Generative Adversarial Networks for ECG generation, translation, imputation and denoising

by

Alaina Mahalanabis

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Artificial Intelligence is increasingly being used in medical applications. One challenge present in AI in medicine is not having high quality datasets available for training machine learning models. In this work, I explore two different methods of generating high quality medical data. In the first approach, I used a cycleGANs as novel method for ECG translation, imputation and denoising. In the second method, I present a novel algorithm for generating high quality ECG data that uses a machine learning framework called Generative Adversarial Networks and explanation AI systems. Through empirical evaluation, I show that both methods improve over state-of-the-art methods in their respective applications. This thesis demonstrates that machine learning methods can be used to address the data scarcity problem in the medical setting.

Acknowledgements

I would like to thank the following friends, family and mentors:

First, I would like to thank Dr. Vijay Ganesh, for his mentorship and guidance. Thank you for working with me to explore research areas that are most interesting to me. Thank you so much for giving me the opportunity to pursue my degree under your supervision.

Dr. Krzysztof Czarnecki and Dr. Mark Crowley, thank you for agreeing to be a reader for my thesis and for your valuable feedback.

Thank you to everyone I collaborated with as part of Dr. Ganesh's research group especially to Vineel Nagisetty for his guidance and feedback even after he has oficially graduated from the program.

Finally, I would like to thank my parents for their continued support. I could not have completed my masters degree without them.

Dedication

This is dedicated to my beloved family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial intelligence has the potential to transform the field of medicine. Multiple different applications of AI are being explored in the medical field including AI for medical image interpretation, AI in drug discovery, personalized medicine, natural language processing applications to prepare clinical reports and many more applications [17].

There are many different types of medical data including imaging data such as magnetic resonance imaging and x-rays, time series data including biomedical signals such as electrocardiograms (ECG), electroencephalography (EEG) and sensor data, multimodal data sets such as electronic health records (EHR) and big data sources such as genomics and other "omics" data. Each type of dataset poses unique challenges. Medical images can contain billions of pixels too large to fit neural networks without preprocessing. Multimodal datasets are difficult to work with for xAI systems that typically work with continuous data. Time series data require machine learning models to learn the temporal relationships in the data.

One of the major technical challenges facing medical AI is the scarcity of high-quality training sets. Supervised learning techniques require large, labeled, high-quality datasets that are difficult to obtain. The accuracy of the model depends directly on the amount of training data available. The scarcity of medical data exists for several reasons: privacy and the sensitive nature of medical data make data sharing difficult, the shortage of rare medical datasets, the cost of obtaining datasets, and low quality data.

There are two mechanism that allow for medical data sharing: informed consent or data anonymization. Both mechanisms present challenges. If informed consent was only obtained for a specific context, the data cannot be used in another context. There is also

bias when getting informed consent where there are important differences between consenters and non-consenters. Alternatively, data can be anonymized but current anonymization techniques are inadequate to protect the privacy of people [6].

The cost of obtaining some types of medical data is another barrier to getting access to medical datasets. MRI scanners cost millions of dollars and are also expensive to maintain. Advanced EEG methods that are relatively inexpensive can be several thousand dollars.

In the case of rare diseases, any researcher will not have access to sufficient data on the rare disease. To address this issue, several open science initiatives have been founded. However, challenges still remain and getting access to rare disease data is hard.

Fractional amounts of low quality data can negatively impact outcomes. Imaging data may be of poor quality. Time-series data often have missing values due to distortions and faults with the instrumentation. Biomedical signals can have noise.

1.1 Related Work

Several recent works have explored the use of GANs for the generation of ECG [8, 5]. The first approach explored several CNN and LSTM architectures for generated ECGs. This paper highlights some of the challenges with ECG generation, including instability in training and no consensus on how to evaluate the quality of synthetic time series data. The second approach used a CNN architecture and evaluated the quality of the synthetic data by using the synthetic data to augment a heartbeat classifier. In our work, we address the challenges highlighted in the papers.

In Chapter 4, we incorporate explanation AI output into our GAN architecture. This idea was initially proposed by [14] and we have extended it to the time series domain. The main idea behind this approach is to improve the feedback from the discriminator to the generator. The discriminator loss is used to compute the gradients to update the generator. The xAI output indicates which input features were important for the discriminator's classification. We multiplied the xAI output with the gradient to give more weight to the important features.

One of the most widely used GAN based imputation methods is called GAIN, Generative Adversarial Imputation Net [24]. In this model, the discriminator outputs a value for each feature that indicates whether the value is real or imputed. Normally, in a GAN, the discriminator would output a single value for an input, but in GAIN, the discriminator output length is equal to the number of features. The discriminator is also given a "hint" matrix which provides it with additional information about the missingness of the input.

Machine learning models have been successfully applied to the problem of ECG signal denoising. Specifically, RNN models and autoencoders have been used for ECG desnoising and outperformed traditional filtering methods for denoising [19]. In Chapter 3 of this thesis, we apply the cyclegan architecture to the problem of ECG denoising.

Problem Statement: In this thesis, I address the following question *Is it possible to use Machine Learning for the generation, imputation, denoising and translation of ECG data?*

1.2 Contributions

The main contributions of this thesis are the following:

1. **A novel method for ECG translation, imputation and denoising:** We address the problem of low quality ECG signals due to missing data and noisy data. We also address the problem of imbalanced data and not having enough data from one class. We combine a cycleGAN architecture with a LSTM generator and discrminator and use Wasserstein loss for the adversarial loss. We show that this model can be used to transform an ECG signal from one class to another. We demonstrate the effectiveness of this model as an imputation tool and for denoising a ECG signal that has noise added. We evaluate the imputation and denoising result against state-of-the-start methods
2. **A novel algorithm for ECG Generation:** We develop a novel algorithm for ECG generation that leverages GANs, explanation AI systems, and Wasserstein Loss function. We evaluate the results of our experiments using multiple metrics including dynamic time warping, Pearson's correleation coefficient, qualitative analysis and augmenting a state-of-the-art classifier with synthetic examples.

1.3 Thesis Organization

The rest of this thesis is organized as follows:

1. Chapter two covers **background** information necessary to understand the rest of the material covered. In particular, we describe the morphology and important features of an ECG signal. We then provide information on Generative Adversarial Networks (GANs), improvements to GANs, explanation AI systems and cycleGAN.

2. Chapter three covers a novel application of cycleGAN for the translation, imputation and denoising of ECG data. We use a LSTM generator and discriminator and use wasserstein loss as the adversarial loss. In this chapter, we present a detailed overview of the cycleGAN architecture. We show the results after increasing the percent of missing data from 10% missing data to 50%. We evaluate the performance of cycleGAN against several other imputations methods by comparing the root mean squared error, mean absolute percent error and by comparing AUC scores after augmenting a classifier using imputed data from both of these approaches. We compared cycleGAN as a denoising tool against several state-of-the-art approaches and evaluated the results using multiple metrics. We also use cycleGAN to translate normal class ECGs to arrythmias and evaluate cycleGAN as a translation tool.

3. Chapter four covers generating high quality ECGs by using explanation AI systems feedback and Wasserstein GAN with an interpolation method of imposing gradient penalty. In this chapter, I test a CNN and LSTM architecture augmented by 4 different explanation AI systems. I explore multiple evluation metrics. I report on experiments where the explanation augmented GAN outperforms standard GAN. I discuss how the xAI feedback can be used to generate high quality medical datasets especially when data is scarce.

4. Finally, chapter 5 concludes the thesis summarizing the contributions of this work. I also identify future research in the area of using GANs to generate medical data.

Chapter 2

Background

2.1 ECG

An Electrocardiogram (ECG) is an electrical recording of the heart that shows the depolarization and repolarization of cardiac muscle cells during cardiac cycles. ECGs are a common diagnostic and monitoring tool for arrhythmias and cardiac abnormalities. The ECG is the summed electrical signal from many cells.

An ECG cardiac cycle shown in Figure 2.1 has the following features: The P wave corresponds to depolarization of the atria which starts the mechanical contraction of the atria. The QRS complex corresponds to ventricular depolarization and the T wave corresponds to ventricular repolarization. Typically, the morphology of the ECG signal is used to diagnose abnormalities [21], and in this thesis, qualitative evaluation is an important evaluation tool.

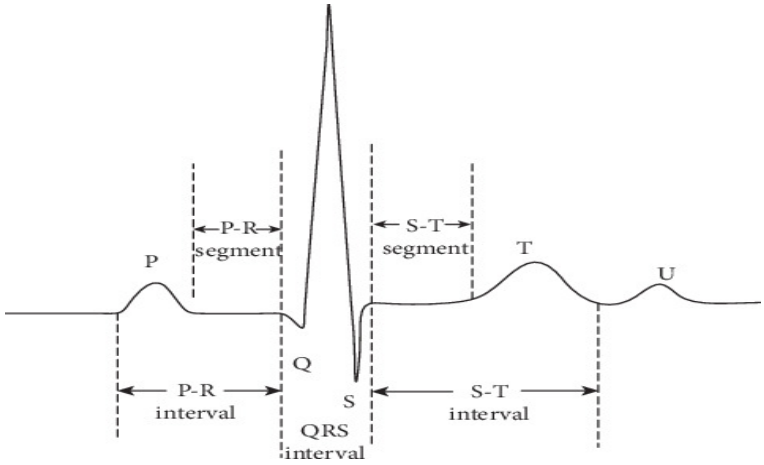


Figure 2.1: Classic Labeled ECG

2.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) consist of two neural networks, a generator and a discriminator. The goal of the generator is to learn a dataset's probability distribution and produce data that are identical to that distribution. A discriminator is a binary classifier that outputs whether an input is from the training set or an output from the generator. The generator and discriminator are trained together. The generator and discriminator play a minimax game where the discriminator tries to maximize this objective function while the generator tries to minimize it [1].

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data(x)}}\ [logD(x)] + \mathbb{E}_{z\sim p_z(z)}\ [log(1-D(G(z)))]$$

2.3 Wasserstein GAN

One of the main problems of GAN training is that careful balance must be maintained between discriminator and generator training. If the discriminator is significantly better than the generator, the output of the discriminator will always be close to 0 or 1 and will not provide meaningful feedback to the generator. The loss function will output a value close to 0 and result in a 0 gradient where the generator will not learn. Conversely, if the generator is more powerful than the discriminator and learns how to fool the discriminator, the generator can fail to learn the entire data distribution and only produces a small subset of the training sample. This is known as mode collapse.

In a Wasserstein GAN (WGAN) architecture, the Wasserstein distance is used to calculate the difference between the generator data distribution and the real data distribution [2]. The Wasserstein loss function gives smoother gradients and allows greater stability of GAN training. The discriminator wants to maximize this expression and the generator wants to minimize the expression. The Wasserstein loss is given below.

