M.Sc. in Computer Science and Engineering Thesis

Inferring Health Conditions Through Applying a Fusion of Machine Learning and Biomedical Signal Processing

by

Nabil Ibtehaz

Submitted to

Department of Computer Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)

Dhaka, Bangladesh

8 February 2020

Declaration of Authorship

I, Nabil Ibtehaz, declare that the work presented in this thesis, titled, "Inferring Health Conditions Through Applying a Fusion of Machine Learning and Biomedical Signal Processing", is the outcome of the investigation and research carried out by us under the supervision of Dr. M. Sohel Rahman.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Nabil Ibtehaz

Nabil Ibtehaz

Candidate

The thesis titled "**Inferring Health Conditions Through Applying a Fusion of Machine Learning and Biomedical Signal Processing**", submitted by Nabil Ibtehaz, Roll No. 1017052037 F, Session: October 2017, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering on February 8, 2020.

Board of Examiners

Dr. M. Sohel Rahman — Chairman (Supervisor)
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)

Dr. Md. Mostofa Akbar — Member (Ex-Officio)
Professor and Head
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)

M. M. Islam 8/2/20

Dr. Md. Monirul Islam — Member
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)

Dr. Rifat Shahriyar — Member
Associate Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)

Dr. Swakkhar Shatabda — Member
Associate Professor
Department of Computer Science and Engineering
United International University

Acknowledgement

I would like to convey my heart-felt gratitude and thanks to my thesis supervisor Dr. M Sohel Rahman, Professor of the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka for the continuous guidance and support I have received from him during the course of this work. He was there with his kind help and direction anytime I was in need of them. His inspiration, motivation and friendly attitude helped make my work under him a great experience. It was a great pleasure to be a part of his research group and work under him.

I would also like to express my sincere appreciation to the members of my thesis committee, Dr. Md. Mostofa Akbar, Dr. Md. Monirul Islam, Dr. Rifat Shahriyar and Dr. Swakkhar Shatabda, for their valuable feedback and suggestion on my work.

In addition, my heart-felt thank goes to all the faculty members and my course teachers for their continuous help and guidance throughout the course of my M.Sc. program. I also thank my family members, without whose support and encouragement, it would be impossible to carry on with my study smoothly.

Nabil Ibtehaz

Dhaka
8 February 2020

Dedicated to my loving parents

Contents

List of Figures

List of Tables

Abstract

The human body is the most complicated apparatus, comprising a plethora of mutually connected organs, performing diversely different sophisticated functions. These interconnected organs and processes communicate with each other in forms of signals, which are commonly termed as Biomedical Signals. In addition to playing a vital role in the proper operation of various physiological tasks, these signals can also be used as indications of whether the body is functioning appropriately or is suffering from diseases. For a long time, Doctors have been using ECG (Electrocardiogram) signals to monitor the heart condition of the patients. Moreover, PPG (Photoplethysmogram) signals have been being used to infer blood pressure. Both have resulted in an abundance of academic research works, however, they seem to be disjoint in nature.

About half of the works in the literature revolve around Signal Processing based methods, where after monumental analysis of the signal a rather simpler mathematical model is developed. On the contrary, there are works following the approaches of Machine Learning, but they hardly consider the discernible patterns in the signals during feature engineering. Fusing the ideas from both the schools of Signal Processing and Machine Learning, we, therefore, propose an improved algorithmic pipeline, VFPred, that can detect Ventricular Fibrillation from ECG signals, which catalyzes life-threatening cardiac arrests. VFPred extends upon traditional signal processing based feature extraction and subsequently utilizes a suitable machine learning based classifier to not only demonstrate an outstanding accuracy, but also a balance between sensitivity and specificity.

On the contrary, employing the potential of Deep Learning, we develop PPG2ABP, that is capable of inferring the continuous arterial blood pressure waveform with minimal error from analyzing PPG signals. Use of deep learning emancipates PPG2ABP from the need for handcrafted features, which often restricts the input signals to follow an ideal shape. Furthermore, this enables us to surpass contemporary methods both in terms of reliability and versatility.

Chapter 1

Introduction

The human body continuously produces a tremendous amount of information that reflect the status of the functionalities and actions of the body. Such information can be captured by specialized sensors that measure different types of information in the form of signals, such as EEG (Electroencephalogram) that measures the brain activity, ECG (Electrocardiogram) and PPG (Photoplethysmogram) both of which measure the heart activities and so forth. These signals are of utmost importance as by studying these we can determine whether the body is functioning properly or not.

1.1 Biomedical Signals

Biomedical signals can be defined as inspection of physiological activities and mutual interaction of various organisms [21]. The goal of Biomedical Signal Processing is to extract significant and interesting information from biomedical signals, so that biologists can perform new biological discoveries and physicians can monitor patients for illnesses.

There exist a plethora of Biomedical signals, and the number is continually increasing due to the advancements in Biomedical Sensor Technology. The existing signals can be classified in a number of ways [29]. For example, based on the system of origin, they can be categorized into the signals from cardiovascular system, nervous system, auditory system, respiratory system, musculoskeletal system etc. Some of the well known signals are Electrocardiogram or ECG that measures the electrical activity of heart, Electroencephalography or EEG that records the electrical activity of brain, Photoplethysmogram or PPG that detects the blood volume changes in vessels, Electromyography or EMG that captures the electrical activity produced by skeletal muscles, Electrooculography or EOG that computes the corneo-retinal standing potential existing between the front and the back of eye, Phonocardiogram or PCG listens to sounds made by heart in a cardiac cycle, and the list goes on.

1.2 Scope of this Thesis

In this work, from the vast repertoire of diversified Biomedical Signals, we focus our efforts in processing and analyzing the two most prominent signals, namely, ECG and PPG. The rationale of being limited to these two are manifold with two primary reasons. Firstly, PPG and ECG signals are more widely accessible as compared to other biomedical signals like EEG or EOG. The recent emergence of Smartwatches and their rapidly increasing popularity and usage are contributing the most in this regard. As the smartwatches often contain PPG and ECG sensors, these are becoming more accessible to the common people. Secondly, ECG and PPG allows us to analyze the activities of heart to detect cariac abnormalities, which takes a heavy tole of millions of death worldwide every year. For example, sudden cardiac arrest was responsible for approximately 6 million deaths in Europe and in the United States [73]. Also more than 1.4 billion people worldwide were somehow affected by hypertension [121] in the year of 2014, and hypertension is responsible for a number of cardiovascular diseases. This huge number is alarming as more than 4 million people die of cardiovascular diseases every year only in Europe and when considering the whole world the number of deaths exceeds 17 million [110]. Therefore, we are motivated to study ECG and PPG signals and their potential applications in order to reduce the chances of such unfortunate deaths, as early detection often leads to a timely recovery.

ECG (Electrocardiogram) is by far the most commonly used biomedical signal to capture the activities of the human heart [117]. ECG signals detect the small electrical changes that are a consequence of cardiac muscle depolarization followed by repolarization during each cardiac cycle (heartbeat). Changes in the normal ECG pattern occur in numerous cardiac abnormalities, including cardiac rhythm disturbances (Atrial fibrillation, Ventricular tachycardia), inadequate coronary artery blood flow (Myocardial Ischemia and Myocardial Infarction), and electrolyte disturbances (Hypokalemia and Hyperkalemia).

Photoplethysmogram (PPG) is an optically obtained signal that can be used to detect blood volume changes in the microvascular bed of tissue [118]. The PPG signal is usually sensed by using a pulse oximeter, that can illuminate the skin and measure changes in light absorption [99]. Each cardiac cycle compels the heart to pump blood to the periphery, and despite being quite damped by the time it reaches the skin, this pressure pulse is capable of distending the arteries and arterioles in the subcutaneous tissue. By attaching the pulse oximeter without compressing the skin, a pressure pulse is visible from the venous plexus, resulting in a small secondary peak.

ECG and PPG both being very important biomedical signals have been used for a number of applications. Most notably, ECG has been being widely used for detection of various heart diseases [2,7,85]. On the other hand, PPG being connected to blood volume has been being used to predict blood pressures [37,65,108]. A plethora of algorithms based on signal processing algorithms have been introduced but they fall short when tested on a large amount of data [6], machine learning based approaches on the other hand, despite performing better, treat feature extraction as a black box and are often devoid of any proper rationale.

1.3 Objective of This Thesis

In this work, we will explore possible ways of fusing the ideas of two branches Biomedical Signal Processing and Machine Learning. From an elaborate literature review we have made the observations that although the traditional signal processing based approaches work from the intuition of the doctors and manage to capture some vital properties of the signal, they fail to generalize when applied on a large real world dataset [6]. Machine learning based approaches, on the other hand, performs a mix and match of the features presented in the signal processing literature and applies a machine learning algorithm. Machine learning based approaches are adaptive as they learn from the data directly, therefore the generalize comparatively better. However, these features are often incompatible, hence not only they tend to miss the actual significant properties of the signal but also some combination may lead to worse performance [72]. Based on all these analyses, we believe a proper fusion of Signal Processing and Machine Learning can lead to more robust and accurate algorithms bringing the best of the both worlds.

The primary objectives of our work are as follows:

- To perform a thorough literature review and experiment with the efficacy of various features and attributes of the biological signals
- To implement some modifications over the existing robust signal processing pipelines to improve the feature extraction process
- To apply a suitable machine learning algorithm that can exploit the behaviors of the signals under consideration
- To make all the codes open source, so that they can be utilized in wearables and server side applications

1.4 Thesis Contribution

This thesis makes the following contributions:

- It presents VFPred, a combined Signal Processing and Machine Learning based algorithm to detect Ventricular Fibrillation from ECG signals.

- It presents PPG2ABP, a combined Signal Processing and Deep Learning based algorithm to predict Blood Pressure from PPG signal alone.
- It provides a modular, open source implementation of the proposed algorithms.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 briefly discusses the relevant ideas from both Signal Processing and Deep Learning literature utilized in this research work. In Chapter 3 a detailed description of the algorithmic pipeline VFPred is provided, which fuses the ideas of signal processing and machine learning to detect Ventricular Fibrillation from ECG signals. Chapter 4 presents the algorithmic pipeline PPG2ABP, which predicts the continuous waveform of arterial blood pressure using PPG signals. Finally, Chapter 5 concludes this thesis.

Chapter 2

Preliminaries

In this chapter we briefly discuss the concepts of Signal Processing and Deep Learning which are essential to methods explored and developed in this thesis.

2.1 Signal Processing Techniques

Proper usage of the tools of Signal Processing is of utmost importance when we work with Biomedical Signals. Not only these techniques aid in filtering and pre-processing the signals, but also such algorithms are almost vital to extract features from the raw signals. Two very important concepts of Signal Processing which are used in analyzing Biomedical Signals, namely, Empirical Mode Decomposition and Discrete Fourier Transform, are described as follows.

2.1.1 Empirical Mode Decomposition

Empirical Mode Decomposition (EMD), proposed by Huang et al. [47], is a data-driven, adaptive signal processing method which is suitable for analyzing non-stationary and nonlinear signals, like the ECG signal. This algorithm decomposes a signal into a sum of Intrinsic Mode Functions (IMF). An IMF represents a simple oscillatory function with the following characteristics:

1. The number of extrema and the number of zero crossings must either be equal or differ (with each other) at most by one.
2. At any point, the mean value of the envelopes of the local maxima and minima is zero.

By applying the EMD algorithm we decompose a signal x(t) into a set of IMF functions in decreasing order of oscillatory behavior, and possibly a Residue,

$$x(t) = IMF_1(t) + IMF_2(t) + ... + IMF_n(t) + R(t) \tag{2.1}$$

Given a signal $x(t)$ the steps are as follows:

1. Identify all the local maxima and local minima points of $x(t)$.
2. Connecting all the local maxima and minima points by a cubic spline, we obtain the upper and lower envelope $e_u(t)$ and $e_l(t)$ respectively.
3. Next we compute the pointwise mean of the upper and lower envelopes $m_{10}(t)$ as follows

$$m_{10}(t) = \frac{e_u(t) + e_l(t)}{2} \tag{2.2}$$

4. We perform pointwise subtraction of the mean envelop values from the original signal, and thereby we obtain the first component $h_{10}(t)$ as follows,

$$h_{10}(t) = x(t) - m_{10}(t) \tag{2.3}$$

5. In an ideal case this $h_{10}(t)$ should be an IMF function, i.e. it would possess the two characteristics mentioned above. However in practice this is often not the case as $h_{10}(t)$ may still lack the two characteristics. In such cases, we again use $h_{10}(t)$ as the main signal and repeat the steps 1-5 to obtain $h_{11}(t)$,

$$h_{11}(t) = h_{10}(t) - m_{11}(t) \tag{2.4}$$

6. This process is repeated k times until the IMF conditions are met. In the sequel we get:

$$h_{lk}(t) = h_{l(k-1)}(t) - m_{lk}(t) \tag{2.5}$$

7. Since, this is a numerical method some other stopping criteria may also be used. A commonly used criterion could be based on the value of Standard Deviation (SD) between the two consecutive iterations, as [47] proposes:

$$SD = \sum_{t=0}^{T} \frac{|h_{1(k-1)}(t) - h_{1k}(t)|^2}{h^2_{1(k-1)}(t)} \tag{2.6}$$

The idea is that when SD is smaller than a predefined threshold, we consider $h_{1k}(t)$ as the first IMF component, $IMF_1(t)$

8. Next we separate the IMF component from the original signal $x(t)$ and obtain $x_1(t)$ as follows:

$$x_1(t) = x(t) - IMF_1(t) \tag{2.7}$$

9. However this $x_1(t)$ may contain significant oscillatory behavior. So we take $x_1(t)$ as the original signal and repeat steps 1-9, to obtain successive IMF components IMF_2, IMF_3, ... and so on.

10. This process is continued until we are left with a zero valued function or a monotonic function, i.e. the Residue, R.

2.1.2 Discrete Fourier Transform (DFT)

The Fourier Series proposed by Joseph Fourier in 19^{th} century, was originally developed to represent a periodic signal as an infinite sum of simple harmonic oscillations of different frequencies [88]. This allows us to analyze the impact of individual frequency bands over a signal. Later it was extended to aperiodic signals through Fourier Transform. For finite and discrete signals we sample the Fourier Transform coefficients as a finite sequence, that corresponds to a complete period of the original signal [88]. Thus, Discrete Fourier Transform (DFT) for a discreate signal x of length N is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp(-\frac{2\pi i}{N} kn), \quad for \quad 0 \leq k \leq N-1 \tag{2.8}$$

2.2 Deep Learning Architectures

In recent years, with the advancement of parallel processing capabilities in GPUs (Graphics Processing Unit), Deep Learning has brought about a revolution in almost every domains, including healthcare and medical informatics. Convolutional Neural Networks, being unparalleled in analyzing images, also performs remarkably well when working with signals. In the following subsections we briefly discuss two novel deep learning networks, namely U-Net and MultiResUNet.

2.2.1 U-Net

Similar to FCN [75] and SegNet [11], U-Net [94] uses a network of convolutional layers entirely to perform the task of semantic segmentation. The network architecture is symmetric, having an *Encoder* that extracts spatial features from the image, and a *Decoder* that constructs the segmentation map from the encoded features. The *Encoder* follows the typical formation of a convolutional network. It involves a sequence of two 3×3 convolution operations, followed by a max-pooling operation with a pooling size of 2×2

and stride of 2. This sequence is repeated four times, and after each down-sampling, the number of filters in the convolutional layers is doubled. Finally, a progression of two 3×3 convolution operations connects the *Encoder* to the *Decoder*.

On the other hand, the *Decoder* first up-samples the feature map using a 2×2 transposed convolution operation [124], reducing the feature channels by half. Then a sequence of two 3×3 convolution operations is performed again. Similar to the *Encoder*, this succession of up-sampling and two convolution operations is repeated four times, halving the number of filters at each stage. Finally, a 1×1 convolution operation is performed to generate the final segmentation map. All convolutional layers in this architecture, except for the final one, use the *ReLU* (Rectified Linear Unit) activation function [68]; the final convolutional layer uses a *Sigmoid* activation function.

Perhaps, the most ingenious aspect of the U-Net architecture is the introduction of skip connections. In all the four levels, the output of the convolutional layer, prior to the pooling operation of the *Encoder* is transferred to the *Decoder*. These feature maps are then concatenated with the output of the up-sampling operation, and the concatenated feature map is propagated to the successive layers. These skip connections allow the network to retrieve the spatial information lost by pooling operations [34]. The network architecture is illustrated in Figure 2.1.

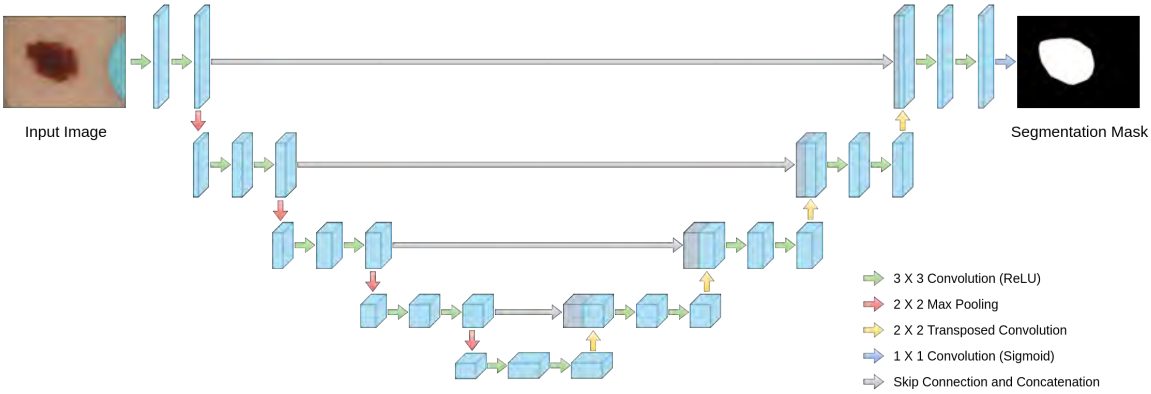

Input Image
Segmentation Mask
3 X 3 Convolution (ReLU)
2 X 2 Max Pooling
2 X 2 Transposed Convolution
1 X 1 Convolution (Sigmoid)
Skip Connection and Concatenation

Figure 2.1: The classic U-Net architecture. The model comprises an encoder and a decoder pathway, with skip connections between the corresponding layers (Figure borrowed from [49]).

Subsequently, the U-Net architecture was extended through a few modifications to 3D

U-Net for volumetric segmentation [26]. In particular, the two-dimensional convolution, max pooling, transposed convolution operations were replaced by their three-dimensional counterparts. However, in order to limit the number of parameters, the depth of the network was reduced by one. Moreover, along with using batch normalization [52], the number of filters was doubled before the pooling layers to avoid bottlenecks [107]. The original U-Net [94] did not use batch normalization. However, when experimented with it later, the results revealed, perhaps astonishingly, that batch normalization may even hurt the performance sometimes [26].

2.2.2 MultiResUNet

U-Net has been a remarkable and the most popular deep network architecture in the medical imaging community, defining the state of the art in medical image segmentation [34]. However, as will be clear shortly, U-Net is not enough to handle the problem we handle here. On the other hand a very recent proposal of MultiResUNet seems more promising. Through deep contemplation of the U-Net architecture and drawing some parallels to the recent advancement in the field of deep computer vision, some insightful and useful observations can be made. These observations lead to the development of MultiResUNet, a novel architecture, whose the two core components are as follows:

MultiRes Block

In the U-Net architecture, after each pooling layer and transposed convolutional layer, a sequence of two 3 × 3 convolutional layers are used. As explained in [107], this series of two 3 × 3 convolutional operation actually resembles a 5 × 5 convolutional operation. Therefore, following the approach of the Inception network, the simplest way to augment U-Net with a multi-resolutional analysis capability is to incorporate 3 × 3, and 7 × 7 convolution operations in parallel to the 5 × 5 convolution operation, as shown in Figure 2.2a.

Therefore, replacing the convolutional layers with Inception-like blocks should facilitate the U-Net architecture to reconcile the features learnt from the image at different scales. Another possible option is to use strided convolutions [116], but in our experiments, it is overshadowed by the former. Despite the gain in performance, the introduction of additional convolutional layers in parallel extravagantly increases the memory requirement. Therefore, we improvise with the following ideas borrowed from [107]. We factorize the bigger, more demanding 5×5 and 7×7 convolutional layers, using a sequence of smaller and lightweight 3×3 convolutional blocks, as shown in Figure 2.2b. The outputs of the 2nd and the 3rd 3×3 convolutional blocks effectively approximate the 5×5 and 7×7 convolution operations respectively. We hence take the outputs from the three convolutional blocks and concatenate them together to extract the spatial features from different scales. From our experiments, it is seen that the results of this compact block closely resemble that of the memory-intensive Inception-like block described earlier. This outcome is in line with the findings of [107], as the adjacent layers of a vision network are expected to be correlated.

