Deep Learning-based Signal Processing Approaches for Improved Tracking of Human Health and Behaviour with Wearable Sensors

Eoin Brophy

M.Eng

A thesis presented for the degree of Doctor of Philosophy

School of Computing

Dublin City University

Supervisor: Prof. Tomás E. Ward

August 2022

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge, breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

	18214163	August 31, 2022
Signature	Student ID	Date

Dedication

Mam, Dad and Clare.

“We love the things we love for what they are.”

List of Publications

The following are **journal/book chapter papers** that have been submitted/published during the course of my PhD.

- **E. Brophy**, B. Hennelley, M. De Vos, G. Boylan and T. E. Ward, "Improved Electrode Motion Artefact Denoising in ECG using Convolutional Neural Networks and a Custom Loss Function," in IEEE Access, 2022. [**Published**]
- **E. Brophy**, P. Redmond, A. Fluery, M. De Vos, G. Boylan and T. E. Ward, "Denoising EEG signals for Real-World BCI Applications using GANs," Frontiers in Neuroergonomics, 2022. [**Published**]
- **E. Brophy**, Z. Wang, Q. She, T. E. Ward, "Generative Adversarial Networks in Time Series: A Systematic Literature Review", ACM Computing Surveys, August 2022. [**Accepted**]
- **E. Brophy**, M. De Vos, G. Boylan and T. E. Ward,"Multivariate Generative Adversarial Networks and their Loss Functions for Synthesis of Multichannel ECGs," in IEEE Access, 2021. [**Published**]
- **E. Brophy**, M. De Vos, G. Boylan and T. E. Ward, "Estimation of Continuous Blood Pressure from PPG via a Federated Learning Approach". Sensors (Basel), 2021. [**Published**]

The following are **conference papers** that have been published during the course of my PhD.

- M. Habiba, **E. Brophy**, B. Pearlmutter and T. E. Ward. "ECG Synthesis with Neural ODE and GAN Models". International Conference on Electrical, Computer and Energy Technologies (ICECET), South Africa, 2021.
- **E. Brophy**, 2020. "Synthesis of Dependent Multichannel ECG using Generative Adversarial Networks." In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20). Association for Computing Machinery, New York, NY, USA, 3229–3232.
- D. Gunasekaran, G. Venkatraj, **E. Brophy** and T. E. Ward. "Improved Speech Synthesis using Generative Adversarial Networks". Irish Conference on Artificial Intelligence and Cognitive Science (AICS), Ireland, 2020.
- **E. Brophy**, J. J. Dominguez, Z. Wang, A. F. Smeaton and T. E. Ward. "An Interpretable Machine Vision Approach to Human Activity Recognition using Photoplethysmograph Sensor Data". Irish Conference on Artificial Intelligence and Cognitive Science (AICS), Ireland, 2018.

- **E. Brophy**, J. J. Dominguez, Z. Wang and T. E. Ward. "A Machine Vision Approach to Human Activity Recognition using Photoplethysmograph Sensor Data". 29th Irish Signals and Systems Conference (ISSC), UK, 2018.

The following are **workshop papers/manuscripts** that have been published during the course of my PhD.

- J. Weldon, **E. Brophy** and T. E. Ward, "Generation of Synthetic Electronic Health Records Using a Federated GAN", arXiv preprint arXiv:2109.02543, 2021.
- **E. Brophy**, W. Muehlhausen, A. F. Smeaton and T. E. Ward, "CNNs for Heart Rate Estimation and Human Activity Recognition in Wrist Worn Sensing Applications," 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2020, pp. 1-6.
- A.M. Delaney, **E. Brophy**, and T. E. Ward, "Synthesis of realistic ECG using generative adversarial networks," arXiv preprint arXiv:1909.09150, 2019.
- **E. Brophy**, Z. Wang and T. E. Ward. "Quick and Easy Time Series Generation with Established Image-based GANs". arXiv preprint arXiv:1902.05624, 2019.
- W. Muehlhausen, L. Zhang, **E. Brophy,**, and T. E. Ward, "A novel approach to using machine learning algorithms for fraud detection in real world data from wrist worn wearable devices," Quality of Life Research, Springer, 2019, (Vol. 28, pp. S177-S178).

The following are **additional articles** that have been published during the course of my PhD.

- C. Zhenshuo, **E. Brophy** and T. E. Ward. "Malware Classification Using Static Disassembly and Machine Learning". Irish Conference on Artificial Intelligence and Cognitive Science (AICS), Ireland, 2021.
- H. Bae, **E. Brophy**, R. H. Chan, B. Chen, F. Feng, G. Graffieti, V. Goel, X. Hao, H. Han, S. Kanagarajah, S. Kumar, S.-K. Lam, T. L. Lam, C. Lan, Q. Liu, V. Lomonaco, L. Ma, D. Maltoni, G. I. Parisi, L. Pellegrini, D. Piyasena, S. Pu, Q. She, D. Sheet, S. Song, Y. Son, Z. Wang, T. E. Ward, J. Wu, M. Wu, D. Xie, Y. Xu, L. Yang, Q. Yang, Q. Zhong, and L. Zhou, "IROS 2019 Lifelong Robotic Vision: Object Recognition Challenge [Competitions]," IEEE Robotics & Automation Magazine, vol. 27, no. 2, pp. 11–16, jun 2020.
- N. Crone, **E. Brophy** and T. E. Ward, "Exploration of Algorithmic Trading Strategies for the Bitcoin Market", arXiv preprint arXiv:2110.14936, 2021.

