

   (iii) Compute RR-interval rr_i using (5.1).

   (iv) Check to see if this RR-interval is less than a pre-defined threshold *thr_sveb*. This threshold is defined as:

$$thr\_sveb = thr * rr\_med \tag{5.11}$$

   where,

   - rr_med is the median of the ten most recent values in the vector *rr_series*. If *rr_series* vector contains less than ten values, then take the median of all the values in the vector to compute rr_med.

Table 5.7. Performance evaluation of SVEB algorithm on MITDB Dataset

Dataset	Se	PPV	FSc
MITDB	92.11%	83.77%	87.74%

- *thr* is a scaling factor set to 0.875 by default. Note that changing this value will accordingly change the balance between false positives and false negatives.

(v) If the value of *rr_i* is less than *thr_sveb*, then mark *r_i* as `SVEB` i.e., indicating that this R-peak is a supraventricular beat and go to Step 6. Else, mark *r_i* as `non-SVEB` and append the value *rr_i* to *rr_series*.

7. Return the *r_peak* vector with R-peak locations along with `SVEB/non-SVEB` labels for each R-peak.

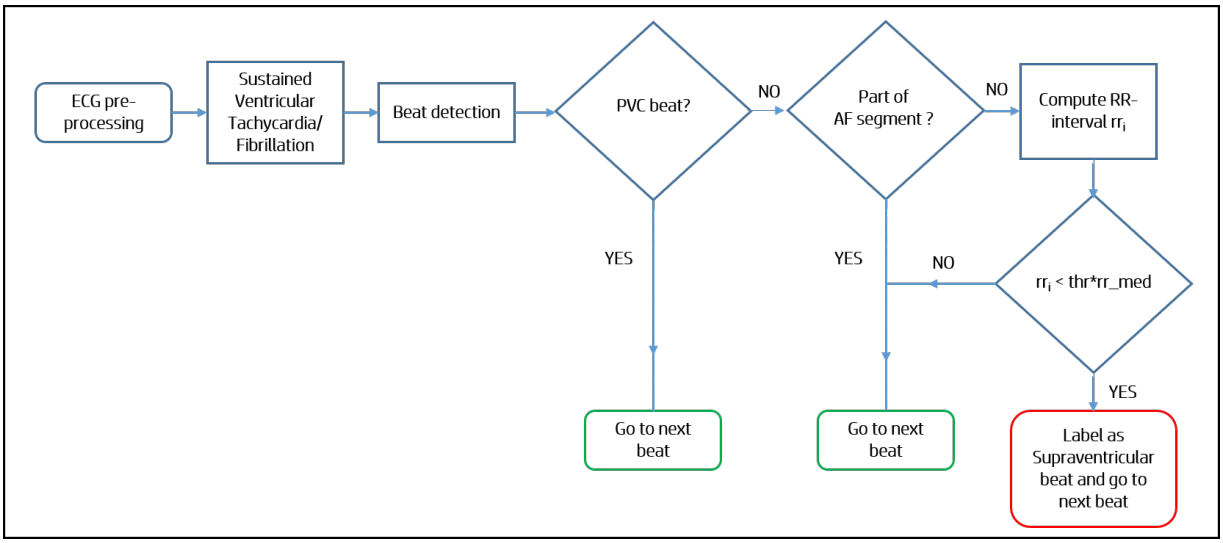

Figure 5.6. Flow diagram depicting SVEB detection algorithm.

This concludes the section on SVEB detection algorithm description and the results for the same are tabulated in Table 5.7. Figure 5.6 shows a simple flow diagram depicting the steps involved in SVEB detection. The occurrence of SVEBs in certain specific patterns gives rise to different supraventricular arrhythmias. Since these patterns are clearly defined in medical literature (Aehlert, 2018), simple logical analysis that searches for these patterns is implemented to detect the presence of these arrhythmias. It must be remembered that this logical analysis step is based on beat label information obtained in the SVEB classification stage and hence the detection accuracy for these arrhythmias inherently depends on the accuracy of SVEB detection. These arrhythmias and the logic required to detect them are briefly discussed below.

5.3.1 Supraventricular Bigeminy

Supraventricular Bigeminy (SVBI) refers to the occurrence of alternating SVEB and non-SVEB beats. To confirm supraventricular bigeminy, presence of atleast three consecutive pairs of Non-SVEB/SVEB beats is required. If SVEB beats are denoted as `S` and non-SVEB beats are denoted as `N`, then the algorithm searches for the pattern `[NSNSNS...]`. Hence atleast six beats are required to detect presence of SVBI episodes. The occurrence of two consecutive `N` beats marks the end of a supraventricular bigeminy episode. To detect another episode, the algorithm once again searches for the sequence `[NSNSNS...]` in the remaining beats. Once the sequence is found, it marks the onset of another SVBI episode and continues scanning until it encounters two consecutive `N` beats which marks the offset of this new episode. This process is repeated for all the remaining beats in the input ECG signal.

5.3.2 Supraventricular Trigeminy

Supraventricular Trigeminy (STRI) refers to the occurrence of a SVEB beat after every two consecutive non-SVEB beats. To confirm supraventricular trigeminy, presence of atleast three consecutive triplets of the form `[NNS]` is required. To detect a supraventricular Trigeminy episode, the algorithm searches for the pattern `[NNSNNSNNS...]`. Hence atleast nine beats are required for detection of SVTRI episodes. The occurrence of any beat-triplet other than `[NNS]` beat-triplet marks the end of a supraventricular trigeminy episode. To detect another episode, the algorithm once again searches for the sequence `[NNSNNSNNS...]` in the remaining beats. Once the sequence is found, it marks the onset of another SVTRI episode and continues scanning until it encounters a beat-triplet other than `[NNS]` which marks the offset of this new episode. This process is repeated for all the remaining beats in the input ECG signal.

