Augmenting Electrocardiogram Datasets using Generative Adversarial Networks

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Santhosh Alladi

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Junaed Sattar

May, 2020

Acknowledgements

I would first like to thank my advisor, Professor Junaed Sattar for providing me an opportunity to work on this project and for his continuous support and guidance. This thesis would not be coined without him.

I would like to thank Ben and Mike from Preventice for providing me with a real clinical dataset and for their feedback and support.

I would also like to thank the members of my thesis committee, Professor Catherine Qi Zhao and Professor Julianna Abel.

Finally, my deep and sincere gratitude to my family and friends for their continuous and unparalleled love, help and support.

Dedication

To my family and friends.

Abstract

In this work, we explore and propose various techniques for synthesis of realistic Electrocardiogram (ECG). In the last two decades, several methods were developed to automatically classify the ECG data. Deep learning has evolved rapidly and now demonstrates state-of-art performance in many fields, including the ECG domain. In this thesis, we first summarize the existing deep learning models in the context of ECG, providing the details about their architecture, hyper-parameters, performance results, training and testing datasets used, the number of classification classes, etc. Many of these models are trained and tested on very small datasets. However, deep learning models have to be trained on large datasets for accurate ECG classification. In this work, we propose a few generative models to augment the existing ECG datasets and demonstrate its effectiveness under various network configurations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Artificial Intelligence (AI) has been rapidly evolving with its applications in almost every field. Notably, the applications of AI in Medicine has been very empowering and useful to the society. In 2016, among the investments made towards AI research, healthcare applications received the highest allocation compared to other AI categories [15]. AI-powered medical technologies are advancing into applicable solutions for clinical practice [16, 17]. Today, various health-monitoring wireless devices and wearables, smartphones, mobile monitoring sensors are being widely used by individuals and in different areas of medicine. Deep learning algorithms can deal with increasing amounts of data provided by these devices. However these deep learning models need large datasets to train in order to perform efficiently [18]. Despite the wide spread use of automated ECG interpretation in medical diagnosis and care, the diagnostic accuracy of the automated interpretations is limited, there is significant room for important improvements [19]. One of the major factors for this is the limited availability of open-source ECG datasets. Since medical data is sensitive [20], it is often very difficult to find it for research and developmental purposes. The goal of this work is to augment the existing but scarce ECG datasets with synthetic data to assist in training deep classification models.

Cardiovascular Diseases(CVDs) account for about 31% of all the deaths world-wide, and they are the primary cause of mortality globally [21]. A majority of deaths due to CVDs are caused by arrhythmia, a condition in which the rhythm of the heart is irregular. The timely prediction of arrhythmia reduces the fatality rate of cardiac disease patients [22]. ECG is a crucial tool in the diagnosis of arrhythmia. They measure the electrical properties of the heart on a temporal scale. They generate periodic data which helps in estimating the heart rate (see Section 2.1).

There are many different types of arrhythmias, each of which is associated with a pattern in the ECG [23]. Cardiologists and technicians carefully examine an ECG recording to identify the type of arrhythmia. Analyzing a long ECG recording or a real-time diagnosis is tedious, painstaking and error-prone. Hence, automated, computer-assisted tools are needed to analyze and classify the ECG signals. The task of automation of the ECG classification has been evolving since the late 1990s. Starting from the basic decision tree kind of approach to the latest deep learning techniques, automated ECG interpretation has advanced rapidly. There are many deep learning models developed for this task (see Chapter 3). A majority of works use small datasets like the MIT-BIH dataset(refer Section 2.5) for training and testing their models.

Since the MIT-BIH dataset contains more number of Normal rhythms, there is an imbalance in the amount of data obtained for other types(see Table 2.2). A machine learning algorithm works well when there is an equal distribution of data among all the classes [24]. Additionally, the model will not be robust. As a result, the model will not work well when deployed in an industry, though it works well according to the published paper. These issues become bottlenecks to the widespread use of machine learning algorithms. To prevent it from happening, the models should be trained with large amounts of data.

Medical data is scarce, and it is challenging to find and collect clean data. Some of the solutions for the small dataset problem are to employ transfer learning, self-supervised learning, and data augmentation. Data augmentation is a strategy used to enlarge the training dataset by synthesizing new data. Generative Adversarial Networks (GANs), a class of machine learning algorithms, is one of the widely used techniques for data augmentation. GANs have already proved their worth, giving excellent results in generating realistic images with high resolution for various vision problems (see results of deblurGAN [35], ESRGAN [36]). In this thesis, we have used GANs to generate ECG data to augment the existing datasets. We have used the ECG dataset from our industry partner, Preventice Solutions® (Rochester, MN) to train our models.

These techniques protect the privacy of the patients from whom the data was collected. Also, the data generated by our model does not match with any of the training data.

1.1 Application Domain

This section discusses one of the use cases where this thesis work would be useful.

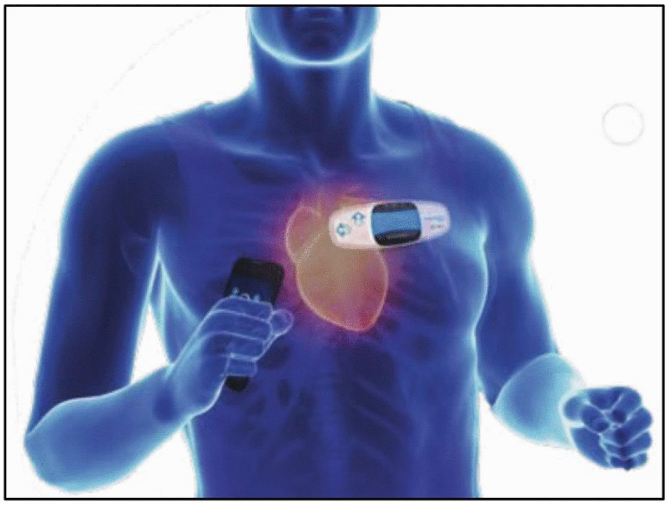

Figure 1.1: BGH device and it's placement (Source: [2])

There are typically two types of mobile devices that measure the ECG: a holter device and a Mobile Cardiac Telemetry (MCT) device. These devices are often prescribed to the patients who require continuous monitoring of their heart, like the ones who have undergone recent heart surgery or the ones having any heart condition. A holter device is designed only to measure the ECG, and has very limited storage. Wires from this device will be attached to the electrodes planted on the patient's chest. The locally stored data from the device is extracted after the monitoring period and then analyzed for any abnormal heartbeats or rhythms.

On the other hand, using Mobile Cardiac Telemetry (MCT) devices, the ECG information can be continuously uploaded to a receiving center(generally a cloud server) throughout the day. Usually, the ECG data is sent to a cloud processing center using any gadget like a smartphone. The analysis of the heartbeat information at the cloud processing center can be automated. A well trained deep learning model can be used to classify the heartbeats. If there is any abnormal rhythm detected, physicians and the patients will be notified about the critical findings. Figure 1.1 shows the placement of a MCT device, called BodyGaurdian® Heart (BGH) device from Preventice Solutions®.

The task of automated classification of the heartbeats though promising, is difficult. In the last two decades, many machine learning models were proposed to solve this task. Chapter 3 presents a survey of various works done in this field. Based on the review presented, most of the models developed uses small training and testing datasets. There are not many open datasets available and the ones available are small. A majority of the works use the MIT-BIH dataset, and it has data only from 47 patients. Models trained using these small datasets have a significant risk of overfitting. In these cases, data synthesized using data augmentation techniques will be beneficial.

1.2 Thesis Outline

The rest of this thesis is organized as follows:

- Chapter 2 briefly explains all the concepts used in this thesis.
- Chapter 3 reviews some existing deep learning models for classification of an ECG signal.
- Chapter 4 discusses the different models we implemented in this work, to synthesize ECG data.
- Chapter 5 summarizes the experiments we have conducted and our results.
- Chapter 6 presents the conclusion and discusses some future work directions.

Chapter 2

Background

In this chapter, we briefly explain all the concepts related to our thesis work. We talk about the ECG signal, discuss a few deep learning concepts and a generalized deep learning classification algorithm along with some evaluation metrics. Finally, we talk about some of the open-source ECG datasets available.

2.1 The ECG signal

The electrocardiogram (ECG) is a very popular diagnostic tool and is widely used in the treatment of various heart-related diseases. It is a recording of the changes in the electrical activity of the heart over time and contains essential physiological information that is widely used to analyze heart function. Electrodes are placed on the body, and the electrical potential of the heart is recorded from different angles, called ”leads”. Many electrodes can be used, and based on the number, their relative position on body is determined. In the conventional 12-lead ECG equipment, 10 electrodes are used which are placed on the patient’s chest and limbs to capture the ECG from 12 different angles. We know that heartbeats are continuous cycles of contraction and expansion. As the ECG signals represent the heartbeat, they are periodic too. An ECG signal corresponding to each beat has three main components namely, the P wave, the Q, R and S waves, and the T wave. Very rarely, a U wave can also be detected, and it is part of a normal ECG wave. Figure 2.1 shows a normal ECG wave. The Q, R, and S waves together form the QRS complex, which is the characteristic part of an ECG signal. The beat-to-beat classification can be done by the analysis of the QRS complex. The rhythm of the heart in terms of beats per minute (bpm) is usually calculated by counting the R peaks of the ECG wave during a one-minute time interval. The distance between the R peaks of two consecutive beats is called as the RR-interval and this is an important aspect in determining the rhythm of an ECG recording.