Acknowledgements

Although a PhD is about becoming an independent researcher, I have received a great deal of support and assistance throughout the years of my research. Firstly, I would like to thank my supervisor:

- Professor Tomás Ward of Insight SFI Research Centre for Data Analytics at DCU,

  and advisors:

- Professor Maarten De Vos of the Department of Electrical Engineering and the Department of Development and Regeneration at KU Leuven,
- Professor Geraldine Boylan of INFANT Research Centre at UCC,
- Dr. Lorcan Walsh of Digital Endpoint Capability Centre at Novartis.

Your pragmatic research approach proved invaluable in establishing me as a successful researcher. In addition, your insightful feedback helped me to formulate my research questions and methodologies. You have elevated my critical thinking to a new level and provided me with the skills and knowledge that I needed to develop professionally and complete my dissertation.

I want to acknowledge my colleagues for their helpful collaboration throughout my PhD studies. Zhengwei, for your expert insights into machine learning and bringing my knowledge forward. José, for helping me edit my first couple of academic papers and understanding what it takes to structure them. Lili, for learning alongside me and transferring that knowledge back and forth, not to mention helping build that GPU machine. Finally, Aaron, for training alongside me after hours which helped keep my mind clear, and of course, for sharing your NLP expertise.

In addition, I would like to thank my parents and siblings for their sympathetic ear. You are always there for me. Finally, I could not have completed my studies and dissertation without the support of my best friend, Clare; you provided every support I could only hope for. Your unwavering support and faith in me helped grow my own academic and personal confidence. Here's to the next chapter.

Finally, I would like to acknowledge the support of NVIDIA Corporation with the donation of a Titan Xp GPU used for my research. I could not have even begun to dream of a PhD without funding, and Science Foundation Ireland funded this PhD under grant number 17/RC-PhD/3482, to whom I extend my sincere gratitude.

Contents

List of Figures

List of Tables

List of Abbreviations

AAMI - Association for the Advancement of Medical Instrumentation

ABP - Arterial Blood Pressure

ACF - Autocorrelation Function

ADC - Analogue-to-Digital Converter

AE - Autoencoder

AI - Artificial Intelligence

AR-FNN - Autoregressive Feed-Forward Neural Network

ARR - Arrhythmia

BCE - Binary Cross-Entropy

BCI - Brain-Computer-Interface

BIDMC - Beth Israel Deaconess Medical Center

BiGridLSTM - Bidirectional Grid Long-Short Term Memory

BP - Blood Pressure

BPTT - Back Propagation Through Time

CAE - Convolutional Autoencoder

CGAN - Conditional GAN

CNN - Convolutional Neural Network

CNNR - Convolutional Neural Network with Regression

CRISP-ML(Q) - Cross-Industry Standard Process for the development of Machine Learning applications with Quality assurance methodology