5.3.3 Supraventricular Quadrigeminy

Supraventricular Quadrigeminy (SVQUAD) refers to the occurrence of a SVEB beat after every three consecutive non-SVEB beats. To confirm supraventricular quadrigeminy, presence of atleast three consecutive quadruplets of the form `[NNNs]` is required. To detect a supraventricular trigeminy episode, the algorithm searches for the pattern `[NNNSNNNSNNNS...]`. Hence atleast twelve beats are required for detection of SVQUAD episodes. The occurrence of any beat-quadruplet other than `[NNNS]` beat-triplet marks the end of a supraventricular quadrigeminy episode. To detect another episode, the algorithm once again searches for the sequence `[NNNSNNNSNNNS...]` in the remaining beats. Once the sequence is found, it marks the onset of another SVQUAD episode and continues scanning until it encounters a beat-quadruplet other than `[NNNS]` which marks the offset of this new episode. This process is repeated for all the remaining beats in the input ECG signal.

5.3.4 Supraventricular Runs

Supraventricular runs refer to occurrence of three or more consecutive SVEBs at heart rates greater than 100 bpm. At heart rates greater than 120 bpm, these rhythms are also known as Supraventricular Tachycardia episodes. This includes sinus tachycardia, atrial tachycardia and junctional tachycardia. To detect supraventricular runs, the algorithm searches for groups of three or more consecutive SVEB beats. For every group of three or more consecutive SVEB beats, the mean heart rate is computed as follows:

$$RR_{group} = \frac{1}{n-1} \sum_{k=1}^{n-1} rr_k \tag{5.12}$$

$$HR_{group} = \frac{Fs * 60}{RR_{group}} \tag{5.13}$$

where,

- n is the number of SVEBs in the beat group.
- rr_k is the k^{th} RR-interval in the group, computed using (5.1).
- Fs is the sampling frequency.

If the value of HR_{group} exceeds 100 bpm, that group of beats is labeled as a supraventricular run episode.

5.3.5 Supraventricular Couplets

Supraventricular couplets or pairs refer to occurrence of exactly two consecutive SVEBs at heart rates greater than 100 bpm. To detect supraventricular couplets, the algorithm simply searches for the presence of `[SS]` pattern in the beats and each such pair is marked as a supraventricular couplet episode.

5.3.6 Sinus Bradycardia

Sinus bradycardia (SBR) refers to the condition where the SA node fires slower than normal for a patient's age (Aehlert, 2018). Heart rates lower than 50 bpm usually fall under the category of SBR. When the heart rates goes below 40 bpm, it is termed *Extreme Bradycardia* (Clifford et al., 2015) or *Severe Sinus Bradycardia* (Aehlert, 2018). To detect SBR, the algorithm searches for groups of five or more consecutive beats whose mean heart rate (computed using (5.13)) is less than 50 bpm. For every such group found, the algorithm does the following:

(i) Check if the beats in the group include VF/VT segments or PVC beats or AF beats.

(ii) If none of the above conditions are met, then label that group of five or more consecutive beats as a Sinus bradycardia episode.

5.4 Conclusion

Automated techniques for accurate AF classification represent a continually improving area of biomedical research. A stable AF classifier is expected to be robust to noise while remaining extremely precise in distinguishing AF rhythms from other arrhythmias as well as normal rhythms. The work described in this chapter is a contribution in that direction. Probabilities from an eight-state Markov matrix (modelling transitions in RR-intervals) along with other RR-interval features in 60s epochs were used to train a new Random Forests model for the purpose of AF detection. From Table 5.2.6, it can be observed that the AF detection algorithm described in this chapter achieves consistently high values of sensitivity, specificity and F-Scores on different sets of databases with different sampling frequencies - MITDB (360 Hz) and AFDB (250 Hz). This confirms the ability

of the proposed algorithm to generalize well across multiple databases irrespective of the underlying sampling rate as long as this information is provided as input to the algorithm.

This concludes the description of supraventricular arrhythmias. The next chapter presents a cloud-based ECG analysis platform for performing online cardiac arrhythmia detection.

References

Aehlert, B. J. (2018). *ECGs Made Easy* (6 ed.). Elsevier.

Asgari, S., A. Mehrnia, and M. Moussavi (2015). Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine. *Computers in biology and medicine 60*, 132–142.

Babaeizadeh, S., R. E. Gregg, E. D. Helfenbein, J. M. Lindauer, and S. H. Zhou (2009). Improvements in atrial fibrillation detection for real-time monitoring. *Journal of electrocardiology 42*(6), 522–526.

Carrara, M., L. Carozzi, T. J. Moss, M. de Pasquale, S. Cerutti, M. Ferrario, D. E. Lake, and J. R. Moorman (2015). Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy. *Physiological measurement 36*(9), 1873.

Clifford, G. D., C. Liu, B. Moody, H. L. Li-wei, I. Silva, Q. Li, A. Johnson, and R. G. Mark (2017). Af classification from a short single lead ecg recording: the physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, pp. 1–4. IEEE.

Clifford, G. D., I. Silva, B. Moody, Q. Li, D. Kella, A. Shahin, T. Kooistra, D. Perry, and R. G. Mark (2015). The physionet/computing in cardiology challenge 2015: reducing false arrhythmia alarms in the icu. In *2015 Computing in Cardiology Conference (CinC)*, pp. 273–276. IEEE.