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{f \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta} [D(x)]$$

In a WGAN architecture, the discriminator is replaced by a critic. The critic does not use a sigmoid function at the end which does not limit the output to be between 0 and 1. A constraint is placed on the critic to be Lipschitz continuous. This imposes the restriction that the norm of the gradient of the critic should be less than or equal to 1. In the WGAN paper, the authors use gradient weight clipping to enforce the Lipschitz constraint. After each gradient update, the weights are clamped to a fixed box. For example, the weights would be between [-0.01, 0.01].

2.4 Improved Wasserstein GAN Training

In the original Wasserstein GAN paper, the authors state that weight clipping is a terrible way to enforce the Lipschitz constraint. If the weight clipping parameter is too large, it takes a long time to train the GAN and makes training to optimality hard. If the weight clipping parameter is too small, this can lead to vanishing gradients when the number of layers is large or in cases where batch normalization is not used such as in RNNs and LSTMs. In the paper titled "Improved Training of Wasserstein GANs", Gulrajani et al. show that a better way to enforce the Lipschitz constraint is by inputting an interpolated image to the critic and taking the norm of the gradient of the interpolated image [9].

2.5 Explanation AI (xAI) Systems

Neural network classifiers are often treated as a black box where the user does not know which input features were important for a classifier's output. Explanation AI systems attempt to show the user which elements of an input feature were the most important to the classifier in making its decision. In this thesis, we explore several xAI systems as outlined below.

1. **Adversarial Explanations for AI systems (AXAI)** An adversarial attack occurs when a small change in the input causes a DNN to misclassify the image. In AXAI,

the projected gradient descent method (PGD) is used to find an adversarial example. Pixels that have not been changed much are filtered out. The Quickshift method is used to segment the input. The segments with the highest altered inputs are used.

2. **Integrated Gradients** Integrated gradients compute the integral of the gradient of the discriminator's output to its input features along the path from a baseline to the input.

3. **Saliency** Saliency returns the gradient of the discriminator output with respect to the input. Saliency takes a first-order Taylor expansion of the network for a given input. The gradients are the coefficients of each feature in the linear representation of the model. These coefficients represent the importance of the feature.

2.6 CycleGAN

CycleGAN is another class of GANs that was originally developed for image-to-image translations [26]. The cycleGAN architecture consists of two generators G_1 and G_2 and two discriminators D_1 and D_2. G_1 learns the mapping function $G_1 : X \rightarrow Y$ where X and Y are two different domains, like paintings and photographs. G_2 simultaneously learns the reverse mapping functions $G_2 : Y \rightarrow X$. Instead of inputting a random noise vector into the generator, an image is input and the generator transforms the image from one domain into another. CycleGAN has 3 loss functions, the adversarial loss from the discriminator, the cycle consistency loss, and an identity loss. The cycle consistency loss imposes the constraint that $G_2(G_1(x)) \sim x$ and $G_1(G_2(y)) \sim y$ and the identity loss $G_1(y) \sim y$ and $G_2(x) \sim x$.

The cycleGAN architecture and forward and backward cycle-consistency loss are shown in Figure 2.2. The cycleGAN architecture works on unpaired images where no information is provided on which x_i matches which y_i.

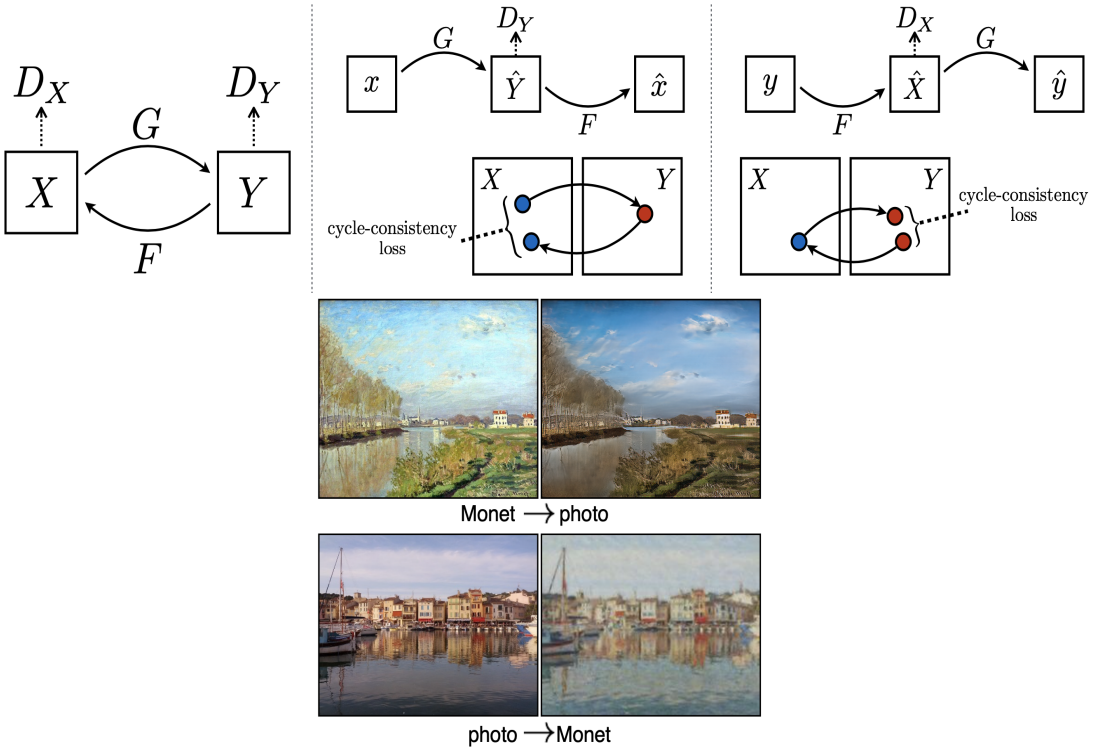


Figure 2.2: cycleGAN architecure and image translation example

Chapter 3

cycleGAN for time series medical data translations, imputation and denoising

3.1 Introduction

There have been several applications of cycleGAN in the medical domain, including image translations from CT to MRI [12], image denoising and enhancement [25], and image data imputation [20]. The architecture used in the original cycleGAN [26] paper consisted of several convolutions and residual blocks. We used this model to translate CT images to MRI and vice versa to illustrate the function of cycleGAN and to visually examine its feasibility in the translation of medical image data. The visual results of this experiment are shown in 3.1.

In the medical setting, one class of data may be more abundant than another. In the ECG setting, normal heartbeats are more abundant than arrhythmias. For the MIT-BIH database described below, the normal (N) class heartbeats has 90632 samples compared to a type of arryhmias called the fusion (F) class of heartbeats that has only 803 samples.

In practice, ECG datasets often have missing values due to faults and distortions [23]. ECG data are equispaced data, meaning that the time increments between successive data points are equal. $|t_1 - t_2| = |t_2 - t_3| = |t_n - t_{n-1}|$. Although the ECG data are periodic, we are considering one heartbeat at a time and in this case the signal is not considered periodic. Many imputation techniques rely on inter-attribute correlations, which makes

univariate time-series data more challenging than multivariate. In the univariate case, we cannot look at the relationship among variables to do imputation.

There are three main types of missing values: data that is missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR). In MCAR, missing data points occur completely at random. This means that the probability of a certain observation being missing is independent of the point of time of the observation. In the case of MAR, the probability that an observation is missing is dependent on the point of time in the observation of the series. One example of MAR is sensor data where the sensor malfunctions for a period of time resulting in missing data points spanning several time steps. In NMAR, the probability of an observation being missing depends on the value of the observation [13].

Imputation methods for univariate time series data fall into three broad categories: univariate algorithms, univariate time series algorithms, multivariate algorithms on lagged data [13]. Examples of univariate algorithms are mean, median, and mode. These do not consider the temporal relationship of the data. Univariate time series algorithms like last observation carried forward (locf) and linear inteprolation take into account the time series characteristic of the data. Multivariate algorithms consider time to be an implicit variable in the data and add time information as covariates.

Noise in an ECG signal can come from multiple sources, with the most common causes being baseline wander, power-line interference and muscle artefacts [3]. Baseline wander are low-frequency disturbances that come from poor contact of the electrode with the skin, body movements, or respiration. Power line interference is high-frequency noise caused by the circuity. Typically a high-pass filter is used to remove baseline wander and a low-pass filter is used to remove power line interference. Muscle artefacts are caused by electrical activity in the muscles.

There are several ECG denoising techniques that fall into five main categories: empirical mode decomposition (EMD), deep learning based autoencoder models, wavelet based methods, adaptive filtering and bayesian filtering.

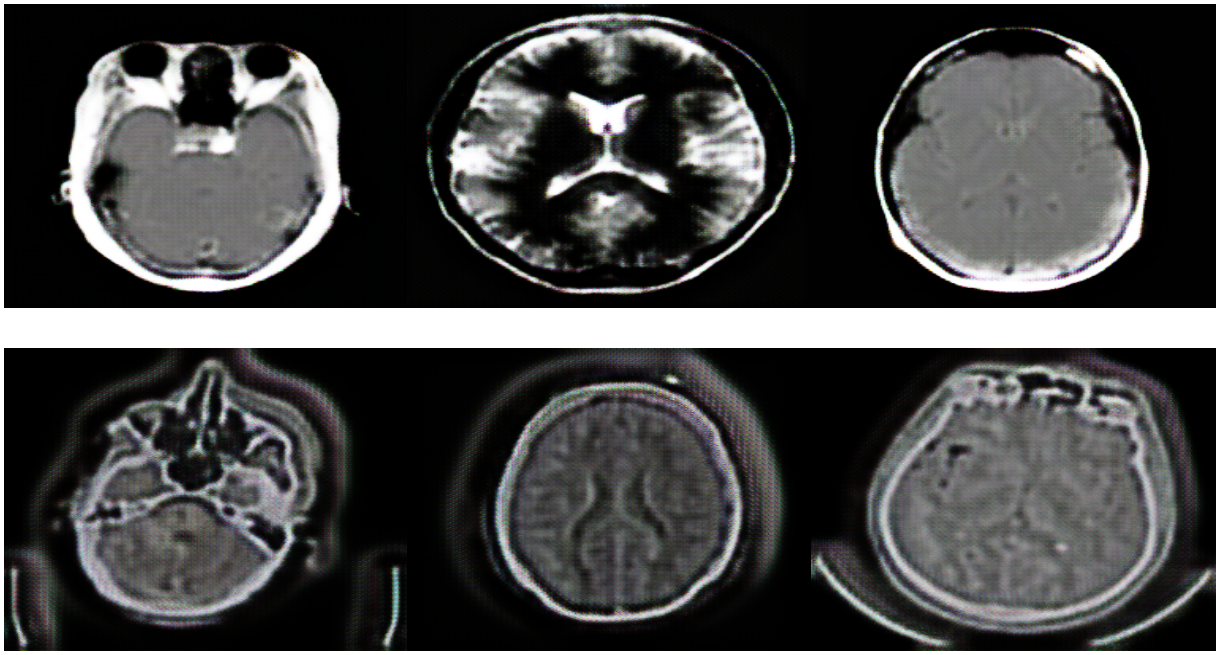


Figure 3.1: CT (top) to MRI (bottom) Image translation using original model architecture

3.1.1 Contributions

1. **CycleGAN for univariate time series data translation** Integration of a CycleGAN architecture with a LSTM generator and discriminator and Wasserstein Loss to translate univariate time series data from one class to another

2. **CycleGAN for time series data imputation** We show that cycleGAN is an effective method for imputation of univariate time series data

3. **CycleGAN as a denoising method for ECG** Use cycleGAN as a novel method to denoise ECG

3.2 Experimental Setup and Evaluation Metrics

The imputation and translation experiments were performed on ECG data from the MIT-BIH arrhythmia database. This is considered to be the gold standard for ECG classification

tasks [8]. The database contains 48 half-hour ECG recordings from 47 individuals. The sampling frequency is 360 Hz. There are 5 classes of heartbeats: Normal beats (N), supraventricular ectopic beats (S), ventricular ectopic beats (V), fusion beats (F), and unknown beats (Q). For this project, we have omitted the unknown beats. The number of samples for each heartbeat class is shown in table 3.1. Each heartbeat is represented as a vector of 1x216 where each element is a millivolt at a time step.