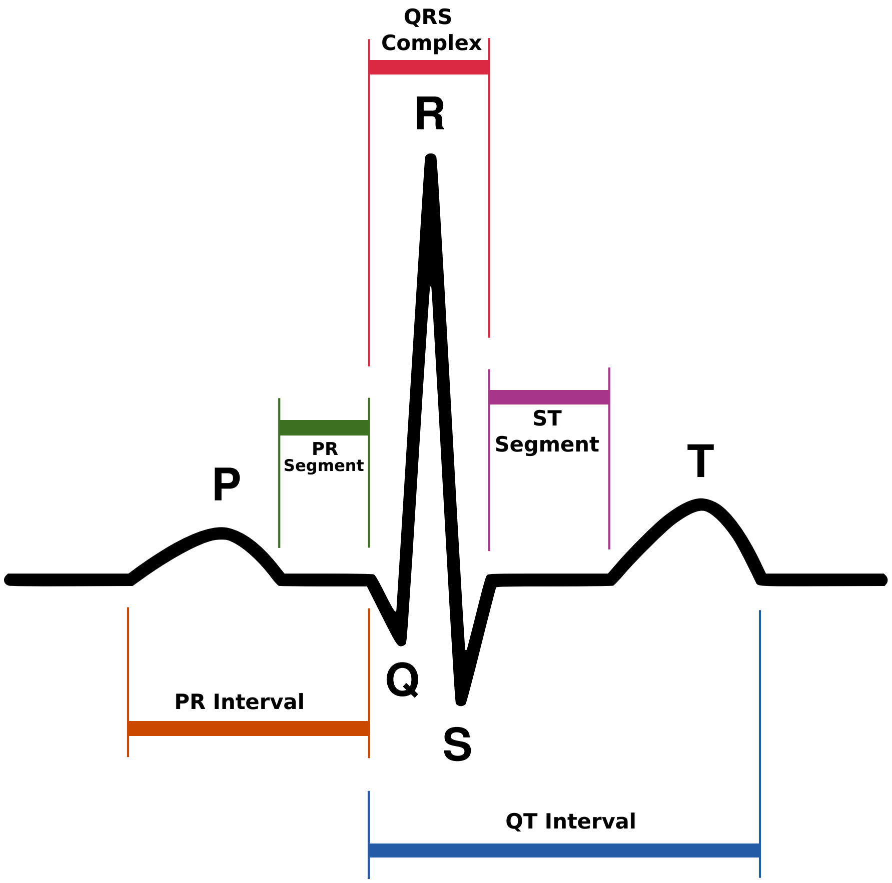

Figure 2.1: A Normal ECG wave (source: [3])

2.2 Deep Learning

Deep Learning can be considered as a subset of machine learning in Artificial Intelligence. Deep learning architectures such as Deep Neural Networks (DNN), Recurrent

Neural Networks (RNN) and Convolutional Neural Networks (CNN) have brought about breakthrough performance in almost all the fields [25]. Neural Networks are inspired by the human nervous system. Figure 2.2 shows the analogy between the neural network and the human nervous system. Neural Networks are computational algorithms that solves prediction problems.

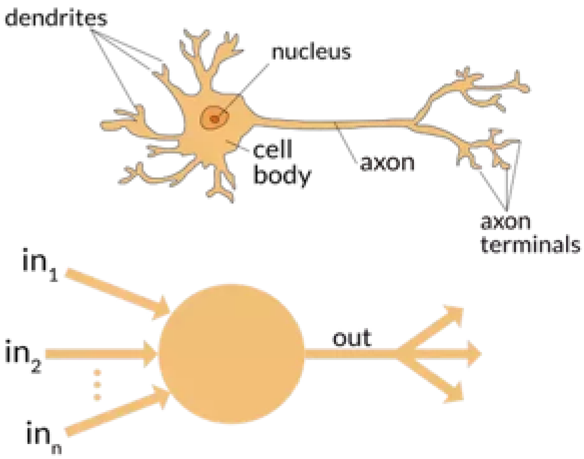

Figure 2.2: A Neuron and NN basic structure (source: [4])

2.2.1 Deep Neural Networks

A Deep Neural Network (DNN) is a particular network, which fuses the feature extraction and classification into a single learning model and constructs a decision-making function [26]. These types of neural networks have achieved great success in pattern recognition. A DNN consists of an input layer for the raw data descriptors, hidden inner layers for computation, and an output layer for the prediction(see Figure 2.3). The DNN finds a mathematical manipulation to turn the input into the output for any linear or non-linear relationship existing between them.

The output values are obtained by a series of sequential computations along the network layers on the input data. The input to a layer is the output of its previous layer. At every layer, the input vector is multiplied by the weight vector in the current layer to produce the weighted sum. On this weighted sum, a non-linear function, such as a sigmoid, hyperbolic tangent, or rectified linear unit (ReLU), is applied to compute the output values of the current layer. As we go along the layers, the vectors transform and lead to slightly more abstract representations as compared to the previous layers [25]. Based on the types of layers used in DNNs and the corresponding learning method, DNNs can be classified as Multi Layer Perceptron (MLP) or Auto Encoders (AE) [5].

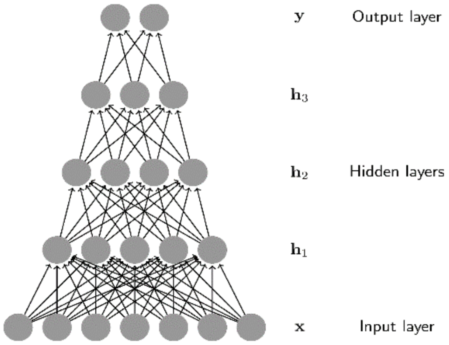

Figure 2.3: Basic structure of DNN (Source: [5])

2.2.2 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is tailored towards using images as inputs, which through a series of steps, are flattened into numerical inputs that can be fed into the same type of system. Conceptually, a CNN resembles a MLP. Every single neuron in the MLP has an activation function that maps the weighted inputs to the output. An MLP becomes a deep MLP when more than one hidden layer is added to the network. There are three basic layers– convolutional layer, pooling layer, and fully-connected layer with a rectified linear activation function in a CNN architecture (see Figure 2.4). Many improvements of CNN happened in different aspects, including layer design, activation function, loss function, regularization, optimization, and fast computation [27].

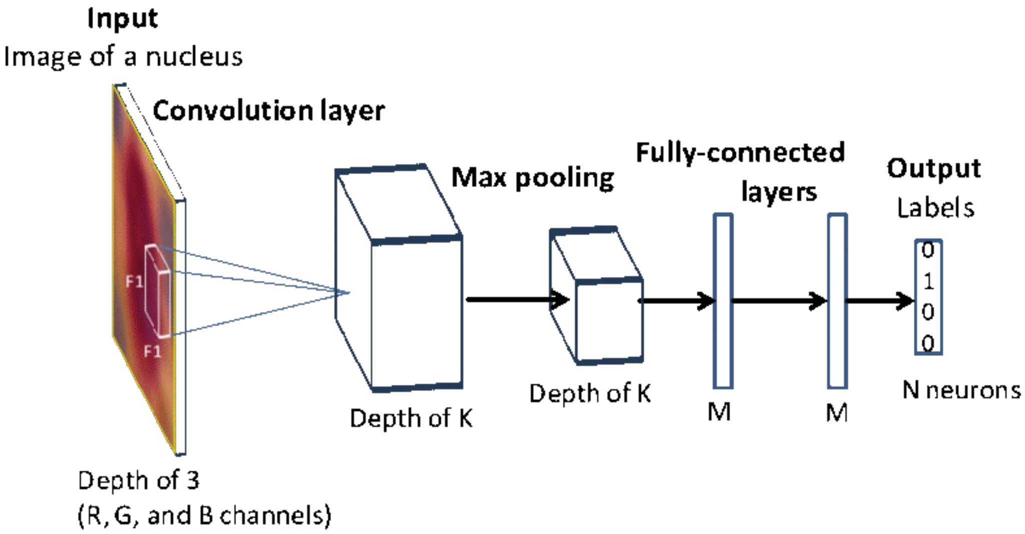

Figure 2.4: Basic architecture of a CNN (Source:[6])

2.2.3 Recurrent Neural Networks

Recurrent Neural Networks (RNN) are designed to utilize sequential information in the data. They have a basic structure with a cyclic connection (see Figure 2.5). As the input data is processed sequentially, the recurrent computation is performed in the hidden units whenever a cyclic connection exists. The past information is implicitly stored in the hidden units, which are called state vectors. The output for the current input is computed considering all previous inputs using these state vectors [25, 5].

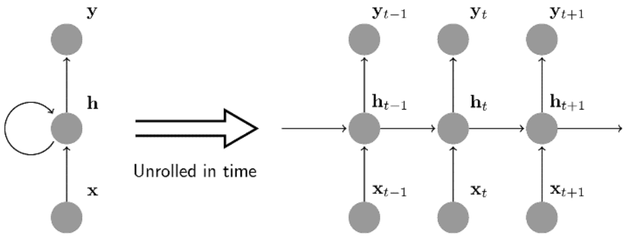

Figure 2.5: Basic structure of a RNN (Source: [5])

2.2.4 Generative Adversarial Networks

Generative Adversarial Networks (GANs) are one of the generative modeling techniques which use deep learning methods such as CNNs and LSTMs [28]. A GAN model has two main components, Generator and Discriminator. The generator network creates new data examples while the discriminator classifies these examples as real or fake. The training strategy of GAN models is quite interesting. We train both of these networks simultaneously. The training continues until nash equilibrium is attained.

The following analogy for GANs helps to understand this concept better. Consider the generator as a counterfeiter and the discriminator as a police. The counterfeiter produces fake currency notes, and the police detect and seize counterfeit money. To win, one must outdo the other - the counterfeiter must continuously learn to forge money so that it closely resembles its genuine counterpart, and the police must constantly become better at identifying the techniques used by the counterfeiter and thereby, detect the fraud.

2.3 Generalized Deep Learning Classification Algorithm

In the process of applying machine learning for analyzing ECG data, meaningful feature extraction or representation has to be done. Conventionally, meaningful or task-related features are mostly designed by human experts based on their knowledge about the target domains, which made it challenging for non-experts to exploit machine learning techniques for their own purposes. However, deep learning has relieved such obstacles by absorbing the feature engineering step into a learning step. That is, instead of extracting features in a hand-designed manner, deep learning requires only a set of data with minor pre-processing, if necessary, and then discovers the informative representations in a self-taught manner. The burden of feature engineering has shifted from human-side to computer-side, thus allowing non-experts in machine learning to effectively use deep learning for their own applications [29].

Currently, no standard methodology exists for the construction of an optimal neural network with the right number of layers and the neurons for each layer. For these reasons, DNNs are built empirically by performing a wide set of trials. In each trial, DNN is manually configured by changing the following parameters: the number of hidden layers, the activation function, the number of learning steps, and for each of the hidden layers, the number of neurons making up the layer. For each manual configuration, the classification accuracy is evaluated over the testing set. After this laborious manual phase, the best classification performance with a DNN can be composed.

The DNN classifiers have the neuron layers with an activation function. Widely used activation functions is the Rectified Linear Unit (ReLU). The ReLU is an activation function defined as:

$$f(x) = max(0, x)$$

where x is the input to a neuron.