Couceiro, R., P. Carvalho, J. Henriques, M. Antunes, M. Harris, and J. Habetha (2008). Detection of atrial fibrillation using model-based ecg analysis. In *2008 19th International Conference on Pattern Recognition*, pp. 1–5. IEEE.

Dash, S., K. Chon, S. Lu, and E. Raeder (2009). Automatic real time detection of atrial fibrillation. *Annals of biomedical engineering 37*(9), 1701–1709.

Datta, S., C. Puri, A. Mukherjee, R. Banerjee, A. D. Choudhury, R. Singh, A. Ukil, S. Bandyopadhyay, A. Pal, and S. Khandelwal (2017). Identifying normal, af and other abnormal ecg rhythms using a cascaded binary classifier. In *2017 Computing in Cardiology (CinC)*, pp. 1–4. IEEE.

Du, X., N. Rao, M. Qian, D. Liu, J. Li, W. Feng, L. Yin, and X. Chen (2014). A novel method for real-time atrial fibrillation detection in electrocardiograms using multiple parameters. *Annals of Noninvasive Electrocardiology 19*(3), 217–225.

Freund, Y. and R. E. Schapire (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pp. 23–37. Springer.

Goldberger, A. L., L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation 101*(23), e215–e220.

Hong, S., M. Wu, Y. Zhou, Q. Wang, J. Shang, H. Li, and J. Xie (2017). Encase: An ensemble classifier for ecg classification using expert features and deep neural networks. In *2017 Computing in Cardiology (CinC)*, pp. 1–4. IEEE.

January, C. T., L. S. Wann, J. S. Alpert, H. Calkins, J. E. Cigarroa, J. C. Cleveland, J. B. Conti, P. T. Ellinor, M. D. Ezekowitz, M. E. Field, K. T. Murray, R. L. Sacco, W. G. Stevenson, P. J. Tchou, C. M. Tracy, and C. W. Yancy (2014). 2014 aha/acc/hrs guideline for the management of patients with atrial fibrillation. *Circulation 130*(23), e199–e267.

Kalidas, V. and L. S. Tamil (2019). Detection of atrial fibrillation using discrete-state markov models and random forests. *Computers in biology and medicine 113*, 103386.

Kirchhof, P., S. Benussi, D. Kotecha, A. Ahlsson, D. Atar, B. Casadei, M. Castella, H.-C. Diener, H. Heidbuchel, J. Hendriks, G. Hindricks, A. S. Manolis, J. Oldgren, B. A. Popescu, U. Schotten, B. Van Putte, P. Vardas, and E. S. D. Group (2016, 08). 2016 ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS. *European Heart Journal 37*(38), 2893–2962.

Ladavich, S. and B. Ghoraani (2015). Rate-independent detection of atrial fibrillation by statistical modeling of atrial activity. *Biomedical Signal Processing and Control 18*, 274–281.

Lake, D. E. and J. R. Moorman (2011). Accurate estimation of entropy in very short physiological time series: the problem of atrial fibrillation detection in implanted ventricular devices. *American Journal of Physiology-Heart and Circulatory Physiology 300*(1), H319–H325.

Liu, C., J. Oster, E. Reinertsen, Q. Li, L. Zhao, S. Nemati, and G. D. Clifford (2018). A comparison of entropy approaches for af discrimination. *Physiological measurement 39*(7), 074002.

Moody, G. (1983). A new method for detecting atrial fibrillation using rr intervals. *Computers in Cardiology*, 227–230.

Moody, G. B. and R. G. Mark (2001). The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine 20*(3), 45–50.

Petrenas, A., V. Marozas, L. Sornmo, and A. Lukosevicius (2012). An echo state neural network for qrst cancellation during atrial fibrillation. *IEEE transactions on biomedical engineering 59*(10), 2950–2957.

Rajpurkar, P., A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng (2017). Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*.

Ródenas, J., M. García, R. Alcaraz, and J. J. Rieta (2015). Wavelet entropy automatically detects episodes of atrial fibrillation from single-lead electrocardiograms. *Entropy 17*(9), 6179–6199.

Sarkar, S., D. Ritscher, and R. Mehra (2008). A detector for a chronic implantable atrial tachyarrhythmia monitor. *IEEE Transactions on Biomedical Engineering 55*(3), 1219–1224.

Stephens, M. A. (1974). Edf statistics for goodness of fit and some comparisons. *Journal of the American statistical Association 69*(347), 730–737.

Tateno, K. and L. Glass (2001). Automatic detection of atrial fibrillation using the coefficient of variation and density histograms of rr and δrr intervals. *Medical and Biological Engineering and Computing 39*(6), 664–671.

Teijeiro, T., C. A. García, D. Castro, and P. Félix (2017). Arrhythmia classification from the abductive interpretation of short single-lead ecg records. In *2017 Computing in Cardiology (CinC)*, pp. 1–4. IEEE.

Veenhuyzen, G., C. Simpson, and H. Abdollah (2004). Atrial fibrillation. *CMAJ 171*(7), 755–760. cited By 29.

Xia, Y., N. Wulan, K. Wang, and H. Zhang (2018). Detecting atrial fibrillation by deep convolutional neural networks. *Computers in biology and medicine 93*, 84–92.

Zabihi, M., A. B. Rad, A. K. Katsaggelos, S. Kiranyaz, S. Narkilahti, and M. Gabbouj (2017). Detection of atrial fibrillation in ecg hand-held devices using a random forest classifier. In *2017 Computing in Cardiology (CinC)*, pp. 1–4. IEEE.

Zhao, L., C. Liu, S. Wei, Q. Shen, F. Zhou, and J. Li (2018). A new entropy-based atrial fibrillation detection method for scanning wearable ecg recordings. *Entropy 20*(12), 904.