CV - Computer Vision

CVD - Cardiovascular Diseases

DAT-CGAN - Decision Aware Time Series Conditional GAN

DCGAN - Deep Convolutional

DJI - Dow Jones Index

DL - Deep Learning

DP - Differential Privacy

DPGAN & GAN-DP - Differentially Private GAN

DTW - Dynamic Time Warping

DWT - Discrete Wavelet Transform

ECG - Electrocardiogram

ED - Euclidean Distance

EEG - Electroencephalogram

EHR - Electronic Health Record

EMD - Empirical Mode Decomposition

EMG - Electromyography

EOG - Electrooculography

ETF - Exchange-traded Fund

FedGAN - Federated Learning GAN

FFT - Fast Fourier Transform

FID - Frechet Inception Distance

GAN - Generative Adversarial Network

GARCH - Generalised Autoregressive Conditional Heteroskedasticity

GDP - Gross Domestic Product

GDPR - General Data Protection Regulation

GMMN - Generative Moment Matching Networks

HAR - Human Activity Recognition

HIPAA - Health Insurance Portability and Accountability Act

HRE - Heart Rate Error

HRV - Heart Rate Variability

IBI - Interbeat Interval

ICA - Independent Component Analysis

ICD - International Classification of Diseases

IIIS - International Institute for Integrative Sleep Medicine

IMU - Inertial Measurement Unit

IS - Inception Score

JS - Jensen-Shannon

KF - Kalman Filter

KL - Kullback–Leibler

LED - Light Emitting Diodes

LMI - Legates and McCabe Index

LS-GAN-DTW - Loss Sensitive GAN with DTW distance measure

LS-GAN - Loss Sensitive GAN

LSGAN-DTW - Least Squares GAN with DTW term

LSGAN - Least Squares GAN

LSTM - Long-short term memory

MA - Motion Artefact

MAE - Mean Absolute Error

MAP - Mean Arterial Pressure

MBD - Minibatch Discrimination

MC - Monte Carlo

MIMIC - Medical Information Mart for Intensive Care

ML - Machine Learning

MLE - Maximum Likelihood Estimation

MLII - Modified Limb lead II

MMD - Maximum Mean Discrepency

MNIST - Modified National Institute of Standards and Technology database

MRE - Mean Relative Error

MSE - Mean Square Error

MVDTW - Multivariate (in)dependent DTW

NLP - Natural Language Processing

NN - Neural Network

NR-GAN - Noise Reduction GAN

NS - Nash-Sutcliffe model efficiency coefficient

NSDR - Normalised Source-to-Distortion Ratio

NSR - Normal Sinus Rhythm

OECD - Organisation for Economic Co-operation and Development

PCA - Principal Component Analysis

PCC - Pearson Correlation Coefficient

PG-BLEU - Policy Gradient with bilingual evaluation understudy

PII - Personal Identifiable Information

PIPL - Personal Information Protection Law

PPG - Photoplethysmography

PRD - Percent Root mean square Difference

PReLU - Parametric Rectified Linear Units

PSD - Power Spectral Density

PTT - Pulse Transit Time

RCGAN - Recurrent Conditional GAN

ReLU - Rectified Linear Units

RF - Random Forrest

RGAN - Recurrent GAN

RMSE - Root Mean Square Error

RNN - Recurrent Neural Network

RRMSE - Relative Root Mean Squared Error

RQ - Research Question

SAR - Source-to-Artifact Ratio

SC-GAN - Sequentially Coupled GAN

SeqGAN - Sequence GAN

SigCWGAN - Conditional Sig-Wasserstein GAN

SIR - Source-to-Interference Ratio

SNR - Signal-to-Noise Ratio

SPX - S&P 500 Index

SSIM - Structural Similarity Index

SVC - Support Vector Classification

SVD - Singular Value Decomposition

SVM - Support Vector Machine

SVNN - Stochastic Volatility Neural Network

SynSigGAN - Synthetic Biomedical Signals GAN

t-SNE - t-distributed Stochastic Neighbour Embedding

T2T-GAN - Time-series-to-Time-series GAN

TCN - Temporal Convolutional Networks

TRTS - Train on Real, Test on Synthetic

TSTR - Train on Synthetic, Test on Real

TTS - Text-To-Speech

UAR - Unweighted Average Recall

VAE - Variational Autoencoder

WA - Weighted Accuracy

WHO - World Health Organisation

WI - Willmott Index of agreement

XAI - Explainable AI

Abstract

Deep Learning-based Signal Processing Approaches for Improved Tracking of Human Health and Behaviour with Wearable Sensors

Eoin Brophy

This thesis explores two lines of research in the context of sequential data and machine learning in the remote environment, i.e., outside the lab setting - using data acquired from wearable devices. Firstly, we explore Generative Adversarial Networks (GANs) as a reliable tool for time series generation, imputation and forecasting. Secondly, we investigate the applicability of novel deep learning frameworks to sequential data processing and their advantages over traditional methods. More specifically, we use our models to unlock additional insights and biomarkers in human-centric datasets.

Our first research avenue concerns the generation of sequential physiological data. Access to physiological data, particularly medical data, has become heavily regulated in recent years, which has presented bottlenecks in developing computational models to assist in diagnosing and treating patients. Therefore, we explore GAN models to generate medical time series data that adhere to privacy-preserving regulations. We present our novel methods of generating and imputing synthetic, multichannel sequential medical data while complying with privacy regulations. Addressing these concerns allows for sharing and disseminating medical data and, in turn, developing clinical research in the relevant fields.

Secondly, we explore novel deep learning technologies applied to human-centric sequential data to unlock further insights while addressing the idea of environmentally sustainable AI. We develop novel deep learning processing methods to estimate human activity and heart rate through convolutional networks. We also introduce our 'time series-to-time series GAN', which maps photoplethysmograph data to blood pressure measurements. Importantly, we denoise artefact-laden biosignal data to a competitive standard using a custom objective function and novel application of GANs. These deep learning methods help to produce nuanced biomarkers and state-of-the-art insights from human physiological data.

The work laid out in this thesis provides a foundation for state-of-the-art deep learning methods for sequential data processing while keeping a keen eye on sustainable AI.

Chapter 1

Introduction

This thesis showcases the capability of machine learning (ML) and deep learning (DL) systems in remote monitoring of human health and performance. Furthermore, the research presented herein demonstrates the significant advantage that our novel ML and DL approaches offer over classical signal processing methods. We also benchmark our techniques against other ML methods from recent literature, which previously have been compared to classical machine learning with signal processing derived features. We centre this work around human health monitoring from a medical perspective. It is essential to set out that these ML techniques aim to augment and not replace decision-making protocols currently in place.

1.1 Motivation and Contribution

The Organisation for Economic Co-operation and Development (OECD) has indicated that Ireland (as of 2020) had 3.5 practising doctors per 1000 inhabitants, around the OECD average. Although the population uses hospitals more efficiently, healthcare expenditure has risen from €12.7 billion in 2006 to €23.8 billion in 2019 (pre-COVID), with chronic health accounting for over 75% of spending in 2017 [1].

In most countries, year on year, healthcare spending is increasing as a percentage of Gross Domestic Product (GDP) [2]. Around the globe, Health Departments have signalled a need to shift the paradigm of treating ill people from a reactive treatment-based model to one where the focus is increasingly on keeping people healthy, preventing illness and promoting health and wellbeing [3].

Future healthcare will be guided by core principles, including greater emphasis on patient-centredness and pressures to move toward new payment models, such as outcomes-based charging. Advances in digital health technologies, including mHealth and MedTech, can contribute significantly to a transformation in healthcare delivery, e.g. enabling proactive care through low-cost, low-power monitoring devices and applications using advanced data analytics that enables greater personalisation of treatments. These new digital health technologies have the potential to reduce the burden on the finite and relatively expensive resources of clinical institutions and care centres whilst providing very cost-effective modes for diagnostics, monitoring and treatment of larger population sets.

There is a growing public sentiment concerning data privacy and, in particular, how sensitive data is shared [4], [5]. As a result, privacy has moved to the forefront of the population's thinking, and recently individuals want control over how their information is collected, stored and processed [6], [7].