An approximation to the rectifier is the analytic function:

$$f(x) = \ln(1 + \exp x)$$

which is called the *softplus* function.

The construction, training, and validation of a DNN can be described as follows:

- load the training and testing datasets
- build a classifier
- fit the model
- evaluate the accuracy of the model over the training set
- calculate the predictions of the model over the testing set
- calculate the classification performance of the model over the entire dataset

2.4 Evalution Metrics

For the evaluation of a DNN, different metrics like accuracy, sensitivity, specificity, precision, and F1 score are used. The definitions of these parameters are as follows:

$$accuracy = \frac{(TP + TN)}{(TP + TN) + (FP + FN)}$$

$$sensitivity or recall = \frac{TP}{(TP + FN)}$$

$$specificity = \frac{TN}{(FP + TN)}$$

$$precision = \frac{TP}{(TP + FP)}$$

$$F1\,score = \frac{2 * precision * recall}{precision + recall}$$

where:

True Positive (TP) is the number of instances where the model correctly predicts the positive class;

True Negative (TN) is the number of instances where the model correctly predicts a negative class;

False Positive (FP) is the number of instances where the model incorrectly predicts the positive class;

and False Negative (FN) is the number of instances where the model incorrectly predicts a negative class

After training, the model can be overfitting, underfitting, or balanced. In the case of underfitting, the model learns an improper trend or distribution of the data. Consequently, it will not be able to make right classification on testing or new data. In the case of a balanced model, the model learns the distribution of the data, and models it accuarately. This model will be able to successfully classify on testing and new data. In the case of overfitting, the model learns the distribution of the data and accurately classifies the training data. But, it will fail to classify the testing and new data because it learned the training data too well [7]. The Figure 2.6 depicts the three cases just discussed.

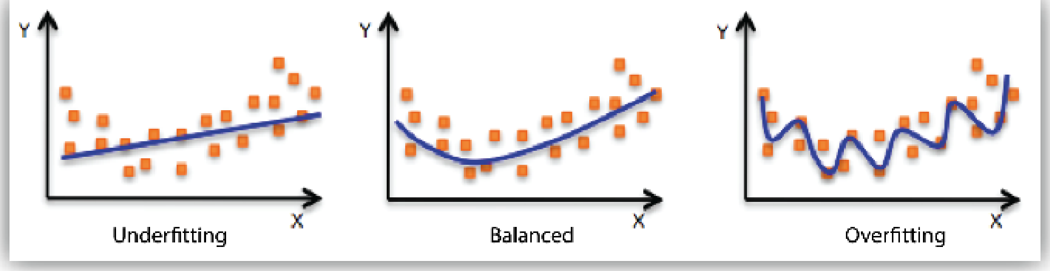

Figure 2.6: Different kinds of fitting (Source: [7])

2.5 The datasets and the AAMI standard

In this section, we discuss the various open-source ECG datasets available. These datasets can be used for research purposes to train models.

Various datasets are available, each of them containing heartbeats grouped by patients records. Most of them are freely available and also contain detailed instructions on how to preprocess the data before use. These datasets follow the standards specified by the Association for the Advancement of Medical Instrumentation (AAMI). These standards are available in ANSI/AAMI EC57:2012 (Revision of ANSI/AAMI EC57:1998/(R)2008) [1]. The standard also defines the protocol to perform the evaluations to make sure the automated testing methods are reproducible and comparable [30]. Table 2.1 summaries the five ECG beat classes based on recommendation from AAMI.

The AAMI standard specifies the use of five databases:

- AHA: The American Heart Association Database for Evaluation of Ventricular Arrhythmia Detectors (80 records of 35 minutes each)
- MIT–BIH: The Massachusetts Institute of Technology–Beth Israel Hospital Arrhythmia Database (48 records of 30 minutes each) [31]

N	**S**	**V**	**F**	**Q**
Normal	Atrial premature	Premature ventricular contraction	Fusion of ventricular and normal	Paced
Left bundle branch block	Aberrant atrial premature	Ventricular escape		Fusion of paced and normal
Right bundle branch block	Nodal (junctional) premature			Unclassifiable
Atrial escape	Supra-ventricular premature			
Nodal (junctional) escape				

Table 2.1: Summary of ECG beat classifications as per ANSI/AAMI EC57:2012 (Source: [1])

- ESC: The European Society of Cardiology ST-T Database (90 records of 2 hours each) [32, 33]
- NST: The Noise Stress Test Database (12 ECG records of 30 minutes each plus three records of noise only— supplied with the MIT–BIH database) [34, 33]
- CU: The Creighton University Sustained Ventricular Arrhythmia Database (35 records of 8 minutes each—supplied with the MIT–BIH database with incomplete annotations) [35, 33]

The first four datasets (AHA, MIT–BIH, ESC, and NST) contain digitized excerpts of two-channel Holter-type recordings. Every beat is labeled with its type. The CU dataset consists of digitized single-channel ECG recordings with rhythm changes labeled.

Of these five datasets, the MIT-BIH Arrhythmia database is the most commonly used dataset. It was the first openly available dataset used for the evaluation of arrhythmia detectors [31]. It contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 people. Randomly, 23 recordings were chosen from a set of 4000 24-hour ambulatory ECG recordings. This set was collected from patients at Boston's Beth Israel Hospital. The remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias [33]. Table 2.2 provides information on MIT-BIH database.

Type	Number of Beats
N	90,592
S	2,781
V	7,235
F	802
Q	8,039
Total	109,449

Table 2.2: A summary of MIT-BIH database (Source: [1])

Chapter 3

A review: Deep Learning models for ECG classification

We have discussed the generalized algorithm for a deep learning classifier and the different metrics to evaluate a learning model. We have also talked about some of the open-source datasets used for training and testing the models. This chapter reviews some of the existing deep learning architectures for ECG classification.

Awni Et al. published a deep neural network model to classify 12 rhythm classes using 91,232 single-lead ECG data from 53,549 patients. The data used by them was from a single-lead ambulatory ECG monitoring device. The DNN achieved an average area under the Receiver Operating Characteristic curve (ROC) of 0.97. The average F1 score, for the DNN (0.837), exceeded that of average cardiologists (0.780). With specificity fixed at the average achieved by cardiologists, the sensitivity of the DNN exceeded the average cardiologist sensitivity for all rhythm classes [8].

They obtained very impressive results, and they used a large dataset as opposed to the standard MIT-BIH dataset, which is the most commonly used dataset for training and testing. This model can classify ten arrhythmias as well as sinus rhythm and noise for a total of 12 output rhythm classes. The model has a 34-layer network in an end-to-end manner to simultaneously output probabilities for a wide range of distinct rhythm diagnoses, all of which are enabled by their dataset. The dataset is larger than most of the other datasets of its kind. Distinct from some other DNN approaches, no substantial preprocessing of ECG data, such as Fourier or wavelet transforms, is needed to achieve strong classification performance. Figure 3.1 shows the architecture they used.

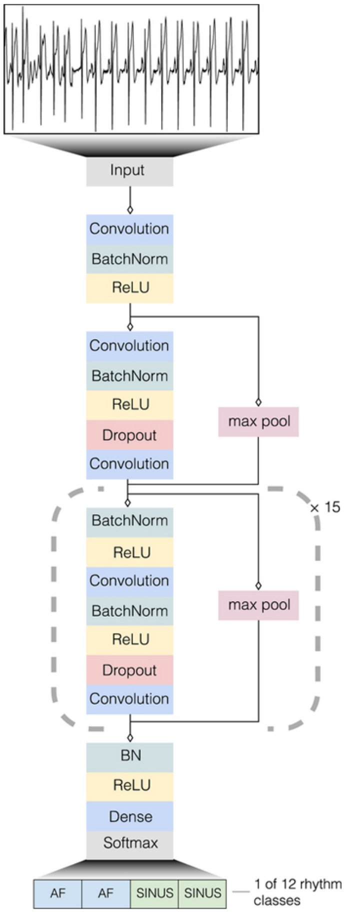

Figure 3.1: Awni Et al. DNN architecture (Source: [8])

Cardiologs® has developed its first deep neural network able to detect multiple heart conditions simultaneously, using only the raw ECG signal as input. It is a 12-lead ECG interpretation algorithm. They take ECG input, which is mapped to one of the 16 pre-specified groups, and classify whether it is abnormal or not. Also, based on this outcome, they calculated the proportion of all the ECGs for which all groups were identified, with no false negative or false positive groups [36].

A model by Erdenebayar Et al. automatically predicts atrial fibrillation (AF) rhythms using a CNN based model. The ECG signal was preprocessed and segmented before training and evaluating the proposed model. Their model achieved high performance with a sensitivity of 98.6%, specificity of 98.7%, and an accuracy of 98.7%. They only classify AF vs. other rhythm types. The model is trained on the MIT-BIH database, Atrial Fibrillation Database (AFDB), the Paroxysmal Atrial Fibrillation Prediction Challenge Database (PAFDB), and the MIT-BIH Normal Sinus Rhythm Database (NSRDB) provided by PhysioNet [33, 37].

Wenbo Et al. proposed a 12-layer 1D CNN to classify single-lead ECG signal into five classes of heart diseases. The proposed method was tested on the MIT-BIH arrhythmia database, and the results were measured. The model obtained a positive predictive value of 0.977, a sensitivity of 0.976, and an F1 score of 0.976. They compared the results obtained with that of the other four methods proposed by Kiranyaz [38], Hu [39], Jiang [40], and Ince [23], on the same database, and obtained superior results on all the three measures. Figure 3.2 shows their architecture [9] .

Rajput Et al. in [41] uses 1D data from a single-lead device. However, this 1D data is converted to 2D Spectrotemporal images after processing wavelets transform and short-time Fourier transform. These 2D images are used for training the deep learning model. The dataset consists of 121,346 single lead episodic ECG records, which are collected from patients with a history of cardiac arrhythmias.