Zoni-Berisso, M., F. Lercari, T. Carazza, and S. Domenicucci (2014). Epidemiology of atrial fibrillation: European perspective. *Clinical epidemiology 6*, 213.

CHAPTER 6

AUTOECG

6.1 Introduction

The research presented in Chapters of this dissertation is translational in nature and can be used in real-world settings. This is demonstrated by deploying the algorithms described in Chapter 2 through Chapter 5 in a fully functional cloud platform - **AutoECG**. **AutoECG** is a first-of-its-kind web service that facilitates automated cardiac arrhythmia detection using information extracted from single-lead electrocardiogram (ECG) signals. As described in Chapter 1, the algorithms developed in this dissertation, and therefore deployed in **AutoECG**, are capable of detecting the following fifteen cardiac arrhythmias:

1. Ventricular Fibrillation and Sustained Ventricular Tachycardia
2. Premature Ventricular Complexes
3. Ventricular Bigeminy
4. Ventricular Trigeminy
5. Ventricular Quadrigeminy
6. Ventricular Couplets
7. Ventricular Runs
8. Atrial Fibrillation
9. Supraventricular Ectopy
10. Supraventricular Bigeminy

11. Supraventricular Trigeminy

12. Supraventricular Quadrigeminy

13. Supraventricular Couplets

14. Supraventricular Runs and Sinus Tachycardia

15. Sinus Bradycardia

More than four million Americans are affected by some form of arrhythmia. Atrial Fibrillation is prevalent in about 2.7 million Americans and is expected to reach 12.1 million by 2030. Prevalence of Premature Ventricular Complexes (PVCs) is about 69% in adults over 75 years. These arrhythmias are highly correlated with congestive heart failure, stroke, high blood pressure and possible sleep apnea. Oftentimes, delayed diagnosis can increase severity of these arrhythmias. Under these circumstances, it imperative to develop a system such as **AutoECG** that aids in continuous ECG monitoring to help provide timely care and treatment in out-of-hospital settings. According to the American Heart Association (AHA) Heart Disease and Stroke Statistics 2015 Update (Mozaffarian et al., 2015), there were 326,000 Out-of-Hospital Cardiac Arrest (OHCA) incidents in the United States in 2014 with a median age of 66 years. Approximately only 60% of these OHCA incidents are treated by Emergency Medical Services (EMS) personnel out of which 25% of them do not report any symptoms. In fact, 69.5% of OHCA incidents occur at residences according to the report. The services of **AutoECG** gain immense significance in the context of these facts. More importantly, the software deployed in **AutoECG** includes algorithms that are capable of accurately detecting ventricular arrhythmias such as PVC, VF, VT and other ventricular rhythms and it has been reported that 23% of OHCA incidents exhibit an initial irregular rhythm of VF or VT. (see Chapter 4). This reiterates the significance and utility of the **AutoECG** platform.

The codebase deployed in **AutoECG**, pertaining to arrhythmia detection algorithms, is written in Python 3.6. The entire **AutoECG** software (in beta testing stage) is implemented on a 64-bit dual-core Intel(R) Xeon(R) Platinum 8175M CPU processor operating at a clock frequency of 2.50 GHz. This hardware platform is provided by Amazon Web Services (`https://aws.amazon.com/`) in the form of Infrastructure as a Service (IaaS). The services of **AutoECG** are available for testing at `https://autoecg.utdallas.edu`.