Although this modification greatly reduces the memory requirement, it is still quite demanding. This is mostly due to the fact that in a deep network if two convolutional layers are present in a succession, then the number of filters in the first one has a quadratic effect over the memory [106]. Therefore, instead of keeping all the three consecutive convolutional layers with an equal number of filters, we gradually increase the filters in those (from 1 to 3), to prevent the memory requirement of the earlier layers from exceedingly propagating to the deeper part of the network. We also add a residual connection because of their efficacy in biomedical image segmentation [34] as well as to introduce 1×1 convolutional layers, which may allow us to comprehend some additional spatial information. We call this arrangement a '*MultiRes block*', as shown in Figure 2.2c.

Res Path

An ingenious contribution of the U-Net architecture was the introduction of shortcut connections between the corresponding layers before the max-pooling and after the de-

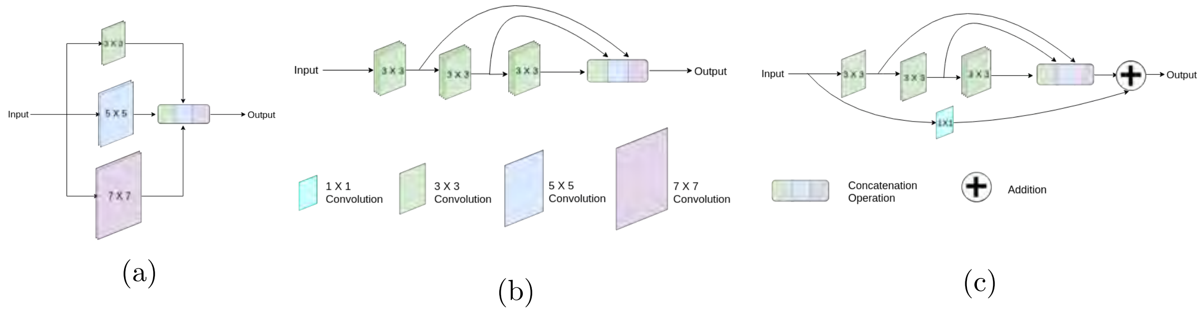


Figure 2.2: Developing the proposed *MultiRes* block. We start with a simple Inception-like block by using 3×3, 5×5 and 7×7 convolutional filters in parallel and concatenating the generated feature maps (Fig. 2.2a). This allows us to reconcile spatial features from different context size. Subsequently, instead of using the 3×3, 5×5 and 7×7 filters in parallel, we factorize the bigger and more expensive 5×5 and 7×7 filters as a succession of 3×3 filters (Fig. 2.2b) . Fig 2.2c illustrates the *MultiRes* block, where we have increased the number of filters in the successive three layers gradually and added a residual connection, along with 1×1 filters for conserving dimensions (Figure borrowed from [49]).

convolution operations. This enables the network to propagate the spatial information that gets lost during the pooling operation from encoder to decoder.

Despite preserving the dissipated spatial features, a flaw of the skip connections may be speculated as follows. For instance, the first shortcut connection bridges the encoder before the first pooling with the decoder after the last deconvolution operation. Here, the features coming from the encoder are supposed to be lower level features as they are computed in the earlier layers of the network. On the contrary, the decoder features are supposed to be of much more higher level, since they are computed at the very deep layers of the network, thereby, going through more processing. Hence, we observe a possible semantic gap between the two sets of features being merged. We conjecture that the fusion of these two arguably incompatible sets of features could cause some discrepancy throughout the learning thereby adversely affecting the prediction procedure. It may be noted that the amount of discrepancy is likely to decrease gradually as we move towards the succeeding shortcut connections. This can be attributed to the fact that in later stages, not only the features from the encoder are going through more processing, but also we are fusing them with decoder features of much juvenile layers.

Therefore, to alleviate the disparity between the encoder-decoder features, we propose to incorporate some convolutional layers along the shortcut connections. We hypothesize that these additional non-linear transformations on the features propagating from the encoder stage should account for or somewhat balance the possible semantic gap (alluded to above) introduced by the higher degree of processing by the deeper decoder stages. Furthermore, instead of using the usual convolutional layers, we introduce residual connections to them as they make the learning easier [105] and is proven to have great potential in medical image analysis [34]. This idea is inspired from the image to image conversion using convolutional neural networks [78], where pooling layers are not favourable for the loss of information. Thus, instead of simply concatenating the feature maps from the encoder stages to the decoder stages, we first pass them through a chain of convolutional layers with residual connections and then concatenate them with the decoder features. We call this proposed shortcut path '*Res path*', illustrated in Fig. 2.3. More specifically, 3×3 filters are used in the convolutional layers and 1×1 filters accompany the residual connections.

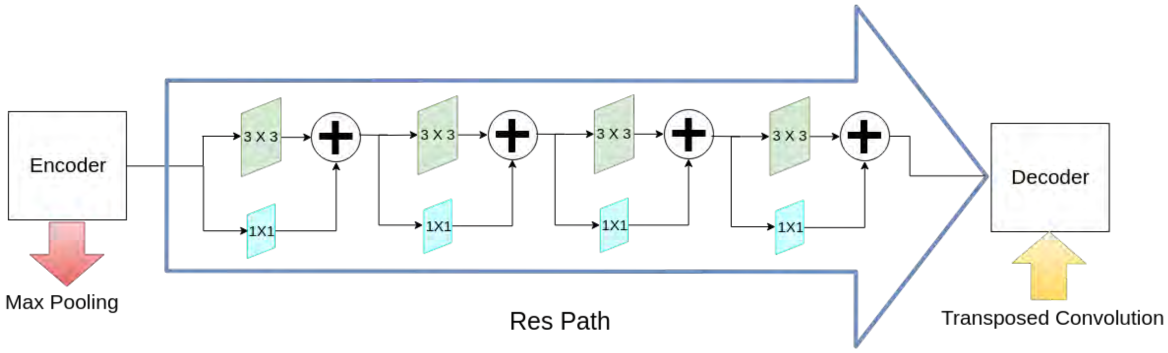


Figure 2.3: Proposed *Res* path. Instead of combining the encoder feature maps with the decoder feature in a straight-forward manner, we pass the encoder features through a sequence of convolutional layers. These additional non-linear operations are expected to reduce the semantic gap between encoder and decoder features. Furthermore, residual connections are also introduced as they make the learning easier and are very useful in deep convolutional networks (Figure borrowed from [49]).

Model Architecture

In the MultiResUNet model, we replace the sequence of two convolutional layers with the proposed *MultiRes* block as introduced in Section 2.2.2. For each of the *MultiRes* blocks, we assign a parameter W that controls the number of filters of the convolutional layers inside that block. To maintain a comparable relationship between the numbers of parameters in the original U-Net and the proposed model, we compute the value of W as follows:

$$W = \alpha \times U \tag{2.9}$$

Here, U is the number of filters in the corresponding layer of U-Net and α is a scalar coefficient. This provides us with a convenient way to both control the number of parameters and keep them comparable to that of U-Net. We compare our proposed model with an U-Net, having $\#filters = [32, 64, 128, 256, 512]$ along the levels, which are also the values of U in our model. We set $\alpha = 1.67$ as it keeps the number of parameters in our model slightly below that of U-Net.

In Section 2.2.2, we have pointed out that it is beneficial to gradually increase the number of filters in the successive convolutional layers inside a *MultiRes* block, instead of keeping them the same. Hence, we assign $*\frac{W}{6}$, $*\frac{W}{3}$ and $*\frac{W}{2}$ filters to the three successive convolutional layers respectively, as this combination has achieved the best results in our experiments. Also, it can be noted that similar to the U-Net architecture, after each pooling or deconvolution operation, the value of W gets doubled or halved respectively.

In addition to introducing the *MultiRes* blocks, we also replace the ordinary shortcut connections with the proposed *Res* paths. Therefore, we apply some convolution operations on the feature maps propagating from the encoder stage to the decoder stage. In Section 2.2.2, we hypothesized that the intensity of the semantic gap between the encoder and decoder feature maps are likely to decrease as we move towards the inner shortcut paths. Therefore, we also gradually reduce the number of convolutional blocks used along the *Res* paths. In particular, we use $4, 3, 2, 1$ convolutional blocks respectively along the four *Res* paths. Also, in order to account for the number of feature maps in encoder-decoder, we use $32, 64, 128, 256$ filters in the blocks of the four *Res* paths respectively.

All the convolutional layers in this network, except for the output layer, are activated by the *ReLU* (Rectified Linear Unit) activation function [68], and are batch-normalized [52]. Similar to the U-Net model, the output layer is activated by a *Sigmoid* activation function. We present a diagram of the proposed MultiResUNet model in Fig. 2.4.

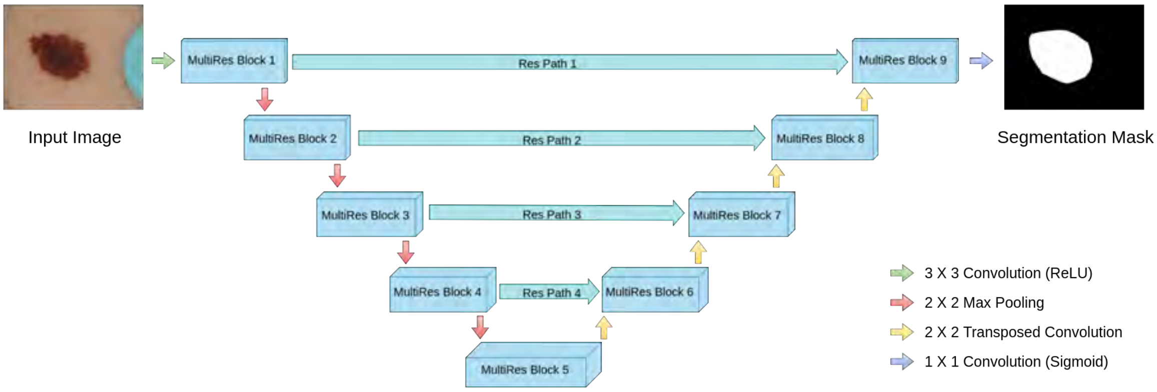


Figure 2.4: Proposed MultiResUNet architecture. We replace the sequences of two convolutional layers in the U-Net architecture with the proposed *MultiRes* blocks. Furthermore, instead of using plain shortcut (skip) connections, we use the proposed *Res* paths (Figure borrowed from [49]).

Chapter 3

Detection of Ventricular Fibrillation from ECG Signals

Ventricular Fibrillation (VF), one of the most dangerous arrhythmias, is responsible for sudden cardiac arrests. Thus, various algorithms have been developed to predict VF from electrocardiogram (ECG). In the literature, we find a number of algorithms based on signal processing, where, after some robust mathematical operations the decision is given based on a predefined threshold over a single value. On the other hand, some machine learning based algorithms are also reported in the literature; however, these algorithms merely combine some parameters and make a prediction using those as features. Both the approaches have their perks and pitfalls; thus our motivation has been to coalesce them in order to get the best out of the both worlds.

3.1 Introduction

Ventricular Fibrillation (VF) is a type of cardiac arrhythmia which occurs when the heart quivers instead of pumping due to some disturbance in electrical activity in the ventricles [92]. This arrhythmia may result in a cardiac arrest leaving the patient unconscious without any pulse. Ventricular Fibrillation is found initially in about 10% of people in cardiac arrest [13] and sudden cardiac arrest is responsible for approximately 6 million deaths in Europe and in the United States [73]. Therefore, fast and accurate detection of Ventricular Fibrillation can save a lot of lives. Electrocardiograph (ECG) signal captures the electrical activities of the human heart. Trained, experienced doctors can analyze these ECG signals and understand the heart condition. However, for the detection and prevention of sudden cardiac arrests caused by arrhythmias like Ventricular Fibrillation, a continuous monitoring of the ECG signal of a patient is essential; this would require a specialized doctor to examine the ECG signal relentlessly. Unfortunately, it is neither practical nor possible for a doctor to continuously monitor ECG signals of a number of patients. This is the primary motivation for developing algorithms to analyze patterns from ECG signals of patients to detect arrhythmias.

Ventricular Fibrillation, being one of the most severe life-threatening arrhythmias, has been studied diligently for over four decades. A number of methods have been proposed for the detection of Ventricular Fibrillation over this time period. Very early works include simple signal processing [15,23,64,109,125]. Subsequently, more advanced signal processing based methods were introduced [5,7–9]. Unfortunately, these methods fail to maintain accuracy when tested on a large dataset [6] (further explained in Section 3.6.2). In recent past, with the emergence of machine learning techniques, better performing algorithms were introduced that extracted features from previously established ECG parameters and employed a machine learning classifier [3,4,10,27,72,104,113]. Though the performance improved, still it is far from perfection and several algorithms have certain limitations.

The signal processing based algorithms for VF prediction actually originate from intuition, observation, and experience of the doctors; then the monumental mathematical tools of signal processing are used to exploit those patterns, and finally based on a threshold or two the decision is made. On the contrary, machine learning based algorithms treat signal characteristics as a black box, they take a number of ECG parameters (collected from those signal processing based algorithms) and train a classification model using those as features. These are more robust as learning from the pattern of a huge variety of data can potentially outperform simple and constrained rule based approaches. Hence our motivation was to blend the two: we started with the intuition of the doctors, formulated those observations using signal processing methods and finally followed a machine learning protocol to acknowledge the diversity in real medical data. Our algorithm is based on the fact that QRS complexes are absent in VF class ECG signals [54]. We exploited this property and extracted this pattern using EMD (Empirical Mode Decomposition) with DFT (Discrete Fourier Transform) based feature engineering. We select the best set of features using Random Forests, and after making several attempts with Logistics Regression, Random Forest, Neural Networks, we finally train a SVM model and evaluate our model on benchmark datasets.

Notably, there exist a plethora of prior works exploiting EMD, DFT, and SVM, albeit mostly as separate methods. Our current work, VFPred [50], stands out in this context as we have been able to make a proper fusion of the methods from different domains. This is evident from the remarkable performance of VFPred as reported in a later section. More specifically, the prior SVM based works treated feature engineering as a black box and exercised a mix-and-match strategy on some arbitrary parameters. On the contrary, we have used intuition and rationale to investigate the effectiveness and efficacy of various features at length, followed by justification through extensive and thorough experimentation to attain the true potential of our chosen classifier algorithm. Also to the best of our knowledge, no prior work made an ensemble of EMD and DFT to interpret the patterns of the ECG signals.

3.2 Datasets

For developing and evaluating VFPred algorithm, we used the following two benchmark datasets of Ventricular Fibrillation detection as has been commonly used in most other works.

1. The MIT-BIH Malignant Ventricular Arrhythmia Database (VFDB) [43]: This database includes 22, 30 minutes long ECG recordings of subjects who experienced

episodes of sustained Ventricular Tachycardia, Ventricular Flutter, and Ventricular Fibrillation.

The ECG signals of this database were sampled with a sampling frequency of 250 Hz.

2. Creighton University Ventricular Tachyarrhythmia Database (CUDB) [87] : This database includes 35, 8 minutes long ECG recordings of human subjects who experienced episodes of sustained Ventricular Tachycardia, Ventricular Flutter, and Ventricular Fibrillation.

   The ECG signals of this database were filtered by an active second order Bessel low-pass filter with a cutoff frequency of 70 Hz. Then they were sampled at 250 Hz and quantized with 12-bit resolution over 10 V range.

These datasets are hosted on PhysioNet [41], and are publicly available.

Following the practice in the literature, we experimented on ECG episodes of length, $T_e = 2$ sec, 5 sec and 8 sec. In order to extract the ECG episodes, first a window of length T_e sec is taken, and the window is shifted by 1 sec to collect the consequent episodes, till the end of the signal. Finally, the episodes were annotated as ‘VF’ or ‘Not VF’ using the expert annotations provided in the dataset. We considered the entire dataset except for the few segments annotated as noise. We only used channel 1 data from VFDB to avoid redundancy. Notably, the dataset is highly imbalanced: we roughly have only 9% of the data as VF (Please refer to Table 3.1).

Table 3.1: Overview of the Combined VF Dataset. The individual continuous signals have been windowed to construct episodes of 2s, 5s ad 8s long respectively. Overall, the number of VF and Not VF episodes have been listed as well as their ratio, poining to the data imbalance.

Episode Length	Number of VF episodes	Number of Not VF episodes	Ratio (VF : Not VF)
2s	5692	51368	9.97 : 90.03
5s	5320	51087	9.43 : 90.57
8s	5013	50823	8.97 : 91.03

3.3 VFPred: Algorithmic Pipeline

The proposed VFPred algorithm takes an ECG signal of T_e seconds long, and through the following pipeline predicts whether Ventricular Fibrillation (VF) is present or not.

3.3.1 Signal Preprocessing and Filtering

ECG signals are usually corrupted by various kinds of noises and interferences. To overcome this, the input ECG signal is preprocessed following the filtering process originally proposed in [6] and slightly modified in [8]. The modified filtering procedure is as follows:

1. The mean value is subtracted from all the samples, thus making the mean zero.
2. Next, a moving average filter of order 5 is applied. For an ECG signal, this should remove most of the interspersions and muscle noise.
3. Then, the signal is filtered using a high pass filter of cut-off frequency, $f_c = 1$ Hz. This imposes drift suppression on the signal.
4. Finally a low pass butterworth filter of order 12 and cut-off frequency, $f_c = 20$ Hz is applied to attenuate the unnecessary high frequency information.

Figure 3.1 illustrates the outcome of preprocessing and filtering on the signal. As can be observed this step removes a great deal of noises from the ECG signal.

3.3.2 Analyzing the Oscillatory Characteristics

Empirical Mode Decomposition (EMD)

Empirical Mode Decomposition (EMD), proposed by Huang et al. [47], is a data-driven, adaptive signal processing method which is suitable for analyzing non-stationary and nonlinear signals, like the ECG signal. This algorithm decomposes a signal into a sum

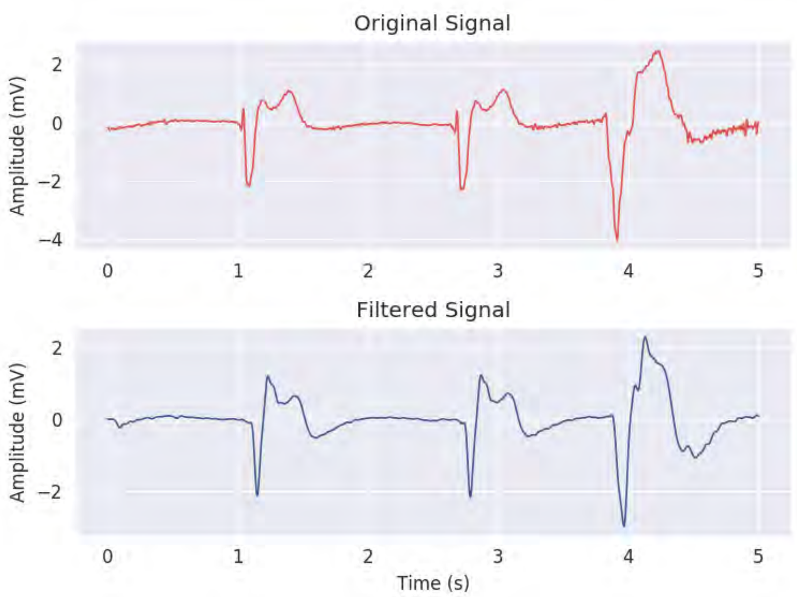


Figure 3.1: Signal Preprocessing and Filtering Step. This process removes most noises and artifacts from the ECG signal and smoothens the signal. Moreover this step also makes the signal zero mean.

of Intrinsic Mode Functions (IMF). An IMF represents a simple oscillatory function with the following characteristics:

1. The number of extrema and the number of zero crossings must either be equal or differ (with each other) at most by one.