In [10], Sherin Et al. demonstrates the application of the Restricted Boltzmann Machine (RBM) and deep belief networks (DBN) for ECG classification following the detection of ventricular and supraventricular heartbeats using single-lead ECG. MIT-BIH database was used to illustrate the model performance. They have achieved a classification accuracy of 93.78% for SVEB class signals and 96.94% for VEB class signals at a sampling rate of 360 Hz. Their experimental results also demonstrate that their framework provides similar classification accuracy (93.63% for SVEB and 95.87% for VEB) when sampled at only 114 Hz, based on which they conclude that the lower

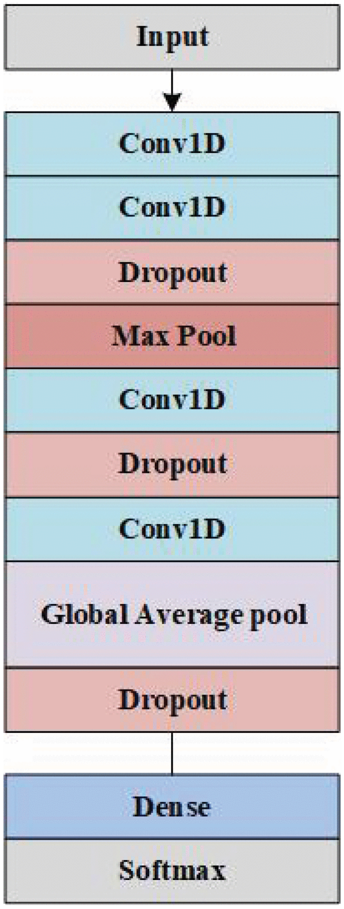

Figure 3.2: The model by Wenbo Et al. (Source: [9])

sampling rate of 114 Hz is also sufficient for accurate classification using their model. Figure 3.3 shows a block diagram of their proposed methodology.

A stacked CNN-LSTM model is proposed by Jen Et al. in [11] to classify the ECG data into their respective classes. Layers 1 to 4 are convolutional layers, and layers 5 to 7 are LSTM layers. The last layer is a fully-connected layer, which makes the classification. The CNN was used as a feature extractor, and the LSTM network was used for sequential learning. The model architecture can be found in Figure 3.4. The recall (sensitivity) values for normal and abnormal are 0.9984 and 0.9985, respectively. The dataset used was an open-source PhysioNet database.

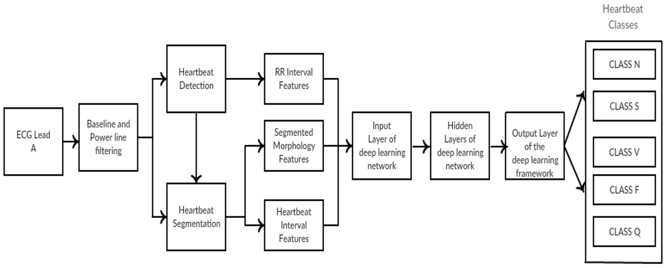
Heartbeat Classes
ECG Lead A
Baseline and Power line filtering
Heartbeat Detection
Heartbeat Segmentation
RR Interval Features
Segmented Morphology Features
Heartbeat Interval Features
Input Layer of deep learning network
Hidden Layers of deep learning network
Output Layer of the deep learning framework
CLASS N
CLASS S
CLASS V
CLASS F
CLASS Q

Figure 3.3: Block diagram of methodology proposed by Sherin Et al. (Source: [10])

Another model based on a combination of CNN and LSTM is proposed by Shu Et al. [42]. It classifies five arrhythmia classes using variable-length heartbeats. They could obtain 98.10% accuracy, 97.50% sensitivity, and 98.70% specificity. The dataset used was the MIT-BIH arrhythmia physio bank dataset.

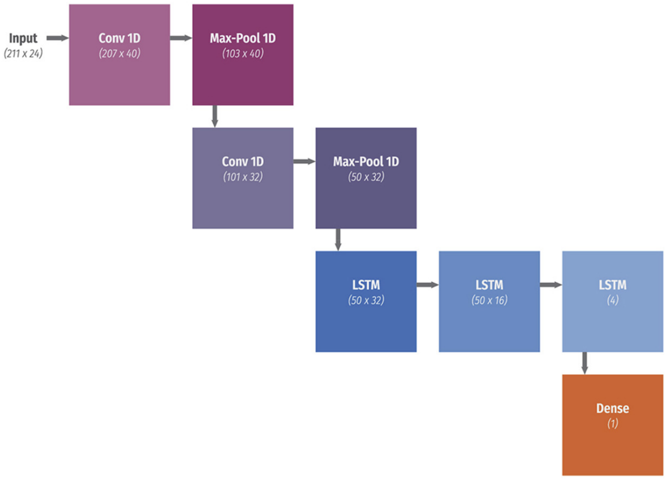

Figure 3.4: The proposed network structure by Jen Et al. (Source: [11])

Acharya Et al. in [12] developed a 9-layer CNN model, comprising of three convolution layers, three max-pooling layers, and three fully-connected layers to identify five different categories of heartbeats in ECG signals. They conducted experiments on two datasets, original and noise attenuated sets of ECG signals derived from the MIT-BIH arrhythmia database. This dataset was artificially augmented to even out the number of instances the five classes of heartbeats and filtered to remove high-frequency noise. The CNN was trained using the augmented data and achieved an accuracy of 94.03% and 93.47% in the diagnostic classification of heartbeats in original and noise-free ECGs, respectively. When a CNN was trained with highly imbalanced data, i.e., without augmenting to the original dataset, the accuracy of the CNN reduced to 89.07% and 89.3% in noisy and noise-free ECGs respectively. Figure 3.5 shows the architecture of the DNN [12].

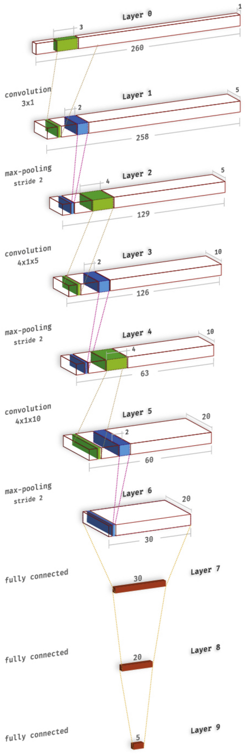

Figure 3.5: The proposed architecture by Acharya Et al. (Source: [12])

In the automated task of ECG interpretation and classification, it is essential to get reliable signals from the ECG devices. The entire learning and prediction can go wrong if we are training the model with improperly measured data. Hence artifact detection is very important [43]. Park et al. propose a method based on three characteristics of ECG artifacts: the spike-like property, the periodicity, and the lack of correlation with the

EEG. The method described involves a two-step process: ECG artifact detection using the energy interval histogram (EIH) method and ECG artifact elimination using a modification of average ensemble subtraction. The artifact detection algorithm discussed in [44], has three main steps: Artifact level, which indicates the intensity of artifacts in the ECG signal, is calculated. In this step, an adaptive filter is used to estimate the artifact in the ECG signal from the impedance signal [45]. Then the artifact levels are post-processed, and finally, based on a threshold value, the ECG signal is marked as an artifact.

Chapter 4

Augmenting ECG Datasets with Deep Generative Models

In the previous chapters, we have argued that the automated classification of ECG data is essential, and the latest advancements in machine learning technologies can help achieve this complicated task. Unfortunately, due to the scarcity of openly available datasets, machine learning models cannot be adequately trained, resulting in poor classification accuracy. To address this scarcity, we take the approach of augmenting existing datasets with synthetic but realistic data generated using deep machine learning techniques. This chapter discusses the details of our methods.

In the followings sections, we discuss our training dataset, the GAN models we developed, our training strategies and results. With sufficient training, the GAN generated signals are expected to mimic actual ECG records taking as either noise or real ECG samples as input.

4.1 Data

Our industry partner, Preventice Solutions® (Rochester, MN), provided the ECG dataset used for training our model. This data was captured using their single-lead

MCT BodyGuardian Heart (BGH) device™. ECG recordings in this dataset were collected from 3,493 subjects. The total number of beats are 661,509. Every record is annotated and adjudicated by three certified, experienced ECG technicians. This dataset follows the standards suggested by the AAMI ANSI: EC 57 [1]. The data captured from the BGH device was at a frequency of 256 Hz, which results in 0.5-second ECG records. This dataset has all types of ECG rhythms.

4.2 Data Preprocessing

We have preprocessed the raw ECG signals from the BGH device to remove the DC offset and Baseline Wander. DC offset is the mean displacement of an ECG signal from zero. Baseline Wander (BW) is a low-frequency artifact, commonly present in the ECG signal recordings. BW removal is an essential step in the preprocessing of the ECG signals because BW makes interpretation of ECG recordings difficult. We have used the eighth order Daubechies (DB-8) wavelet [46] high pass filter to remove the DC offset and the BW.

4.3 Model Architectures

Below, we describe the architectures of the various GAN models we have developed. Discriminators in all our models are CNNs. For Generator networks, we use CNNs or LSTMs depending on the model.

4.3.1 PulseGAN model

Our first model, PulseGAN, is a Deep Convolutional Generative Adversarial Network (DCGAN) [47] with some changes. The details of the model are discussed below.

As with any GAN model, there are two main components, a Generator network and a Discriminator network. The Generator network in this model is a deep convolutional network with 92 layers. The network architecture is as shown in Figure 4.1. The input to this network is a random sequence of 1024 data points (noise), denoting a noisy signal. The first layer of the generator, whose output is reshaped to (8, 1024), is a dense layer

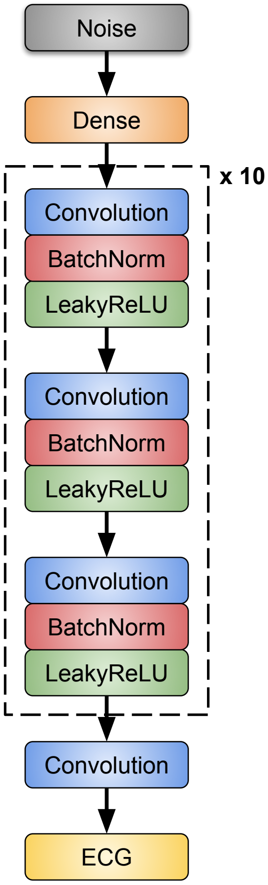

Figure 4.1: PulseGAN Generator Network

with 8*1024 units. The network has ten blocks, each consisting of a repeating series of 1D-convolutional layer, batch normalization layer, and Leaky-ReLU layer. Each block has three 1D-convolution layers, of which the first two have a stride length of 1. The third convolutional layer has a stride length of 2, and hence, the number of filters drop to half by the end of each block. The filter size is 1024 at the beginning, and at the end of the network, it drops to 1. We use a kernel size of 8 for all the layers. The output of this generator network is a sequence of 1024 data points. The generated output is expected to look like a real ECG record after proper training. Since the frequency of the ECG signal that we are using is 256 Hz, the output corresponds to a 4 second ECG strip. We pass this generated ECG signal as input to the discriminator network, along with real ECG samples.