Data protection regulations exist worldwide to protect individuals' sensitive data, such as the General Data Protection Regulation (GDPR) in Europe, the Health Insurance Portability and Accountability Act (HIPAA) in the USA and Personal Information Protection Law (PIPL) in China. Medical data is considered personal sensitive data and generally afforded a higher level of protection under these regulations. HIPAA is perhaps the looser of the three regulations listed here as the HIPAA Privacy Rule states: "De-identified health information created following these methods is no longer protected by the Privacy Rule". In other words, when the data is de-identified, it is no longer protected and may be used for secondary purposes [8]. PIPL and GDPR present a more robust framework for protecting personal health data. There are many steps required for the dataset processor to consider when processing sensitive personal information, such as privacy impact assessment, encryption, consent and notification to the data subjects about secondary uses of their data [9], [10].

While adhering to the more stringent data protection regulations ensures we act in good faith with the principles set out in front of us, Rocher *et al.* found that anonymised datasets are unlikely to satisfy the standards set out in the GDPR [11]. This is concerning for medical data records, particularly when discussing physiological time series data (we often discuss ECG in this thesis). Electrocardiogram (ECG) characteristics made it suitable for subject identification even with de-identification methods in place. It is unique to each person and possible to identify subjects using only the ECG waveform [12].

Recent work has suggested that many patients disapprove of having their health data shared with anyone, even when the primary identifiers have been removed; instead, they want control over what can be accessed [13]. Overall this is essential to the thesis as a significant driver comes from a clinical collaborator who has access to large digitised datasets of human biomedical signals. These datasets have massive value to the collaborator in that they can use the data for training and upskilling clinicians and use the data to train in-house machine learning systems for the classification and diagnosis of various pathologies. Data like this is precious but often is constrained in how it may be used and ends up with a lot of unrealised value. Furthermore, due to privacy regulations, the institutions that collected the datasets cannot easily share them for secondary research purposes unless they received explicit consent from participants in the form of an 'opt-in' clause in the original data collection experiment. It is essential to state here that more than just an 'opt-in' clause is required to disseminate these personal sensitive datasets; anonymisation is also one such principle that must be adhered to. However, many of the datasets were collected before data protection regulations without the foresight of these problems. These legacy issues begin to create a bottleneck for sharing these datasets and lead to stagnation in clinical and research progress and form a large part of our motivation for the work carried out in this thesis. It is necessary to state that other important factors in modern machine learning, such as bias, inclusion, fairness, transparency etc., should be considered when designing a trustworthy AI system. However, these are not a focus for our clinical driver laid out above.

Our contributions towards the above are twofold. Firstly, our pioneering work developing recurrent GANs for time series generation addresses the issue surrounding disseminating sensitive medical data. With appropriate safeguards, this type of sensitive personal data can be used for further upskilling of clinicians and medical professionals, along with improved machine learning models for augmenting decision-making processes in clinical and remote environments. To that end, we contribute the only review of existing GAN technologies concerned with processing time series data in the literature. This allows us to firmly place ourselves among the current state-of-the-art and only further developments in this space. Secondly, we contribute novel deep learning technologies for discovering new pathological biomarkers. These improved analytics, combined with our first contribution, allow for the development of better decision-making systems. Both push state-of-the-art deep learning methods for human physiological data processing and improve end-user and patient outcomes.

1.2 Research Questions and Hypothesis

This thesis delves into the areas of human physiology, machine learning and, by extension, deep learning. We investigate the processing of human physiological data using novel artificial intelligence models to unlock additional insights in the data, be it pathological biomarkers or otherwise, that would not be available before applying AI-based methods described in this thesis. We explore machine learning techniques (i.e., feature extraction and representation learning) that efficiently perform tasks such as classification on physiological datasets. In terms of deep learning, we explore the possibility of using physiological signals to synthesise and impute data with GANs and denoise noisy physiological signals using GANs and convolutional neural networks (CNNs). We use GANs to structure our hypothesis, given that GANs are a new development and have been underexplored in the context of biomedical time series data (compared to image data progress in the computer vision domain).

Hypothesis: We hypothesise that GANs can be used to give competitive performance in a range of applications, from data generation to privacy preservation to data cleaning.

Research questions: Based on our hypothesis, we have set out two research questions (RQ) that arise in these areas and shape the content of this thesis:

1. Can we provide a foundation for time series generation with Generative Adversarial Networks? This question addresses the effectiveness of GANs as a generative model and how the technology can be applied to physiological data in terms of synthesising, imputing and cycling time series modalities. Chapters 2, 3 and 5 visit these areas.

2. Can we successfully leverage novel deep learning-based models to process human physiological signals and return state-of-the-art insights from the data? Chapters 4–6 showcase the capability of our models to do such.

An important aspect of GAN research is that time series signals have not been given significant attention until relatively recently. As a result, most GAN-related developments have taken place in the computer vision domain with image and media data. We are among the initial researchers in the GAN domain focused on time series developments. With growing focus on data privacy and the introduction of the GDPR in Europe, studies involving sensitive personal data such as human physiological signals have become increasingly complex to commence. The study often requires anonymisation of the data and navigation of complex regulatory frameworks. By synthesising physiological time series signals, we can facilitate the sharing and dissemination of synthetic physiological data that adhere to strict privacy regulations, often a burden for researchers and clinicians alike, which bottlenecks clinical research and development. In essence, we retain the essential characteristics of the datasets while side-stepping the associated privacy issues.

1.3 Deep Learning for Signal Processing

We explore deep learning and machine learning in this thesis and provide a background to the fundamentals of this technology. The main approaches for solving problems with AI are supervised and unsupervised learning. The significant difference between these two approaches is that one uses labels to predict outcomes and the other does not. Generally, supervised learning is applied to the problems of classification and regression. On the other hand, unsupervised learning is commonly used for clustering and dimensionality reduction. This thesis primarily applies supervised learning to train deep learning architectures. To establish a foundation of the architectures this thesis uses, we describe feedforward neural networks and the backpropagation algorithm for supervised learning.