For the denoising experiments, we use the physionet QT database which has 105 ECG signals. To add real baseline noise to the ECGs from this dataset, we follow the procedure used by [19]. The MIT-BIH Noise Stress Test Database (NSTDB) has noise-containing recordings. The ECG recordings from the QT database are contaminated with noise from the NSTDB. To compare against other methods, for the denoising experiments, we used an ECG that has 516 time steps.

We used the same cycleGAN architecture for the time series translation and imputation experiments. The generators are a 2 layer bi-directional LSTM network with 50 hidden dimensions followed by a final linear layer. The two-layer LSTM has a dropout value of 0.2. The discriminators consist of a 2-layer bi-directional LSTM network with dropout = 0.2 followed by a linear layer and sigmoid activation function. We used the Adam optimizer with a learning rate of 1e-3 for both generators and discriminators. For the denoising experiments, we changed the LSTM input dimension to 516 and used a hidden dimension of 100. We used the Adam optimizer with a learning rate of 1e-3, batch size 64 and ran for 50 epochs.

We gave the adversarial loss, forward and backward cycle consistency loss an equal weight of 1 and multiplied the identity loss by 10. For the adversarial loss, we used the wasserstein loss function. For the identity loss and cycle consistency loss, we used L1 loss. The interpolation method of gradient penalty was used and we multiplied the gradient penalty by a factor of 10 and summed it with the wasserstein loss.

Heartbeat class	N	S	V	F	total
Total Set	90632	2779	7235	803	101449
Train Set	45868	942	3787	415	51020
Test Set	44258	1837	3221	388	49711

Table 3.1: Partition of the MIT-BIH dataset

3.2.1 Imputation Experiments

For the imputation experiments on the ECG dataset, we used the 'V' class heartbeats because there were over 7000 examples of this type which allowed us to have a sufficiently large train and test set. We changed the missing data rate from 10% to 50% increasing in increments of 10. We tested our cycleGAN approach against 4 other imputation techniques: Last observation carried forward (locf), moving window, kNN, and cubic spline interpolation. We used the impyute package and the scipy interpolation function. To simulate MCAR, we randomly set data points to zero. To simulate MAR, we selected a point at random and set the next several consecutive points to 0. We started by randomly selecting an index and setting the next 22 indices, representing 10% of one ECG signal, to 0. We repeated this process up to 50% of the signal.

To evaluate the performance of cycleGAN for imputation, we looked at root mean squared error (RMSE), mean absolute percent error (MAPE) and AUC scores. RMSE is the root of the squared difference between the original signal and denoised signal or between the original signal and imputed signal. MAPE is the absolute difference between the original and denoised signal expressed as a percent of the original signal. We use a deep residual convolutional neural network for ECG classification. The network was first proposed by [11] and also used by [8]. The classifier uses the RMSProp optimizer with a learning rate of 1e-3 and cross entropy loss function. For each experiment, we trained the classifier for 10 epochs using batch size of 64. The AUC scores were calculated by training the classifier on the real dataset and testing on the imputed signal.

3.2.2 Denoising Experiments

To evaluate the denoising results, we used three metrics, sum of the square of distances (SSD), maximum absolute difference (MAD), percentage root-mean-square difference (PRD).

The metrics for imputation and denoising are defined below:

1. RMSE $(s_1, s_2) = \frac{1}{N}\sum_{n=1}^{N}\sqrt{(s_2(n) - s_1(n))^2}$
2. MAPE $(s_1, s_2) = \frac{|s_1(n)-s_2(n)|}{s_1} * 100$
3. SSD$(s_1, s_2) = \sum_{n=1}^{N}(s_2(n) - s_1(n))^2$
4. MAD$(s_1, s_2) = max|s_1(n) - s_2(n)|$
5. PRD$(s_1, s_2) = \sqrt{\frac{\sum_{n=1}^{N}(s_2(n)-s_1(n))^2}{\sum_{n=1}^{N}(s_2(n)-\bar{s_1})^2}}$

3.2.3 Translation Experiments

We tested the transformation from the N class heartbeats to F, S, and V. We evaluated the results qualitatively and measured the increase in classification accuracy after adding synthetic examples from cyclegan to the training set.

The method we use for the experiments is described by algorithm 1.

Algorithm 1 CycleGAN pseudocode for translation, imputation and denoising

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration, n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2

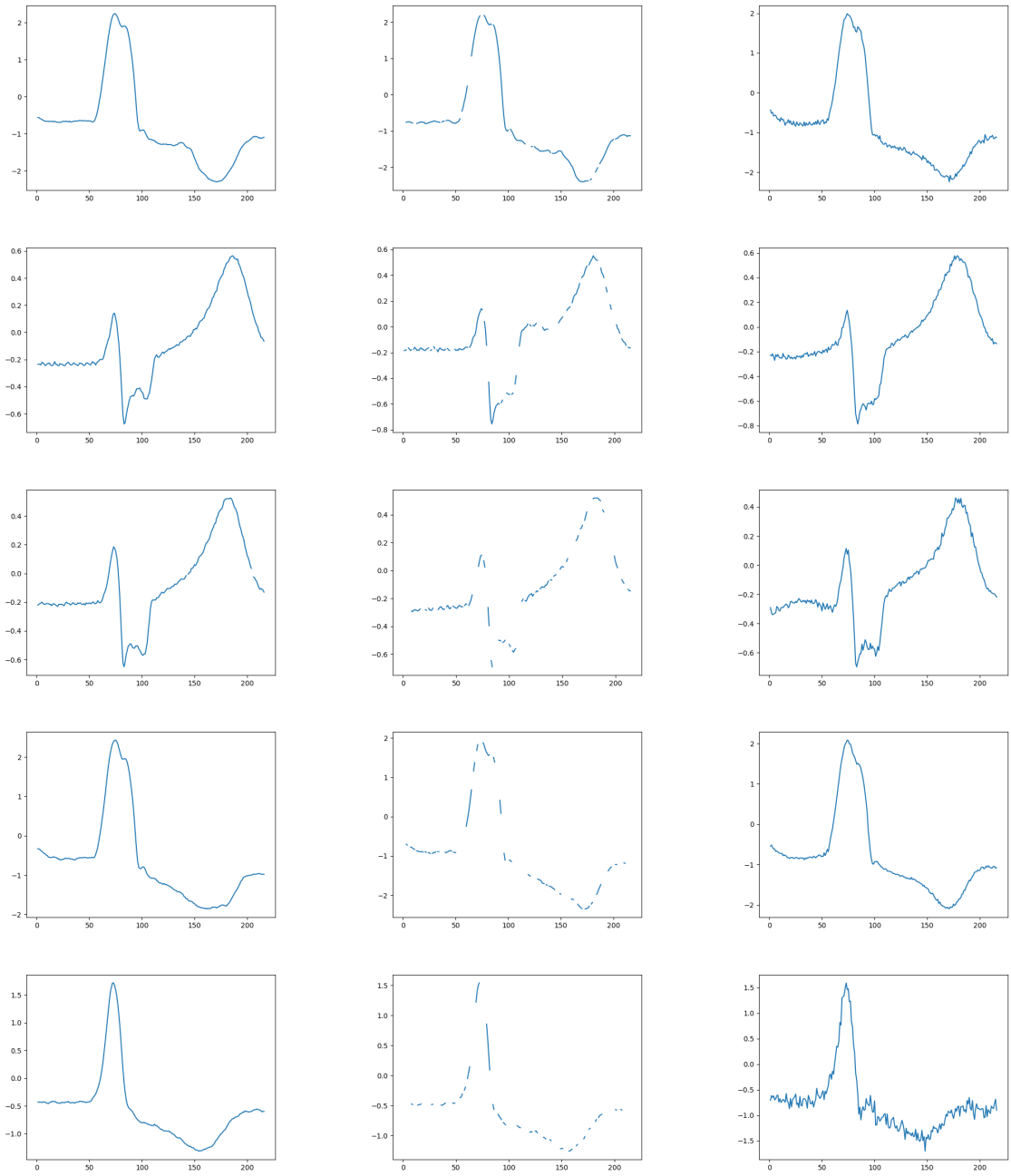
Require: initial critic parameters w_0, initial generator parameters θ_0

while θ has not converged do **do**
  for $t = 1, \ldots, n_{critic}$ **do**
    for $i = 1, \ldots, m$ **do**
      Sample from domain X, sample from domain Y, a random number $\epsilon U[0, 1]$.
      $\tilde{x} \leftarrow G_{y \to x}(y)$
      $\tilde{y} \leftarrow G_{x \to y}(x)$
      $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$
      $\hat{y} \leftarrow \epsilon y + (1 - \epsilon)\tilde{y}$
      $L_1 \leftarrow D_x(\tilde{x}) - D_x(x) + \lambda(||\nabla_{\hat{(x)}} D_x(\hat{x}||_2 - 1)^2$
      $L_2 \leftarrow D_y(\tilde{y}) - D_y(y) + \lambda(||\nabla_{\hat{(y)}} D_y(\hat{y}||_2 - 1)^2$
    end for
    Update the discriminators
    $w \leftarrow Adam(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L_1, \frac{1}{m} \sum_{i=1}^{m} L_2, w, \alpha, \beta_1, \beta_2)$
  end for
  Compute the Adversarial Loss
  $L_A = \frac{1}{m} \sum_{i=1}^{m} (D_x(G_{y \to x}(y^i)) + \sum_{i=1}^{m} (D_y(G_{x \to y}(x^i))$
  Compute the Cycle Consistency Loss
  $L_C = \frac{1}{m} \sum_{i=1}^{m} (x^i - G_{y \to x}(G_{x \to y}(x^i)))^2) + \frac{1}{m} \sum_{i=1}^{m} (y^i - G_{x \to y}(G_{y \to x}(y^i)))^2)$
  Compute Identity Loss
  $L_I = \frac{1}{m} \sum_{i=1}^{m} (x^i - G_{y \to x}(x) + \frac{1}{m} \sum_{i=1}^{m} (y^i - G_{x \to y}(y))$
  update the generators
  $w \leftarrow Adam(L_A, L_C, L_I, \alpha, \beta_1, \beta_2)$
end while