In simple terms, if we consider the generator network to be a black box, we send a noisy signal as input, and we get a fake ECG signal as output. This is shown in Figure 4.2.

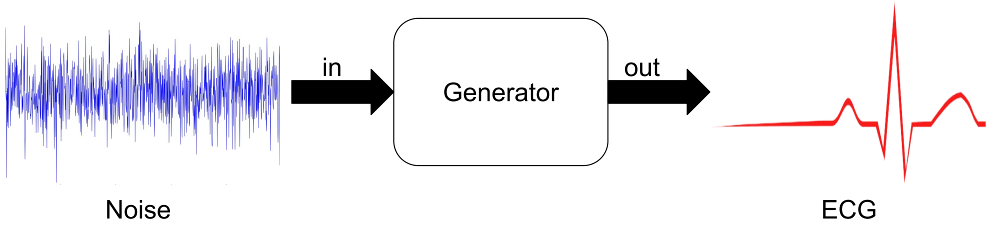

Figure 4.2: Generator functionality in a nutshell. It takes a noisy signal as input and outputs an ECG signal, after proper training. (Source: Noise Image: [13], ECG Image: [14])

The discriminator network is also a convolutional network with eight 1D convolutional layers. The number of filters in the first convolutional layer is 8, and it gradually increases to 1024, increasing 2-fold at each layer. The kernel size in each convolutional layer is 8. After each convolutional layer, there is a leakyReLU and a drop out layer. The drop out rate used is 0.3. There is also a batch normalization layer after every convolutional layer except after the first and the last layers. After this series of layers, we flatten the data and send it to a dense layer with 1 unit. Figure 4.3 shows the discriminator model. The output of the discriminator network is either 0 or 1, indicating whether the input realistically resembles a valid ECG signal or not.

In short, if we input an ECG signal to the discriminator network, it gives an output indicating whether the given signal is real or fake. This is shown in Figure 4.4.

The generator network synthesizes data that is close to realistic data and tries to outsmart the discriminator's attempts to classify it as fake. The discriminator network continuously learns the deception strategies used by the generator to identify real and

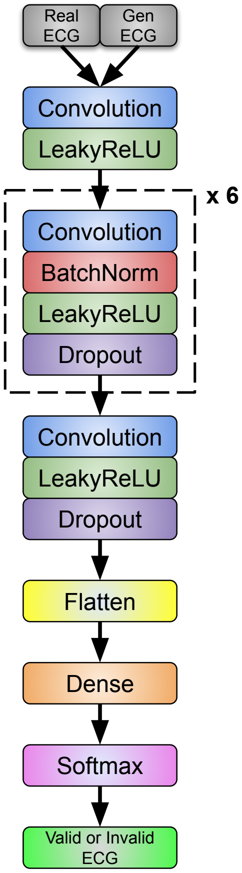

Figure 4.3: PulseGAN Discriminator Network

fake ECG signals. Over a series of epochs, both networks learn the modalities in the ECG data, and the generator model synthesizes new ECG datapoints, which appears to be real.

4.3.2 LPulseGAN model

The results from the PulseGAN model show that the RR-interval (see Section 2.1) is not constant. Also, the consecutive beats do not look similar. Hence to solve these two issues, we have used an LSTM layer [28] in the generator network. The discriminator network is the same as described in the previous model.

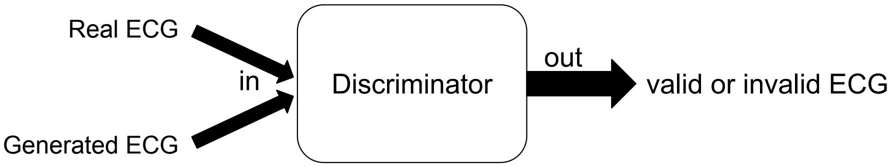

Figure 4.4: Discriminator functionality in a nutshell

4.3.3 RhythmGAN model

Inspired by the concept of StyleGAN [48], we have modified our PulseGAN model to generate different ECG profiles for the same person. For example, given a normal sinus rhythm, our model generates an arrhythmic condition of the rhythm, e.g., Sinus tachycardia (ST). The input for the generator is the normal sinus rhythm, and the output of the model is the corresponding ST rhythm. The discriminator model is trained with the output from the generator and some real ST rhythms. Figure 4.5 shows an overview of our RhythmGAN model.

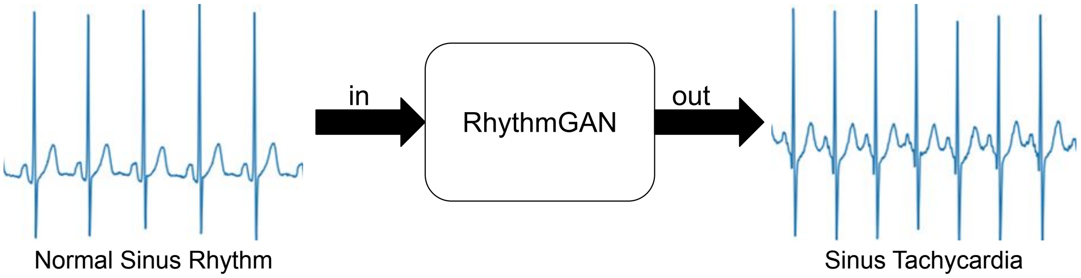

Figure 4.5: RhythmGAN model overview

All the models use Adam's optimizer [49] with a learning rate of 0.0001. The loss function used is binary cross-entropy loss [50]. We have done extensive hyperparameter tuning by varying learning rates, optimization functions, kernel size, stride length, activation functions, and the number of layers. We have explained these details in Section 5.3.

4.4 Training Strategies

In this section, we will discuss two different strategies for training our models. In the first training strategy, we have used beat-centered ECG records. These records have their R-peaks at the center of the signal. We have padded the signal with zeroes, wherever required. In the second strategy, we have used the ECG rhythms. In both these strategies, we have used a noise of dimension 128 as input. We begin with 128 filters in the output layer. In other words, we generate ECG records of 0.5 second duration containing one beat. Once we have tuned the model with different hyperparameters as mentioned above, we double the dimensionality of the output. The model is then tuned again. We repeated this until the output generated has 1024 data points.

Chapter 5

Experiments

In this chapter, we describe all the experiments we have conducted. We talk about the specifications of the system we have used for developing and training our models. We show a few ECG records from our training dataset and some samples generated by our models. Finally, we provide an analysis of our results.

5.1 System Configuration

This section describes the configuration of the system we have used in our work. The machine has AMD Ryzen™ Threadripper™ 1920X 12-Core Processor with HyperThreading cpu with 1.957 GHz base clock. It has two GPUs, one is the NVIDIA® TITAN V with 12 GB HBM2 memory, 5120 CUDA® cores and 640 Tensor cores. The second one is the NVIDIA® TITAN Xp with 12 GB GDDR5X memory and 3840 CUDA® cores. The system has 32 GB RAM. In our work, we have used the 2.0.0-beta1 tensorflow libraries. As already mentioned, the dataset was provided by Preventice Solutions®.

5.2 Qualitative Results

Generally, ECGs are analyzed by cardiologists, physicians or certified ECG technicians for any kind of abnormalities. To determine whether a synthesized signal mimics the morphology of a real ECG signal or not, is tricky and requires the expertise of a trained medical professional. We were not able to conduct the qualitative assessment of the synthesized ECG data by our industry partner due to the current COVID-19 situation.

5.3 Hyperparameter Tuning

We have tuned our models with all the possible combinations of values mentioned below for the following parameters: learning rate, optimization functions, kernel size, stride length, activation functions, input noise dimension, and the number of hidden layers. We have varied the learning rates from 1e-7 to 0.3. The optimizers used are Adam [49], SGD [51], and RMSProp [52], and the activation functions used are Tanh, Sigmoid, ReLU, and Leaky-ReLU [53]. The stride lengths and kernel sizes used are from the sets {1,2,4,8} and {2,4,8,16} respectively. The input noise dimension used was either 128 or 256. The number of layers was adjusted according to the duration of the ECG signal being generated.

5.4 Results

In this section, we discuss the results of our work. We present a few ECG data samples generated by our models. Before we get to the results, it should be borne in mind that whether a synthesized ECG signal mimics the morphology of a real ECG signal or not, is not a straightforward visual comparison (for similarities in the waveforms) and requires the expertise of a trained medical professional. The ECG samples in Figures 5.1, 5.4, 5.7 and 5.10 are a few ECG samples from our training dataset. As mentioned in Section 4.1, we used all types of rhythms for training our PulseGAN and LPulseGAN models. So accordingly, various rhythm patterns were generated.

As mentioned in section 4.4, we first trained our models to generate 0.5 second ECG samples. For this, we used a noisy signal, consisting of 128 random sequence of numbers as input to the generator. The discriminator was trained with 0.5 second ECG records from the training dataset and the output from the generator. The generated 0.5 second ECG data from PulseGAN and LPulseGAN models are shown in Figures 5.2 and 5.3.

Once we have successfully generated 0.5 second ECG samples, we have increased the dimension of the output layer of the generator to 256. Correspondingly we have increased the dimension of the input to the discriminator to 256. We have tuned our model with all the possible combinations of the hyperparameters, as mentioned in the Section 4.4. This process was repeated, doubling the dimension of the output layer of the generator each time, until it reached 1024. Since the frequency of the input ECG data is 256 Hz., 1024 data points correspond to a four second ECG signal. Figures 5.2, 5.5, 5.8 and 5.11 show some of the ECG samples generated by PulseGAN model. Figures 5.3, 5.6, 5.9 and 5.12 contain samples synthesized by the LPulseGAN model.

We have trained our RhythmGAN model to generate only four second samples as it is difficult to identify the rhythms from samples of duration less than four seconds. As discussed in Section 4.3.3, the input to the generator model is an ECG signal from the training dataset. Figure 5.14 shows some samples which appear like Sinus Tachycardia (ST) generated by our RhythmGAN model taking normal sinus rhythms as input.