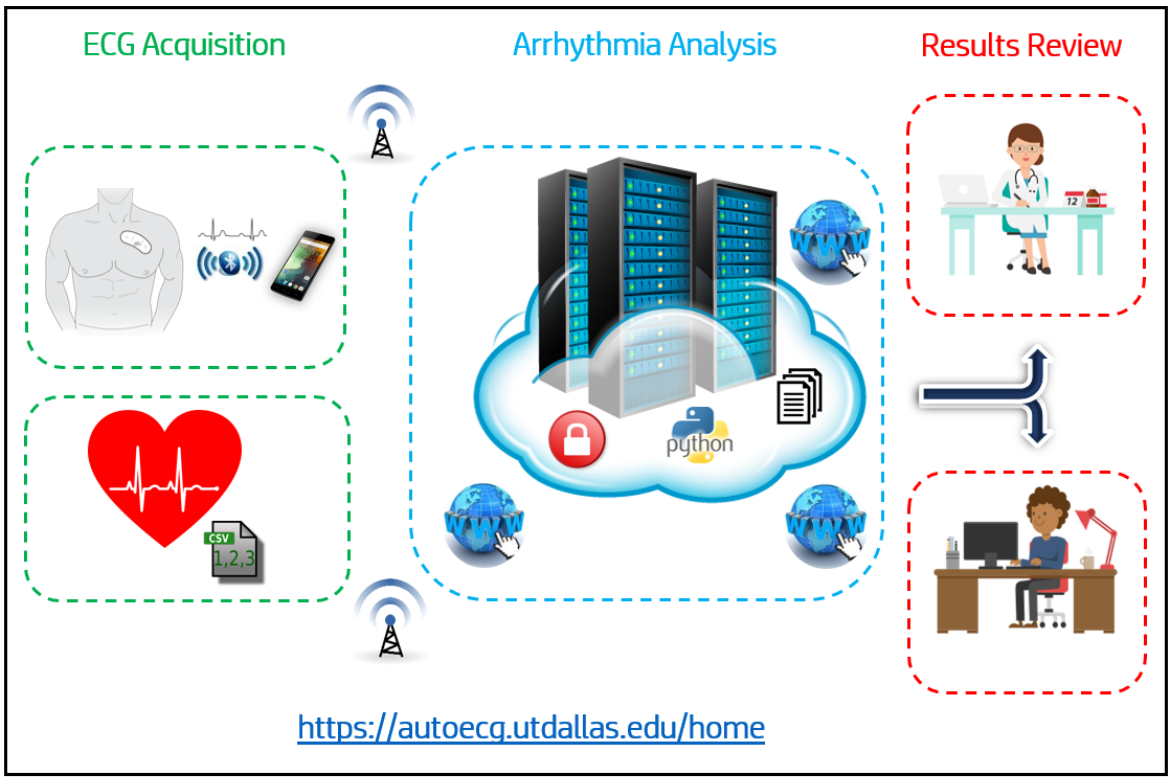

Figure 6.1. AutoECG Platform Architecture

6.2 System Architecture

The AutoECG architecture, depicted in Figure 6.1, consists of (i) an ECG acquisition stage, (ii) an arrhythmia analysis stage and (iii) a results review stage. They are briefly described below:

Prior to using the AutoECG platform, users are required to login using their username/password credentials. All user accounts (users, doctors, technicians, etc.) hosted in AutoECG are securely protected via Auth0 and data access is compartmentalized depending on user type and access level.

1. **ECG Acquisition:** In this stage, the ECG data is read from the user. The AutoECG platform accepts ECG data in the form of a single-column comma-separated values (.csv) file. Upon uploading the ECG, the user is required to enter the sampling frequency which is required for computing heart rate metrics. Once this is completed, the ECG data is processed for arrhythmia analysis.

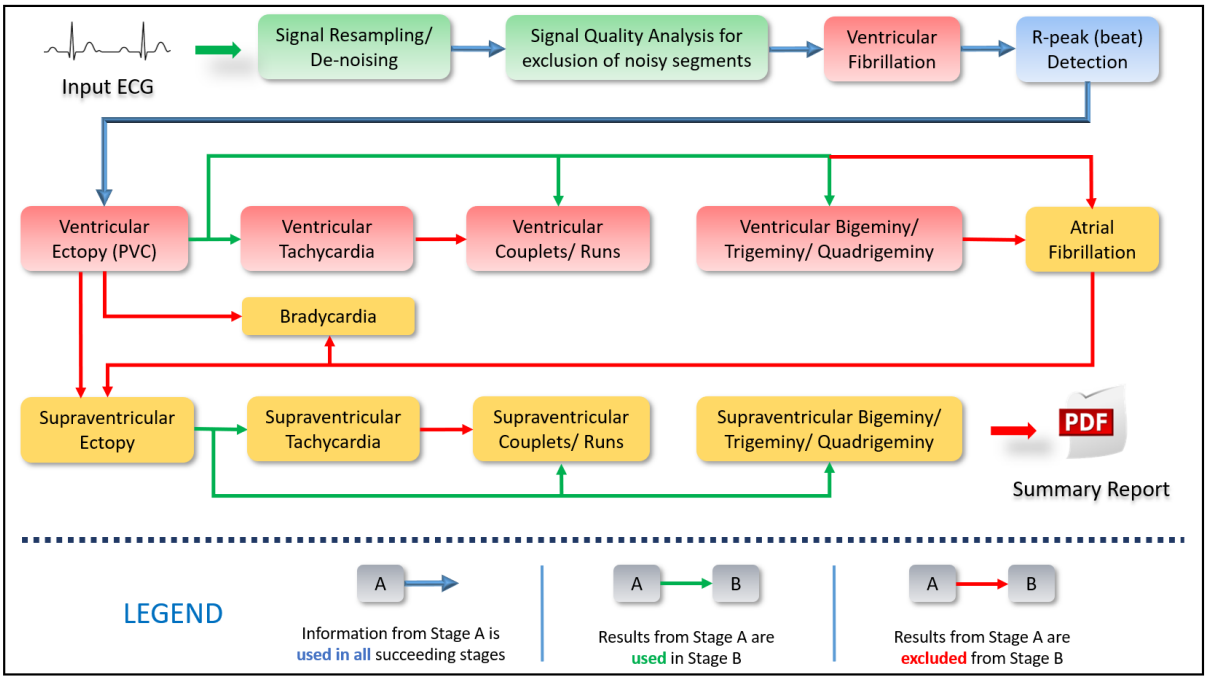

Figure 6.2. AutoECG Algorithm Flow

2. **Arrhythmia Analysis:** In this stage, raw ECG data is first extracted from the uploaded .csv file. Following this, arrhythmia analysis for detection of the above mentioned fifteen arrhythmias is performed as follows:

(i) **ECG denoising:** ECG de-noising is performed using the techniques described in Section 2.2.2 for low frequency noise removal and Section 2.3.2 (high-frequency noise suppression.

(ii) **EM noise classification:** Following ECG denoising, signal quality analysis is implemented using the techniques described in Section 2.4. In this stage, ECG segments with very high grade noise artifacts are identified and suppressed from any further arrhythmia analysis.

(iii) **Ventricular Fibrillation detection:** Next, detection of ventricular fibrillation/sustained ventricular tachycardia (VF/VT) is carried out using the methodology described in Section 4.3. This stage does not require beat detection and hence is done prior to beat detection.

(iv) **Beat detection:** After VF/VT detection is completed, beat detection is performed using the approach laid out in Chapter 3. The subsequent steps utilize this beat information for achieving accurate arrhythmia detection performance.

(v) **Detection of PVCs and associated arrhythmias:** Following beat detection, detection of PVC beats is implemented using the methods explained in Section 4.4. Using this PVC beat information, additional PVC-dependent arrhythmias are subsequently detected using logical analysis as described towards the end Section 4.4.