2. At any point, the mean value of the envelopes of the local maxima and minima is zero.

By applying the EMD algorithm we decompose a signal x(t) into a set of IMF functions in decreasing order of oscillatory behavior, and possibly a Residue, as shown in Figure 3.2.

$$x(t) = IMF_1(t) + IMF_2(t) + ... + IMF_n(t) + R(t) \tag{3.1}$$

Observing IMF components to Distinguish VF from Not VF

IMF components describe the oscillatory characteristics of a signal. From the studies of ECG signal, it has been found that except for 'VF' signals all other classes of ECG signals

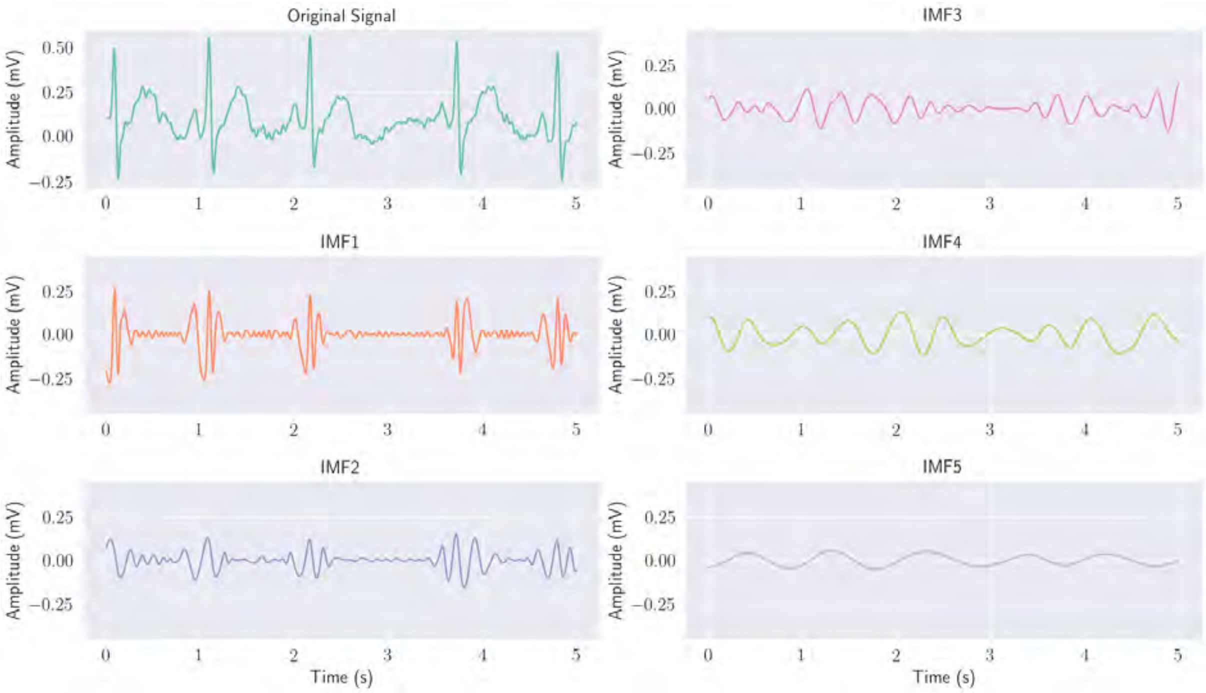


Figure 3.2: Empirical Mode Decomposition of an ECG Signal. Here we can observe that the top IMF components are more oscillatory in nature and gradually their oscillation diminishes. IMF_1 and IMF_2 captures most of the oscillations from the original signal and it progressively fades in IMF_3, IMF_4 and IMF_5.

contain a QRS complex [54]. Thus, we have a property that apparently separates the 'VF' from the 'Not VF'. An attempt can be made to find a proper formulation to distinguish 'VF' signals exploiting this characteristic.

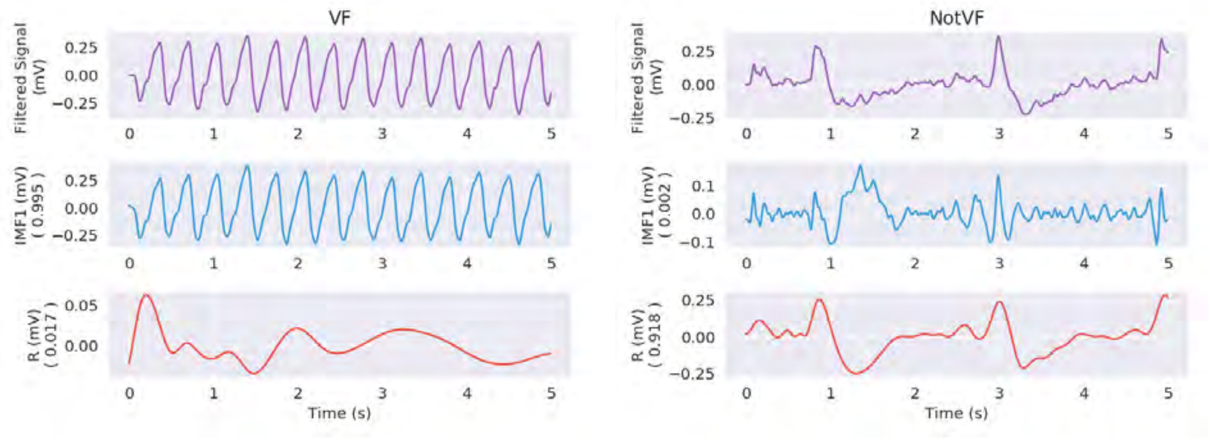
The presence of a QRS complex distorts the upper and lower envelopes as they introduce additional local maxima and local minima points. On the contrary due to the absence of any QRS complex, 'VF' class ECG signals have envelopes that are symmetric in nature. The small interval QRS complexes in the 'Not VF' class ECG signals result in higher frequency oscillations. Thus their 1st IMF component captures those oscillations missing the actual ECG signal itself. On the other hand, for the lack of QRS complex, the 'VF' class ECG signals are not supposed to have major high frequency oscillations compared to 'Not VF' class ECG signals. Thus the 1st IMF component of 'VF' class is likely to follow the original ECG signal.

Thus, it should be possible to distinguish the two classes by observing how much the 1st IMF component correlates with the ECG signal itself. Moreover, the remaining signal can also be analyzed to increase the robustness, which will be referred as the Residue component [8]. Since the IMF and Residue components are disjoint in nature it is expected that the Residue of 'Not VF' signal will match the original ECG signal more than that of 'VF' class ECG signal. The cosine similarity metric can be taken as a measure of similarity between the two signals [8]. Thus $IMF_{similarity}$ and $R_{similarity}$ can be defined as follows:

$$IMF_{similarity} = \frac{Signal \bullet IMF_1}{\mid Signal \mid \; \mid IMF_1 \mid} \tag{3.2a}$$

$$R_{similarity} = \frac{Signal \bullet R}{\mid Signal \mid \; \mid R \mid} \tag{3.2b}$$

These facts are illustrated in Figure 3.3,



(a) (b)

Figure 3.3: Relation of the 1st IMF and R components with the original signal, the cosine similarities are presented inside the parentheses. In (a) for a 'VF' class signal, we observe that the 1st IMF and the original signal are very similar (cosine similarity = 0.995) while R deviates quite a bit (cosine similarity = 0.017). (b) on the other hand, shows an example from 'Not VF' class. Here, we can observe distinct QRS peaks, and they prevent the 1st IMF from capturing any useful information (cosine similarity of 0.002 with the original signal). Rather they are quite oscillatory accounting for the non-uniform upper and lower envelops. This compels the residue to closely follow the original signal (cosine similarity = 0.918).

Extracting Oscillatory Characteristics from ECG Signals

Even after preprocessing and filtering, some high frequency noises may still prevail in the signal. In such cases, the 1st IMF component would actually represent those noises. To determine whether the 1st IMF component contains useful information or noise we follow the scheme proposed in [8]. In order to extract the oscillatory features from the ECG signals we perform the following steps:

- Empirical Mode Decomposition is performed on the filtered signal and the first two IMF components along with the Residue are computed as follows:

$$x(n) = IMF_1(n) + IMF_2(n) + R(n) \tag{3.3}$$

where:

x(n)	=	Filtered Signal
IMF$_1(n)$	=	1st IMF component
IMF$_2(n)$	=	2nd IMF component
R(n)	=	The Residue

- The noise level (V_n) is calculated as a percentage of the maximum ECG signal amplitude as follows:

$$V_n = \alpha(\max(x(n))) \tag{3.4}$$

Where, α is a constant that should be chosen wisely. We took $\alpha = 0.05$.

- The samples n_L of $IMF_1(n)$ that falls within $-V_n$ to V_n are identified, i.e.,

$$n_L = \{t : t \in n, |IMF_1(t)| \leq V_n\} \tag{3.5}$$

- The noise level crossing ratio ($NLCR$) is calculated using the following formula,

$$NLCR = \frac{\sum_{n \in n_L} IMF_1^2(n)}{\sum_{n \in n_L} x^2(n)} \tag{3.6}$$

- Finally, the appropriate IMF is selected as follows :

$$IMF = \begin{cases} IMF_1(n) + IMF_2(n) & \text{, if } NLCR \leq \beta \\ IMF_1(n) & \text{, otherwise} \end{cases} \tag{3.7}$$

  Here, β is a properly chosen constant. We considered $\beta = 0.02$.

Thus, the appropriate IMF component and the Residue is taken for further analysis.

Observing Oscillatory Characteristics on Dataset

Based on our analysis above, the expected value of $IMF_{similiarity}$ for 'VF' is greater than 0.5 whereas it is less than 0.5 for 'Not VF'. On the contrary, the expected value of $R_{similarity}$ is just the opposite, i.e., less than 0.5 for 'VF' and greater than 0.5 for 'Not VF'. However, it was found that these theoretical observations are not always found to be true in practice, especially, when checked against a large amount of diverse data. In particular, a lot of overlaps are observed between the two classes.

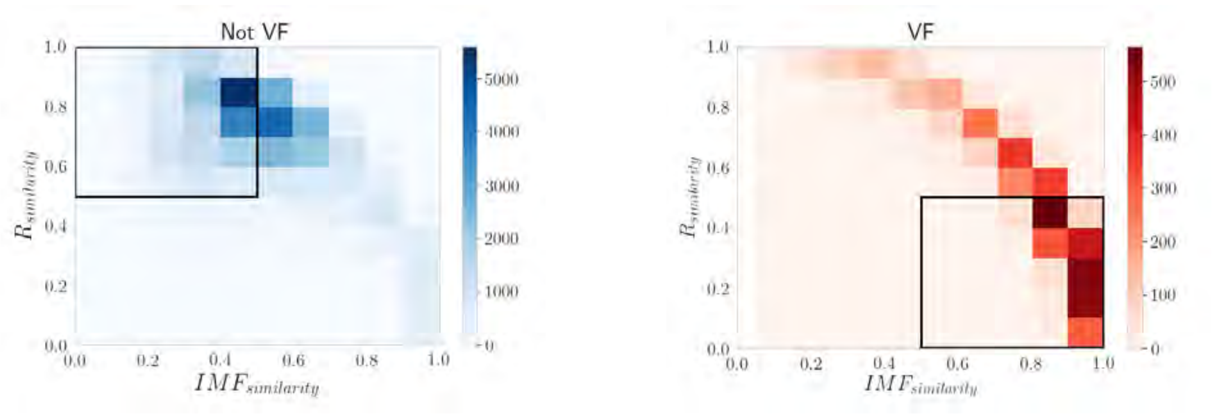
In Figure 3.4, we have presented a 2D histogram that shows the distribution of $IMF_{similarity}$ and $R_{similarity}$ of the two classes. For both the classes, according to our theoretical analyses, the data points should have been within the bounding boxes. However, this is not the case as we can observe a great amount of overlapping.

Hypothesis

From further experimentation with data we came up with the following hypothesis:

> *Even after filtering and preprocessing, there may still remain some undesired frequency components in our data, preventing us from separating the two classes.*

In other words, there may be certain frequency components of the IMF and certain (not necessarily the same) frequency components of the Residue which may allow us



(a)

(b)

Figure 3.4: 2D Histograms representing the distribution of $IMF_{similarity}$ and $R_{similarity}$ of ‘VF’ and ‘Not VF’. From our theoretical analyses, the IMF component should be similar to the original signal for a ‘VF’ class signal and it is likely to be different for a ‘Not VF’ class signal. On the contrary, the Residue component should diverge from the original signal for a ‘VF’ class signal, but should closely follow the original signal for ‘Not VF’. Hence, bounding boxes have been drawn where the points are expected to reside. Here the two classes should have been confined within the black bounding boxes, but a lot of overlapping is clearly visible.

to distinguish the two classes accurately. Hence, we need to prioritize these frequency components while making a decision.

So instead of taking the straightforward cosine similarity of the Signal with IMF and R as features, as is done in [8], we focus on the frequency information.

3.3.3 Extracting Frequency Information from Oscillations

Discrete Fourier Transform (DFT)

The Fourier Series proposed by Joseph Fourier in 19^{th} century, was originally developed to represent a periodic signal as an infinite sum of simple harmonic oscillations of different frequencies [88]. This allows us to analyze the impact of individual frequency bands over a signal. Later it was extended to aperiodic signals through Fourier Transform. For finite and discrete signals we sample the Fourier Transform coefficients as a finite sequence, that corresponds to a complete period of the original signal [88]. Thus, Discrete Fourier Transform (DFT) for a discreate signal x of length N is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp(-\frac{2\pi i}{N}kn), \quad for \quad 0 \leq k \leq N-1 \tag{3.8}$$

Observing The Frequency Components

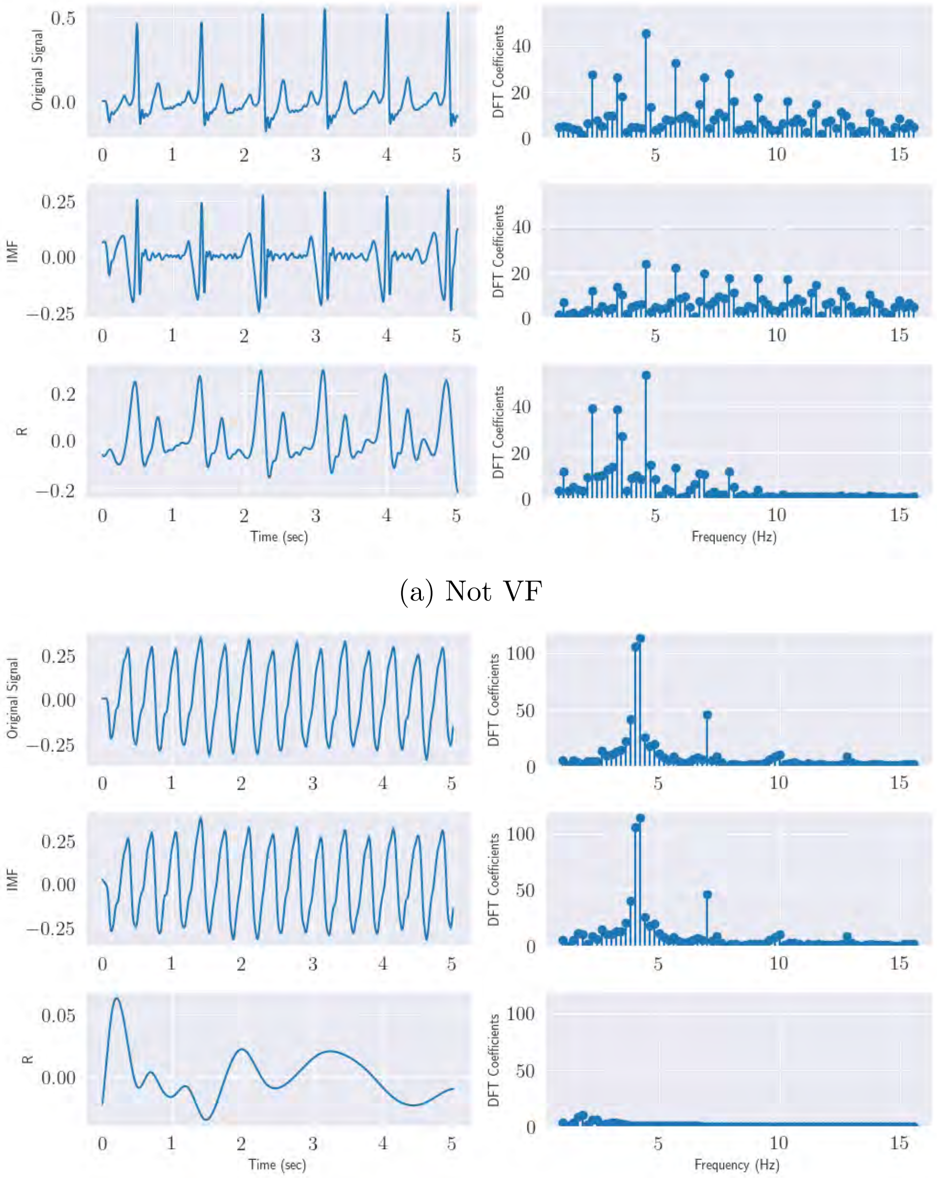
Our theoretical analyses indicate that for a 'VF' class ECG signal, the IMF component should be similar to the original signal, and the Residue component is likely to deviate from it. The opposite is expected for a 'Not VF' class ECG signals. But in studies on real world dataset, a lot of fluctuations are observed. Thus, according to our hypothesis we resolve the Signal, IMF and Residue to frequency components using DFT, and then analyze their relations.

In our data analysis it was found that the frequency components lying between 1 - 5 Hz were more prominent while separating the two classes. Also, if we plot the frequency components of the two classes and observe them we get a somewhat intuitive idea of classifying the signals (Figure 3.5).

In Figure 3.5a, we observe the example of a 'Not VF' class. Here it is evident from the DFT coefficients, that the original signal is a wide-band signal, due to the presence of QRS complex. For the case of IMF, the DFT coefficients merely captures the distributed wideband components caused by QRS complex. On the other hand, the DFT coefficients of the R signal seems to represent a quite narrow-band signal and it captures the pattern of the signal. This satisfies our expectation.

On the other hand, in Figure 3.5b, i.e., for the example of 'VF' class, we can observe that the IMF frequency components, as we expected, almost completely match with that of the original signal. Additionally, a very tiny frequency components of R can be observed, which is in line with our analysis.

In both the examples we can observe that there are some additional rather unexpected, unusual frequency components. This fact becomes more apparent as we study more data. These frequency components disturb our expected waveshape, thus affecting the validation of our assumptions. So we need to determine which frequency components should be



(a) Not VF

(b) VF

Figure 3.5: Frequency domain observation of the two classes. In (a) for 'Not VF' the DFT coefficients represent the wide band nature of the signal due to the presence of QRS complex. Here, we can observe that the IMF component accounts for the fluctuations, instead of following the signal, also the DFT coefficients appear to be representing the wide band characteristics of the signal as well. On the contrary, the R component seems to follow the pattern of the signal. On the other hand, in (b) we can observe that for 'VF' class the DFT coefficients of the IMF component almost completely match with that of the signal whereas DFT coefficients of R are pretty tiny, thus they cover almost no information at all. In both the figures it can be seen that the DFT components between 1-5 Hz offer the most useful insights (here, the plots are cropped to 1-15 Hz range for visualization purposes.

considered. A machine learning algorithm can be used to interpret the significance of the individual frequency components and set weights accordingly to distinguish the two classes.

3.3.4 The Machine Learning Classifier

Feature Extraction

Instead of taking the DFT coefficients as features, we are actually interested in how much the individual DFT coefficients of the signal matches with the corresponding DFT coefficients of IMF or R. Thus, similar to cosine similarity we multiply the two terms and normalize the product with an appropriate value.

More precisely, for both IMF and R, we multiply each of the DFT coefficients with the corresponding DFT coefficient of the original signal. Then, we normalize them by the product of the magnitude of the DFT vector of the signal with the magnitude of the DFT vector of IMF and R respectively. This gives us the similarity IMF, R and the signal in the frequency domain, which we use as features for our machine learning algorithm.

$$IMF_{similarity}[i] = \frac{Signal_{DFT}[i] \bullet IMF_{DFT}[i]}{||\ Signal_{DFT}\ ||\ \ ||\ IMF_{DFT}\ ||}, \quad 1 \leq i \leq N \tag{3.9}$$

$$R_{similarity}[i] = \frac{Signal_{DFT}[i] \bullet R_{DFT}[i]}{||\ Signal_{DFT}\ ||\ \ ||\ R_{DFT}\ ||}, \quad 1 \leq i \leq N \tag{3.10}$$

Feature Selection using Random Forest

By far, we have considered all the frequency components as features. However, not all features may be equally useful for our prediction and some may even hamper our prediction. This motivates us to perform a feature ranking, i.e., analyzing the importance of an individual feature.

Random Forest [18, 45] is an ensemble learning algorithm that employs a collection of decision tree classifiers. However, along with solving the classification problem, a random forest can also be used to determine the importance of the features and thereby rank them accordingly. [30, 83].

Thus using the Random Forest algorithm, we identify the most promising subset of the features and use them for the final classification.

SVM Classifier

Support Vector Machines (SVM), proposed by Vapnik [16, 28] compute a hyperplane between the data points that separates them into two classes. Using a quadratic optimization, the hyperplane, $w^T x + b = 0$, is constructed to maximize the distance or margin between the hyperplane and the nearest points. SVM is inherently a linear classifier but with nonlinear mappings of the input space using an appropriate kernel, SVM can be employed for nonlinear classification purposes as well. After selecting the useful features, we use an SVM classifier to classify the ECG episodes to be either 'VF' or 'Not VF'. We use a Gaussian Radial Basis Function (RBF) as our kernel because it reliably finds the optimal classification solutions in most practical situations [58]. The value of a Radial Basis Function (RBF) is a function of distance from the origin ($\phi(x) = \phi(||x||)$), or some other predefined point ($\phi(x, c) = \phi(||x - c||)$). In particular, we used the Gaussian variant of RBF, which for two vectors x and x' is defined as follows:

$$K(x, x') = \exp(-\gamma \mid\mid x - x' \mid\mid^2) \tag{3.11}$$

where, γ is the inverse of the standard deviation of the gaussian distribution.