3.3 Results on ECG imputation

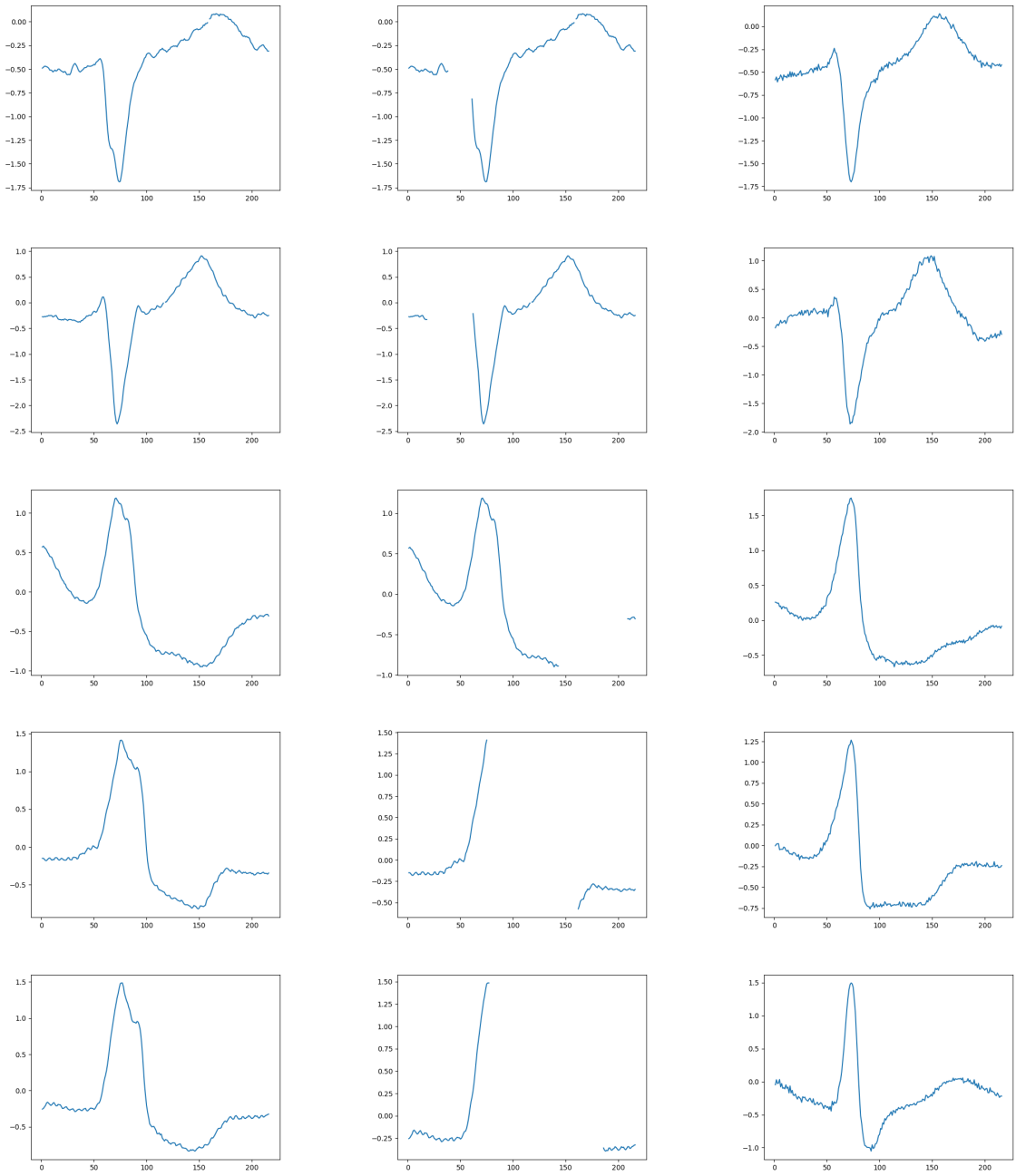
		MCAR				MAR	
	%	**AUC**	**RMSE**	**MAPE**	**AUC**	**RMSE**	**MAPE**
LOCF	10	0.9518	0.0036	1.67	0.943	0.0296	11.11
	20	0.9452	0.0077	3.66	0.9212	0.0966	29.6
	30	0.9403	0.0133	6.21	0.9019	0.1913	54.04
	40	0.9342	0.0201	9.54	0.8768	0.2878	75.16
	50	0.9258	0.0296	13.83	0.8413	0.3981	103.91
Moving Window	10	0.9818	0.0017	0.72	0.954	0.0284	10.01
	20	0.9797	0.004	1.73	0.9367	0.0962	31.83
	30	0.9564	0.0076	3.34	0.9257	0.1875	54.39
	40	0.9517	0.0128	5.777	0.8974	0.2863	75.61
	50	0.9476	0.021	9.489	0.8622	0.3972	103.72
kNN	10	0.6205	0.0164	4.97	0.9462	0.0365	10.06
	20	0.5248	0.0428	12.95	0.9305	0.0917	24.33
	30	0.5402	0.1256	36.76	0.9175	0.1477	38.34
	40	0.4057	0.126	35.72	0.8983	0.2053	53.88
	50	0.4531	0.1823	52.6	N/A	N/A	N/A
Interpolation	10	0.5916	0.0362	11.169	0.4034	0.6164	98.7
	20	0.5838	0.0731	20.87	0.4089	0.6175	98.99
	30	0.4292	0.186	29.93	0.4151	0.6174	98.93
	40	0.4201	0.2462	38.28	0.4067	0.6177	99
	50	0.391	0.3089	46.89	0.4073	0.6167	98.96
CycleGAN	10	0.8958	0.0101	3.36	0.9592	0.0141	8.21
	20	0.9021	0.0234	7.68	0.9596	0.0261	12.21
	30	0.9264	0.0384	12.19	0.9564	0.0509	23.47
	40	0.9404	0.055	16.67	0.9126	0.0872	43.48
	50	0.9696	0.0824	26.75	0.9068	0.1104	48.55

Table 3.2: Imputation Results



(a) Original Heartbeat (b) Missing Data (c) Imputation result

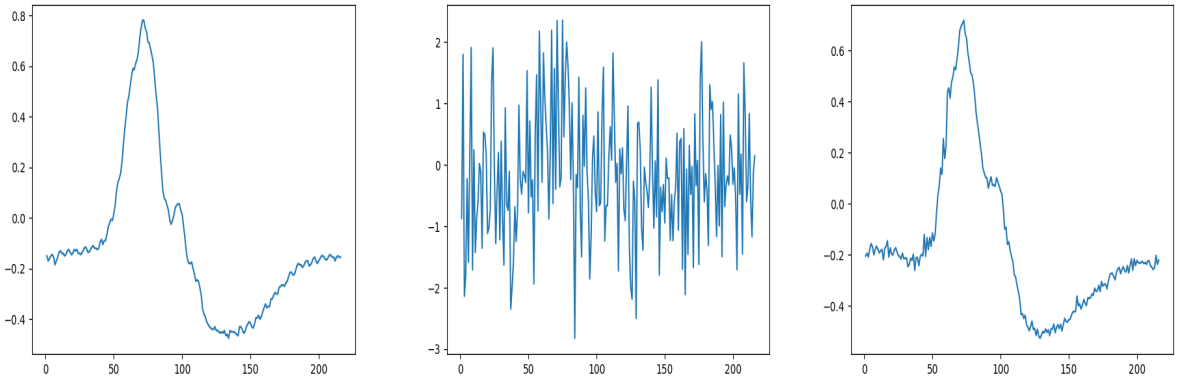
Figure 3.2: Imputation Results MCAR 10 to 50 %



(a) Original Heartbeat (b) Missing Data (c) Imputation Result

Figure 3.3: Imputation Results MAR 10 to 50 %

3.4 Results on ECG denoising

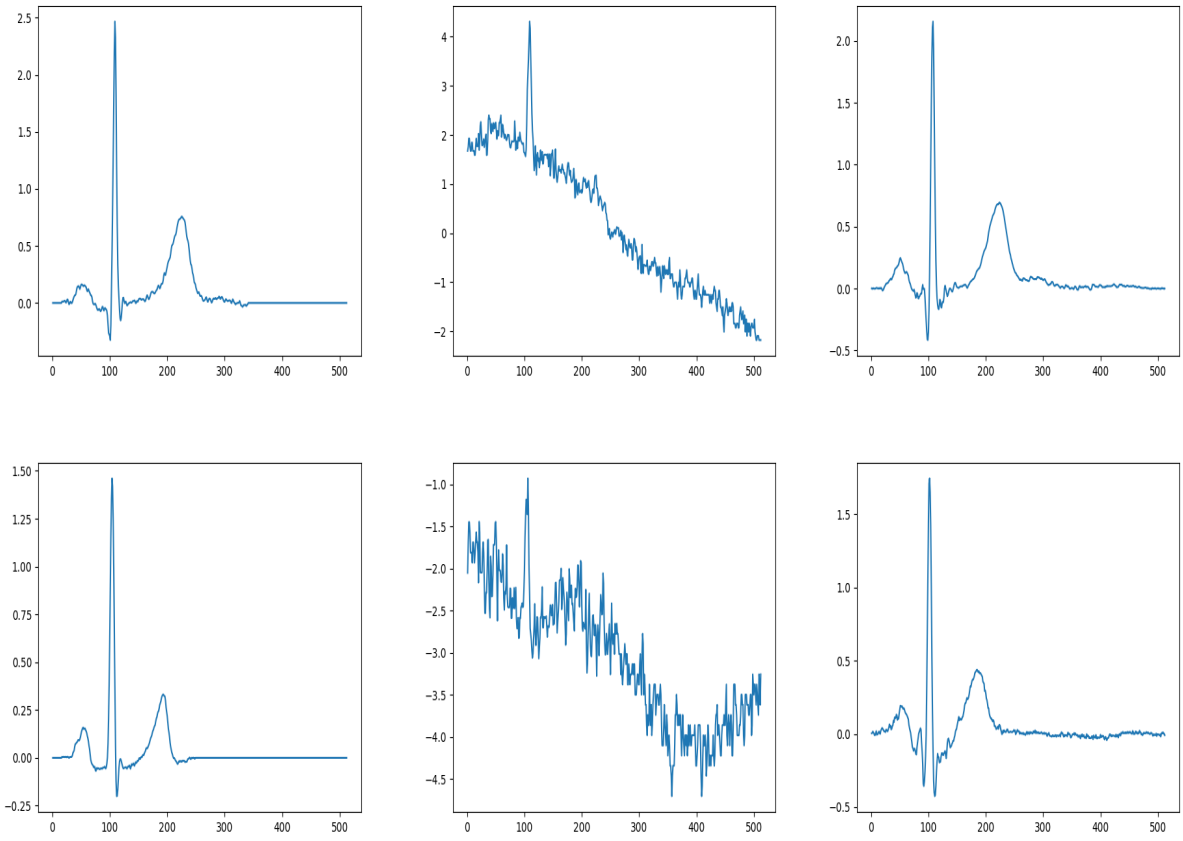


(a) Original Heartbeat (b) Gaussian Noise Added (c) Denoising Result

Figure 3.4: ECG denoising gaussian noise using cycleGAN

Method/Model	SSD	MAD	PRD
FIR filter	44.97	0.69	65.77
IIR Filter	35.63	0.62	61.62
DRNN	5.85	0.44	49.91
FCN-DAE	6.79	0.48	62.18
Vanilla L	13.565	0.54	88.47
Vanilla NL	6.9	0.41	63.55
Multibranch NL	5.362	0.39	55.59
cycleGAN	5.1391	0.2394	51.1849