1.3.1 Neural Networks

Neural Networks (NN) are comprised of several layers, with each layer holding a specified number of neurons. Neurons in machine learning are computational units designed to be loosely analogous to neurons in our brain. They take an input(s) $\boldsymbol{a}^{(0)}$ and output $\boldsymbol{a}^{(1)} = f(W\boldsymbol{a}^{(0)}+b)$, where W is the weight matrix, b the bias and f is the activation function. The activation function is a function used by the neurons/nodes in the network to compute a weighted sum on the input and biases and is used to get an output of the neuron. Some of the more common activation functions are the rectified linear units, sigmoid, and tanh functions. These are generally referred to as the non-linear element of the neuron and are computed for the output of it. Equation (1.1), shows the mathematical expression of the sigmoid function, and we can observe that it is a non-linear function. It is common for these activation functions to squash/ transform their outputs; hence choosing activation functions becomes an important part of NN architecture design.

$$f = \frac{1}{(1+e^{-x})} \tag{1.1}$$

A neural network then becomes a collection of these neurons where the output of one neuron in the previous layer is used as the input of another neuron in the next layer. Figure 1.1 illustrates a simple feedforward Neural Network with two hidden layers.

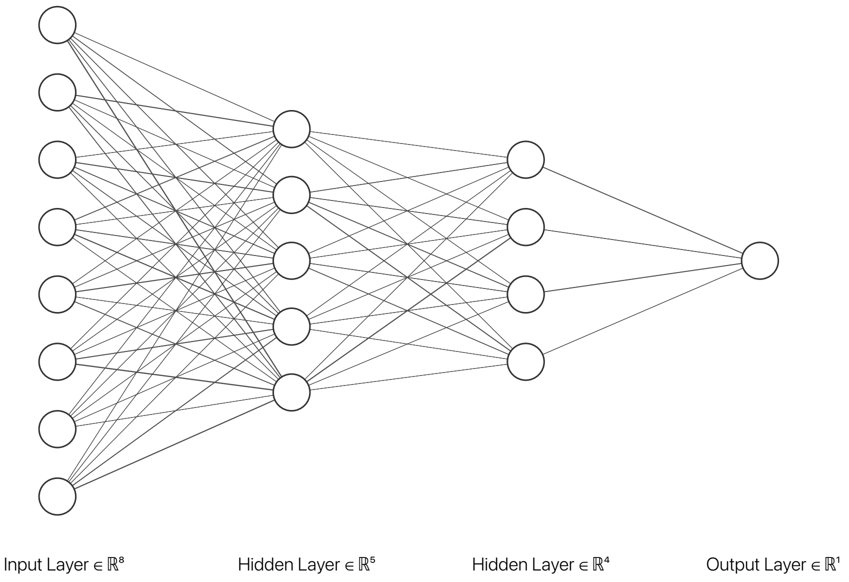
Input Layer ∈ $\mathbb{R}^8$
Hidden Layer ∈ $\mathbb{R}^5$
Hidden Layer ∈ $\mathbb{R}^4$
Output Layer ∈ $\mathbb{R}^1$

Figure 1.1: A simple feedforward Neural Network with two hidden layers and one neuron in the output layer.

These networks are essentially a highly complex function, often with over tens of thousands of parameters capable of learning very difficult tasks. For the networks to become problem solvers, they must be trained using an optimisation algorithm and a defined cost function.

In simple terms, the cost/loss function's purpose is to quantify to the network how far or near its prediction is from the actual value. Some common loss functions for training NNs are binary cross entropy (commonly used for classification problems) and mean squared error (commonly used for regression problems). The mean squared error is the most commonly used cost function for the works in this thesis and is measured as the average of the sum of squared differences between predicted and actual values, see equation (1.2).

$$MSE = \frac{\sum^{n}(y-\hat{y})^2}{n} \tag{1.2}$$

Where n is the number of samples, y is the actual value and $\hat{y}$ is the predicted values. Loss functions are of no practical use to the network without an optimisation algorithm to minimise their value.

Gradient descent [14] is often the optimisation algorithm chosen for finding the local minimum of a function, and the algorithm for computing this gradient efficiently is called backpropagation [15]. The derivative of the function $f(x)$ allows us to find the slope of the function at point x. Gradient descent uses this derivative, $f'(x)$, to minimise a function as it tells us how a small change in the input will result in an improvement in the output. On the other hand, backpropagation (backprop) calculates the gradient of the cost function with respect to the weights and biases of the network by allowing the information from the cost function to flow backwards through the network. Hence, when we think about a network learning, it can be simplified to minimising a cost function by calculating the gradients on its information (backprop), then performing gradient-based optimisation, and finally, updating the respective parameters (θ) of the network.

We have covered the basics of architecture and learning in neural networks and will now discuss the deep learning neural architectures used in this thesis. They provide a foundation for several data processing techniques, but we will look at them with respect to signal denoising, imputation and synthesis.

1.3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have contributed tremendously to the success of machine learning since their introduction in the 1990s. They are an example of neuroscientific principles influencing deep learning [16], in that they can be designed to mimic the processing of images in the visual cortex of the human brain [17]. Fully automatic learning of a CNN allows the neural network to extract features that are salient in the input data across different layers.