The entire workflow of VFPred prediction system is presented in the following flow diagram:

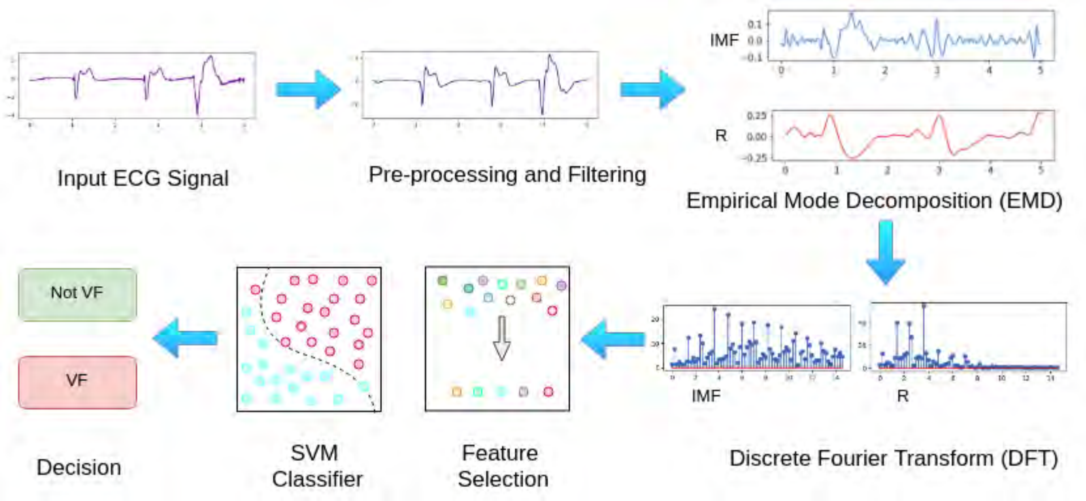

Input ECG Signal
Pre-processing and Filtering
IMF
R
Empirical Mode Decomposition (EMD)
IMF
R
Discrete Fourier Transform (DFT)
Feature Selection
SVM Classifier
Not VF
VF
Decision

Figure 3.6: Flow Diagram of VFPred Algorithm. The algorithm takes an ECG signal of length T_e sec, then performs some pre-processing and filtering in order to remove noise and artifacts from the signal. Next, the IMF and R component of the signal is analyzed using Empirical Mode Decomposition. After that, the

3.4 Implementation

We have implemented the VFPred algorithmic in Python programming language [111]. We used NumPy [114] for efficient numerical computation, SciPy [53] and PyEMD [67] for signal processing. We used SVM and Random Forest implementation from scikit-learn [91] library and used imbalanced-learn [70] for SMOTE . Moreover, we used matplotlib and seaborn [48] for visualization purposes. We also used the WFDB package [101] to fetch data from Physionet. All the codes are available in the following github repository:

`https://github.com/nibtehaz/VFPred`

The experiments were performed on a Dell-Inspiron 5548 Notebook (with a 5th generation Intel core-i5 CPU @2.2 GHz having 8 GB DDR3 RAM).

3.5 Experiments

In the following subsections, the experimental protocols used to tune the SVM classifier, in addition to studying and measuring the significance the features are described briefly.

3.5.1 Evaluation Metrics

Our problem reduces into a two class classification problem:

1. Positive Class : VF
2. Negative Class : Not VF

We use the following commonly used evaluation metrics:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.12}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.13}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.14}$$

However, since our dataset is hugely imbalanced, taking the Accuracy as the metric is not sufficient. This is because Accuracy understandably will follow the accuracy of the larger class (i.e., in our case specificity). So if an algorithm correctly identifies 'Not VF' but fails to detect 'VF', the Accuracy would still be high. This trend is disturbingly observed in many of the works in the literature, as most of the works prioritize the 'Not VF' class [2,4,5,7,8,10,104].

Thus we need an evaluation metric that accounts for the imbalance in the dataset. Geometric Mean Accuracy (G-Mean Accuracy) is one such metric [14]. G-Mean Accuracy is defined as follows:

$$G - Mean\ \ Accuracy = \sqrt{Sensitivity \times Specificity} \tag{3.15}$$

This metric will become high only when both the sensitivity and specificity are high. So giving priority to the majority class as seen in some previous works will result in poor performance.

3.5.2 SVM Parameter Tuning

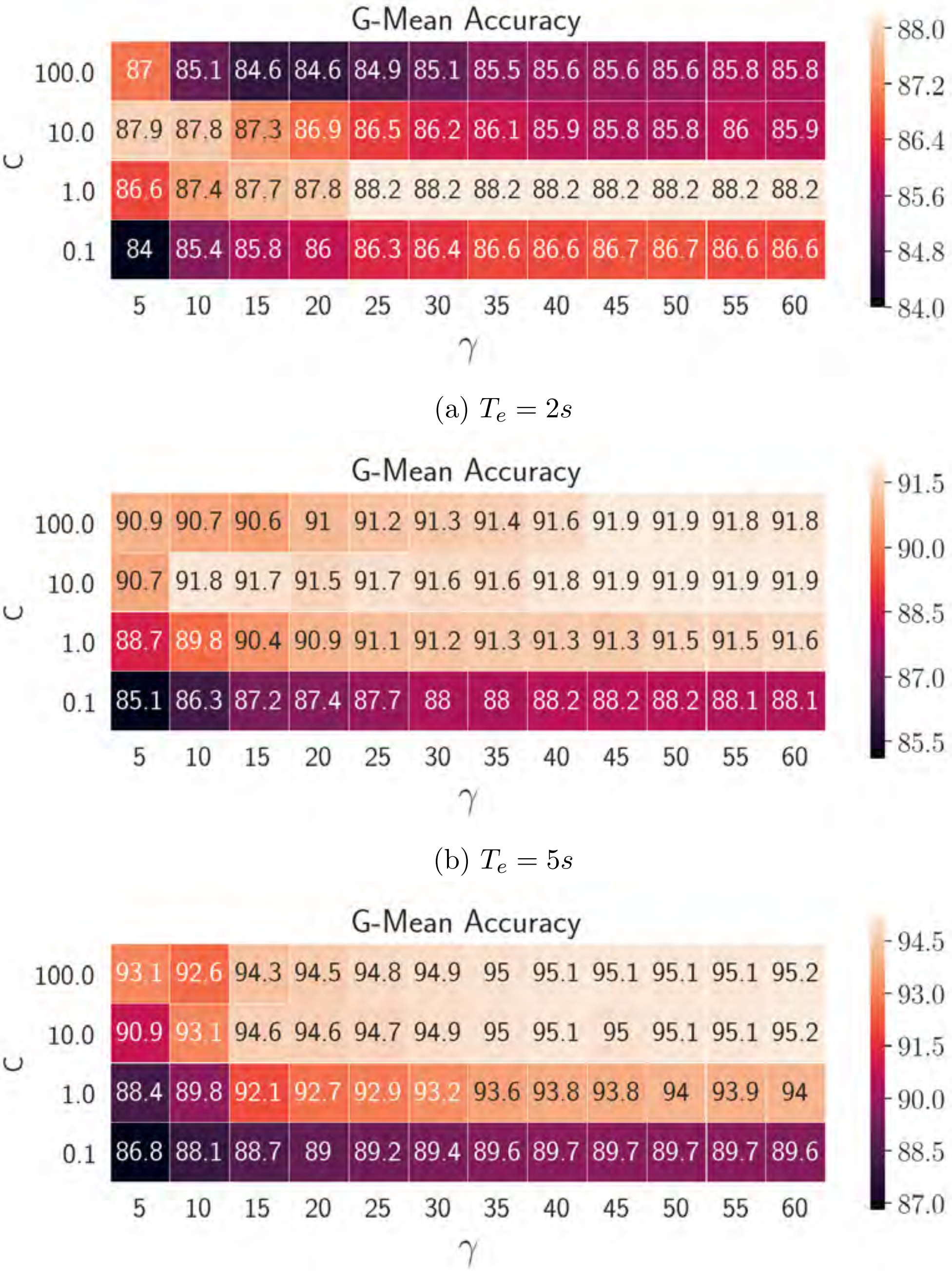
Performance of any machine learning algorithm depends highly on the proper selection of parameters. For our algorithm, we have used an SVM classifier with 'RBF' kernel. This SVM classifier has two parameters:

1. Regularization Constant, C: C is the parameter that defines whether the SVM classifier margin is a soft margin or a hard margin [28].
2. Kernel Hyperparameter, γ : γ is the parameter of Gaussian Radial Basis Function [28].

For tuning the parameters, initially, we took a small portion of the data as the training data and the rest as the validation data. Since the dataset is highly imbalanced, it was quite tricky to train our classifier. We randomly shuffled the entire dataset, and then took 3000 samples from 'VF' class and 5000 samples from 'Not VF' class as the training set. More samples were taken from the 'Not VF' class because it contains more variations as compared to the 'VF' class. The rest of the dataset was kept as the validation set (VF=2320, Not VF=46087).

To tune the parameters we performed an exhaustive grid search. We tried out 3 different episode lengths, $T_e = 2$ s, 5 s, and 8 s, as seen in literature. We trained the classifier using the training data and observed the performance on the validation data. The performance on the validation set is illustrated in Figure 3.7. Here we have shown the values of G-Mean Accuracy for different combinations of γ and C, as it nicely balances both Sensitivity and Specificity.

From the experiments we observe that as we increase the episode length, the performance improves. G-Mean Accuracy improves from 88.244% for $T_e = 2$s to 91.966% for $T_e = 5$s and to 95.101% for $T_e = 8$s. Thus, an increase of T_e by 3s roughly improves G-Mean accuracy by 4%. From a practical consideration, we should set the episode length, T_e such that the detection is both fast and accurate. However, there is a trade-off here: as



(a) $T_e = 2s$

(b) $T_e = 5s$

(c) $T_e = 8s$

Figure 3.7: SVM parameter Tuning. In order to select the best set of parameters, we performed experiments for different combinations of the parameters C and γ, and observed the G-Mean Accuracy. These experiments were performed taking only a small portion of the data as training data and the rest as validation data, and they were repeated for T_e = 2s, 5s, and 8s. Here it can be observed from three cases that as the episode length increased from 2s (3.7a) , to 5s (3.7b) , and even to 8s (3.7c) the performance improves. Thus the bigger windows are taken the better accuracy is obtained. For $T_e = 5s$ and $T_e = 8s$, the best results are obtained for $\gamma = 45$ and $C = 100$.

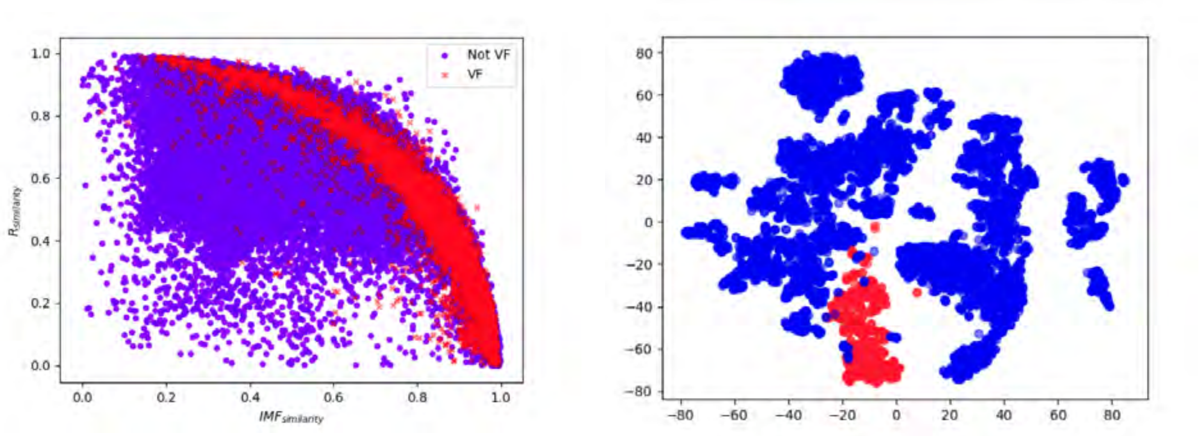
we increase the episode length T_e the detection accuracy improves but the computation becomes slower. So as a middle point we consider an episode length, T_e = 5s, for our further experiments.

Also, it can be observed that for $T_e = 2s$, $C = 1$ yields the best results, but for $T_e = 5s$ and $T_e = 8s$ the most promising results are obtained for $C \in [10, 100]$. On the other hand, the best performing values of γ are between 45 and 60. By performing some additional random experiments, it was seen that for $T_e = 5s$, the set of parameter values $C = 100$ and $\gamma = 45$ consistently outperformed all other combinations. Thus, we selected them as the parameters of our SVM model.

3.5.3 Evaluation of Feature Eminence

Existing works used the values of $IMF_{similiarity}$ and $R_{similiarity}$ directly as features [8]. However, when considering a large number of diverse data, relying on only two parameters may fail to capture sufficient resolution. Moreover, it is quite possible that there may exist some overlap between the two classes when considering only two features. Fig. 3.8a demonstrates a scatter plot of the feature points of the two classes, and it can be clearly observed that there exists a lot of overlap between the two classes. In fact, the samples of the VF class are completely surrounded by the samples of the Not VF class. Therefore, it would be almost impossible to distinguish the two classes when the features are almost identical. However, the original authors conveniently subsampled the dataset, which led to better accuracy values. But when considering the enitre dataset, such a method is very likely to fail.

To overcome these shortcomings, we have proposed features based on DFT on top of IMF and R similarities. Since our proposed feature set spans over a high-dimensional space it is difficult to visualize them. Therefore, we have made an attempt to perform a PCA (Principal Component Analysis) [119] to reduce the dimensions to two for the ease of better visualization. But, this did not work well, as it is quite challenging for PCA to achieve such drastic reduction in dimension. T-SNE algorithm [77] is quite promising in this aspect, but the problem is that it is computationally very expensive. Thus we were able to perform the dimensionality reduction on only a small subset (10%) of the data, as illustrated in Fig. 3.8b. As can be noticed from Fig. 3.8b, the overlaps of the two classes in the DFT version are much lesser, and when we are considering the complete feature vector it can be conjectured that such distinction will be even higher. Therefore, the proposed DFT based features are capable of better segragation of the two classes.



(a)

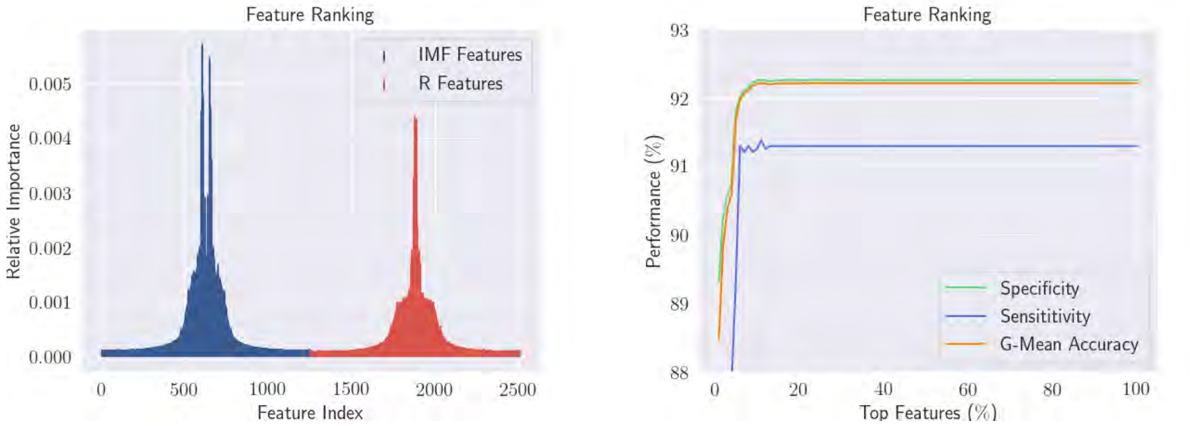
(b)

Figure 3.8: Evaluating the quality of extracted features. Fig. 3.8a presents a scatter Plot of the traditional features, namely, $IMF_{similiarity}$ and $R_{similiarity}$. The samples from VF and Not VF class are colored in red and blue respectively. From the figure, it is obvious that there does exist an alarming amount of overlapping between the two classes. On the contrary Fig. 3.8b demonstrates a two dimensional t-SNE plot of the proposed features based on DFT (of 10% random samples). It can be observed that the amount of overlapping has mitigated remarkably.

3.5.4 Feature Ranking by Random Forest

We used a random forest comprising 750 decision trees to rank our features. After computing the feature importance using the random forest algorithm, we normalized the values and plotted them in Figure 3.9a. Here the first half consists of the IMF features (colored in blue) and the last half corresponds to the R features (colored in red).

Now, from the plot, we can observe that most of the features do not contribute at all in making an accurate prediction. These features originate from the DFT coefficients and since DFT coefficients for a real signal is symmetric, so are their significances. Moreover,



(a) Relative Importance of the Features

(b) Effectiveness Different % of Features

Figure 3.9: Feature Ranking using a Random Forest. In Figure 3.9a we have plotted the relative importance of the features. The features coming from IMF and R are colored in blue and red respectively. From the figure it is evident that a great deal of features is actually quite trifling, thus omission of them should make the classification more efficient. Thus, in Figure 3.9b we have shown the outcome of taking only a percentage of the top features, in terms of Specificity, Sensitivity, and G-Mean Accuracy. The performance gradually improves as we increase the number of features, but after taking 16% of the best features, it reaches a plateau. However, it was seen that taking the top 24% features yields a slight improvement of 0.002%.

the importance of the leftmost and the rightmost features of both IMF and R are negligible, they actually correspond to the high frequency components. The most impactful features are the features in the middle, and the importance gradually decreases as we move away from the center.

After ranking the features, we conduct our experiments on a reduced number of features based on their relative importance. We again take 3000 random samples from ‘VF’ and 5000 random samples from ‘Not VF’ as the training set and treat the rest as the validation set, as we did to tune the parameters. The ECG episodes are taken to be 5 seconds long, and the SVM parameters were selected as mentioned in the previous section.

We had selected the percentage of features to be considered, sorted by their importance and observed the Sensitivity, Specificity, and G-Mean accuracy on the validation set (Figure 3.9b). From the experiments, it was seen that only about top 16% features were sufficient to sensibly distinguish between the two classes, and a slight improvement of 0.002% was observed if we had taken the top 24% features. Thus, we have selected the subset of only the top 24% features, as our final feature set.

This observation greatly reduces the dimensionality of the problem and makes both the training easier and the prediction faster. Also, this feature ranking exercise using Random Forest supports our hypothesis, that not all frequency components are needed for classification.

3.5.5 Statistical Significance Assessment of the Classifiers

Typically in machine learning based approaches, we select the best classifier based on metric scores like precision, recall or f1 score, as obtained from a cross validation process. However, based on the property of the data along with the splitting process, this may sometimes result in a false perception. Therefore, often the high performance of a classifier during validation, does not directly translate to a high performance in real world test scenario. This can be attributed to the fact that, the splitting of training and validation data acts favorably for the classifier, which also prevents the model from generalization. Thus, to overcome such shortcomings, statistical tests hold more potential in assessing the superiority of a model over others. Thomas Dietterich, in his highly influential study [31], recommended the McNemar's test [81] as the most suitable test with an acceptable Type I error, specially when the classifiers are complex and difficult to train. As a result, we train a number of different classifiers in addition to SVM, namely, Logistic Regression, Adaboost, Neural Network, KNN, Decision Tree and Random Forest. We observe both their GM Accuracy value and the corresponding χ^2 statistics from the McNemar's test. From our experiments it was observed that although models like Neural Network and KNN did come close to the SVM in terms of GM Accuracy, their χ^2 statistics corresponded to a p value of almost zero. This completely nullifies the null hypothesis, which states that the two classifiers should have the same error rate and the only deviations observed in experimental outcomes are by chance. Therefore, McNemar's test establishes that our SVM classifier indeed is significant statistically.

The summary of the statistical significance analysis of the classifiers is presented in

Table 3.2.

Table 3.2: Statistical Significance Assessment of the Classifiers. Here we present the GM Accuracy value obtained by the classifiers using the smaller set of training data. In addition we report the χ^2 statistics value and the corresponding p value. It is evident from the results that the improvement in using SVM classifier is statistically significant.

Method	GM Accuracy	χ^2 Statistics	p value
SVM	0.92	-	-
Logistic Regression	0.63	239.975	≈ 0.00
Adaboost	0.73	1155.278	≈ 0.00
Neural Network	0.87	86.629	≈ 0.00
KNN	0.8	74.630	≈ 0.00
Decision Tree	0.66	1934.581	≈ 0.00
Random Forest	0.62	269.069	≈ 0.00

3.5.6 Overcoming the Imbalance in the Dataset

As mentioned earlier, a major complication is that the dataset is highly imbalanced. This imbalance in the dataset prevents us from training our classifier properly. To overcome this limitation, we oversampled our data [44]. We avoided random oversampling as it merely replicates some data from the minority class and as a result, often tends to overfit [82]. Instead, we generated some synthetic data using Synthetic Minority Over-Sampling Technique (SMOTE) developed by Chawla et al. [22] which avoids over-fitting by distributing the probability over the neighborhood of the minority class points in lieu of imposing too much bias on the given minority class points. Now with balanced SMOTE'd dataset, we train our machine learning classifier

3.6 Results

In this section we present the results obtained from 10-Fold Cross-Validation. Furthermore, we compare our proposed method VFPred with contemporary algorithms. Furthermore, it should be noted that in this section we consider the entire dataset after upsampling using SMOTE.