(vi) **Atrial Fibrillation detection:** Following PVC detection, detection of atrial fibrillation (AF) episodes is carried out using the methodology presented in Section 5.2.

(vii) **Detection of other supraventricular arrhythmias:** This is the last phase of the arrhythmia detection cycle and in this phase, detection of supraventricular

ectopic beats (SVEBs) and SVEB-dependent arrhythmias is carried out using logical analysis techniques as described in Section 5.3.

This concludes the arrhythmia analysis stage. Figure 6.2 shows the flow diagram representing the arrhythmia analysis stage.

3. **Results Review:** The results from the arrhythmia analysis stage are formatted in the form of a user-readable *ECG Analysis Report* that contains useful information as follows:

   (a) Summary statement stating all the arrhythmias detected by the algorithm.

   (b) Average heart rate (in bpm)

   (c) Number of PVCs and SVEBs

   (d) Overall duration of high grade noise episodes

   (e) Onset and offset of individual high grade noise episodes

   (f) Overall duration of individual arrhythmias

   (g) Onset and offset of all episodes for each arrhythmia that is detected in the arrhythmia analysis stage.

   This summary report can be downloaded by the users in .pdf format. An example summary report is presented at the end of the chapter, following Section 6.3. This report shows arrhythmia analysis results for the uploaded ECG file - `tri.csv` (see `Filename` field in the report). The **AutoECG** software detects the presence of ventricular bigeminy (VBI) and ventricular trigeminy (VTRI) in the extracted ECG and this is highlighted in the `ECG Analysis Summary` section of the report. The `Analysis Results` section provides more detailed information such as processing

time, ECG duration, average heart rate, arrhythmia episodes, etc. More importantly, the subsections `Ventricular Bigeminy` and `Ventricular Trigeminy` in the report provide the corresponding onset and offset timestamps for each of the VBI and VTRI episodes detected by the algorithms along with reporting the overall time duration for each arrhythmia. It has to noted that the findings in the *ECG Analysis Report* should not be used in any diagnostic capacity by the user and that the user must contact a physician/medical practitioner for further review of the report.

6.3 Conclusion

In conclusion, this chapter presents a cloud-based ECG analysis platform, termed **AutoECG**, that hosts the arrhythmia analysis algorithms developed as part of this dissertation (Chapter 2 through Chapter 5). The algorithms deployed in **AutoECG** are device-agnostic and are well-equipped to analyze data from a diverse range of ECG acquisition devices. Additionally, the AutoECG system is designed to handle ECG signals spanning 30 seconds to 24 hours, thus facilitating real-time analysis (wearables) as well as overnight monitoring (hospitals, bedside monitoring, etc.). Implementing a fully automated arrhythmia system such as **AutoECG** minimizes burden on physicians and helps them prioritize their patients thus allowing them to attend to more patients as well as speeding up diagnosis and treatment. Such systems also have the advantage that they can be easily integrated into telemedicine platforms for remote delivery of healthcare services.

05/26/2020

ECG Analysis Report

Powered by Quality of Life Technology LABORATORY

**** Disclaimer: We DO NOT store any information regarding user ECG data or the corresponding analysis results on our server. This work is part of academic research and hence the results contained in this document should not be used in any diagnostic capacity.*

----x----x----

Filename : tri.csv
Source : Web Upload

ECG Analysis Summary

This ECG shows signs of ventricular bigeminy and ventricular trigeminy.

Analysis Results

Processing time (hh:mm:ss) : 00:00:10

ECG duration (hh:mm:ss) : 00:00:30

Noisy segments : 0 (0 hours, 0 minutes and 0 seconds)

Number of beats : 33
Average heart rate : 66.37 bpm

Number of probable PVCs : 11
Number of probable PACs : 0

Sustained VT/VF episodes : 0 (0 hours, 0 minutes and 0 seconds)

Short VT episodes : 0 (0 hours, 0 minutes and 0 seconds)

Ventricular couplet episodes : 0 (0 hours, 0 minutes and 0 seconds)

Ventricular bigeminy episodes : 1 (0 hours, 0 minutes and 6 seconds)

>> Episode 1 : 00:00:03 to 00:00:10

Ventricular trigeminy episodes : 1 (0 hours, 0 minutes and 12 seconds)

>> Episode 1 : 00:00:10 to 00:00:23

Ventricular quadrigeminy episodes : 0 (0 hours, 0 minutes and 0 seconds)

Other ventricular episodes : 0 (0 hours, 0 minutes and 0 seconds)

Atrial fibrillation episodes : 0 (0 hours, 0 minutes and 0 seconds)

Sinus bradycardia episodes : 0 (0 hours, 0 minutes and 0 seconds)

Supraventricular tachycardia episodes : 0 (0 hours, 0 minutes and 0 seconds)

Supraventricular couplet episodes : 0 (0 hours, 0 minutes and 0 seconds)

continued on next page ...


```
Supraventricular bigeminy episodes      : 0 (0 hours, 0 minutes and 0 seconds)

Supraventricular trigeminy episodes     : 0 (0 hours, 0 minutes and 0 seconds)

Supraventricular quadrigeminy episodes  : 0 (0 hours, 0 minutes and 0 seconds)
```

References

Mozaffarian, D., E. J. Benjamin, A. S. Go, D. K. Arnett, M. J. Blaha, M. Cushman, S. de Ferranti, J.-P. Després, H. J. Fullerton, V. J. Howard, M. D. Huffman, S. E. Judd, B. M. Kissela, D. T. Lackland, J. H. Lichtman, L. D. Lisabeth, S. Liu, R. H. Mackey, D. B. Matchar, D. K. McGuire, E. R. Mohler, C. S. Moy, P. Muntner, M. E. Mussolino, K. Nasir, R. W. Neumar, G. Nichol, L. Palaniappan, D. K. Pandey, M. J. Reeves, C. J. Rodriguez, P. D. Sorlie, J. Stein, A. Towfighi, T. N. Turan, S. S. Virani, J. Z. Willey, D. Woo, R. W. Yeh, and M. B. Turner (2015). Heart disease and stroke statistics 2014 - 2015 update. *Circulation 131*(4), e29–e322.