5.5 Analysis of the results

In this section, we analyze the results of our models. As mentioned above, we are unable to get a qualitative analysis of our results. However we would like to discuss some evident observations in our results.

5.5.1 Analysis of the PulseGAN output samples

The half-second and one-second ECGs generated by the PulseGAN resemble real ECG signals. Every sample has a P-wave, well-shaped QRS complex, followed by a T-wave. Also, all the images presented here have different shapes for the QRS complex; this is very usual and expected as ECG beats have different wave-forms. Some of the samples have a noisy pattern before and after the QRS complex, but it is not abnormal as it can represent fibrillation of the heart. Many samples in our training dataset have them.

For a two-second ECG record, on an average, there can be two to three beats. In general, it can have one to six beats, depending on the rhythm of the heart. In the second sample, the last beat appears to be a bit skewed, and it represents an artifact.

The third beat in the third sample, is not equally spaced from the second and fourth beat, but the other beats seems to be equally spaced. So it represents a valid ECG sample. Also, for the fifth beat in the same sample, the deflection is positive, which is usual as sometimes the left side of the heart generates more deflection as compared to the right side. In the last sample in Figure 5.8, the pattern between two consecutive beats is close but not the same. In an original and clean ECG record, the repeating pattern would be the same.

In the four-second ECG samples from Figure 5.11, the beats towards the end of the sample do not match with the initial beats. The same is the case with the first two beats in the second sample. These represents artifacts.

5.5.2 Analysis of LPulseGAN output samples

The 0.5 and one-second samples are normal and realistic. The beat in the fourth sample in Figure 5.6 is also usual. It represents a QRS fragmentation. Observe that there is a small p-wave before this. This sample also represents a valid ECG, though it is a bit noisy.

The two-second samples generated by LPulseGAN are a bit noisy. There is no clear beat pattern in the third sample and it might represent an artifact. The first sample in this figure is noisy.

Among the four-second samples in Figure 5.12, the third sample contains an artifact as the second and the last beats are not similar.

5.5.3 Analysis of RhythmGAN output

For the experiment we have conducted on this model, the input was Normal Sinus Rhythm and output was Sinus Tachycardia (ST), which is a rhythm during which there is an increased heart rate. Most of the output samples of this model exhibit the desired quality, an increase in the frequency of the beats. However, the fourth sample in Figure 5.14 has around 15 beats and doesn't represent ST. It can represent Atrial Fibrillation as there are no P waves and the heart rate is around 225 beats per minute (15 beats in 4 seconds implies 225 in one minute).

On the whole, majority of the samples generated exhibit the well-defined characteristics of an ECG waveform. Samples published in this section are those that are seemingly similar to a real ECG waveform and where the wave-forms appear cleaner.