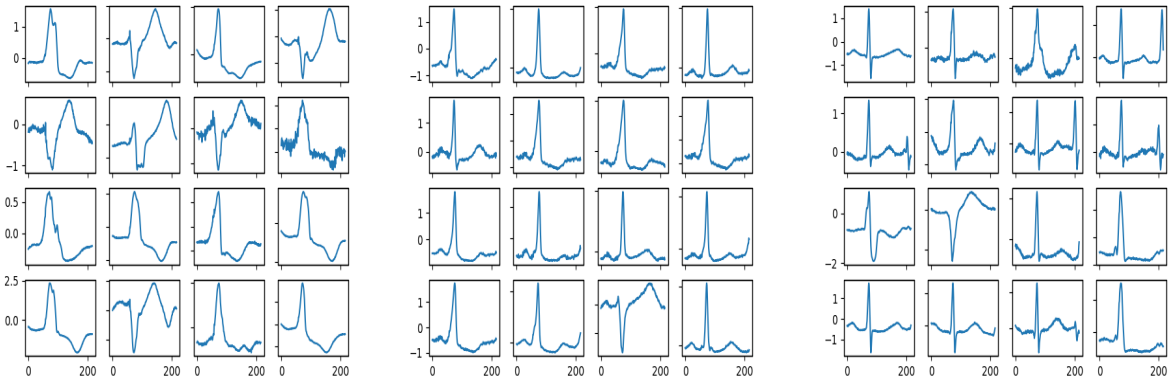
Table 3.3: Quantitative Denoising Results



(a) Original Heartbeat (b) Noise from NSTDB (c) Denoising Result

Figure 3.5: Denoising noise from NSTDB using cycleGAN

3.5 Results on ECG translation from normal class to arrythmias



(a) N → V transformation (b) N → F transformation (c) N → S transformation

Dataset class	Original Dataset	cycleGAN examples added
N	99.3%	88%
S	38%	88.4%
V	77%	87.4%
F	23%	81.7%

Table 3.4: Accuracy Results after adding cycleGAN results

3.6 Analysis of Results

Table 3.2 shows the AUC scores, rmse and mape results after imputation with various techniques. We show the results for imputation after missing values being added at random and the results for imputation with extended gaps. In the case of adding MCAR gaps, locf and moving window outperformed cycleGAN. In the case of MAR, cycleGAN far outperformed all existing methods in all metrics.

In Figure 3.2, the left column is the original heartbeat, the middle column shows the heartbeat with the missing data ranging from 10% to 50% and the right column shows the results after imputation with cycleGAN. Figure 3.3 shows the results when longer gaps are introduced in the data. In this figure, the middle column shows the longer gaps. Visually, we see that in both cases, cycleGAN is an effective imputation tool and can reconstruct the ECGs in cases of missing data. The results show that the optimal imputation tool depends on the distribution of the missing data. When indices are drawn from a normal

distribution and set to NULL, moving window is the best choice for imputation. When there are longer gaps present in the data, cycleGAN is the most effective imputation tool. Our experimental results show that cycleGAN can generate high quality ECGs and outperforms existing methods as an imputation tool.

Figure 3.5 shows that cycleGAN is an effective denoising tool that can remove baseline wander noise. The quantitative results shown in 3.3 demonstrates that cycleGAN outperforms state-of-the-art methods.

For both the imputation and denoising experiments, cycleGAN learned how to impute and denoise the data but did not learn the reverse function of removing data and adding noise. For these 2 experiments, cycleGAN gave the best results when we gave more weight to the identity loss compared to the cycle consistency loss. This is in contrast to the original image translation experiments where more weight was given to the cycle consistency loss.

Our results show that cycleGAN can be used to translate ECGs from the normal class to the different arrythmia classes. As shown in Table 3.4, we significantly improved the clssification accuracy of the F and S class heartbeats by adding synthetic examples produced by cycleGAN. The qualitative results show that cycleGAN produced smooth F class and S class heartbeats without mode collapse. We note that for the N to V translation, some of the signals are noisy and the P wave and the T wave are not visible in some signals.

3.7 Future Work

ECG denoising is a critical preprocessing step in many applications and affects further downstream analysis. In this work, we used cycleGANs as a novel method to do ECG denoising. Our experimental approach was to add baseline wander noise to the ECGs from the QT database. In future work, we want to simulate other sources of noise and test our method as a denoising method for other kinds of noise.

One future direction is to combine the approach from a state-of-the-start multivariate imputation tool called GAIN: Generative Adversarial Imputation Nets [24]. In this approach, the discriminator outputs a value for each time step indicating whether the value is real or fake. This is in contrast to the current approach where the discriminator outputs a single value indicating whether the heartbeat is real or fake. By giving us a value for each time step, this gives us more feedback from the discriminator and might improve generator learning.

Chapter 4

GAN based generation of univariate time series medical data

4.1 Introduction

Generative Adversarial Networks (GANs) have been used extensively in the field of image generation. The use of GANs for time series data is a more recent development. Previous methods that use GANs for ECG generation note that GAN training is highly unstable and lacks suitable evaluation measures.

In this chapter, we propose a new method of ECG generation described in Algorithm 2 that we call xai-wgan. We used the wassertsein loss function and imposed the Lipschitz constraint by using the interpolation method of gradient penalty. We trained the GAN for 50 epochs and during the last 10 epochs of training, we added the xAI feedback by multiplying the xAI output with the generator gradient.

4.1.1 Contributions

1. **ECG generation using Wasserstein GAN with gradient penalty and xAI Feedback** A novel way of generating ECGs that combines Wasserstein GAN with gradient penalty, xAI feedback

2. **Explanation AI with univariate time series medical data** A key contribution of this chapter is the incorporation of xAI systems with time series data. Several different types of xAI are used to explain the discriminator's prediction.

3. **Evaluation Methods** We evaluated synthetic ECGs using multiple metrics, including time series similarity, Pearson's's correlation coefficient, data utility, qualitative analysis and timing.

4.2 Method

Algorithm 2 WGAN with gradient penalty and xAI Feedback. We use default values $\lambda = 10$, $n_{critic} = 5$, $\alpha = 0.001$, $\beta_1 = 0$, $\beta_2 = 0.9$

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration, n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2, explanation matrix M, weight of XAI system δ

Require: initial critic parameters w_0, initial generator parameters θ_0

while θ has not converged do **do**
  for $t = 1, \ldots, n_{critic}$ **do**
    for $i = 1, \ldots, m$ **do**
      Sample real data x $\mathbb{P}_r$, latent variable z $p(z)$, a random number $\epsilon U[0, 1]$.
      $\tilde{x} \leftarrow G_\theta(z)$
      $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$
      $L^{(i)} \leftarrow Dw(\tilde{x}) - Dw(x) + \lambda(||\nabla_{\hat{(x)}} D_w(\hat{x}||_2 - 1)^2$
    end for
    $w \leftarrow Adam(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of latent variables
  $w \leftarrow SGD(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2, \nabla_w * M * \delta)$
end while

4.2.1 Experimental Setup

For our experiments, we try to generate F class heartbeats because of the low availability of this class of heartbeat. Some examples of F class heartbeats from the MIT-BIH database are shown in 4.1.

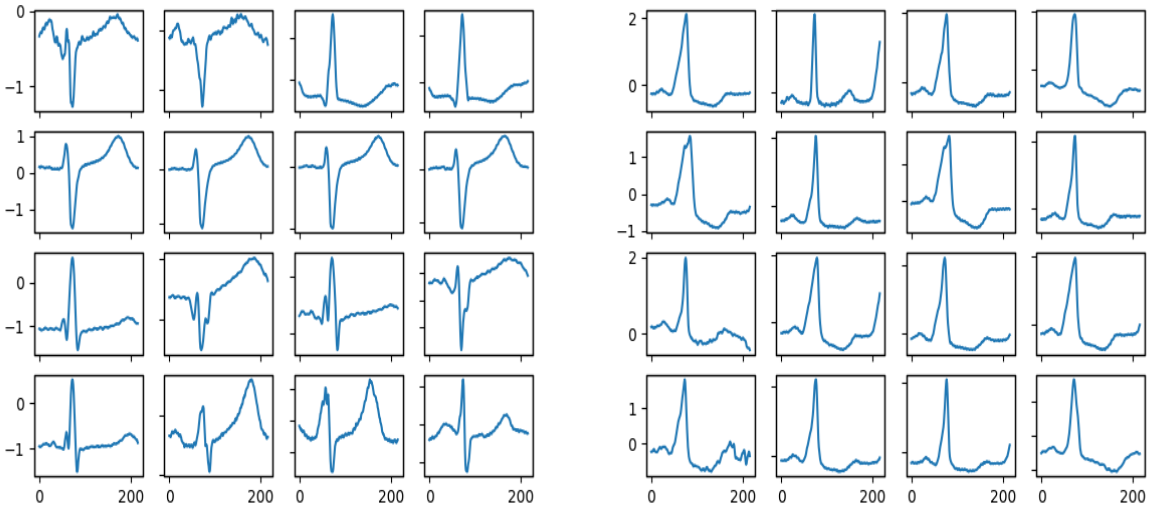


Figure 4.1: F class heartbeats from MIT-BIH Heartbeats

In this setting, a learning rate of 1e-3 gave optimal results. In order to get better results using xAI systems, we used the SGD optimizer. The Adam optimizer makes its own changes to the gradients and we found was suppressing the effects of the xAI system.

We used the same classifier used in Section 3 to test our ECG generation method. We measured accuracy scores for the F class beats after adding 1000, 3000, 10000 and 150000 synthetic examples. The results are shown in 4.1.