3.6.1 K-Fold Cross-Validation

Cross-Validation is an evaluation test that determines how well a model can generalize on an independent dataset. This removes any human biases that may have been introduced and also accounts for the variance in the dataset. Experiments on real-world datasets show that the best cross validation scheme is a 10-fold cross-validation [61]. Thus we have performed a 10-fold cross validation to evaluate out algorithm. In the experimentation stages to reduce the time of computation we used a smaller set of data as the training data, but in this stage we use the entire dataset. Moreover, we make the data balanced by upsampling using SMOTE, and perform random shuffling. In addition, We also performed a stratified 10-fold cross validation which maintains a uniform distribution of the classes [61]. The results are summarized in Table 3.3.

Table 3.3: Experimental Results. Here we have shown both the results of the 10-Fold Cross Validation and Stratified 10-Fold Cross Validation tests. It should be noted that in this case we have used the entire dataset after upsampling using SMOTE which has eliminated the imbalance from the dataset.

10-Fold Cross Validation				
Data	Sensitivity (%)	Specificity (%)	Accuracy (%)	G-Mean Accuracy (%)
Training Data	100	100	100	100
Test Data	99.988 ± 0.016	98.401 ± 0.19	99.194 ± 0.092	99.191 ± 0.095
Stratified 10-Fold Cross Validation				
Data	Sensitivity (%)	Specificity (%)	Accuracy (%)	G-Mean Accuracy (%)
Training Data	100	100	100	100
Test Data	99.992 ± 0.01	98.395 ± 0.187	99.194 ± 0.092	99.190 ± 0.096

Firstly, from the results, it can be observed that our proposed algorithm accurately classifies the training data. Thus, it may appear that the algorithm is overfitting the training dataset. However, it is not the case since our algorithm performs remarkably on the test data as well. The obtained value of Specificity is slightly higher in the ordinary 10-fold cross validation than in the stratified 10-fold cross validation. The opposite scenario is seen for Sensitivity. However, in both the cases both Sensitivity and Specificity is equally high, and it is reflected by the high value of G-Mean Accuracy. All these points to the effectiveness of our feature engineering and learning algorithm optimization.

It appears that our model predicts a number of false positives. But from further analysis it turned out that most of these false positives actually contained a small segment of VF within them, becoming VF after 1-2 seconds. Thus, these false positives are actually not harmful rather beneficial if we are to develop a real time predictive system.

3.6.2 Comparison with Other Methods

Ventricular Fibrillation, being one of the most severe life threatening arrhythmias, is an exceedingly studied area. Research works have been undergoing in this area starting from the early '70s to even to date. In this section, we compare our work with some other well established works. During the study of other works, the following observations were apparent:

1. Some researchers made a pre-selection of ECG signals by hand [8,9,15,23,64,71,90, 109,125]. This resulted in better performance of their algorithms, but the accuracy drastically falls when tested on the entire dataset [6].

2. All the datasets we have are imbalanced. It was observed in some research works that priority was given to the majority class ('Not VF') resulting in a higher Specificity and lower Sensitivity, albeit with a high accuracy [2,4,5,7,8,10,104].

3. Surprisingly, a few authors tested their algorithms on the training set, resulting in a higher but clearly misleading accuracy [104].

Amann et al. [6] presented a comparison of ten algorithms for the detection of VF. It was shown that none of the algorithms, namely, TCI [109], ACF [23], VF Filter Method [64], Spectral Algorithm [15], Complexity Measure Algorithm [125], Li Algorithm [71], Tompkins Algorithm [90] etc., performed well over the entire dataset. The actual accuracy measures were very low compared to what was specified in the original papers since a pre-selection of the signals was done. Other algorithms in the literature that use only Signal Processing techniques also show a selectively high but overall poor performance. Arafat et al. [9] presented a method based on EMD and Bayes Decision Theory, which shows a sensitivity (Se) of 99.00% , specificity(Sp) of 99.88% and accuracy(Ac) of 99.78%, with only 'VF' and 'NSR' (Normal Sinus Rhythm) signals in the dataset. But the performance falls drastically when other beats and rhythms are considered. Anas et al. [8] presented another algorithm based on the EMD which obtains a sensitivity of 82.89%, specificity of 99.02% and accuracy of 98.62%, on the test set. The authors not only made a pre-selection of the signals but also gave priority to the Not VF class while selecting the threshold which results in a high accuracy but low sensitivity. Two other algorithms, Hilbert transform (HILB) [5] and Phase Space Reconstruction (PSR) [7] algorithms use the phase space of the ECG signal for VF detection. But they don't consider the shape of this signal. Thus, they fail to differentiate VT (Ventricular Tachycardia) from VF when other arrhythmias are present (HILB: Se = 79.73, Sp =98.83, Ac = 98.40; PSR: Se = 78.07, Sp= 99.01, Ac = 98.53).

Machine learning approaches show great promises as they improve the detection accuracy. Almost all the machine learning algorithms in the literature extract features from other existing VF detection algorithms based on signal processing, and then use them as features to detect VF. Atienza et al. [4] studied a number of ECG signal parameters and used them as features to train an SVM model (Se=74.1, Sp=94.7). In a later work [3], they considered ECG segments of 8s long to compute temporal and spectral parameters as features and then they developed a classifier using SVM. This obtained an accuracy of 91.1% while detecting VF. Qiao et al. [72] presented quite a similar pipeline; some parameters were computed as features and a genetic algorithm was used to rank the features. Then, weights were set to classes to consider the imbalance in the dataset and finally, an SVM model was trained (Se=96.2%, Sp=96.2%, Ac=96.3%). However, a drawback of the algorithm is that it considered Ventricular Fibrillation, Ventricular Tachycardia, and Ventricular Flutter all as the VF class. As these three classes are quite similar in appearance, an algorithm should be robust enough to distinguish among these three. Verma et al. [113] used similar approaches to extract features and subsequently used a Random Forest classifier to detect VF signals. This algorithm obtained Ac = 94.79, Se = 95.04, Sp = 94.78 for 5s long episodes and better results for 8s long episodes (Ac = 97.20, Se =

95.05, Sp = 97.02). Mi et al. [104] experimented with different dimensionality reduction and machine learning algorithms. Strangely, their models were tested on the training set, still, it obtained poor sensitivity (Se = 92.396, Sp = 99.121, Ac = 99.350). Asl et al. [10] did some preprocessing on the ECG signals before extracting the features and performing a dimensionality reduction using Gaussian Discriminant Analysis (GDA); finally, an SVM model was trained. This algorithm is by far the best performing one (Se = 95.77, Sp = 99.40, Ac = 99.16), but the drawback is that it works on the window length of 32 R-R interval, which is roughly 30 seconds. This makes the algorithm extremely slow and delays the prediction. Clayton et al. [27] used a (shallow) neural network to classify 'VF', which unfortunately obtained poor performance (Se = 86, Sp = 58). Acharya et al. [2] very recently used a Convolutional Neural Network (CNN) to detect VF along with some other arrhythmias. Their algorithm obtained a high specificity (98.19) but very low sensitivity (56.44).

In Table 3.4 we present a comparison among a number of methods. Here we are only showing the algorithms based on machine learning techniques as they outperform traditional signal processing based algorithms. Note that the results reported in Table 3.4 should be interpreted carefully as different works have reported their results using different validation techniques. As has already been mentioned, in our work, we employed 10-fold cross validation, and avoided independent testing as the idea of the latter has been doubted by some researchers (e.g., [25]). More specifially, as argued by Chou in [25], the way of independent test instance selection to test the predictor could be quite arbitrary resulting in arbitrary conclusions. A predictor achieving a higher success rate than the other predictor for a given independent testing dataset might fail to keep so when tested by another independent testing dataset. Accordingly, independent testing is not a fairly objective test method although it has often been used to demonstrate the practical application of a predictor in different domains. Unfortunately most of the works we are comparing with, employed independent testing but without any specific information about the test dataset; also they have not published the code of their implementation making it impossible to reproduce the test results and ensure a level playing field.

Table 3.4: Comparison among different methods. Results are taken from the respective papers. '-' indicates that the score was not available/reported in the respective paper. Here, we have only included the machine learning based works, as the outperforms the traditional approaches based solely on signal processing (as shown in Section 3.6.2.)

Algorithm	Sensitivity (%)	Specificity (%)	Accuracy (%)	G-Mean Accuracy (%)
Atienza et al. [4]	74.1	94.7	-	83.77
Atienza et al. [3]	-	-	91.1	-
Qiao et al. [72]	96.2	96.2	96.3	96.2
Verma et al. [113]	95.04 (5s)	94.78 (5s)	94.79 (5s)	94.91 (5s)
	95.05 (8s)	97.02 (8s)	97.20 (8s)	96.03 (8s)
Mi et al. [104]	92.396	99.121	99.350	95.70
Asl et al. [10]	95.77	99.40	99.16	97.57
Clayton et al. [27]	86	58	-	70.63
Acharya et al. [2]	56.44	98.19	97.88	74.44
VFPred	99.988 ± 0.016	98.401 ± 0.19	99.194 ± 0.092	99.191 ± 0.95

3.7 Discussion

Ventricular Fibrillation is a dangerous life threatening arrhythmia that can cause sudden cardiac arrest resulting in a sudden death. The inability of a doctor to continuously monitor the heart conditions of all the patients motivates us to design and develop efficient and accurate automated systems to perform the task. Immediately after Ventricular Fibrillation is detected a shock treatment can be given to the patient. A lot of research works have been done to detect Ventricular Fibrillation using either signal processing or machine learning techniques separately. In some works, the authors even proposed automated systems to give shock treatment to the patients. They argued that the system should have higher specificity, i.e., the accuracy of detecting patients not affected by Ventricular Fibrillation should be higher. This imposes priority on detecting 'Not VF' class correctly and in most cases the accuracy of detecting 'VF' was low. We find this contradictory, as the objective is to detect VF, not the other. Moreover, such a system would be vulnerable if it fails to detect VF consistently which may result in deaths of patients. We do not deny the importance of high specificity but at the same time, we believe that the sensitivity should be high as well. Note that, our goal is to aid the doctors and in no way to replace them. Hence, instead of developing automated shocking systems that fail in detecting ‘VF’ properly with the excuse of preventing the shock treatment of unaffected patients, we propose an automated monitoring system which will continuously monitor the patients and when the possibility of ‘VF’ arises, it will activate an alarm and an experienced doctor would examine the case and decide whether shock treatment is indeed required. Thus the sensitivity of the detection/prediction system should be high to avoid unfortunate deaths. We also gave importance to specificity, as our algorithm can classify both the classes with near equal performance.

In this work, we combined the strengths of both signal processing and machine learning. We analyzed the pattern of the ‘VF’ class ECG signals and proposed a novel feature engineering scheme. Next, we trained machine learning classifiers on the extracted features. As the dataset is highly imbalanced we adopted appropriate techniques to overcome the imbalance. Also, we computed feature importance and experimentally found that similar performance can be obtained by using a much smaller subset of the computed features, which can be very helpful for constructing a real time system as it would make our algorithm much faster. We have compared our algorithm with the state of the art and found out that our algorithm outperforms them. The only algorithm that comes close to our algorithm is presented by Asl et al [10]. However, the downside of their algorithm is that it works on windows of 32 R-R intervals (roughly 30 seconds), obtaining a sensitivity of 95.77%, specificity of 99.40%, accuracy of 99.16% and G-Mean Accuracy of 97.57%. On the other hand, our algorithm obtains a sensitivity of 99.99%, specificity of 98.40%, accuracy of 99.20% and G-Mean Accuracy of 99.19%, on a small 5 seconds long window. It was seen experimentally that our performance improves as we increase the window length.

In conclusion, this work has shown that the combination of signal processing based feature engineering and machine learning based decision making can greatly improve the performance of algorithms in biomedical signal processing domain. While existing algorithms emphasize either on signal processing or on machine learning, leaving the other one neglected, our objective was to make a fusion of these two approaches to take the best of both techniques. Thus we have developed a robust algorithm, VFPred, for the detection of Ventricular Fibrillation that is both fast and accurate. One of the most noteworthy features of VFPred is that it can classify both the classes equally accurately, even when using a short ECG signal. This is a significant improvement as existing works, while can identify the majority class ('Not VF') very well, fall much short in identifying the minority class ('VF'). Also, the success of VFPred with short ECG signals opens up the possibilities for developing more responsive, real time cardiac monitoring systems.

Chapter 4

Predicting Blood Pressure from PPG Signals

Cardiovascular diseases, being one of the most life-threatening diseases, take a toll of millions of deaths worldwide annually. Hypertension, which is one of the major reasons behind cardiovascular diseases, also provides us with a preventive measure. Thus, by analyzing the blood pressure continuously, we can predict hypertension and this early monitoring can save a lot of lives. In this chapter we present a method to infer blood pressure from PPG signals using Deep Learning. Development of such method holds a great promise as PPG being the cheapest biomedical sensor is present even in the least expensive smartwatches and fitness bands.

4.1 Introduction

Even in today's world of technological advances, cardiovascular diseases (CVD) are one of the most menacing causes of morbidity and mortality, crippling the ageing population [66]. More than 4 million people die of cardiovascular diseases every year only in Europe and when considering the whole world the number of deaths exceeds 17 million [110]. Hypertension is one of the leading reasons for cardiovascular diseases. Alarmingly in the year

2014, more than 1.4 billion people worldwide were somehow affected by hypertension [121], and the number is only expected to increase. Only in the USA, there are around 67 million patients of hypertension, which covers almost one-third of the population, and more alarmingly only less than half of them try to control their blood pressure [20]. Hypertension has thus been termed as 'Silent Killer' for dormant nature that eventually leads to untimely death [122]. Therefore, continuous blood pressure monitoring is essential. However, owing to the lack of experts physicians opposed to the ever-increasing number of patients, the development of automated monitoring methods is the only feasible mean to confront the crisis.

Several methods have been introduced, capable of measuring blood pressure (BP) accurately. However, this accuracy comes at the cost of invasiveness of such methods, which are often cumbersome to conduct. For catheter-based approach [51], not only the intervention of an expert is required, but also such procedures cause pain to already delicate patients. Clinics nowadays relies on cuff-based methods to measure blood pressure, but those as well are inconvenient and intrusive and not suitable for continuous blood pressure monitoring [84]. Therefore, for a significant amount of time, it has been the interest of researcher community to develop methods and apparatus to determine blood pressure from biomedical signals in a continuous, cuff-less, non-invasive manner [76, 97, 100].

Photoplethysmography (PPG) has gained a lot of popularity in recent times due to its widespread inclusion in smartwatches and fitness bands for its simplicity and cheapness. The idea behind PPG is uncomplicated, it works based on the illumination of skin and detection of the light absorption of skin. Hence, a PPG sensor generally comprises an led light source and a photodetector [19], the led emits light to the skin tissue and the photodetector keeps track of how much light is reflected i.e., the degree of absorption. It is empirically established that the amount of reflected light is proportional to the volume of blood flowing in the illuminated region [115]. Since the volume of blood is coupled to the speed of blood flow which harmonizes to the pressure exerted on the arteries, PPG signal has been being prominently used for the measurement of blood pressure [98]. Besides, PPG signals are also used for calculating absorption of Oxygen as well as the level of

Hemoglobin in blood [57] and for diagnosis of events like Hyperemia [96]. Despite having versatile applications, PPG signals also fall short in certain aspects, mostly because they get easily corrupted by movements [59].

In recent years there have been a plethora of academic studies to assess the state of blood pressure, using biomedical signals, mainly PPG often in conjunction with several other ones. The primary rationale behind measuring blood pressure from ppg is the association of the speed of blood flow and blood pressure. Overall, when blood vessels are contracted, blood flows rapidly, enforcing more pressure [103]. The opposite scenario is observed when the vessels are relaxed, as blood flows steadily and the pressure is diminished. Therefore, studies have been conducted investigating the rate of blood flow, which is popularly termed as Pulse Wave Velocity (PWV) [17]. Based on PWV, blood pulses require a time delay to reach the periphery of the body from the heart, which is denoted as Pulse Transit Time (PTT) [38]. Two other related terms Pulse Arrival Time (PAT) and Pre-Ejection Period (PEP) are also relevant in such analyses [98]. A lot of research work has been conducted to develop mathematical models of these various delays to infer blood pressure values [12, 39, 79, 93, 120].

While, prior works revolved around fitting the various delays corresponding to the rate of blood flow, of late several machine learning based approaches has been introduced [55, 56, 84, 86, 103]. The methods usually take the PPG signal (along with ECG signal in most cases), and predict the values of Diastolic Blood Pressure (DBP), Systolic Blood Pressure (SBP) and Mean Arterial Pressure (MAP). These methods generalize comparatively better, nevertheless, they are afflicted with a few limitations recurring throughout the literature. Firstly, most of these methods require ECG signals, which may be sometimes difficult to include in wearable cuff-less systems. Again, some of them rely on handcrafted features to predict the BP, but to compute such features the algorithms often demand the signal to be following ideal configuration.

This work presents PPG2ABP, a novel approach based on deep learning to predict the waveshape of the continuous Arterial Blood Pressure (ABP) signal from the Photoplethysmogram signal. The existing works limit themselves to inferring Systolic and Diastolic blood pressures. Though some efforts have been made to investigate the correlation between PPG and ABP signals [80], to the best of our knowledge no research work has been performed to predict the actual ABP signal from the PPG signal. In this regard, our study is a pioneering one, the first algorithmic pipeline that is capable of predicting the actual waveform of the ABP signal. Furthermore, being a deep learning based pipeline, PPG2ABP is free from the need for handcrafted features, therefore the requirement of signals maintaining a standard shape is redundant. Moreover, the different values of interest in the literature, i.e., SBP, DBP and MAP can be calculated from the predicted ABP waveform, and even in this criteria, our method outperforms the existing works, despite not being explicitly trained to do so.

The rest of this paper is organized as follows: Section **??** briefly explains the dataset we used as well as the proposed methodology, PPG2ABP. Section 4.4 presents the experimental setup and considerations. Section 4.5 elaborates the significant outcomes of PPG2ABP on the test data and draws a comparison with contemporary works. Finally, Section 4.6 concludes the paper.

4.2 Datasets

In order to develop and evaluate our proposed algorithm, the Physionet's MIMIC II dataset (Multi-parameter Intelligent Monitoring in Intensive Care) [42,95] was used. Fortunately, this database not only provides the PPG signals, but also it contains the simultaneous Arterial Blood Pressure (ABP) signal continuously, which is a crucial part of our algorithm. The sampling rate for both the signals are 125 Hz and they are recorded with 8-bit precision.

In this work, we actually utilized the data compiled from MIMIC II by Kachuee et al. [55, 56]. The primary reasoning behind this was that not only the data was presented in a convenient form to analyze, but also the raw signals are already pre-

Table 4.1: Statistics of the Dataset. Here we present the minimum, maximum and average values of DBP, MAP and SBP respectively. In addition we also list their standard deviation. It can be observed that SBP values have the highest variance, which makes their prediction the most difficult one, comparatively.

	Min	Max	Mean	Std
DBP	50	165.17	66.14	11.45
MAP	59.96	176.88	90.78	14.15
SBP	71.56	199.99	134.19	22.93

processed followed by their algorithm. Their compiled dataset is present in the UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/datasets/Cuff-Less+Blood+Pressure+Estimation`), from where it was downloaded on July 2019.

To ease their analyses Kachuee et al. ignored the signal episodes with too tiny or too large values of blood pressure values. They only considered signals with $60mmHg \leq DBP \leq 130mmHg$, $80mmHg \leq SBP \leq 180mmHg$. We, however, did not follow this scheme, firstly because we wished to test our algorithm on a broader range of signals and secondly it may be the case that in a real-world application scenario such small and high values of blood pressure are encountered. Therefore, we considered even signals with $DBP \approx 50mmHg$ and $SBP \approx 200mmHg$. The statistics of the dataset is presented in Table 4.1. It can be observed that SBP has a comparatively greater value of standard deviation, this extensive range causes difficulties when predicting the SBP values as hypothesized by Kachuee et al [56].

For our analysis, we considered signal episodes of 8.192 seconds long, i.e., we predicted 8.192 s long arterial blood pressure signal from ppg signals of 8.192 s long. Picking this episode length of 8.192 seconds allowed us to train sufficiently deep neural networks without being jeopardized by extensive computational complexity. However, in the downloaded database we have signals of a total duration of 741.53 hours. To mitigate the computational requirements, we hence undersampled the data. The following scheme was followed. First of all, the 8.192 seconds long signals were arranged in bins based on their SBP and DBP values. Next, from all the bins randomly 25% of the data was selected. However, if for some bins selecting one-fourth exceeds 2500 episodes, only 2500 episodes are included randomly. In this way, a total of 127260 random episodes are obtained, counting up to a duration of 353.5 Hours. From this 100000 samples at random are chosen to be the training data (roughly 78.58%) and the remaining 27260 samples are kept as the independent testing data. It should be noted that proper care and attention were given to make sure signal data from all the patients are included and also the training and test data remain disjoint. Furthermore, we did not omit ppg signals of sub-ideal quality, rather the random selection process led to the inclusion of a high number of merely acceptable (G2) and unfit (G3) signals [35].