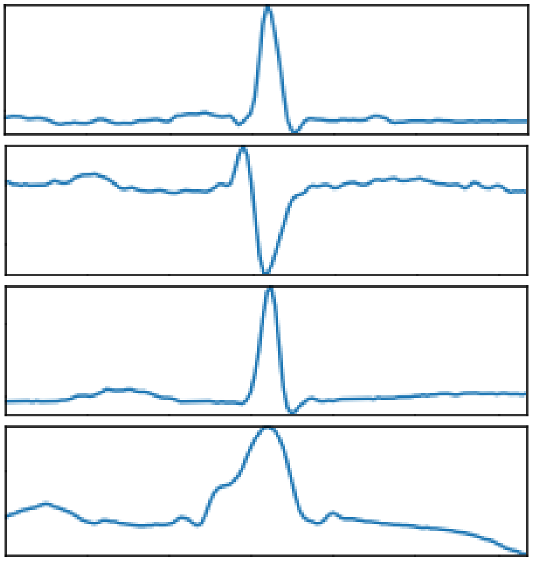

Figure 5.1: 0.5 second ECG samples from the training dataset

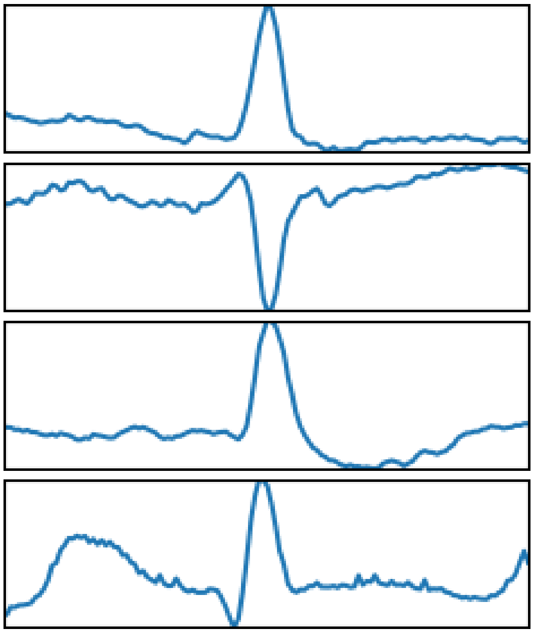

Figure 5.2: 0.5 second ECG samples generated by the PulseGAN dataset

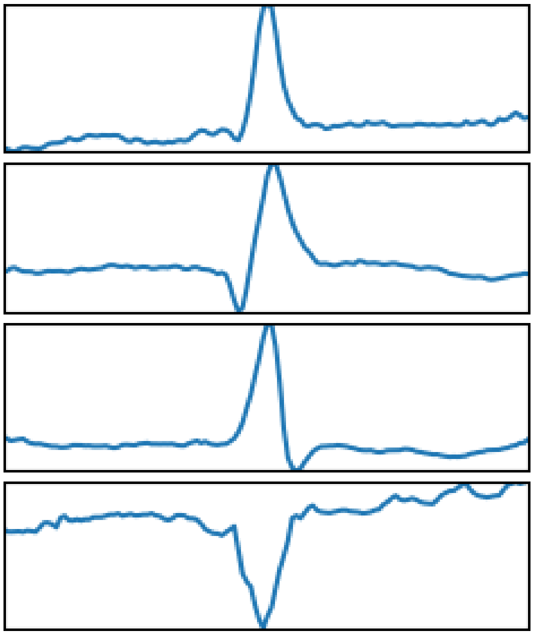

Figure 5.3: 0.5 second ECG samples generated by the LPulseGAN dataset

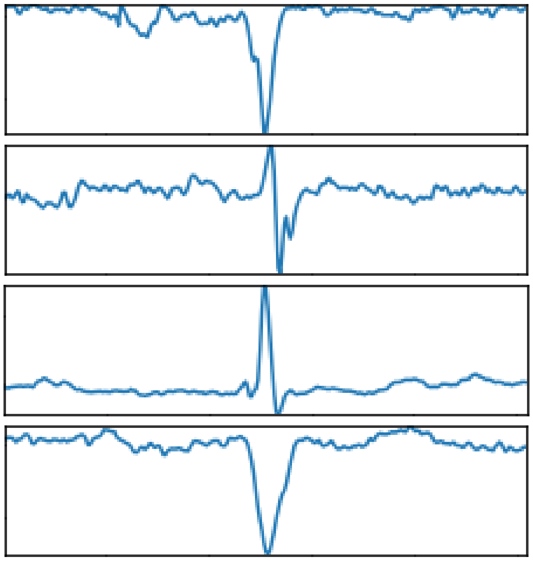

Figure 5.4: One second ECG samples from the training dataset

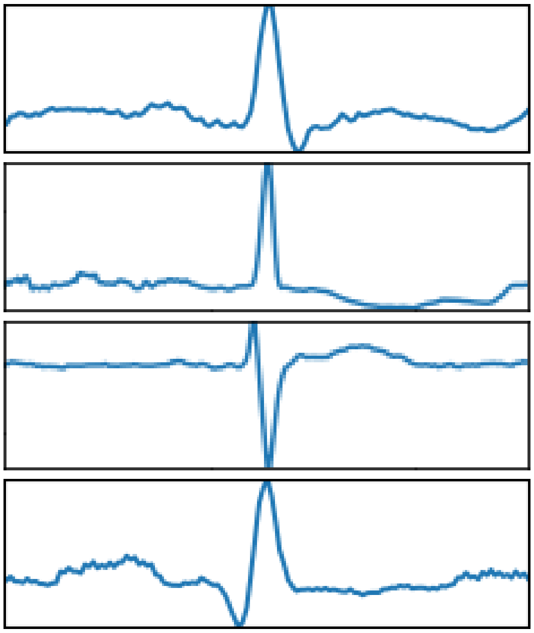

Figure 5.5: One second ECG samples generated by the PulseGAN dataset

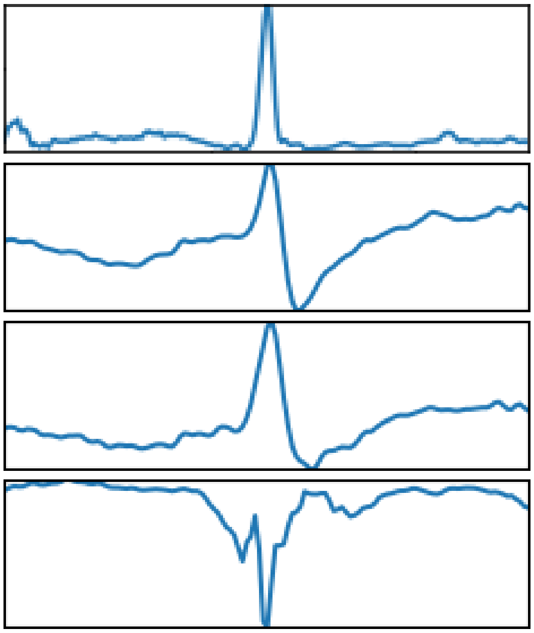

Figure 5.6: One second ECG samples generated by the LPulseGAN dataset

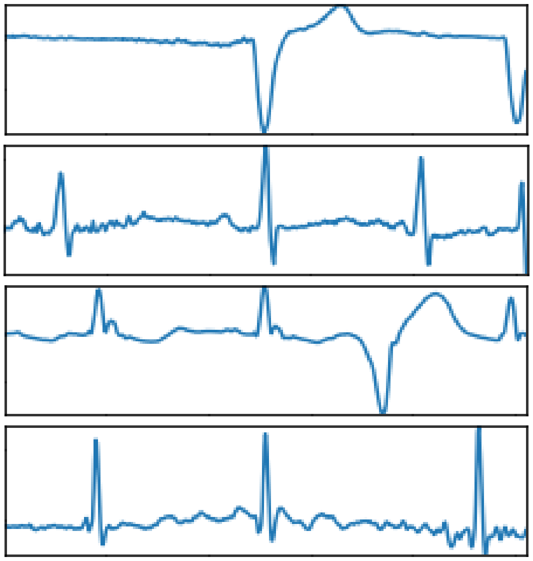

Figure 5.7: Two second ECG samples from the training dataset

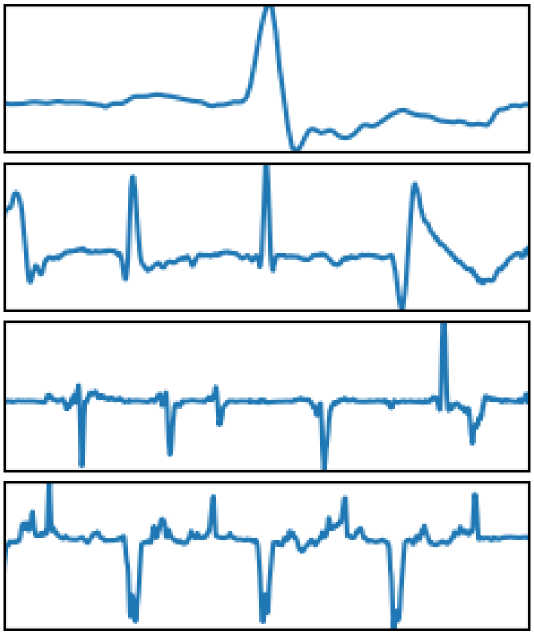

Figure 5.8: Two second ECG samples generated by the PulseGAN dataset

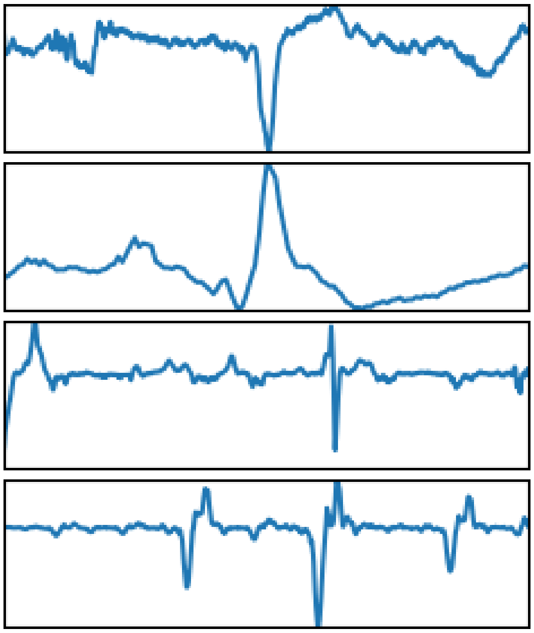

Figure 5.9: Two second ECG samples generated by the LPulseGAN dataset

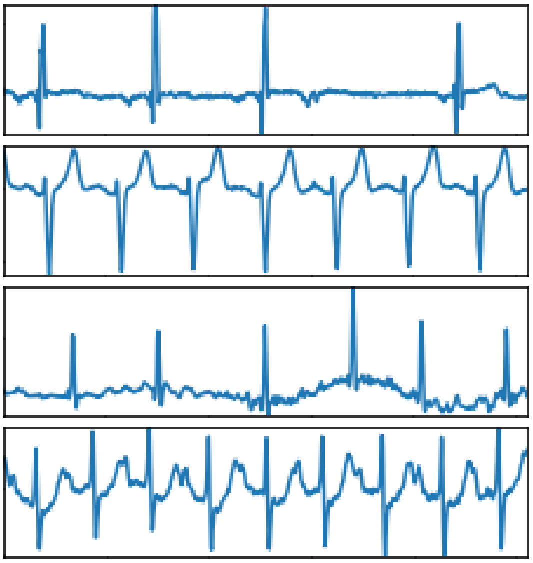

Figure 5.10: Four second ECG samples from the training dataset

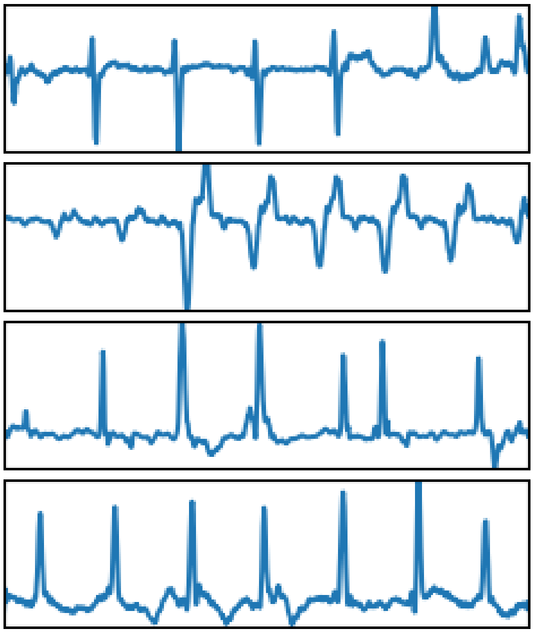

Figure 5.11: Four second ECG samples generated by the PulseGAN dataset

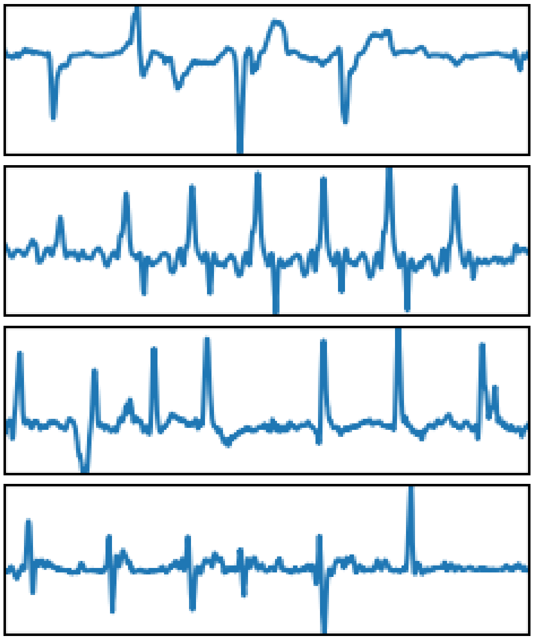

Figure 5.12: Four second ECG samples generated by the LPulseGAN dataset

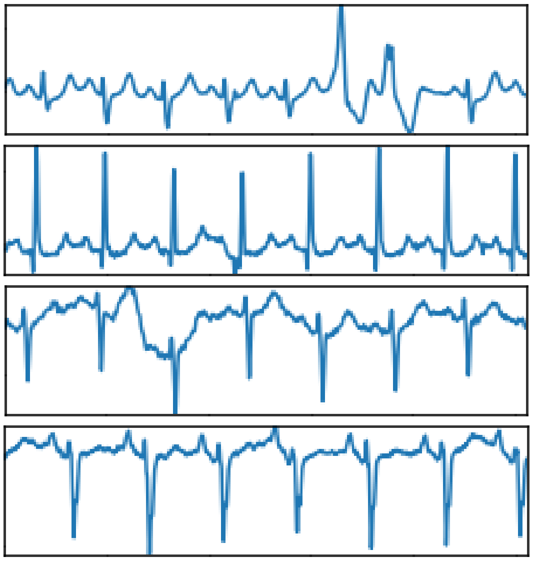

Figure 5.13: Four second ECG samples of Sinus Tachycardia samples from the training dataset.

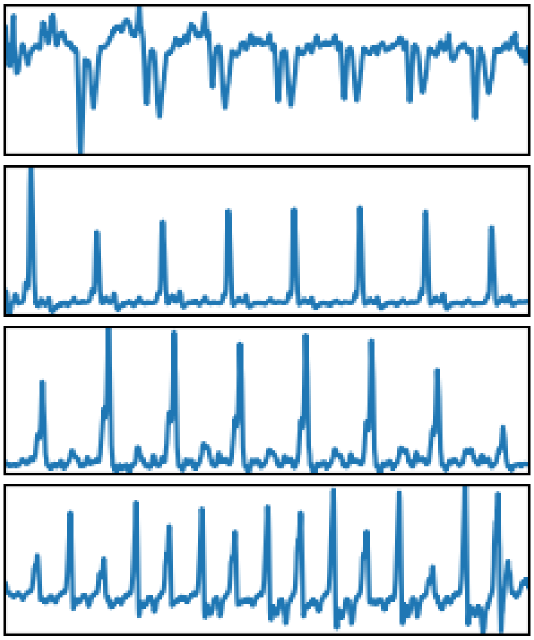

Figure 5.14: Four second ECG samples generated by the RhythmGAN model trained to generate ST rhythms.

Chapter 6

Conclusion and Discussion

To solve the problem of scarcity of ECG datasets, we have explored the data augmentation technique. In this thesis, we have developed a few GAN models to synthesize ECG data. We have used two types of generator networks, CNNs and LSTMs. For discriminator, we have used CNNs. For the RhythmGAN models, we can vary the input rhythms, and ECG samples of required rhythm can be generated by giving similar rhythm samples from the training dataset as input to the discriminator. The generated ECG samples exhibit the characteristics of the ECG waveform. However, some of the samples are noisy, and some contain artifacts.

Further research is necessary to stabilize the GAN training process and produce better ECG samples. In our work, we have used a mix of ECG samples from different leads for training the models. Training the model with ECG samples from the same lead can produce better results. The dataset we used for training in this work contains all kinds of rhythms. It has some noisy ECG samples as well. Better results can be obtained if we train the model only on one type of rhythms, collected by placing the MCT in a similar position on the body and eliminating the noisy beats and artifacts. Also, other generative techniques have to be explored like Variational Auto Encoders (VAE) [54]. VAEs are also showing good results in many generative tasks [55].

Once the model is generating realistic ECG data, we could combine this model with an ECG classification model to generate annotated data. This method significantly reduces the load on ECG technicians to annotate and adjudicate the synthesized data. Also, a new parameter can be introduced to the model, which specifies the type of rhythms to be generated. Specific ECG rhythm samples can be synthesized by setting that parameter to the required value.