Following the results of Vineel et al.'s paper, we used the Saliency xAI system as it was found to have the best performance. We also tested integrated gradients due to ease of implementation. The final xAI system that we tried is adversary AI. This is a completely new approach to xAI and we wanted to test it in the time series domain. Figure 4.2 and 4.3 visually shows the results of the xAI systems.

We ran all the experiments for 50 epochs. We incorporated xAI feedback into the generator training in the last 10 epochs. This allowed the discriminator to be well trained so that the xAI feedback was meaningful.

All our experiments were run on Compute Canada.

4.3 Experimental Results

4.3.1 Evaluation Criteria

Based on a thorough literature survey of metrics [7] for the time series domain, we use the following criteria to evaluate the quality of time series data generated.

1. **Dynamic Time Warping (DTW):** DTW is a common metric for evaluating time series data. DTW optimally aligns two time series along the temporal axis to measure the distance. DTW can be viewed as the minimum cost to align two time series data. A lower DTW value indicates higher similarity between two time series datasets.

2. **Pearson's's Correlation Coefficient:** The Pearson's's Correlation coefficient ranges from +1 to -1 where +1 indicates that two time series are perfectly correlated while -1 signifies that they are inversely correlated.

3. **Augment classifer using synthetic data:** We augment training data for a ECG classifier by adding the synthetic ECGs and measure accuracy scores.

4. **Qualitative Analysis:** We visually inspect the results after each epoch and report the results at the end of the training phase. ECGs are visual tool where the morphology of the signal is used to make a diagnosis. The visual quality of the ECG signal is an important evaluation metric.

5. **Timing Analysis** Timing data for all experiments is provided. We want to measure the additional time taken for the Wassertein architecture and the cost of the xAI systems.

4.3.2 Visualization of xAI outputs

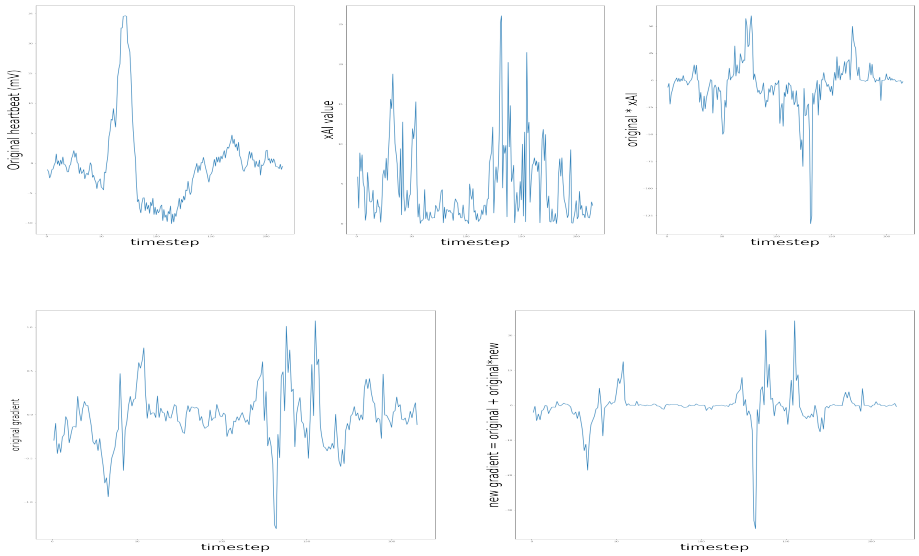


Figure 4.2: Saliency xAI output CNN

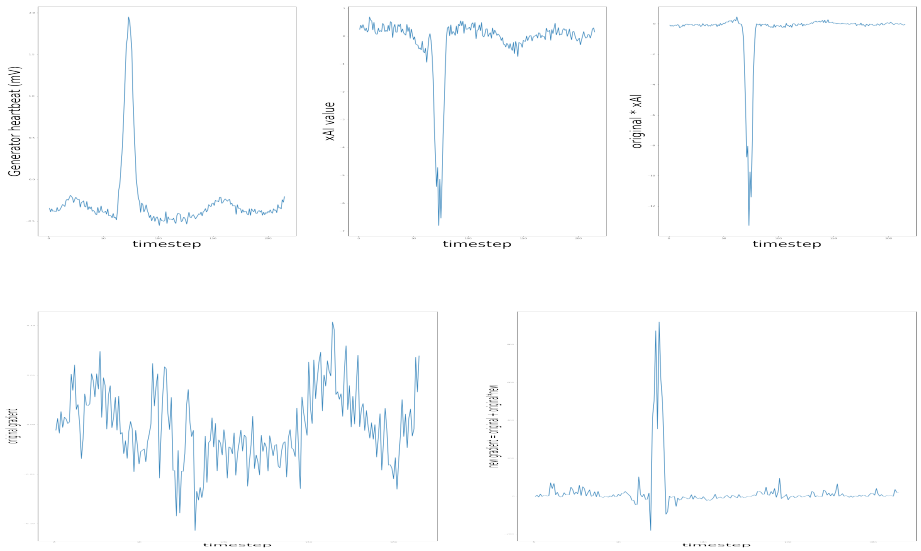
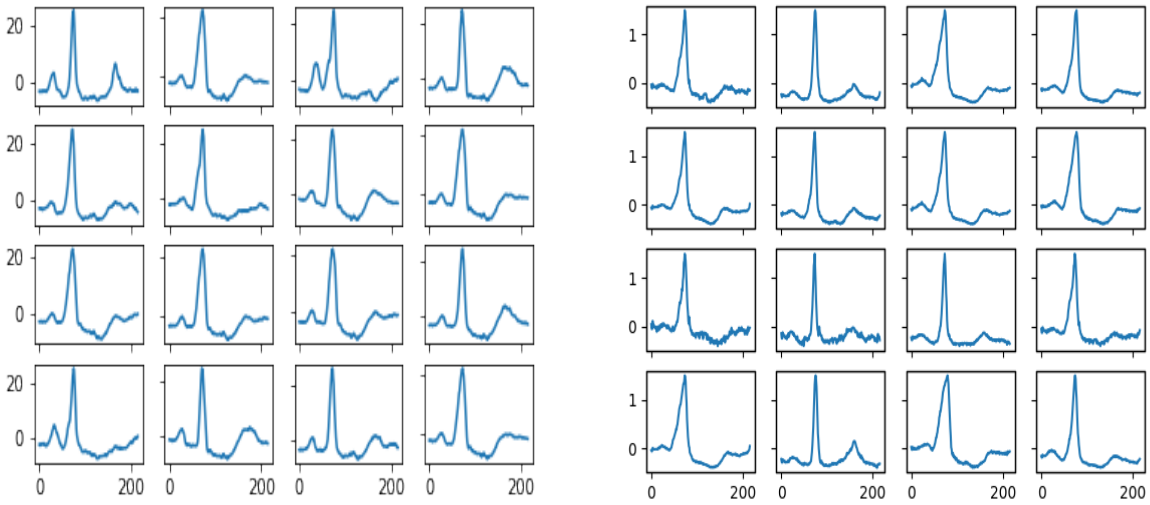


Figure 4.3: Integrated gradient xAI output LSTM

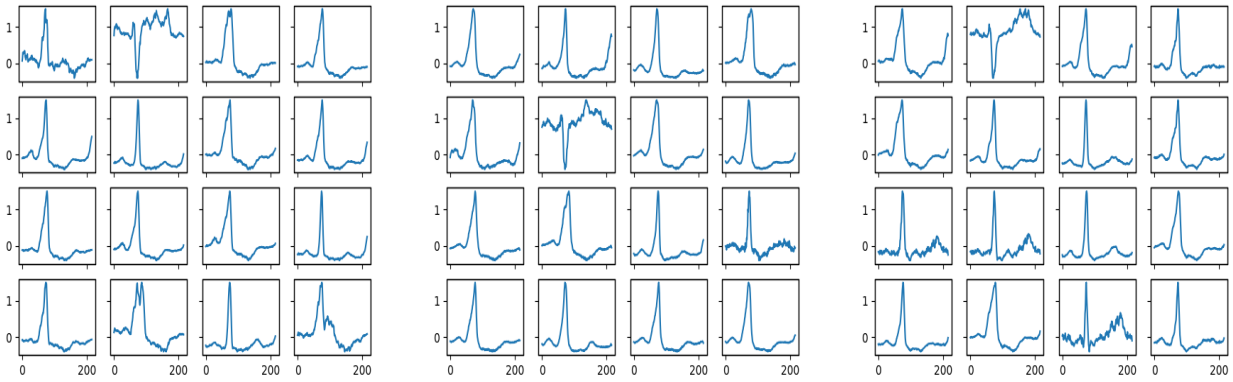
Figure 4.2 and 4.3 allows us to visualize the xAI outputs. The top row of each figure shows the original heartbeat, xAI output and the product of the original time series by the xAI output. The bottom row shows the original gradient and the product of the gradient by the xAI output. This is the modification we're making to the gradient as a way to augment feedback to the generator. These figures show that each xAI system gives weight to different timesteps and there is signifcant variability in the xAI outputs. Based on these results, we would expect that the choice of xAI system has a significant effect on GAN performance.

4.3.3 Qualitative Results



(a) CNN Architecture (b) CNN wgan

Figure 4.4: F class heartbeats CNN



(a) CNN with adversary xAI (b) CNN with Ig xAI (c) CNN with Salinecy xAI

Figure 4.5: CNN Architecture with different xAI Systems

Figure 4.4 visualizes the output of CNN with wgan and 4.5 shows the output of CNN with different xAI systems. Upon visual inspection, we can see that adding xAI systems allows the GAN to learn a greater distribution of the data. Using xAI systems, the GAN produces ECG with an inverted QRS complex.

Figure 4.10 shows the output of a standard LSTM GAN and LSTM with wasserstein loss. Figure 4.11 shows the outputs of LSTM with xAI systems.