4.3 PPG2ABP: Algorithmic Pipeline

We first present a brief description of the steps of PPG2ABP, followed by a detailed discussion of the steps. The proposed algorithm PPG2ABP takes a PPG signal of T_e seconds long, performs some minimal preprocessing operation to attenuate the irregularities and noises. Next, the filtered signal is processed using an Approximation Network, which approximates the ABP waveshape based on the input PPG signal. The preliminary rough estimate of ABP is further refined through a Refinement Network. Finally in addition to predicted ABP waveform, the values of SBP, DBP and MAP are computed straightforwardly. The overall pipeline of PPG2ABP is depicted in Fig. 4.1

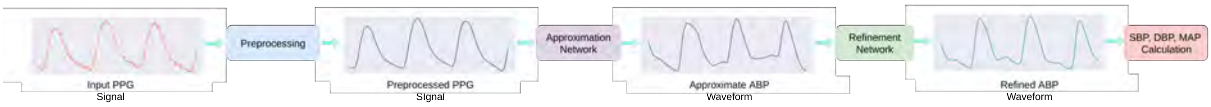


Figure 4.1: Algorithmic Pipeline of PPG2ABP. PPG2ABP takes a PPG signal of T_e seconds long as input and performs some preprocessing and filtering [55]. Next, the filtered signal is passed to the Approximation Network to approximate the ABP waveform. After that, the Refinement Network refines the overall waveform approximation. Finally, in addition to the ABP waveform, values like SBP, MAP and DBP can be computed.

Preprocessing

As mentioned earlier, we have used the signals already preprocessed by the method presented by Kachuee et al. [55]. Therefore, our preprocessing step is identical to theirs.

The preprocessing stage involves primarily wavelet denoising, for its desirable outcomes, such as superior phase response, computational efficiency, adaptiveness in terms of SNR etc. [56]. The wavelet transform is performed to 10 decomposition levels, with Daubechies 8 (db8) as the mother wavelet [102]. Then, the very low (0-0.25 Hz) and very high frequency (250-500 Hz) components are negated by setting the decomposition coefficients to zero. Next, wavelet denoising is performed with soft Rigrsure thresholding [32, 33]. Finally, the signal is retrieved by reconstructing the decomposition.

It should be noted at this stage that, to facilitate the training of the deep learning models, the PPG signals were mean normalized.

Approximation Network

The filtered signals are then processed through an Approximation Network, which approximates the ABP signal based on the input PPG signal. The Approximation Network is actually a one-dimensional deep supervised U-Net model.

U-Net [94] comprises a network built using only convolutional layers to perform the task of semantic segmentation. The network structure is built using a symmetric pair of Encoder Network and Decoder Network. The Encoder Network extracts spatial features from the input which are utilized by the Decoder Network to produce the segmentation map. The most innovative idea behind U-Net is the use of skip connections, which preserves the spatial feature maps, likely to have lost during pooling operation.

Though the original U-Net is designed to perform semantic segmentation on images, for our purpose we employ it to perform regression based on one-dimensional signals. Therefore, the two-dimensional convolution, pooling and upsampling operations are replaced by their one-dimensional counterparts. Moreover, all the convolutional layers other than the final one use $ReLU$ (Rectified Linear Unit) activation function [68] and are batch normalized [52]. To produce a regression output, the final convolutional layer uses a linear activation function.

Moreover, we use deep supervision in our U-Net network [69]. Deep supervision is a technique proven to reduce overall error by directing the learning process of the hidden layers. In our deeply supervised 1D U-Net, we compute an intermediate output which is a subsampled version of the actual output signal, prior to every upsampling operation. A loss function is computed with gradually declining weight factor as we move deeper into the model. This additional auxiliary losses drive the training of the hidden layers and makes the final output much superior.

Refinement Network

The outputs of the Approximation Network sometimes deviate much from the actual output. Therefore, we use an additional network to refined the output of the Approximation Network. We call this network Refinement Network, which is a one-dimensional MultiResUNet model.

The MultiResUNet model [49] is an improved version of the U-Net model. The primary distinction between the two is the inclusion of MultiRes blocks and Res paths. Multires blocks involve a compact form of multresolutional analysis, using factorized convolutions. On the other hand, Res paths impose additional convolutional operations along the shortcut connections to reduce the disparity between the feature maps of the corresponding levels of Encoder and Decoder networks.

Similar to the Approximation Network, this network also consists of one-dimensional versions of convolution, pooling and upsampling operations. The activation functions are identical as well, i.e. ReLU for all the layers but the final one, which uses a linear activation instead. The layers are also batch normalized but not deeply supervised.

SBP DBP calculation

After constructing the refined waveform of the blood pressure signal using the Refinement Network, the values of interest, namely SBP, DBP and MAP can be computed by taking the max, min and mean value of the signal respectively.

Mathematically,

$$SBP = max(ABP) \tag{4.1}$$

$$DBP = min(ABP) \tag{4.2}$$

$$MAP = mean(ABP) \tag{4.3}$$

Here, ABP is predicted blood pressure waveform from PPG2ABP.

4.4 Experiments

We have used Python (more specifically Python3) programming language [112] in order to perform the experiments. The neural network models have been developed using the Keras [24] library with Tensorflow backend [1]. Moreover, we have made the codes publicly available, which can be found in the following github repository:

`https://github.com/nibtehaz/PPG2ABP`

The experiments have been conducted in a desktop computer with intel core i7-7700 processor (3.6 GHz, 8 MB cache) CPU, 16 GB RAM, and NVIDIA TITAN XP (12 GB, 1582 MHz) GPU.

In the subsequent paragraphs we briefly present various experimental outcomes and insights. It should also be noted that the results presented in this section use only the training data after obtaining a validation set from random splitting. We have elaborately described how the algorithm is evaluated using the independent test data in the next section.

4.4.1 Selection of Models

In preliminary stages of our experiments, in addition to U-Net and MultiResUNet some other deep learning models like SegNet [11] and FCNN [75] were used. However, U-Net and MultiResUNet yielded comparatively better results. Moreover, some experiments were performed by using U-Net as the refinement network and MultiResUNet as the approximation network. Upon analyzing the results, it was observed that when U-Net was used as the refinement network, it failed to reach the performance level of MultiResUNet. On the contrary, when MultiResUNet was used as the approximation network it did perform better than the classical U-Net. Yet, when another MultiResUNet model was used as the refinement network the overall performance remained quite identical. We hypothesize that though MultiResUNet is superior to U-Net and manages to obtain a much better waveform, the refinement network reaches a plateau eventually. Nonetheless, since U-Net is computationally lighter than MultiResUNet, we use U-Net as the approximation network and MultiResUNet as the refinement network.

4.4.2 Selection of Loss Functions

Another potential concern is choosing the loss function. Typically Mean Squared Error (MSE) and Mean Absolute Error (MAE) are most prevalently used loss functions. For predicted values $\hat{Y} = [\hat{y_1}, \hat{y_2}, \hat{y_3}, \ldots, \hat{y_n}]$ and ground truth valyes $Y = [y_1, y_2, y3, \ldots, y_n]$, these two losses are defined as follows.

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y_i})^2}{n} \tag{4.4}$$

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y_i}|}{n} \tag{4.5}$$

In our experiments, we found out that using MAE as the loss function of the approximate network significantly improves the accuracy, whereas training the approximation network with MSE loss function falls much shorter. Upon inspecting samples and outputs we developed the following rationale. Since at the approximation network stage all we care about is getting a rough overall estimate of the waveform, it suffices to put equal weights to all the errors. But if we use MSE as the loss function, the error terms get squared and the bigger errors are more penalized. At this stage we have rather little information regarding the output waveform, therefore putting more emphasis on eliminating the bigger error terms actually vandalizes the overall outcome. As a result, it is more appropriate to use MAE as the loss function in the approximation network stage as it balances all the error terms ensuring a rough projection. On the contrary, in the refinement network, we already have an overall approximation of the waveform. Hence, it becomes beneficial to use MSE in that stage as the larger error terms are likely to be diminished better this way.

4.4.3 Effect of Number of Convolutional Filters

We have also explored the efficacy of using wider variants of both the networks, comprising an increased number of convolutional filters. For the U-Net model, we changed the number of convolutional filters from multiples of 32 to the same of 48, 64, 96 and 128. Similarly, For MultiResUNet we tuned the value of $alpha = [1.67, 2, 2.5, 3]$, which controls the number of filters used in the convolutional layers throughout the model. It was observed that models with a higher number of filters performed better, which is obvious owing to the fact that the inclusion of additional filters would allow the model to learn and capture additional shapes and features. However, as the number of filters increases, the models become computationally more expensive, and after a certain level, the improvement obtained from adding new filters is not worth the rising computational demand. Therefore, owing to this trade-off, we used an U-Net model with number of filters as multiples of 64, i.e., [64,128,256,512,1024] and for the MultiResUNet, we limited the value of α to 2.5.

4.4.4 Effect of Deep Supervision

Additionally, we have experimented with the concept of deep supervision and employing auxiliary losses to facilitate the training. For both the U-Net and MultiResUNet models we have imposed additional loss functions on the outputs of the convolutional layers just before the transposed convolution operations. Moreover, the weights of the losses were selected as $[1, 0.9, 0.8, 0.7, 0.6]$, i.e., the full weight was put on the actual output but was gradually diminished for that of the premature outputs. For U-Net models, a dramatic improvement was observed, but for MultiResUNet models, the improvement was not much significant. Therefore, to establish a trade-off between computational effort and accuracy, deep supervision was employed in the U-Net model but was rejected in the MultiResUNet model.

4.4.5 Training Methodology

As specified in Section 4.4.2, MAE and MSE are used as the loss functions of Approximate and Refinement networks respectively. In order to minimize these losses the Adam optimizer [60] is used, which adaptively computes different learning rates for individual parameters based on the estimates of first and second moments of the gradients. Adam has a number of parameters including β_1 and β_2, which control the decay of first and second moment respectively. However, in our experiments, we have used Adam with the parameters mentioned in the original paper. The models have been trained for 100 epochs using Adam optimizer. The reason for selecting 100 as the number of epochs is due to the fact that after 100 epochs no further improvement was noticed in either of the networks.

4.4.6 K-Fold Cross Validation

Cross-Validation tests tend to approximate the performance of an algorithm on an independent dataset, ensuring a balance between bias and variance. In a k-Fold cross-validation test, the dataset D is randomly split into k mutually exclusive subsets $D_1, D_2, \cdots, D_k$

of equal or near-equal size [62]. The algorithm is then run k times subsequently, each time taking one of the k splits as the validation set and the rest as the training set.

We have performed a 10-fold cross validation using the training data, i.e., 90 % of the training data was used to train the model and the remaining 10 % data was used as validation data. This approach was repeated 10 times using different data splits, and thus 10 models were developed. The best performing model was selected and was ultimately evaluated using the independent test data.

4.5 Results

In the following subsections we present the evaluation of PPG2ABP using different metrics. We compute the error in prediction of the ABP waveform. Furthermore, we analyze the results when noisy signals are involved and score PPG2ABP under BHS and AAMI Standards. Moreover, we observe that the SBP, DBP values, predicted from PPG2ABP, can be used to classify states of hypertension. Finally, we establish the statistical significance of PPG2ABP, in addition to comparing it with contemporary works.

4.5.1 Predicting ABP Waveform

The primary and unique objective of this work was to transform ppg signals to the corresponding blood pressure waveform. Despite some correlation between the two established from previous studies, they are quite different from each other when considering the waveforms. Nevertheless, the proposed PPG2ABP model manages to predict the waveform of blood pressure taking only the ppg signal as input. The output of the approximate network gives an overall rough estimate, which is further refined by the refinement model. Fig. 4.2 illustrates such an example. It can be seen that the predicted waveform closely follows the ground truth waveform of blood pressure.

Therefore, from experimental results it is evident that PPG2ABP can translate ppg signals to corresponding blood pressure signals, preserving the shape, magnitude and

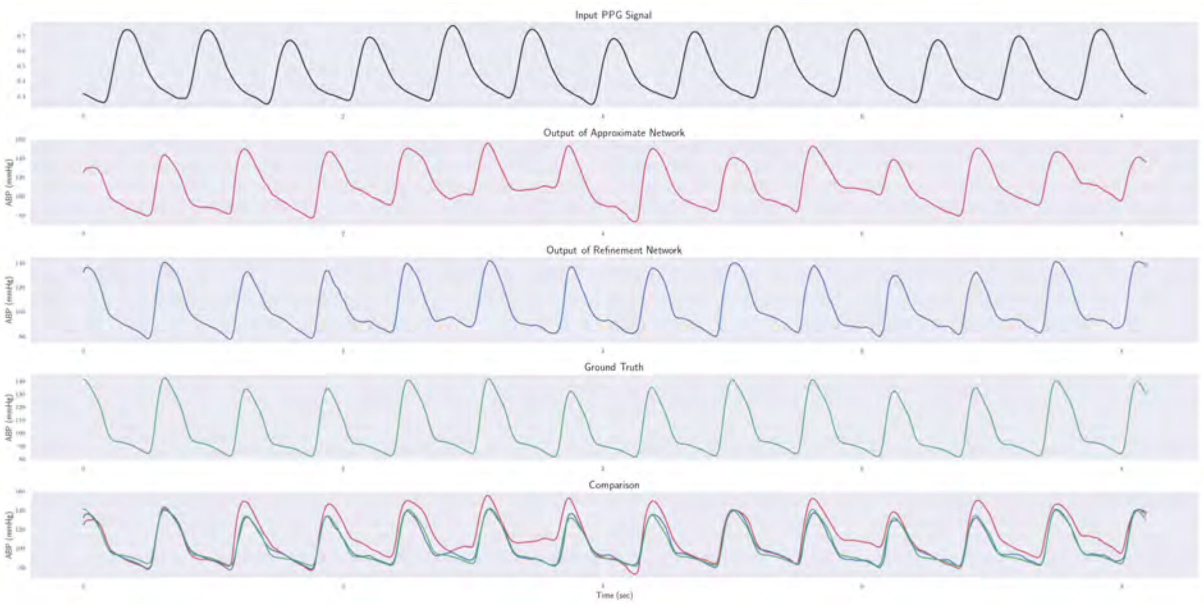
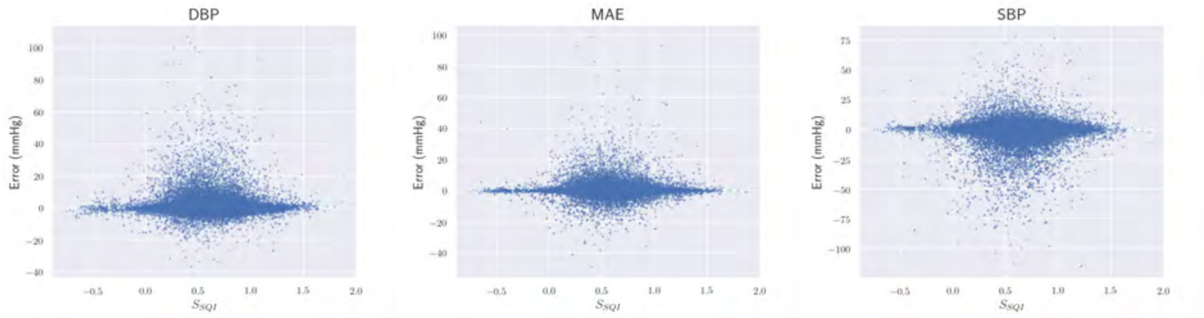


Figure 4.2: Demonstration of the output of the PPG2ABP pipeline, when a sample PPG signal from the test data is given as input. From the figure it can be observed that the output from the Approximate Network, despite roughly following the overall pattern of the ground truth, falls short in certain aspects. The shortcomings are vividly apparent around the peaks. Furthermore the prediction fails to rapidly slope down from the peak regions. However, the prediction from the Refined Network seems to be more satisfactory. It can be observed that in addition to following the overall pattern of the ground truth waveform, the final predicted waveform also successfully mimics the peak regions and subsequent downward inclination. Therefore, the inclusion of the Refinement Network on top of the Approximate Network significantly improves the results, as evident from the drop of mean reconstruction error from 9.52 mmHg to 2.37 mmHg, for this particular example.

phase in unison. Quantitatively, the mean absolute error of this blood pressure waveform construction is 4.604 ± 5.043 mmHg over the entire test dataset. In addition the mean absolute error of DBP, MAP and SBP prediction is 3.449 ± 6.147 mmHg, 2.310 ± 4.437 mmHg, 5.727 ± 9.162 mmHg respectively. Furthermore, previous studies have pointed out that there exists a phase lag between the ppg and bp signals of MIMIC database [123] and some further processing is required to align them. However, in our predicted output, we can observe that our deep learning based model has been able to overcome this issue of phase lag. This is another remarkable point to consider as due to signal acquisition protocols there may exist a phase lag between the two signals in real-world applications as well.

4.5.2 Inappropriate Signals

As mentioned earlier, PPG signals get easily corrupted by different types of artefacts. Unfortunately, cleansing PPG signals of these anomalies is no trivial task [74]. Therefore, often a tendency is observed to ignore the noisy PPG signals, as they also hinders the computation of handcrafted features. Assessment of quality of PPG signals is also challenging albeit having multiple metrics, due to their inconsistent behaviour [35]. From the experimental study, it has been established that Skewness based quality index S_{SQI} is the most effective one [63]. Therefore, we plot the errors in predicting DBP, MAP and SBP against S_{SQI} (Fig. 4.3). From the plot, it appears that for signals with low S_{SQI}, the error is smaller. However, this is due to the fact that very few signals were in that unfit region and somehow the model managed to learn their patterns with comparative ease. The interesting region is in the middle where the error is the highest. This is though the Acceptable region, the variations made the prediction most difficult. On the contrary, for the Excellent region, i.e. with the highest values of S_{SQI}, minimal errors were encountered. Nevertheless, the plots may deceive us as the were some really corrupted signals with a questionably high value of S_{SQI} , the opposite being also true. Furthermore, the outlier signals in each of the range imposed most difficulties.



(a) (b) (c)

Figure 4.3: Evaluation of PPG2ABP in perspective of the presence of inappropriate signals. For the comparative eminence of skewness in assessing PPG signal quality we have used S_{SQI} as the grade of PPG signals. It can be observed that as S_{SQI} increases the overall error of predicting DBP, SBP along with MAE diminishes. Also it should be noted that there were only a few of PPG signals with extremely low S_{SQI} which was learnt well by the model. Besides, even some good quality PPG signals yielded a low S_{SQI} score.

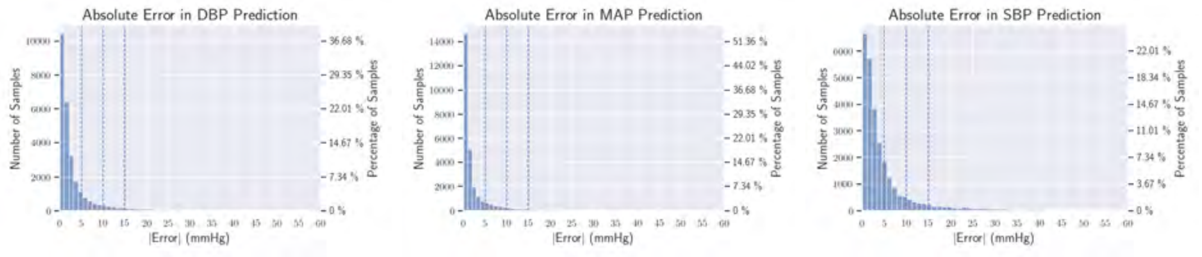
4.5.3 BHS Standard

The British Hypertension Society (BHS) has introduced a structured protocol to assess blood pressure measuring devices and methods [89]. Hence, this standard has been frequently used in the literature as a metric [55,56,84,86]. The accuracy criteria of the BHS standard appraise methods based on the absolute error. More specifically, the grades are provided by counting what percentage of the predictions on the test samples fall under 5 mmHg, 10 mmHg and 15 mmHg absolute error respectively. The three grades are presented in Table 4.2, for an algorithm to obtain a certain grade, it must satisfy all the three thresholds simultaneously. In addition, there is grade D for algorithms failing to meet the requirements of grade C [89].

The absolute error of computing DBP, MAP and SBP on the test data by PPG2ABP is presented in Fig. 4.4. It can be observed that for both DBP and MAP most of the predictions are covered by the 15 mmHg error threshold, a significant part of which actually fall under 5 mmHg error surprisingly. For these two we obtain a grade A score under BHS standard. On the contrary, for SBP though it is apparent from Fig. 4.4 that quite a number of test predictions exceed the 15 mmHg error threshold, still it is good enough to achieve the grade B score. It should be noted that to the best of our knowledge no other algorithms obtained grade B in SBP prediction on MIMIC II dataset (more details on Section 4.5.7). The detailed results are presented in Table 4.2, it can be observed that PPG2ABP outperforms the requirements of 5 and 10 mmHg thresholds by an impressively high margin.