CNNs are simply neural networks at their core; however, their name comes from the fact that they use the convolution operation in place of the matrix multiplication in at least a single layer. Convolution is an integral that expresses the overlap of function g with function f as g is shifted [18] and is defined in equation (1.3).

$$f * g = \int f(a)g(t-a)da \tag{1.3}$$

The motivation behind using convolutional layers is that convolution has the following mechanisms; sparse interactions, weight sharing and equivariant representations that serve to improve machine learning systems. CNNs are also invariant to certain transformations in the input and work with inputs of varying sizes which makes this choice of network more flexible over other NNs [19]. Figure 1.2 illustrates a CNN with typical operations found in these networks, such as; max-pooling and convolution. Max-pooling is one of several operations used to bring features together. It is a way of downsampling the feature space as well as bringing together information from different parts of the network. Pooling introduces favourable properties for tasks such as object classification and detection.

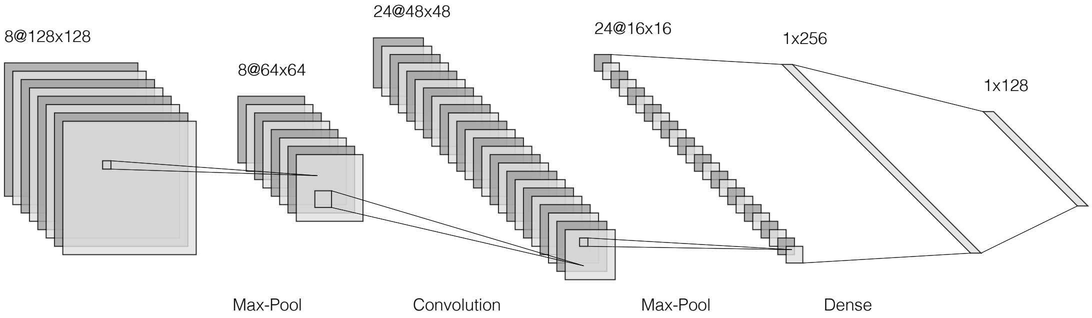

Figure 1.2: Convolutional neural network with max-pooling and convolutional operations followed by a dense or fully connected layer.

Given the proper training, a CNN provides for implementing high accuracy classifiers without the need for signal processing or feature engineering knowledge. In ML architectures, feature learning or representation learning is often used, and it is a set of techniques that allows a network to automatically discover feature patterns in the input data. CNNs specialise in data with a grid-like topology. This has contributed to their success in practical applications, particularly with image classification, where images can be considered a 2-D grid of pixels. Recently, CNNs have been used with time series signals to achieve high-quality signal denoising performance; we can consider a time series signal as a 1-D grid of data dependent on time. We apply CNNs to our work and discuss their uses further in Chapters 4 and 6.

1.3.3 Autoencoders

An autoencoder (AE) is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs in an attempt to replicate the input to its output. It has traditionally been used for dimensionality reduction or feature learning. The network consists of an encoder and decoder. The encoder can be represented by the function $h = f(x)$ where h is a compressed hidden layer that describes a code to represent the input x. Conversely, the decoder is represented via a reconstruction $r = g(h)$. An autoencoder is visualised in Figure 1.3 in its most straightforward terms.

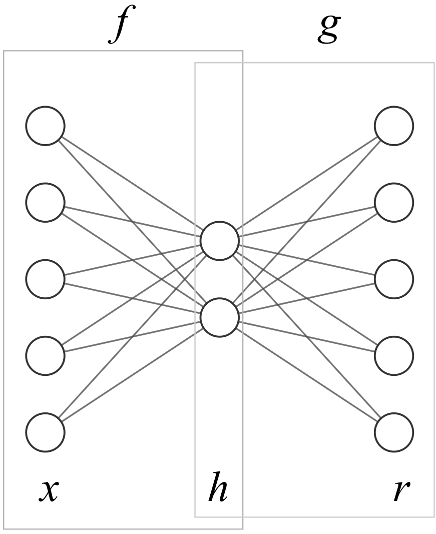

Figure 1.3: Autoencoder Model that maps an input x to an output r through the embedded representation h. The encoder block is designated here as f and the decoder as g.

Autoencoders are limited to what they can copy from input to output because simply learning to set the output of the decoder equal to the input means the model will not be useful. Therefore, these restrictions on the model enable the AE to learn valuable data properties. As mentioned above, AEs have been historically used for dimensionality and feature learning works [20]. Recently, developments in latent variable models have permitted autoencoders into the generative modelling space, as we will look at in more detail in Chapter 2.

1.3.4 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are neural networks specialised in processing sequential data whose values take the form of $x(1), x(2), ..., x(t)$. RNNs take advantage of parameter sharing, making it possible for the model to generalise across samples of different lengths and share statistical strengths across various positions in time [16]. Figure 1.4 illustrates a diagram of an RNN.

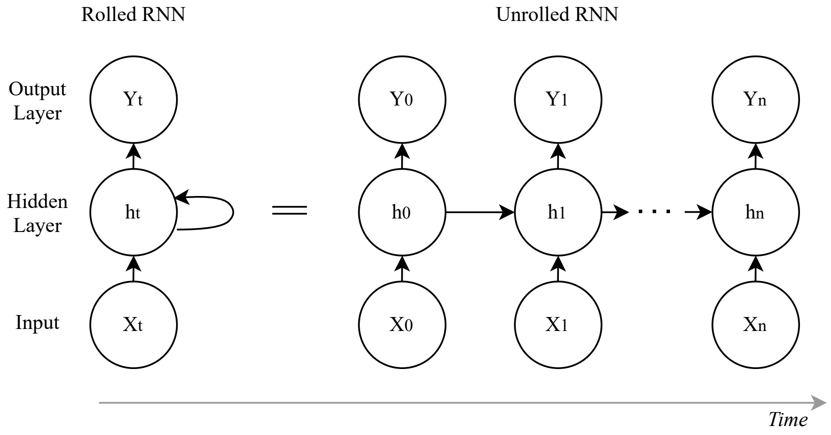

Figure 1.4: Rolled RNN (left) and Unrolled RNN (right). Where X_t is the input, h_t is the hidden state and Y_t is the output, all at time t.