4.3.4 DTW and Pearson's Coefficient

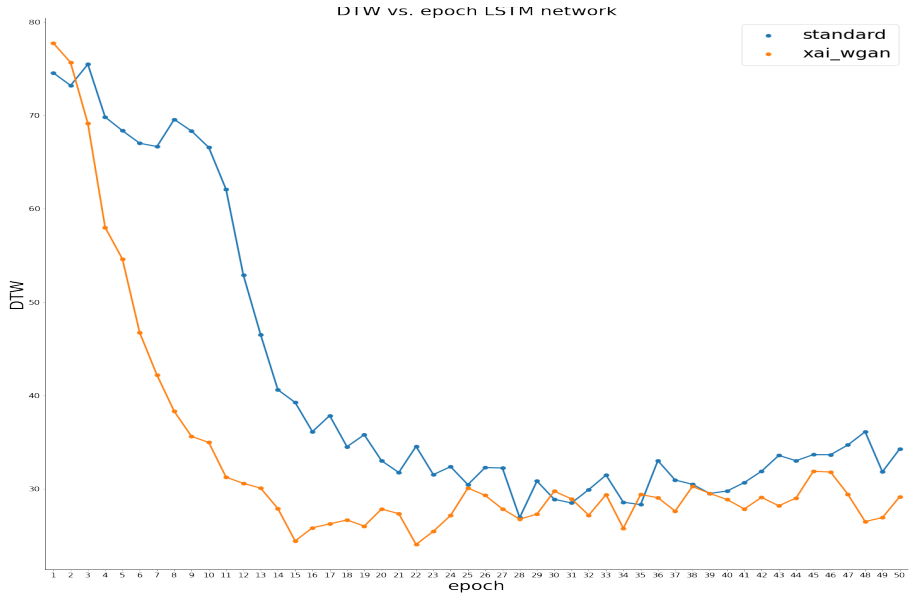


Figure 4.6: DTW LSTM Architecture

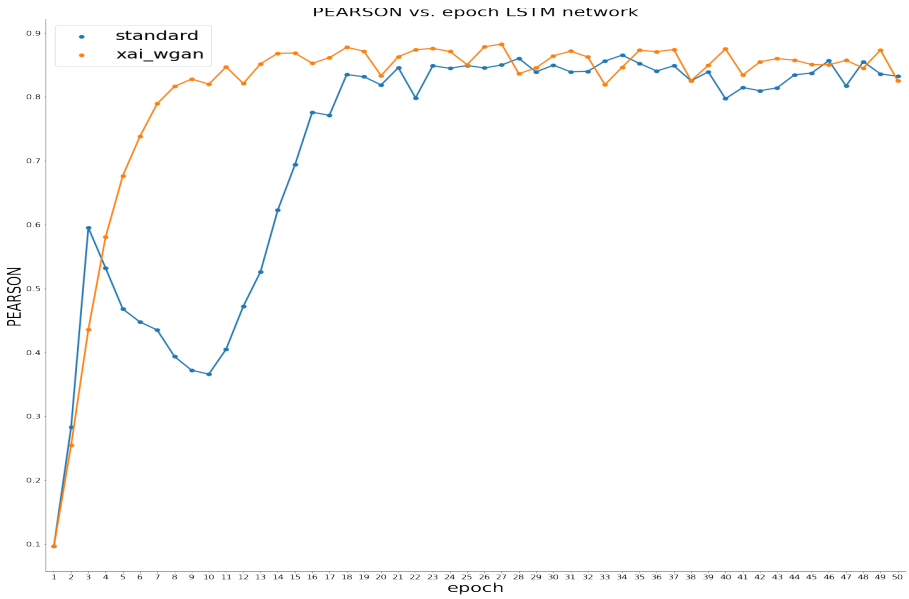


Figure 4.7: Pearson's Correlation LSTM Architecture

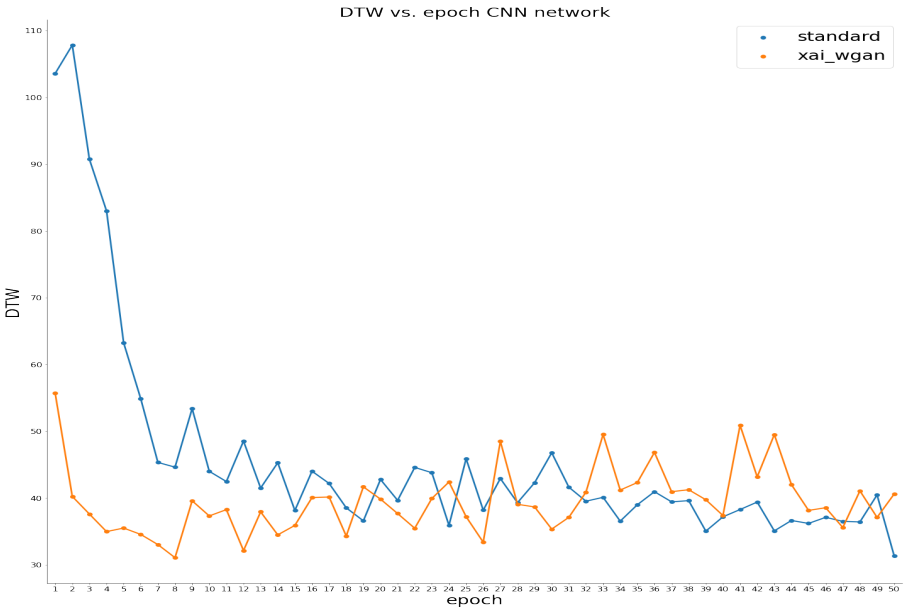


Figure 4.8: DTW CNN Architecture

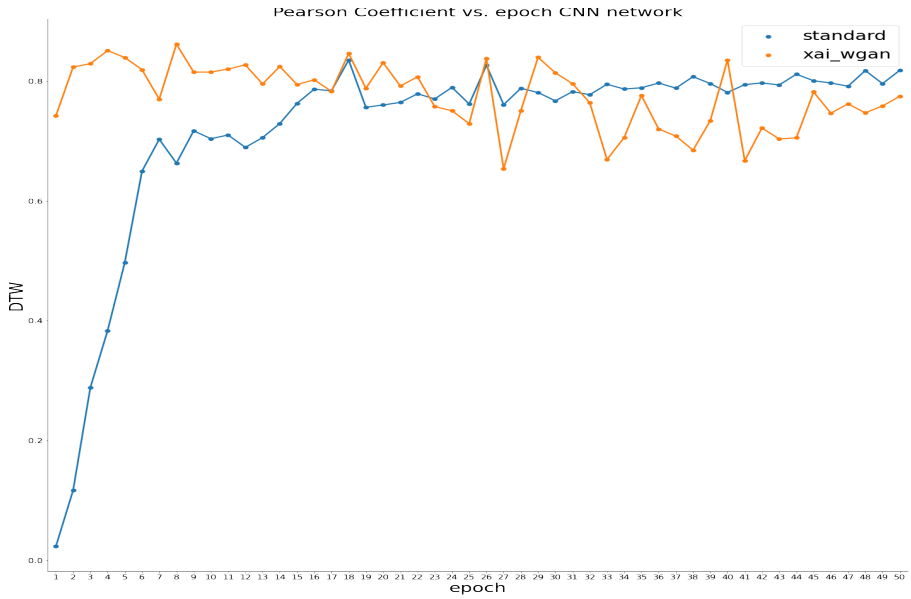


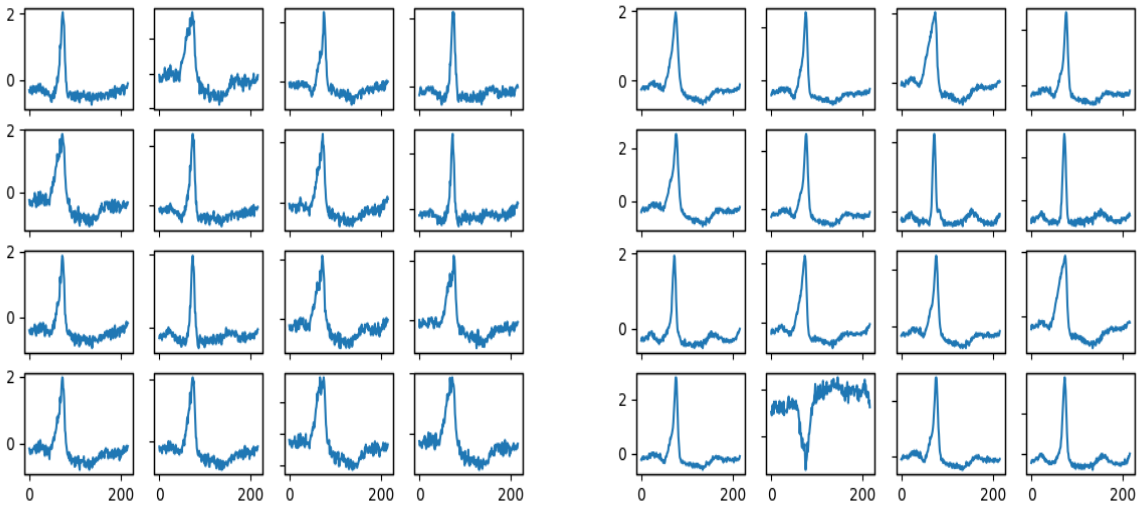
Figure 4.9: Pearson's Correlation CNN Architecture

Figure 4.8 and 4.9 shows the DTW and Pearson's Correlation Coefficient for the CNN architecture respectively and 4.6 and 4.7 shows the DTW and Pearson's Correlation Coefficient for the LSTM architecture. On the x axis is the number of epochs and on the y axis are the unitless DTW and Pearson's values. In all cases, we observe that xai-wgan reaches the final value faster.

4.3.5 Accuracy Scores

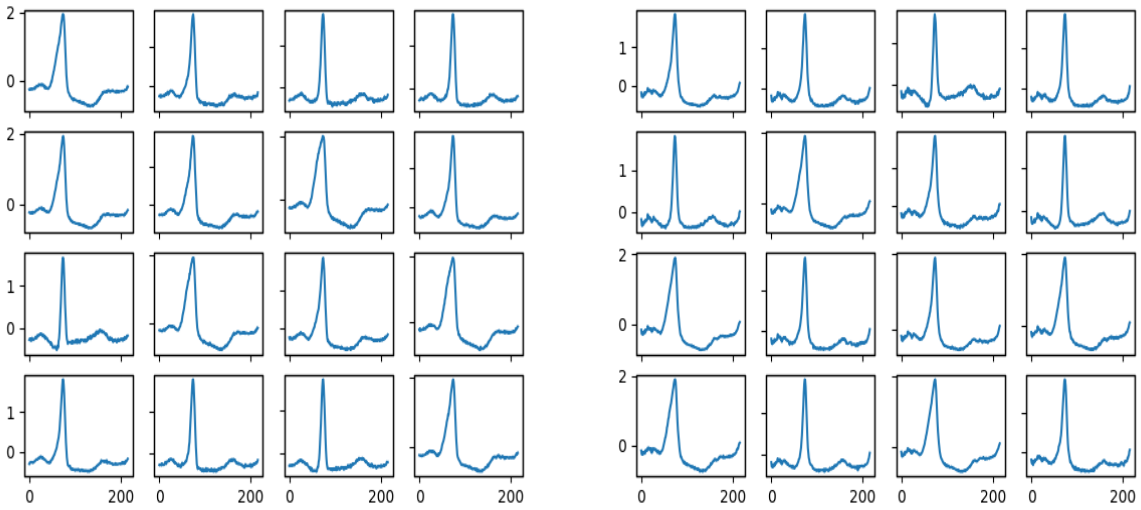
	CNN					LSTM			
	No wgan	wgan	Saliency	Ig	adversary	No wgan	wgan	Ig	Adversary
No synthetic examples	21.90%	21.90%	21.90%	21.90%	21.90%	21.90%	21.90%	21.90%	21.90%
1000 examples added	33.00%	95.10%	93.30%	93.60%	91.00%	5.80%	30.20%	64.90%	74.40%
3000 examples added	96.60%	95.40%	95.30%	97.70%	96.90%	20.10%	73.70%	85.60%	85.60%
10,000 examples added	98.00%	99.20%	98.50%	99.20%	97.90%	41.10%	91.80%	96.40%	90.20%
15,000 examples added	98.00%	99.20%	97.90%	99.50%	98.50%	32.00%	90.20%	95.60%	93.60%