Table 4.2: Evaluation of BHS Standard. Here we present the criteria used in grading the rank of predictions using BHS Standard. We also demonstrate how our results compares with the BHS Standard.

		Cumulative Error Percentage		
		$\leq$ 5 mmHg	$\leq$ 10 mmHg	$\leq$ 15 mmHg
Our Results	DBP	82.836 %	92.157 %	95.734 %
	MAP	87.381 %	95.169 %	97.733 %
	SBP	70.814 %	85.301 %	90.921 %
BHS	grade A	60 %	85 %	95 %
	grade B	50 %	75 %	90 %
	grade C	40 %	65 %	85 %



(a) (b) (c)

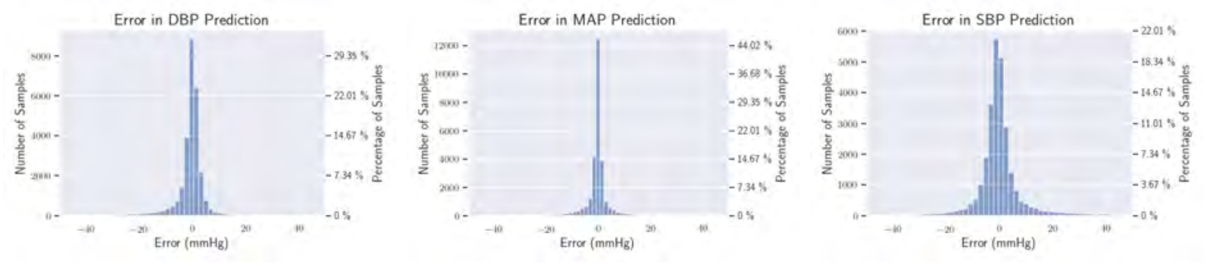
Figure 4.4: Mean Absolute Error Histogram. Here we present how the mean absolute error of predicting DBP, SBP and MAP of the samples are distributed. In addition, we also observe errors of how many samples lie below the 5 mmHg, 10 mmHg and 15 mmHg thresholds, used in the evaluation of BHS Standard.

Table 4.3: Evaluation of AAMI Standard. Here we present the criterion used in grading the rank of predictions using AAMI Standard. We also demonstrate how our results compares with the AAMI Standard.

		ME (mmHg)	STD (mmHg)	Number of Subjects
Our Results	DBP	1.619	6.859	942
	MAP	0.631	4.962	942
	SBP	-1.582	10.688	942
AAMI Standard		≤ 5	≤ 8	≥ 85

4.5.4 AAMI Standard

Similar to the BHS Standard, AAMI Standard is another metric to evaluate blood pressure measuring devices and methods. The criterion set by AAMI standard [36] requires the blood pressure measuring methods to have a mean error and standard deviation of less than 5 mmHg and 8 mmHg respectively. Table 4.3 shows our results under the AAMI criterion. It can be observed that for both DBP and MAP the requirements of AAMI standard are satisfied, they are met with a big margin to be more precise. However, for SBP, although the condition of mean error is fulfilled, the value of standard deviation is a bit higher than the ideal. It may be noted here that other contemporary methods fail to satisfy the AAMI criterion for SBP on the MIMIC dataset as well. The histograms of error for prediction of DBP, MAP and SBP is presented in Fig. 4.5. From the figures, it is again evident that though for DBP and MAP the spread of error is very narrow, it is comparatively outspread for SBP.



(a) (b) (c)

Figure 4.5: Mean Error Histogram. Here we present how the mean absolute error of predicting DBP, SBP and MAP of the samples are distributed. All these errors seem to have a mean of zero and a small value of standard deviation other than for SBP.

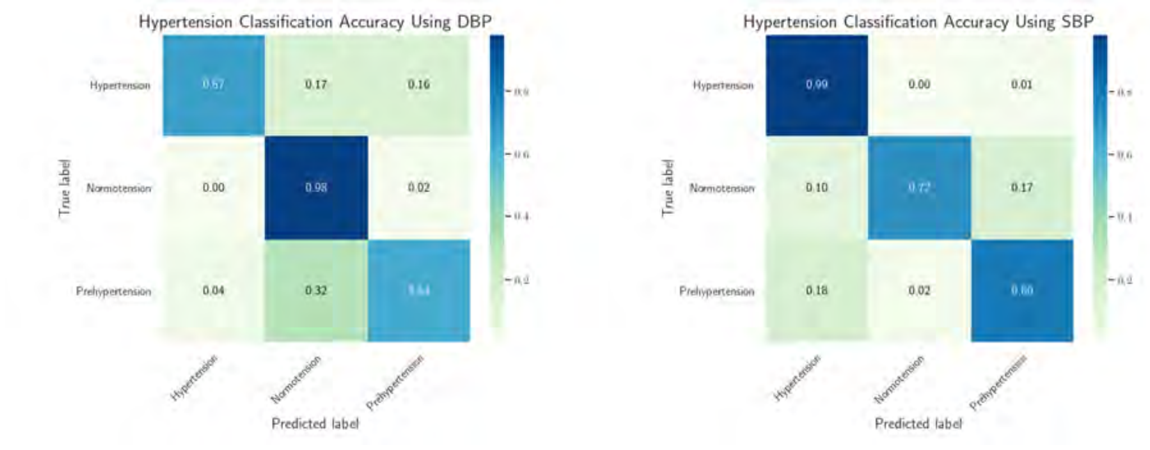
Table 4.4: Hypertension Classification performance using the predicted values of SBP and DBP. It can be observed that DBP values are more potent in determining Normotension, whereas, SBP values show greater promise in identifying Hypertension.

	DBP				SBP			
Class	Range	Precision	Recall	F1-Score	Range	Precision	Recall	F1-Score
Normotension	DBP $\leq$ 80	91.35 %	98.16 %	94.63 %	SBP $\leq$ 120	92.37 %	72.46 %	81.22 %
Prehypertension	80 < DBP $\leq$ 90	76.25 %	63.97 %	69.57 %	120 < SBP $\leq$ 140	86.66 %	79.90 %	83.14 %
Hypertension	90 < DBP	91.66 %	67.46 %	77.77 %	140 < SBP	94.68 %	98.53 %	96.56 %

4.5.5 BP Classification Accuracy

From an application perspective, it is actually more beneficial to be able to classify the state of hypertension of a patient, instead of the exact values of SBP, DBP or MAP. This classification can be done from the values of SBP and DBP in a straight-forward manner [46]. Table 4.4 denotes the ranges for the three most common classes, namely Normotension, Prehypertension and Hypertension. In addition, the performance of classifying the blood pressure states based on SBP or DBP values alone are also listed. It can be observed from the table that SBP values provide the most reliable means to determine Hypertension, i.e. an F1-score of 96.56%, along with nearly balanced Precision and Recall. SBP also manages to diagnose Prehypertension state with an F1-Score or 83.14%. On the contrary, it can be noted that DBP can be used to classify Normotension state with comparatively better accuracy, having an overall F1-Score of 94.63%. DBP, however, scores poorly in determining the other two classes.

Therefore, it'd suitable to classify Hypertension and Prehypertension states using SBP values and the Normotension condition with DBP values. The confusion matrices presented in Fig. 4.6 also strengthens these analyses.



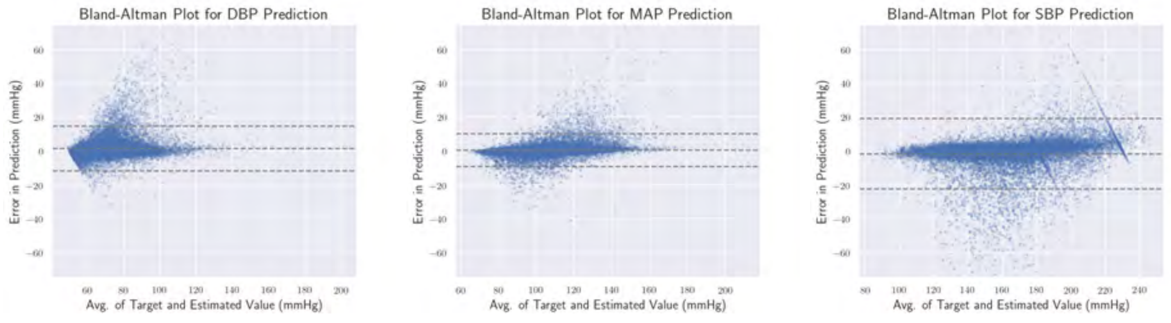
(a) (b)

Figure 4.6: Confusion Matrix of Hypertension Classification using DBP and SBP values. It can be observed that DBP values are more potent in determining Normotension, whereas, SBP values show greater promise in identifying Hypertension.

4.5.6 Statistical Analysis

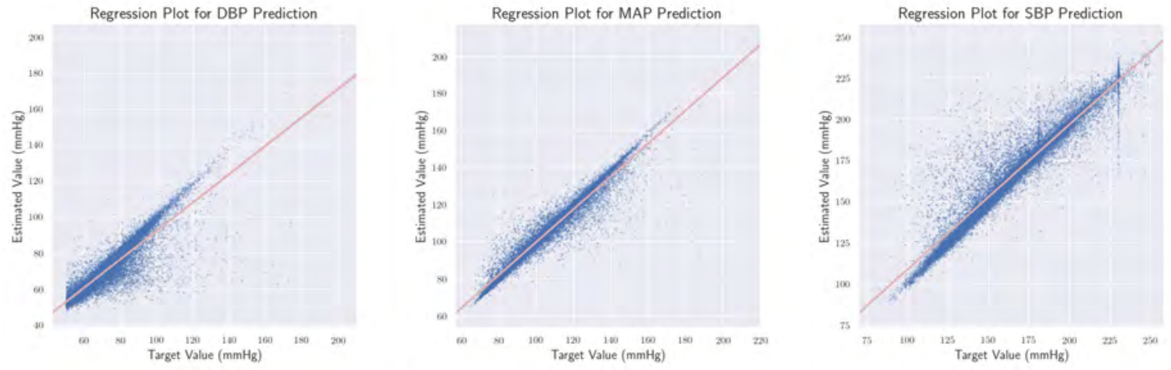
Fig. 4.7 represents the Bland-Altman plots [40] for predicting DBP, MAP and SBP respectively. The 95% limits of agreement span the segment from $\mu - 1.96\sigma$ to $\mu + 1.96\sigma$ (shown using dashed lines), where μ and σ are the mean and standard deviation of the distribution respectively. For DBP, MAP and SBP this limit translates to $[-11.825 : 15.0637]$, $[-9.095 : 10.357]$ and $[-22.531 : 19.367]$ mmHg respectively. Though these numbers may appear to be overwhelming, actually if we observe the plots from Fig. 4.7 it can be seen that most of the error terms fall below 5 mmHg range. It is nevertheless true that all the three plots contain a great chunk of outliers, most specifically the SBP one (Fig. 4.7c).

In addition, Fig. 4.8 depicts the regression plots of predicting DBP, MAP and SBP respectively. From the plots alone, it is evident how much the predictions are correlated with the ground truth values. Moreover, the values of Pearson Correlation Coefficient for DBP, MAP and SBP predictions are 0.8941, 0.9656 and 0.9360 respectively, indicating the strong positive correlation further. Furthermore, such high values of Pearson's coefficient on a sample size of 27260 corresponds to a p value of $p < .000001$. Such a low value of p nullifies the null hypothesis completely, indicating the statistical significance of our results.



(a) (b) (c)

Figure 4.7: Bland-Altman Plot. Here through the Bland-Altman plots it is evident that the error of predicting DBP, MAP and SBP of 95% of the samples lie between [-11.825:15.0637], [-9.095:10.357] and [-22.53119.367] respectively.



(a) (b) (c)

Figure 4.8: Regression Plot for Prediction of DBP, MAP and SBP. In all the three cases we obtain a p value in the range of $p < .000001$, which nullifies the null hypothesis and strenghtens the statistical significance of our method.

Similarly, to assess the overall statistical significance of PPG2ABP, we perform a paired Student's t-test. We compare the PPG2ABP pipeline with an 1D U-Net as baseline, for the lack of implementations of other works. The reasoning behind using paired Student's t-test over Wilcoxon signed-rank test is that, the difference between the errors of the models are normally distributed as shown in Fig. 4.9. From the test we obtain a t-statistics of 70.117 and the corresponding p value is almost zero. This again nullifies the null hypothesis, stating that the two models should perform similarly, and any improvement from using PPG2ABP is merely by chance. Hence, paired t-test establishes the statistical significance of PPG2ABP. Although the superiority of PPG2ABP is evident from both the higher performance metric values and statistical significance, it should also be noted that the 1D MultiResUNet model used in PPG2ABP is a bit more expensive computationally,

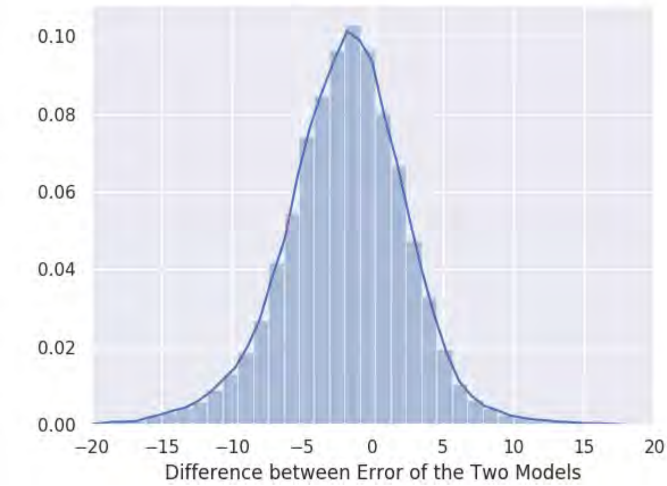


Figure 4.9: Difference between Error of the Two Methods. Here we compare the error of PPG2ABP with a 1D U-Net as baseline. It can be observed that the difference between the errors are normally distributed, which proposes that a paired student's t-test will be the suitable statistical test.

compared to the baseline 1D U-Net. Therefore, this improvement of performance does come at a cost of computation time, which can be considered as a trade-off.

4.5.7 Comparison with Other Methods

Despite there being a lot of research endeavours on this topic, unfortunately, we cannot compare most of the works directly. This is mostly due to the fact that most of the works use a proprietary dataset of their own, which is not shared publicly due to privacy reasons [103]. The handful number of methods reported on a standard public dataset like MIMIC II, also use different numbers of patients which again makes it difficult to ensure a level playing field. Slapnivcar demonstrated that while methods such as that of Kachuee et al. [56] struggles for working with a vast repertoire of patients, works like Xing et al. [123] face less impediments for using a smaller subset of patients. Therefore, presenting a comparison among different methods gets unnecessarily complicated. Nevertheless, we have compiled a list of works evaluated on MIMIC II dataset with a comparable and sufficiently large number of patients.

The comparison is presented in Table 4.5. The results are collected from the respective papers. However, for the work of Mousavi et al. [86] they neglected 1% error of SBP which achieved a Grade of C under BHS standard, but we consider the error which drops it down to Grade D. It can be observed that only PPG2ABP manages to obtain Grade B for SBP prediction under the BHS standard, and also consistently outperforms all the other methods. This is indeed remarkable as PPG2ABP was not explicitly trained at all to predict accuracy SBP, DBP or MAP.

Table 4.5: Comparison among different approaches. Here we list the methods that used the MIMIC II dataset to evaluate their performance. Furthermore, for a fairer comparison we have only included the methods that considers a significant portion of the dataset. We compare the methods using measures like Mean Absolute Error (MAE) of predicting DBP, MAP, SBP, in addition to BHS and AAMI Standard.

Study	Dataset	Input	Results

Kachuee et al. [55]	MIMIC II (942 subjects)	PPG and ECG	BHS Standard:
			DBP = Grade B, MAP = Grade C, SBP = Grade D
			MAE:
			DBP = 6.34 mmHg, MAP = 7.52 mmHg, SBP = 12.38 mmHg

Kachuee et al. [56]	MIMIC II (942 subjects)	PPG and ECG	BHS Standard: DBP = Grade B, MAP = Grade C, SBP = Grade D AAMI Standard met for : DBP, MAP MAE: DBP = 5.35 mmHg, MAP = 5.92 mmHg, SBP = 11.17 mmHg
Mousavi et al. [86]	MIMIC II (441 subjects)	PPG	BHS Standard: DBP = Grade A, MAP = Grade B, SBP = Grade D AAMI Standard met for : DBP, MAP
Slapnivcar et al. [103]	MIMIC II (510 subjects)	PPG	MAE: DBP = 9.43 mmHg, SBP = 6.88 mmHg
PPG2ABP	MIMIC II (942 subjects)	PPG	BHS Standard: DBP = Grade A, MAP = **Grade A**, SBP = **Grade B** AAMI Standard met for : DBP, MAP MAE: DBP = **3.449 mmHg**, MAP = **2.310 mmHg**, SBP = **5.727 mmHg**

4.6 Discussion

In conclusion, in this work, we have made an effort to infer the complete waveform of blood pressure signals using ppg signals alone. There are indeed a number of works where only the various information of interest like DBP, SBP and MAP are computed. However, these studies have been constrained as none of them managed to represent the overall picture of the blood pressure. Moreover, often they required additional signals like ECG. Furthermore, several of the algorithms actually computes some handcrafted features from the signals, which limits the algorithms to only work on perfectly shaped signals, free from noises and artefacts.

Therefore, our work solves the two major issues, namely, restriction of using ideal signals only and involvement additional signals. This has been achieved by employing deep learning. Since deep learning models upon analysis of the data, computes high-level abstract features adaptively from the data, this alleviates the need of computation of manual handcrafted features which may impose additional criteria. In addition, since the very beginning, we have been motivated to develop methods based on ppg signal alone. These considerations paved the way to PPG2ABP, a deep learning based method to predict the continuous arterial blood pressure waveform.

The resulting waveforms generated from PPG2ABP corresponds to the actual waveform of the blood pressure signal, retaining the shape, amplitude and even phase. However, the success of PPG2ABP extends beyond that. As it was mentioned earlier, there a plethora of methods to compute measures like SBP, DBP and MAP. It was found out surprisingly that we can calculate these values using pressure waveform predicted by PPG2ABP with outstanding accuracy. It is certainly remarkable that the accuracy of predicting these values outperforms the existing methods, tailored towards predicting them in the first place. These methods are mostly supervised, trained explicitly to infer these values, on the other hand, our proposed PPG2ABP even without any implicit such training, predicts those values better. For DBP and MAP prediction we have achieved Grade A under BHS standard test and also satisfied the criterion of AAMI standard. Though for SBP the results are not so phenomenal, it is still considerably superior. For SBP we have achieved Grade B on BHS standard test, and to the best of our knowledge, no other algorithms managed to score this using MIMIC dataset. Moreover, PPG2ABP is resilient to noises and imperfections. Also, PPG2ABP can also be utilized as hypertension classifier algorithm, with astounding accuracy. Furthermore, the results of PPG2ABP are statistically significant.

The application of PPG2ABP can be manifold. In addition to predicting the typical systolic and diastolic blood pressure values, the complete profile of the blood pressure can be achieved this way. This will allow the doctors to monitor the blood pressure of their patients continuously. Moreover, the trend and pattern of blood pressure can be mapped to the user behaviour and activities, which may lead to insightful findings. Since nowadays all the smartwatches and fitness bands comprise a ppg sensor, applications based on PPG2ABP can be easily deployed to the mass market. This is due to the wonderful fact that PPG2ABP is free from the need for additional, expensive sensors like ECG. Therefore, to make the results of this research accessible to the general public, the codes have been made open sources and can be found in `https://github.com/nibtehaz/PPG2ABP`. The codes are released under the MIT License, which makes it possible to develop server-side or smartphone/smartwatch applications on top of this.

The future direction of this research may also be thought from a number of different perspectives. Firstly, we have used two networks one to approximate and the other to refined the predictions. We are interested to design and develop more optimized deep learning models that we will be able to replace the two with only one model instead. This is will make the ABP prediction from ppg an end-to-end process. Furthermore, it will be interesting and beneficial to develop applications for wearables using PPG2ABP and conduct clinical studies thereby. Moreover, many of the existing methods have shown the effectiveness of utilizing a personalized calibration, which we too wish to explore in our future works.

Chapter 5

Conclusion

In this thesis, various biomedical signals and there diverse applications have been explored. Notably, for the widespread availability in smartwatches, the ECG and PPG signals in particular, have drawn most of our attention. Inclusion of these two, basically extends all our smartwatches to health monitoring devices. Therefore, they possess tremendous potential in continuous, automated diagnosis.