CHAPTER 7

CONCLUSION

In summary, this dissertation has presented in detail several machine learning based algorithms, primarily Deep Learning and Random Forests, complemented by logical analysis, for accurate detection of fifteen different cardiac arrhythmias using information derived from single lead electrocardiogram (ECG) signals. These algorithms have been thoroughly discussed from Chapter 2 through Chapter 6.

Prior to description of these algorithms, a brief introduction about the heart's electrical system, its significance in regulating cardiac functionality and its manifestation on ECGs is described in Chapter 1 along with a succinct description of the ECG fiduciary points such as P-waves, QRS complexes, etc. The subsequent five chapters provide in-depth explanations pertaining to the arrhythmia analysis approaches developed in this research. The methods presented in these chapters form the core of this dissertation.

Chapter 2 discusses an important ECG pre-processing step that is crucial to achieving robust arrhythmia detection performance. This refers to the task of ECG denoising and signal quality analysis. The former deals with removal of low-frequency and high-frequency noise. Low frequency noise, also known as baseline wander, is suppressed through the application of Stationary Wavelet Transforms (See Section 2.2.2) to the ECG signal. This is complemented by adoption of a deep learning approach to perform high-frequency noise suppression (see Section 2.3.2). This involves implementing a Denoising Convolutional Autoencoder that is trained to extract useful ECG signal information in the presence of high-frequency artifacts. Following this, signal quality analysis is performed to identify ECG segments corrupted by high-grade noise artifacts, primarily electrode motion artifacts. To achieve this, a Convolutional Neural Network based binary classifier is developed and ECG segments deemed to be too noisy by the classifier are omitted from further arrhythmia analysis (See Section 2.4).

Next, Chapter 3 presents a two-stage beat (R-peak) detection algorithm. In the first stage, a Convolutional Autoencoder (CAE) based beat extractor model is developed. This CAE based beat extractor model is trained to extract potential QRS complexes from ECG signals while suppressing other ECG components such as P-waves, T-waves, etc. This is followed by a second stage that applies adaptive thresholding on these potential QRS complexes. This latter stage is implemented in five steps viz., (i) initial peak detection, (ii) missed beat detection, (iii) false beat removal, (iv) threshold updating and (v) final R-peak detection. The CAE based beat extraction stage ensures that P-wave and T-waves are not misclassified as R-peaks while the adaptive thresholding stage ensures that occurrence of false beats (due to noisy ECG segments) and missed beats (due to low amplitudes and/or abnormal beat morphology) are minimized, thus ensuring accurate R-peak detection performance. This is key to performing robust arrhythmia detection cycle as the performance of majority of the arrhythmia detection algorithms described in Chapter 4 and Chapter 5 inherently depends on the accuracy of the beat detection step.

The subsequent chapter (Chapter 4) exclusively focuses on algorithms for detection of ventricular arrhythmias. The first half of the chapter deals with analysis of beat-independent arrhythmias (see Section 4.3). This includes sustained ventricular tachycardia, ventricular fibrillation and ventricular flutter (VT/VF). In this regard, seven time-, frequency- and wavelet-domain features are computed that are input to a Random-Forest based machine learning model for achieving efficient VT/VF classification. The remainder of the chapter discusses a hybrid approach to the detection of premature ventricular complexes (PVCs) and associated beat-dependent ventricular arrhythmias (see Section 4.4). RR-interval and beat morphology based features are computed to perform accurate PVC classification using a combination of semi-supervised Autoencoders and Random Forests based machine learning techniques. The results of PVC classification are then used to detect

five other ventricular arrhythmias viz., (i) ventricular bigeminy, (ii) ventricular trigeminy, (iii) ventricular quadrigeminy, (iv) ventricular runs and (v) ventricular couplets.

Following description of ventricular arrhythmias, Chapter 5 provides an in-depth description of supraventricular arrhythmia detection algorithms. Once again, the first half of the chapter exclusively focuses on detection of atrial fibrillation (AF) episodes (see Section 5.2). A machine-learning based approach employing a combination of discrete-state Markov models and Random forests to detect AF episodes is presented here. The proposed features based on an eight-state Markov model for AF classification result in comparable or better AF detection performance than other state-of-the-art AF detection techniques. The remainder of the chapter then delves into detection of other supraventricular arrhythmias that are primarily dependent of detection of supraventricular ectopic beats (SVEBs) (see Section 5.3). In this section, logical analysis techniques are presented for detection of SVEBs and six other supraventricular arrhythmias viz., (i) supraventricular bigeminy, (ii) supraventricular trigeminy, (iii) supraventricular quadrigeminy, (iv) supraventricular runs, (v) supraventricular couplets and (vi) sinus bradycardia.

Finally, Chapter 6 presents a cloud-based arrhythmia detection platform known as **AutoECG**. The algorithms described in Chapter 2 through Chapter 5 are deployed in **AutoECG** which facilitates online arrhythmia analysis using information from single-lead ECGs that can be uploaded by authorized users via `https://autoecg.utdallas.edu`. The chapter provides a concise description of a three-stage system architecture that captures the essence of **AutoECG's** functionality. A flow diagram depicting the implementation of the arrhythmia algorithms is also presented in the chapter along with an example arrhythmia summary report that is generated by the **AutoECG** platform post-analysis for further review by doctors and other medical practitioners.