Table 4.1: F class accuracy Scores



(a) LSTM Architecture (b) LSTM wgan

Figure 4.10: F class heartbeats LSTM architecture



(a) LSTM wgan Integrated Gradient (b) LSTM wgan Adversary xAI

Figure 4.11: F class heartbeats LSTM with xAI feedback

4.3.6 Timing Analysis

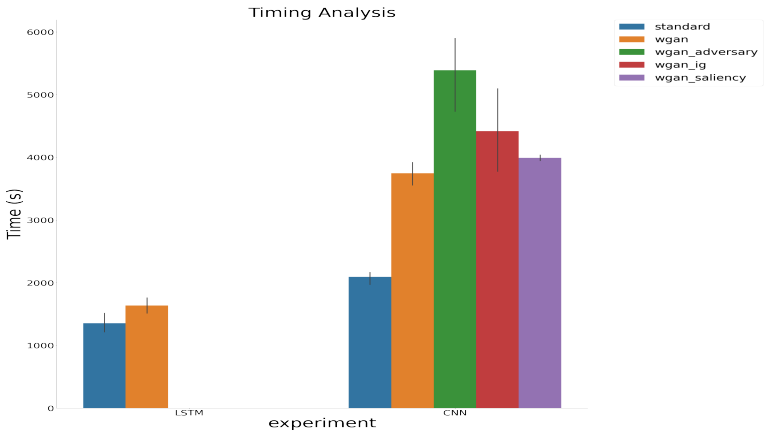


Figure 4.12: Timing

The timing bar graph on figure 4.12 shows the run time using different approaches in seconds. Wasserstein GAN significantly increases the run time compared to standard run time because the discriminator must be run and weights updated for 5 iterations for each iteration of generator training. Adding xAI systems doesn't increase the run time significantly compared to wasserstein GAN because we only use xAI feedback during the last 10 epochs of training.

4.4 Analysis of Results

In this setting, DTW and Pearson's's correlation coefficient do not always agree with the qualitative results. Standard GAN and xai-wgan have similar values for DTW and Pearson's's correlation coefficient, but Figure 4.11 shows that wgan greatly improves the quality of the produced ECGs. The P wave and T wave are not distinguishable in the LSTM alone and the signal is very noisy. Xai-wgan is able to produce higher quality ECGs where we can see the P wave, QRS complex, and T wave. Table 4.1 also shows that augmenting the training set with synthetic examples from xai-wgan gives higher accuracy scores than augmenting the training set with synthetic data from the LSTM alone.

The xAI systems that we used with the LSTM discriminator all gave high values to the QRS complex. Using the CNN architecture, the xAI systems gave higher values to the P wave and the T wave. The difference in output between these two architectures may explain why xAI feedback led to more variability in the CNN output.

Although the CNN was better able to learn the shape of the ECG and produce smoother signals, the CNN network failed to learn the scale of the data. The CNN generated signals fell between the range of +10 to -10 and we used numpy's interpolation function to correct the scale.

4.5 Conclusions and Future Work

We developed a novel method for ECG generation, xai-wgan, and demonstrated that it outperformed standard GAN using several evaluation metrics. There is no consensus on how best to evaluate the quality of the data produced by a GAN [7]. For this project, we used multiple metrics to evaluate GAN performance including DTW to measure the similarity between the real and generated time series, Pearson's's correlation coefficient to measure how correlated the two time series are, the utility of the generated data to

augment a downstream classifier and improve its accuracy, qualitative analysis and timing analysis.

The results of our experiments show that xAI systems can be further developed and improved upon to work on time series data. The xAI systems that we used in our experiments were originally developed for the image domain.

We want to measure the variance among the generated data and measure if there is mode collapse.

For future work, we want to test our method on multivariate time series data including multi-lead ECG data and EEG data. Multivariate time series data will also allow us to test other xAI systems.

Chapter 5

Conclusion

High quality medical data is required for training neural networks that are becoming increasingly widespread in the medical domain.

In this thesis, we have presented two approaches for generating high quality ECGs. In the first, we focus on the generation of ECGs using GANs. In the second, we look at the translating ECGs from the normal class to arrythmias. We also look at using machine learning tools for imputation and denoising of ECGs. The experimental results show that our methods outperform existing state-of-the-art methods.

We have shown the efficacy of these methods in a specific setting, the generation of ECGs. However, there are many time series medical datasets and the applications of our methods to other areas in medicine are vast.

5.1 Future Work

This thesis focused on ECG data, but these methods can be easily applied to other univariate time series data such as sensor data or EEGs from a single lead. The methods described in this thesis can be further extended to multivariate time series data like multi lead EEGs and ECGs.

One future direction of research would be to extend the imputation approach described in Chapter 3 to work on Electronic Health Record (EHR) data. EHR data is challenging to generate because different columns can have mixed data types where one column has categorical data while another column has numeric data. EHR data also has non-gaussian

distributions and highly imbalanced categorical columns. Several approaches have been attempted to generate tabular data. MedGAN [4] uses an autoencoder to transform categorical variables to a latent continuous representation. In TableGAN [15], categorical values are transformed into continuous values and then minimax scaled. TGAN [22] is another approach to generating tabular data where a Gaussian Mixture Model is applied to each column individually. Our imputation method can be applied to EHR data by combing our cycleGAN architecture with one of these methods. Instead of learning the translation from missing data to complete data, cycleGAN will learn the transformation from one column to another. In our architecture, G_1 will learn the transformation from a column that has categorical data to a column that has continuous data and G_2 will learn the reverse the reverse transformation. This can be achieved by using an approach similar to TGAN where each column is preprocessed. Another approach is to add an autoencoder to the architecture. This could work as an imputation technique in a setting where there is missing data in one column but complete data in another column. The proposed cycleGAN has learned the transformation between two columns and could work as an imputation tool.

Another area to explore is to incorporate more xAI feedback into the generator. In this thesis, we used local xAI systems that make predictions sample by sample. Global methods produce one output for the whole batch. An example is shapelets. One future direction is to combine local and global output to enhance the feedback to the generator.

In Chapter 4, we explored several evaluation metrics to assess the quality of the synthetic data generated. One future direction is to do an empirical measure of privacy. Membership inference attack (MIA) is a well-known empirical evaluation of privacy that is typically used to test classifiers. The membership inference attack test asks the following: Given a machine learning model and a record, can we determine whether this record was used as part of the model's training dataset or not?

Using a classifier, MIA aims to infer whether specific data were included in the training dataset by exploiting the confidence vectors returned from the target model. To apply this to a GAN setting, we look at the discriminator's confidence values. The confidence tends to be slightly higher for training data even after GAN training is complete. For membership inference attack in this setting, the attacker does not have access to the GAN discriminator. The attack gets synthetic examples from the GAN and uses these synthetic examples to train a new GAN. The attacker can look at the discriminator confidence of this new GAN to determine if a record was used to train the original GAN.

As the applications of AI in medicine continue to grow, there will be a need to generate more high-quality synthetic data to train better models. This thesis explored different approaches that can be used to create ECGs.

References

[1] Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. 2017.

[3] Shubhojeet Chatterjee, Rini Smita Thakur, Ram Narayan Yadav, Lalita Gupta, and Deepak Kumar Raghuvanshi. Review of noise removal techniques in ecg signals. *IET signal processing*, 14(9):569–590, 2020.

[4] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. 2017.

[5] Anne Marie Delaney, Eoin Brophy, and Tomas E Ward. Synthesis of realistic ecg using generative adversarial networks. 2019.

[6] Khaled El Emam, Sam Rodgers, and Bradley Malin. Anonymising and sharing individual patient data. *BMJ (Online)*, 350(mar20 1):h1139–h1139, 2015.

[7] Ghadeer Ghosheh, Jin Li, and Tingting Zhu. A review of generative adversarial networks for electronic health records: applications, evaluation measures and data sources. 2022.

[8] Tomer Golany, Gal Lavee, Shai Tejman Yarden, and Kira Radinsky. Improving ecg classification using generative adversarial networks. *Proceedings of the ... AAAI Conference on Artificial Intelligence*, 34(8):13280–13285, 2020.

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. 2017.

[10] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. 2017.

[11] Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. Ecg heartbeat classification: A deep transferable representation. 2018.

[12] Yanxia Liu, Anni Chen, Hongyu Shi, Sijuan Huang, Wanjia Zheng, Zhiqiang Liu, Qin Zhang, and Xin Yang. Ct synthesis from mri using multi-cycle gan for head-and-neck radiation therapy. *Computerized medical imaging and graphics*, 91:101953–101953, 2021.

[13] Steffen Moritz, Alexis Sardá, Thomas Bartz-Beielstein, Martin Zaefferer, and Jörg Stork. Comparison of different methods for univariate time series imputation in r. 2015.

[14] Vineel Nagisetty, Laura Graves, Joseph Scott, and Vijay Ganesh. xai-gan: Enhancing generative adversarial networks via explainable ai systems. 2020.

[15] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, 2018.

[16] Arash Rahnama and Andrew Tseng. An adversarial approach for explaining the predictions of deep neural networks. 2020.

[17] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. Ai in health and medicine. *Nature medicine*, 28(1):31–38, 2022.

[18] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz-Rodríguez. Explainable artificial intelligence (xai) on timeseries data: A survey. 2021.

[19] Francisco P Romero, David C Piñol, and Carlos R Vázquez-Seisdedos. Deepfilter: an ecg baseline wander removal filter using deep learning techniques. *Biomedical Signal Processing and Control*, 70:102992, 2021.

[20] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi. Vigan: Missing view imputation with generative adversarial networks. *2017 IEEE International Conference on Big Data (Big Data)*, pages 766–775, 2017.

[21] Romeo Vecht, Nicholas Peters, and Micheal A. Gatzoulis. *Basic principles*, pages 1–9. Springer London, London, 2009.

[22] Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks. 2018.

[23] Fei Yang, Jiazhi Du, Jiying Lang, Weigang Lu, Lei Liu, Changlong Jin, and Qinma Kang. Missing value estimation methods research for arrhythmia classification using the modified kernel difference-weighted knn algorithms. *BioMed research international*, 2020:7141725–9, 2020.

[24] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets. 2018.

[25] Long Zhou, Joshua D Schaefferkoetter, Ivan W.K Tham, Gang Huang, and Jianhua Yan. Supervised learning with for low-dose fdg pet image denoising. *Medical image analysis*, 65:101770–101770, 2020.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251. IEEE, 2017.