The output Y at time t is defined in equation (1.4).

$$Y_t = f(h_t, w_y) = f(w_y \cdot h_t + b_y) \tag{1.4}$$

Where f is the activation function used, h_t is the hidden state at time t, w_y are the weights associated with the hidden to output units/layers, and b_y is the bias associated with the feedforward layer.

It is important to note that for training RNNs, the backprop algorithm is edited to include the unfolding in time to train the parameters of the network, known as backpropagation through time (BPTT). This training algorithm is computationally expensive, $O(\tau)$, and cannot be reduced as the forward and backward passes through the unrolled graph are inherently sequential. Here, τ is the amount of timesteps present in the signal.

RNNs form the basis of the sequence-based neural networks used in this thesis for processing physiological time series data. We discuss RNNs and their cell types in further detail in Chapter 2.

1.3.5 Generative Adversarial Networks

Generative Adversarial Networks (GANs) have been gaining a lot of traction amongst the deep learning research community since their inception in 2014 [21]. Their ability to generate and manipulate data across multiple domains has contributed to their success. While the main focus of GANs to date has been in the computer vision (CV) domain [24], they have also been successfully applied to others, such as natural language processing (NLP). In addition, there has also been a movement toward the use of GANs for time series and sequential data generation and forecasting.

A GAN consists of two neural networks, namely a generator and a discriminator. The generator G takes latent noise $\boldsymbol{z} \in \mathbb{R}^r$ and generates synthetic data. The discriminator D determines if the generated data is real or fake. The generator attempts to maximise the failure rate of the discriminator while the discriminator aims to minimise it, see Figure 1.5. The GAN model converges upon reaching the Nash equilibrium. The two networks are locked in a two-player minimax game defined by the value function *V(G,D)* (1.5), where *D(x)* is the probability that x comes from the real data rather than the generated data [21].

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}\Big[\log\Big(D(x)\Big)\Big] + \mathbb{E}_{z\sim p_z(z)}\Big[\log\Big(1 - D(G(z))\Big)\Big] \quad (1.5)$$

In Equation (1.5), $\mathbb{E}_{x\sim P(x)}[f(x)]$ represents the expectation of $f(x)$ with respect

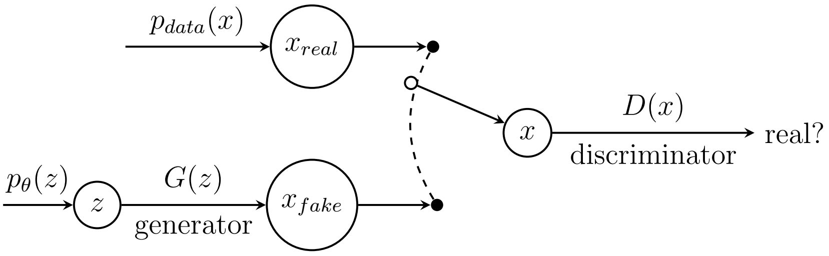

Figure 1.5: Generative Adversarial Network

to $P(x)$. In probability and statistics, the expected value is a generalisation of the weighted average that is calculated by multiplying each of the possible outcomes by its likelihood to occur and summing over all outcomes. $\mathbb{E}[X] = \sum xf(x)$, for discrete random variables and $\mathbb{E}[X] = \int_{-\infty}^{\infty} xf(x)dx$, for continuous random variables.

Given the importance of GANs in subsequent work in this thesis, in the next chapter, we explore this technology in-depth and provide a taxonomy (currently missing in the literature) to enable researchers to understand the variations and applicability of variants better.

1.4 Thesis Outline

We approach our research from the following two perspectives:

1) Applying novel machine learning and deep learning techniques, namely GANs, to synthesise and impute continuous physiological time series data. This line of work aims to solve the issue of privacy and dissemination of personal sensitive data, improving the clinical training and development pipeline that is currently bottlenecked.

2) Use novel AI techniques to unlock further biomarkers from health data collected using simple, low-power wearable devices. The expected outcome from this line of work is improved decision-making systems for medical professionals while keeping an eye on sustainable AI.

The organisation of this thesis is as follows:

- **Chapter 2** provides a review on time series GAN-variants from the continuous and discrete perspectives. We analyse the GAN-variants that deal with common problems encountered in the literature towards improved time series generation and augmentation.

- **Chapter 3** showcases our novel deep learning method for generating clean multichannel physiological data. We present our multichannel GAN architecture and custom loss function with a novel penalisation term for improved data generation compared to more standard methods.

- **Chapter 4** demonstrates, through using a simple optical measurement, used conventionally for heart rate detection in wrist-worn sensors, that we can provide competitive heart rate and human activity recognition simultaneously at low sampling rates.

- **Chapter 5** develops machine learning methods capable of inferring blood pressure from a single optical photoplethysmogram sensor alone. We trained our framework across distributed models and data sources to mimic a large-scale distributed collaborative learning experiment that could be implemented across low-cost wearables. Our time-series-to-time-series generative adversarial network is capable of high-quality continuous blood pressure generation from a single optical signal.