7.1 Limitations and future work

The algorithms described in this dissertation do have a few limitations that can act as fodder for future research. The ECG low frequency noise suppression stage filters out ECG components below 0.7 Hz. Although this does not affect the accuracy of the methods described in this dissertation, this could pose a problem when it comes to analyzing ECGs for other arrhythmias not includes in this dissertation. For instance, detection of ST-segment elevation and ST-segment depression require frequencies as low as 0.05 Hz. Another limitation is that segments classified to be too noisy are omitted from further analysis which may result in skipping detection of arrhythmias in those segments. Next, deep learning models developed in this dissertation are exclusively trained on signals from lead-II configuration such as that of PVC detection. Hence the algorithms have the limitation that ECG signals be acquired from lead-II setup which may not always be possible. Some of the limitations of the AF detection approach include skipping detection of very short AF episodes owing to the use of 60s epochs. Also, since any sort of atrial activity analysis is omitted, the algorithm's accuracy inherently depends on the quality of beat detection although the beat detection technique used here is accurate (Chapter 3) and satisfactory for the purpose of AF classification.

Many improvements can be incorporated in the future that can further increase the accuracy of the algorithms presented in this dissertation. For instance, in the future, information from multi-lead signals can be used instead of single-lead ECGs to create a more rich set of features that could result in better arrhythmia detection performance. Next, algorithms for detection of a more diverse range of arrhythmias could be added to the existing algorithms. This would necessitate more efforts in terms of data collection and annotation, especially in cases of rare arrhythmias. Also, improved signal acquisition setups can result in accurate P-wave detection that could result in efficient atrial activity. This

can further help reduce false alarms and increase overall AF detection accuracy. Currently, the **AutoECG** platform requires users to manually upload ECGs along with the correct sampling frequency which present opportunities for errors. To alleviate this, a bluetooth enabled ECG sensor setup as illustrated in Figure 6.1 in Chapter 6 in the upper half of ECG acquisition stage can result in seamless transfer and subsequent analysis of ECG data and therefore help minimize data upload errors and other human-induced errors. Finally, a technician portal that can be implemented as illustrated in the bottom half of the Results Review stage in Figure 6.1 can serve as a medium for updating annotations and aiding on-the-fly model retraining for improving performance of the algorithms deployed in **AutoECG**.

BIOGRAPHICAL SKETCH

Vignesh Kalidas began his doctoral studies in the Department of Computer Engineering at The University of Texas at Dallas in Fall 2014. Prior to that he completed his MS in Computer Engineering at UTD in 2013 and his B.Tech. in Electronics and Instrumentation Engineering at SASTRA University, India, in 2011.

Vignesh carried out his research under the guidance of Dr. Lakshman S. Tamil in the Quality of Life Technology (QoLT) Laboratory. His dissertation focused on developing machine learning algorithms for automated cardiac arrhythmia detection using electrocardiogram signals. During the course of his PhD at UTD, he also worked as a Graduate Teaching Assistant in the Department of Electrical Engineering. Previously, he worked as an Embedded Software Intern and Engineering (R&D) Intern at Samsung Semiconductors, Inc., in Fall 2012 and Spring 2014, respectively.

CURRICULUM VITAE

Vignesh Kalidas

May 26, 2020

Contact Information:

Department of Computer Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: vignesh.kalidas@utdallas.edu

Educational History:

M.S., Computer Engineering, University of Texas at Dallas, 2013
B.Tech., Electronics and Instrumentation Engineering, SASTRA University, India, 2011

Employment History:

Graduate Teaching Assistant, University of Texas at Dallas, September 2014 – May 2020
Engineering (R&D) Intern, Samsung Semiconductors, Inc., February 2014 – June 2014
Embedded Software Intern, Samsung Semiconductors, Inc., August 2012 – December 2012

Scholarships and Awards:

Runner-up at the Annual Physionet/Computing in Cardiology Challenge, France, 2015
Recipient of Jonsson School Graduate Study Scholarship, UT Dallas, 2014
Recipient of Jonsson School Graduate Study Scholarship, UT Dallas, 2011
Recipient of Dean's List Award, SASTRA University, 2010
Recipient of Dean's List Award, SASTRA University, 2008

Publications:

V Kalidas and LS Tamil, *Detection of Premature Ventricular Complexes using Semisupervised Autoencoders and Random Forests*, 42^{nd} Annual IEEE Engineering in Medicine and Biology Conference, July 2020.

V Kalidas and LS Tamil, *Detection of Atrial Fibrillation using discrete-state Markov models and Random Forests*, Computers in Biology and Medicine, vol. 113, Oct. 2019.

V Kalidas and LS Tamil, *Real-time QRS Detector using Stationary Wavelet Transform for Automated ECG Analysis*, 2017 IEEE Intl. Conference on Bioinformatics and Bioengineering, Washington DC, Oct. 2017.

V Kalidas and LS Tamil, *Cardiac Arrhythmia Classification using Multi-modal Signal Analysis*, Physiological Measurement, vol. 37, no. 8, pp. 1253-72, Jul. 2016.

V Kalidas and LS Tamil, *Enhancing Accuracy of Arrhythmia Classification by Combining Logical and Machine Learning Techniques*, Computing in Cardiology 2015, vol. 42, pp. 733-736.

Professional Memberships:

Institute of Electrical and Electronics Engineers (IEEE), 2014–present