ECG signals have been being used for quite a long time for the detection of various heart arrhythmias by the doctors. Therefore, taking motivations from the medical practice, a plethora of algorithms have been developed to diagnose arrhythmias like Ventricular Fibrillation, Ventricular Tachycardia, Atrial Fibrillation, Atrial flutter etc. While analyzing such methods, we mostly notice two different types of approaches, that differ in the underlying methods/tools used. The former solely focuses on Signal Processing techniques, while the latter utilizes Machine Learning. Actually, the algorithms from the first class are more medically inspired as they diligently audit the signals to extract the patterns studied by the doctors. These methods are developed based on the intuition of the doctors. By analyzing what the doctors actually look for in such signals, relevant attributes are identified. Finally, corresponding mathematical models are developed by utilizing the vast repertoire of Signal Processing techniques. Machine learning based methods, on the other hand, simply select some signal parameters as features and train a learning algorithm to detect the arrhythmias. Despite being more medically consistent, one major shortcoming of the signal processing based methods is that they fail to perform properly when tested on a large amount of data. It is not completely unexpected as the methods are manually handcrafted and it is almost impossible for a person to analyze and consider signals of all shapes and forms. In this regard, machine learning based approaches are more promising as they in fact learn the patterns adaptively from data without any human intervention. Nevertheless, by considering feature engineering as a black box, this type of algorithms sometimes fails to capture the actual important patterns and attributes. Therefore, in this work we have proposed a fusion of the two, effectively combining the best of the both worlds. We develop a novel feature engineering scheme based on the intuitions of the doctors, extending the previous advancements in the former domain. Furthermore, we apply a suitable machine learning pipeline to identify the ECG signals with Ventriuclar Fibrillation. Therefore, we develop VFPred, performing a proper amalgam of the two distant domains, signal processing and machine learning. The most remarkable outcome of VFPred is that the values of sensitivity and specificity are almost balanced, whereas the existing algorithms, despite scoring higher sensitivity, fall short in specificity. We believe that this is due to the reason of us utilizing a feature engineering scheme based on how the doctors actually differentiate the ECG signals. Therefore, instead of performing futile mix-and-match of the numerous possible ECG signal parameters we have focused on the specific attributes that sets the VF class signals apart from the rest. Moreover, most significant features were selected using Random Forest, which resulted in a further superior feature set. Accompanying such an effective feature engineering with a properly tuned, robust SVM classifier, and at the same time, resolving the problem of class imbalance using SMOTE, enabled us to detect ventricular fibrilliation from ECG signals of even as short as 5 seconds long with high accuracy.

On the contrary, PPG signals, being connected with blood volume, is a possible measure of the pressure exerted on the walls of the blood vessels by the blood circulating therein. Therefore, a number of studies have been performed to infer blood pressure from

PPG signals. In this domain, most of the works revolve around the idea that the speed of blood flow reflects the state of the blood pressure. Therefore, these works mostly focus on deriving mathematical parameters to profile the motion of blood flow. In order to do estimate the progress of blood flow in a more detailed way, ECG signals are also used. Hence several parameters like PTT, PEP, PAT have been introduced in the literature and subsequently mathematical models were developed to predict blood pressure using those as features. A conspicuous shortcoming of such methods is the necessity of the inclusion of the ECG signal, which adds the requirement of a dedicated ECG sensor. To overcome this limitation, methods based on PPG signals alone were introduced later. These methods rely on computing some handcrafted features from the shape of the PPG signal. Most notably, the amplitudes and delay between certain points of interest like Systolic Peak, Diastolic Peak and Diastolic Notch are computed, which is followed by mathematical models to predict blood pressure value. However, this approach also has a shortcoming of its own as follows. Since they highly rely on the shape of the PPG signal waveforms, they can not operate on PPG signals that deviate from the ideal configuration. This greatly reduces the utility of such algorithms as PPG signals are quite prone to being distorted. Therefore, we have attempted to develop a method to predict blood pressure with a goal to overcome these two major drawbacks in mind, i.e., inclusion of ECG and requirement of ideally shaped PPG signals. As a result, we have developed PPG2ABP, a deep learning based method that can predict the continuous waveform of the Arterial Blood Pressure (ABP) from the PPG signal alone. Although our primary goal was to construct the ABP waveform, it appeared that the proposed PPG2ABP was also capable of predicting some other quantities of interest like SBP, DBP and MAP. Moreover, not only the waveform construction error was minimal, prediction of such quantities was also almost impeccable. These values even yielded Grade A score under the BHS Standard for the prediction of DBP and MAP, and Grade B score for prediction of SBP. However, for SBP, it should be noted that, Grade B is by far the state of the art except for a questionable work confined to a small dataset. In addition, AAMI standard was also met for the prediction of DBP and MAP. Furthermore, the predicted blood pressure values demonstrated great potential to classify Hypertension levels. All these outstanding outcomes were obtained from a diverse collection of PPG signals which even contained some signals of substandard quality. PPG2ABP was therefore found outperforming contemporary methods by a considerable margin.

The future direction of this research has several branches. In this thesis we have been confined within ECG and PPG signals, but other biomedical signals also hold great promises as well. Most notably, our present interest lies in analyzing EEG signals for their diverse applications, ranging from human emotion recognition to sleep stage detection. In addition, we focused on Ventricular Fibrillation as it is the most life threating one. However, there are other heart arrhythmias that can be diagnosed from ECG signals, Ventricular Tachycardia, Atrial Fibrillation, Atrial Flutter are to name a few. Studying these variants of arrhythmias in conjuction with their corresponding ECG signals is also something we wish pursue in future. Furthermore, our ultimate goals is to simplify these methods as much as possible without sacrificing accuracy, so that we can develop wearable devices to continually monitor the health conditions of the patients.

In conclusion, the methods developed in this thesis hold great promises in continuous assessment of health conditions of patients. Widespread availability of PPG and ECG sensors in wearables and smartwatches, coupled with the gradual advancement of cloud computing in parallel with in-device machine learning holds great potential in the future of constant medical monitoring.

Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, and M. Adam. Automated detection of arrhythmias using different intervals of tachycardia ecg segments with convolutional neural network. *Information sciences*, 405:81–90, 2017.

[3] F. Alonso-Atienza, E. Morgado, L. Fernandez-Martinez, A. García-Alberola, and J. L. Rojo-Alvarez. Detection of life-threatening arrhythmias using feature selection and support vector machines. *IEEE Transactions on Biomedical Engineering*, 61(3):832–840, 2014.

[4] F. Alonso-Atienza, J. L. Rojo-Álvarez, A. Rosado-Muñoz, J. J. Vinagre, A. García-Alberola, and G. Camps-Valls. Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection. *Expert Systems with Applications*, 39(2):1956–1967, 2012.

[5] A. Amann, R. Tratnig, and K. Unterkofler. A new ventricular fibrillation detection algorithm for automated external defibrillators. In *Computers in Cardiology, 2005*, pages 559–562. IEEE, 2005.

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

[7] A. Amann, R. Tratnig, and K. Unterkofler. Detecting ventricular fibrillation by time-delay methods. *IEEE Transactions on Biomedical Engineering*, 54(1):174–177, 2007.

[8] E. M. A. Anas, S. Y. Lee, and M. K. Hasan. Exploiting correlation of ecg with certain emd functions for discrimination of ventricular fibrillation. *Computers in biology and medicine*, 41(2):110–114, 2011.

[9] M. A. Arafat, J. Sieed, and M. K. Hasan. Detection of ventricular fibrillation using empirical mode decomposition and bayes decision theory. *Computers in Biology and Medicine*, 39(11):1051–1057, 2009.

[10] B. M. Asl, S. K. Setarehdan, and M. Mohebbi. Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. *Artificial intelligence in medicine*, 44(1):51–64, 2008.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017.

[12] H. J. Baek, K. K. Kim, J. S. Kim, B. Lee, and K. S. Park. Enhancing the estimation of blood pressure using pulse arrival time and two confounding factors. *Physiological measurement*, 31(2):145, 2009.

[13] A. Baldzizhar, E. Manuylova, R. Marchenko, Y. Kryvalap, and M. G. Carey. Ventricular tachycardias: Characteristics and management. *Critical Care Nursing Clinics*, 28(3):317–329, 2016.

[14] R. Barandela, J. S. Sánchez, V. Garcıa, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.

[15] S. Barro, R. Ruiz, D. Cabello, and J. Mira. 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, 1989.

[16] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[17] J. C. Bramwell and A. V. Hill. The velocity of pulse wave in man. *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 93(652):298–306, 1922.

[18] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[19] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur, and H. Nazeran. A review on wearable photoplethysmography sensors and their potential future applications in health care. *International journal of biosensors & bioelectronics*, 4(4):195, 2018.

[20] Centers for Disease Control and Prevention (CDC and others. Vital signs: awareness and treatment of uncontrolled hypertension among adults–united states, 2003-2010. *MMWR. Morbidity and mortality weekly report*, 61:703, 2012.

[21] H.-H. Chang and J. M. Moura. Biomedical signal processing. *Biomedical engineering and design handbook*, 2:559–579, 2010.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

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

[24] F. Chollet et al. Keras, 2015.

[25] K.-C. Chou. Some remarks on protein attribute prediction and pseudo amino acid composition. *Journal of theoretical biology*, 273(1):236–247, 2011.

[26] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.

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

[28] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[29] N. Dey, A. S. Ashour, W. S. Mohamed, and N. G. Nguyen. *Biomedical Signals*, pages 7–20. Springer International Publishing, Cham, 2019.

[30] R. Díaz-Uriarte and S. A. De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

[31] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

[32] D. L. Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[33] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

[34] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications*, pages 179–187. Springer, 2016.

[35] M. Elgendi. Optimal signal quality index for photoplethysmogram signals. *Bioengineering*, 3(4):21, 2016.

[36] A. for the Advancement of Medical Instrumentation et al. American national standard. manual, electronic or automated sphygmomanometers ansi. Technical report, AAMI SP10-2002, 2003.

[37] P. Fung, G. Dumont, C. Ries, C. Mott, and M. Ansermino. Continuous noninvasive blood pressure measurement by pulse transit time. In *The 26th annual international conference of the IEEE engineering in medicine and biology society*, volume 1, pages 738–741. IEEE, 2004.

[38] L. Geddes, M. Voelz, C. Babbs, J. Bourland, and W. Tacker. Pulse transit time as an indicator of arterial blood pressure. *psychophysiology*, 18(1):71–74, 1981.

[39] H. Gesche, D. Grosskurth, G. Küchler, and A. Patzak. Continuous blood pressure measurement by using the pulse transit time: comparison to a cuff-based method. *European journal of applied physiology*, 112(1):309–315, 2012.

[40] D. Giavarina. Understanding bland altman analysis. *Biochemia medica: Biochemia medica*, 25(2):141–151, 2015.

[41] A. L. Goldberger, 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. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220, 2000.

[42] A. L. Goldberger, 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. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[43] S. D. Greenwald. *The development and analysis of a ventricular fibrillation detector*. PhD thesis, Massachusetts Institute of Technology, 1986.

[44] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[45] T. K. Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

[46] S. W. Holm, L. L. Cunningham, E. Bensadoun, and M. J. Madsen. Hypertension: classification, pathophysiology, and management during outpatient sedation and local anesthesia. *Journal of oral and maxillofacial surgery*, 64(1):111–121, 2006.

[47] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*, volume 454, pages 903–995. The Royal Society, 1998.

[48] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[49] N. Ibtehaz and M. S. Rahman. Multiresunet : Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural Networks*, 121:74 – 87, 2020.

[50] N. Ibtehaz, M. S. Rahman, and M. S. Rahman. Vfpred: A fusion of signal processing and machine learning techniques in detecting ventricular fibrillation from ecg signals. *Biomedical Signal Processing and Control*, 49:349 – 359, 2019.

[51] S. H.-. Investigators. Catheter-based renal sympathetic denervation for resistant hypertension: durability of blood pressure reduction out to 24 months. *Hypertension*, 57(5):911–917, 2011.

[52] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[53] E. Jones, T. Oliphant, and P. Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[54] S. A. Jones. *ECG notes: Interpretation and management guide.* FA Davis, 2009.

[55] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany. Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time. In *2015 IEEE international symposium on circuits and systems (ISCAS)*, pages 1006–1009. IEEE, 2015.

[56] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany. Cuffless blood pressure estimation algorithms for continuous health-care monitoring. *IEEE Transactions on Biomedical Engineering*, 64(4):859–869, 2016.

[57] A. R. Kavsaoğlu, K. Polat, and M. Hariharan. Non-invasive prediction of hemoglobin level using machine learning techniques with the ppg signal's characteristics features. *Applied Soft Computing*, 37:983–991, 2015.

[58] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[59] B. S. Kim and S. K. Yoo. Motion artifact reduction in photoplethysmography using independent component analysis. *IEEE transactions on biomedical engineering*, 53(3):566–568, 2006.

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

[61] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[62] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[63] R. Krishnan, B. Natarajan, and S. Warren. Two-stage approach for detection and reduction of motion artifacts in photoplethysmographic data. *IEEE transactions on biomedical engineering*, 57(8):1867–1876, 2010.

[64] S. Kuo. Computer detection of ventricular fibrillation. *Proc. of Computers in Cardiology, IEEE Comupter Society*, pages 347–349, 1978.

[65] Y. Kurylyak, F. Lamonaca, and D. Grimaldi. A neural network-based method for continuous blood pressure estimation from a ppg signal. In *2013 IEEE International instrumentation and measurement technology conference (I2MTC)*, pages 280–283. IEEE, 2013.

[66] M. A. Laflamme and C. E. Murry. Heart regeneration. *Nature*, 473(7347):326, 2011.

[67] D. Laszuk. Python implementation of empirical mode decomposition algorithm, 2017.

[68] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[69] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570, 2015.

[70] G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

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

[72] Q. Li, C. Rajagopalan, and G. D. Clifford. Ventricular fibrillation and tachycardia classification using a machine learning approach. *IEEE Transactions on Biomedical Engineering*, 61(6):1607–1613, 2014.

[73] Y. Li, J. Bisera, M. H. Weil, and W. Tang. An algorithm used for ventricular fibrillation detection without interrupting chest compression. *IEEE Transactions on Biomedical Engineering*, 59(1):78–86, 2012.

[74] Y. Liang, M. Elgendi, Z. Chen, and R. Ward. An optimal filter for short photoplethysmogram signals. *Scientific data*, 5:180076, 2018.

[75] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[76] N. Luo, W. Dai, C. Li, Z. Zhou, L. Lu, C. C. Poon, S.-C. Chen, Y. Zhang, and N. Zhao. Flexible piezoresistive sensor patch enabling ultralow power cuffless blood pressure measurement. *Advanced Functional Materials*, 26(8):1178–1187, 2016.

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

[78] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921*, 2016.

[79] Z. Marcinkevics, M. Greve, J. I. Aivars, R. Erts, and A. H. Zehtabi. Relationship between arterial pressure and pulse wave velocity using photoplethysmography during the post-exercise recovery period. *Acta Univesitatis Latviensis: Biology*, 753:59–68, 2009.

[80] G. Martínez, N. Howard, D. Abbott, K. Lim, R. Ward, and M. Elgendi. Can photoplethysmography replace arterial blood pressure in the assessment of blood pressure? *Journal of clinical medicine*, 7(10):316, 2018.

[81] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[82] D. Mease, A. J. Wyner, and A. Buja. Boosted classification trees and class probability/quantile estimation. *Journal of Machine Learning Research*, 8(Mar):409–439, 2007.

[83] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10(1):213, 2009.

[84] F. Miao, Z. Liu, J. Liu, B. Wen, and Y. Li. Multi-sensor fusion approach for cuff-less blood pressure measurement. *IEEE journal of biomedical and health informatics*, 2019.

[85] A. Mjahad, A. Rosado-Muñoz, M. Bataller-Mompeán, J. Francés-Víllora, and J. Guerrero-Martínez. Ventricular fibrillation and tachycardia detection from surface ecg using time-frequency representation images as input dataset for machine learning. *Computer methods and programs in biomedicine*, 141:119–127, 2017.

[86] S. S. Mousavi, M. Firouzmand, M. Charmi, M. Hemmati, M. Moghadam, and Y. Ghorbani. Blood pressure estimation from appropriate and inappropriate ppg signals using a whole-based method. *Biomedical Signal Processing and Control*, 47:196–206, 2019.

[87] F. Nolle, F. Badura, J. Catlett, R. Bowser, and M. Sketch. Crei-gard, a new concept in computerized arrhythmia monitoring systems. *Computers in Cardiology*, 13:515–518, 1986.

[88] A. V. Oppenheim. *Discrete-time signal processing*, pages 564–575. Pearson Education India, 1999.

[89] E. O'Brien, J. Petrie, W. Littler, M. de Swiet, P. L. Padfield, D. Altman, M. Bland, A. Coats, N. Atkins, et al. The british hypertension society protocol for the evaluation of blood pressure measuring devices. *J hypertens*, 11(Suppl 2):S43–S62, 1993.

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

[91] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[92] S. G. Priori, C. Napolitano, M. Memmi, B. Colombi, F. Drago, M. Gasparini, L. DeSimone, F. Coltorti, R. Bloise, R. Keegan, et al. Clinical and molecular characterization of patients with catecholaminergic polymorphic ventricular tachycardia. *Circulation*, 106(1):69–74, 2002.

[93] J. Proença, J. Muehlsteff, X. Aubert, and P. Carvalho. Is pulse transit time a good indicator of blood pressure changes during short physical exercise in a young

population? In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 598–601. IEEE, 2010.

[94] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[95] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[96] N. Selvaraj, A. K. Jaryal, J. Santhosh, S. Anand, and K. K. Deepak. Monitoring of reactive hyperemia using photoplethysmographic pulse amplitude and transit time. *Journal of clinical monitoring and computing*, 23(5):315–322, 2009.

[97] P. A. Shaltis, A. T. Reisner, and H. H. Asada. Cuffless blood pressure monitoring using hydrostatic pressure changes. *IEEE Transactions on Biomedical Engineering*, 55(6):1775–1777, 2008.

[98] M. Sharma, K. Barbosa, V. Ho, D. Griggs, T. Ghirmai, S. K. Krishnan, T. K. Hsiai, J.-C. Chiao, and H. Cao. Cuff-less and continuous blood pressure monitoring: a methodological review. *Technologies*, 5(2):21, 2017.

[99] K. Shelley, S. Shelley, and C. Lake. Pulse oximeter waveform: photoelectric plethysmography. *Clinical monitoring*, pages 420–428, 2001.

[100] R. Shriram, A. Wakankar, N. Daimiwal, and D. Ramdasi. Continuous cuffless blood pressure monitoring based on ptt. In *2010 International Conference on Bioinformatics and Biomedical Technology*, pages 51–55. IEEE, 2010.

[101] I. Silva and G. B. Moody. An open-source toolbox for analysing and processing physionet databases in matlab and octave. *Journal of open research software*, 2(1), 2014.

[102] B. N. Singh and A. K. Tiwari. Optimal selection of wavelet basis function applied to ecg signal denoising. *Digital signal processing*, 16(3):275–287, 2006.

[103] G. Slapničar, N. Mlakar, and M. Luštrek. Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network. *Sensors*, 19(15):3420, 2019.

[104] M. H. Song, J. Lee, S. P. Cho, K. J. Lee, and S. K. Yoo. Support vector machine based arrhythmia classification using reduced features. *International Journal of Control Automation and Systems*, 3(4):571, 2005.

[105] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[106] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[107] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[108] X. Teng and Y. Zhang. Continuous and noninvasive estimation of arterial blood pressure using a photoplethysmographic approach. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*, volume 4, pages 3153–3156. IEEE, 2003.

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

[110] N. Townsend, L. Wilson, P. Bhatnagar, K. Wickramasinghe, M. Rayner, and M. Nichols. Cardiovascular disease in europe: epidemiological update 2016. *European heart journal*, 37(42):3232–3245, 2016.

[111] G. Van Rossum et al. Python programming language. In *USENIX Annual Technical Conference*, volume 41, page 36, 2007.

[112] G. Van Rossum et al. Python programming language. In *USENIX Annual Technical Conference*, volume 41, page 36, 2007.

[113] A. Verma and X. Dong. Detection of ventricular fibrillation using random forest classifier. *Journal of Biomedical Science and Engineering*, 9(05):259, 2016.

[114] S. v. d. Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[115] C. Wang, Z. Li, and X. Wei. Monitoring heart and respiratory rates at radial artery based on ppg. *Optik-International Journal for Light and Electron Optics*, 124(19):3954–3956, 2013.

[116] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460. IEEE, 2018.

[117] Wikipedia contributors. Electrocardiography — Wikipedia, the free encyclopedia, 2019. [Online; accessed 28-December-2019].

[118] Wikipedia contributors. Photoplethysmogram — Wikipedia, the free encyclopedia, 2019. [Online; accessed 28-December-2019].

[119] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[120] M. Y.-M. Wong, C. C.-Y. Poon, and Y.-T. Zhang. An evaluation of the cuffless blood pressure estimation based on pulse transit time technique: a half year study on normotensive subjects. *Cardiovascular Engineering*, 9(1):32–38, 2009.

[121] World Health Organization. *World health statistics 2015.* World Health Organization, 2015.

[122] World Health Organization and others. A global brief on hypertension: silent killer, global public health crisis: World health day 2013. Technical report, World Health Organization, 2013.

[123] X. Xing and M. Sun. Optical blood pressure estimation with photoplethysmography and fft-based neural networks. *Biomedical optics express*, 7(8):3007–3020, 2016.

[124] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, June 2010.

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