References

[1] "ANSI/AAMI EC57: 2012 Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," Tech. Rep., 2012. [Online]. Available: www.aami.org

[2] B. A. Teplitzky and M. McRoberts, "Fully-automated ventricular ectopic beat classification for use with mobile cardiac telemetry," in *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2018, pp. 58–61.

[3] "Electrocardiography - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Electrocardiography

[4] T. Velaga, "AI in Medicine: A Beginner's Guide — Hacker Noon." [Online]. Available: https://hackernoon.com/ai-in-medicine-a-beginners-guide-a3b34b1dd5d7

[5] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, p. 851, 2017.

[6] F. Sheikhzadeh, M. Guillaud, and R. Ward, "Automatic labeling of molecular biomarkers of whole slide immunohistochemistry images using fully convolutional networks," *PLOS ONE*, vol. 13, 2016.

[7] "Generalization and Overfitting — Machine Learning." [Online]. Available: https://wp.wwu.edu/machinelearning/2017/01/22/generalization-and-overfitting/

[8] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, "Cardiologist-level arrhythmia detection and classification

in ambulatory electrocardiograms using a deep neural network," *Nature medicine*, vol. 25, no. 1, p. 65, 2019.

[9] W. Zhang, L. Yu, L. Ye, W. Zhuang, and F. Ma, "ECG Signal Classification with Deep Learning for Heart Disease Identification," in *2018 International Conference on Big Data and Artificial Intelligence (BDAI)*, 2018, pp. 47–51.

[10] S. M. Mathews, C. Kambhamettu, and K. E. Barner, "A novel application of deep learning for single-lead ECG classification," *Computers in Biology and Medicine*, vol. 99, pp. 53–62, 2018.

[11] J. H. Tan, Y. Hagiwara, W. Pang, I. Lim, S. L. Oh, M. Adam, R. S. Tan, M. Chen, and U. R. Acharya, "Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals," *Computers in Biology and Medicine*, vol. 94, pp. 19–26, 2018.

[12] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. S. Tan, "A deep convolutional neural network model to classify heartbeats," *Computers in Biology and Medicine*, vol. 89, pp. 389–396, 2017.

[13] "White noise - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/White_noise

[14] "ECG wave image." [Online]. Available: https://toppng.com/ekg-vector-free-download-on-melbournechapter-cover-facebook-cardiology-PNG-free-PNG-Images_176850?search-result=facebook-logo-png

[15] "Healthcare Remains The Hottest AI Category For Deals." [Online]. Available: https://www.cbinsights.com/research/artificial-intelligence-healthcare-startups-investors/

[16] S. Chen and Q. Zhao, "Attention-based autism spectrum disorder screening with privileged modality," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1181–1190.

[17] M. Jiang, S. Francis, D. Srishyla, C. Conelea, Q. Zhao, and S. Jacob, “Classifying individuals with asd through facial emotion recognition and eye-tracking,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019.

[18] G. Briganti and O. Le Moine, “Artificial Intelligence in Medicine: Today and Tomorrow,” *Frontiers in Medicine*, vol. 7, p. 27, 2 2020. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/32118012http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC7012990

[19] A. H. Ribeiro, M. H. Ribeiro, G. M. M. Paixão, D. M. Oliveira, P. R. Gomes, J. A. Canazart, M. P. S. Ferreira, C. R. Andersson, P. W. Macfarlane, M. W. Jr., T. B. Schön, and A. L. P. Ribeiro, “Automatic diagnosis of the 12-lead ECG using a deep neural network,” *Nature Communications*, vol. 11, no. 1, pp. 1–9, 4 2020. [Online]. Available: http://feeds.nature.com/~r/ncomms/rss/current/~3/58yr_TVCeO0/s41467-020-15432-4?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound

[20] “What is HIPAA.” [Online]. Available: https://www.dhcs.ca.gov/formsandpubs/laws/hipaa/Pages/1.00WhatisHIPAA.aspx

[21] “Cardiovascular diseases (CVDs).” [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[22] S. K. Pandey and R. R. Janghel, “Automatic detection of arrhythmia from imbalanced ECG database using CNN model with SMOTE,” *Australasian Physical and Engineering Sciences in Medicine*, vol. 42, no. 4, pp. 1129–1139, 12 2019.

[23] T. Ince, S. Kiranyaz, and M. Gabbouj, “A Generic and Robust System for Automated Patient-Specific Classification of ECG Signals,” *Biomedical Engineering, IEEE Transactions on*, vol. 56, pp. 1415–1426, 2009.

[24] I. Bakkouri and K. Afdel, “Multi-scale CNN based on region proposals for efficient breast abnormality recognition,” *Multimedia Tools and Applications*, vol. 78, no. 10, pp. 12 939–12 960, 5 2019.

[25] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[26] L.-p. Jin and J. Dong, "Ensemble Deep Learning for Biomedical Time Series Classification," *Computational Intelligence and Neuroscience*, vol. 2016, no. 2016, 2016.

[27] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, no. C, pp. 354–377, 2018.

[28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735

[29] D. Shen, G. Wu, and H.-I. Suk, "Deep Learning in Medical Image Analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp. 221–248.

[30] E. J. D. S. Luz, W. R. Schwartz, G. Camara-Chavez, and D. Menotti, "ECG-based heartbeat classification for arrhythmia detection: A survey," *Computer Methods and Programs in Biomedicine*, vol. 127, no. C, pp. 144–164, 2016.

[31] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[32] A. TADDEI, G. DISTANTE, M. EMDIN, P. PISANI, G. B. MOODY, C. ZEELENBERG, and C. MARCHESI, "The European ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography," *European Heart Journal*, vol. 13, no. 9, pp. 1164–1172, 1992. [Online]. Available: https://doi.org/10.1093/oxfordjournals.eurheartj.a060332

[33] G. AL, A. LAN, G. L, H. JM, I. PCh, M. RG, M. JE, M. GB, P. C-K, and S. HE, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2003.

[34] Moody GB, Muldrow WE, and Mark RG, "The MIT-BIH Noise Stress Test Database," in *Computers in Cardiology*, 1984, pp. 381–384. [Online]. Available: https://physionet.org/physiobank/database/nstdb/

[35] N. F. M, B. F. K, C. J. M, B. R. W, and S. M. H, "CREI-GARD. A new concept in computerized arrhythmia monitoring systems." *Proceedings. Annual Scientific Meeting of Computers in Cardiology*, vol. 1986, pp. 515–518, 1986.

[36] S. W. Smith, B. Walsh, K. Grauer, K. Wang, J. Rapin, J. Li, W. Fennell, and P. Taboulet, "A deep neural network learning algorithm outperforms a conventional algorithm for emergency department electrocardiogram interpretation," *Journal of Electrocardiology*, vol. 52, pp. 88–95, 2019.

[37] U. Erdenebayar, H. Kim, J.-U. Park, D. Kang, and K.-J. Lee, "Automatic Prediction of Atrial Fibrillation Based on Convolutional Neural Network Using a Short-term Normal Electrocardiogram Signal," *Journal of Korean Medical Science*, vol. 34, no. 7, 2019.

[38] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2016.

[39] Y. H. Hu, S. Palreddy, and W. J. Tompkins, "A patient-adaptable ECG beat classifier using a mixture of experts approach," *IEEE transactions on bio-medical engineering*, vol. 44, no. 9, p. 891—900, 9 1997. [Online]. Available: https://doi.org/10.1109/10.623058

[40] W. Jiang and S. Kong, "Block-Based Neural Networks for Personalized ECG Signal Classification," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 18, pp. 1750–1761, 2007.

[41] K. S. Rajput, S. Wibowo, C. Hao, and M. Majmudar, "On Arrhythmia Detection by Deep Learning and Multidimensional Representation," 2019.

[42] S. L. Oh, E. Y. K. Ng, R. S. Tan, and U. R. Acharya, "Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats," *Computers in Biology and Medicine*, vol. 102, pp. 278–287, 2018.

[43] H.-J. Park, D.-U. Jeong, and K.-S. Park, “Automated detection and elimination of periodic ECG artifacts in EEG using the energy interval histogram method,” *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 12, pp. 1526–1533, 2002.

[44] J. Ottenbacher, M. Kirst, L. Jatoba, M. Huflejt, U. Grossmann, and W. Stork, “Reliable motion artifact detection for ECG monitoring systems with dry electrodes,” in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2008. IEEE, 2008, pp. 1695–1698.

[45] S. S. Haykin, *Adaptive filter theory*, 4th ed. Upper Saddle River, N.J.: Prentice Hall, 2002.

[46] A. C. H. Rowe and P. C. Abbott, “Daubechies wavelets and Mathematica,” *Computers in Physics*, vol. 9, no. 6, p. 635, 6 1995. [Online]. Available: http://scitation.aip.org/content/aip/journal/cip/9/6/10.1063/1.168556

[47] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 11 2016.

[48] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June. IEEE Computer Society, 6 2019, pp. 4396–4405. [Online]. Available: http://arxiv.org/abs/1812.04948

[49] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 12 2015.

[50] S. Mannor, B. Peleg, and R. Rubinstein, “The cross entropy method for classification,” in *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*. New York, New York, USA: ACM Press, 2005, pp. 561–568. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1102351.1102422

[51] H. Robbins and S. Monro, "A Stochastic Approximation Method," *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[52] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.

[53] C. Enyinna Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," Tech. Rep.

[54] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 6 2019. [Online]. Available: http://arxiv.org/abs/1906.02691http://dx.doi.org/10.1561/2200000056

[55] J. Hong, M. Fulton, and J. Sattar, "A generative approach towards improved robotic detection of marine litter," 2019.

Appendix A

Glossary and Acronyms

Care has been taken in this thesis to minimize the use of jargon and acronyms, but this cannot always be achieved. This appendix contains a table of acronyms and their meaning.

Table A.1: Acronyms

Acronym	Meaning
ECG	Electrocardiogram
DNN	Deep Neural Network
DBN	Deep Belief Network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
AAMI	Association for the Advancement of Medical Instrumentation
MIT-BIH	The Massachusetts Institute of Technology–Beth Israel Hospital
GAN	Generative Adversarial Network
LSTM	Long Short Term Memory
ST	Sinus Tachycardia
NSR	Normal Sinus Rhythm
MCT	Mobile Cardiac Telemetry