- **Chapter 6** presents an overview of two frameworks to denoise physiological signals. We showcase the first example of a GAN capable of EEG artefact removal that is generalisable to more than one artefact type. Our model provides a competitive performance in advancing the state-of-the-art deep learning EEG denoising techniques. We also propose a custom loss function capable of denoising electrode motion artefact in ECG data to a higher standard than other, more common loss functions. We implement our custom loss function with a convolutional neural network to return high-quality ECG, suitable for calculating key cardiac metrics from a previously unobtainable state.

- **Chapter 7** concludes the thesis. Future directions are also discussed with respect to the different research areas explored in this thesis.

Chapter 2

Generative Adversarial Networks in Time Series: A Survey and Taxonomy

2.1 Introduction

Generative adversarial network studies have grown exponentially in the past few years. Their impact has been seen mainly in the computer vision field with realistic image and video manipulation, especially generation, making significant advancements. While these computer vision advances have garnered much attention, GAN applications have diversified across disciplines such as time series and sequence generation. As a relatively new niche for GANs, fieldwork is ongoing to develop high-quality, diverse and private time series data. In this chapter, we contribute a review towards GAN variants designed for time series related applications. This allows us, in later chapters, to place ourselves at the forefront of the developments with this technology.

We provide a review of current state-of-the-art and novel time series GANs and their solutions to real-world problems. GANs are a flexible solution to both of our presented issues in Chapter 1 that may help us address these challenges in everything from data sharing and signal processing to training downstream signal processing models. The applicability of GANs to time series data can solve many issues that current dataset holders face. Data shortage is often an issue, and GANs can augment smaller datasets by generating new, previously unseen data. Data can be missing or corrupted in cases; GANs can impute data, i.e., replace the artefacts with information representative of clean data. GANs are also capable of denoising signals in the case of corrupted data. Another issue is that data protection, privacy, and sharing have become heavily regulated; GANs can ensure an extra layer of data protection by generating synthetic and differentially private datasets containing no risk of linkage from source to generated datasets.

Several methods have been used in the past to generate synthetic data. One such method is the autoencoder which is designed to efficiently learn an informative representation of an input in a small dimensional space and reconstruct the encoded data back such that the reconstructed input is similar as possible to the original one, see Chapter 1 for further details. The AE model is made of an encoder and decoder neural network, as shown in Figure 2.1. However, other generative models have emerged as front-runners due to the quality of the generated data and inherent privacy protection measures.

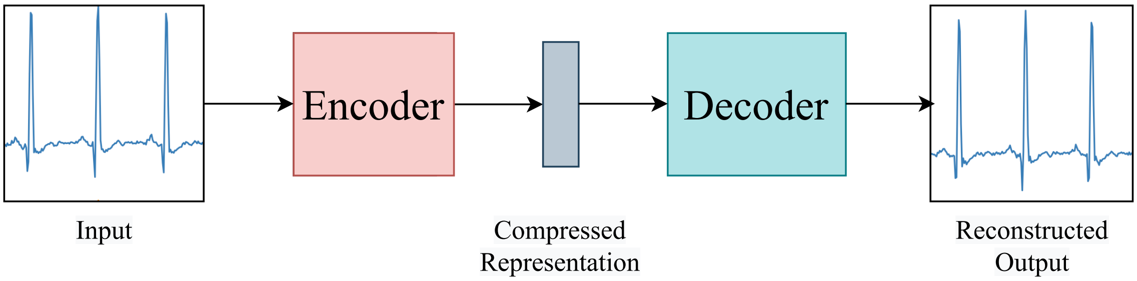

Figure 2.1: Autoencoder Model

Generative Adversarial Networks have been gaining a lot of traction amongst the deep learning research community since their inception in 2014 [21]. Their ability to generate and manipulate data across multiple domains has contributed to their success. While the main focus of GANs to date has been in the CV domain, they have also been successfully applied to others, such as NLP. There has also been a movement towards the use of GANs for time series and sequential data generation, and forecasting.

A GAN is a generative model consisting of a generator and discriminator, typically two neural networks models. In recent years GANs have demonstrated their ability to produce high-quality image and video generation, style-transfer, and image completion. They have also been successfully used for audio generation, sequence forecasting, and imputation.

Time series data generation is not a novel concept in that it has long roots seeded in regression. Furthermore, it initially began as forecasting of timesteps rather than whole sequence generation. One of the most used time series forecasting methods was autoregressive (AR) models. Aside from forecasting data points, AR models focus on preserving the temporal dynamics of a sequence. However, they are inherently deterministic in that no randomness is involved in the calculation of future states of the system. This means that AR models are not genuinely generative or probabilistic. For an AR model, the goal is to produce the next time step (x_{t+1}) in a sequence as a function of the previous n time steps, where n is the order of the model. The formula for a classic AR model is given in Equation (2.1).

$$x_{t+1} = c + \theta_1 x_t + \theta_2 x_{t-1} + \epsilon \tag{2.1}$$

Here, x_t is the value of the sequence at time t, θ is the model parameters, c is a constant, and ϵ is the error term usually chosen as normally distributed noise.

Autoregression was a step shy of time series synthesis. That ultimately came in the form of directed generative networks. When using the term directed, we mean a model where the edges are directed and thus, indicates which variable's probability distribution is defined in terms of another. In other words, this is a structured probabilistic model with conditional probability distributions.

There are several generative models, from Variational Autoencoders (VAEs) and RNN variants to GANs, which have their pros and cons. For example, VAEs use learned approximate inference to produce synthetic samples efficiently. An inference problem is simply using the value of some variables or probability distributions to predict other values or probability distributions. Approximate inference is when we seek to approximate a true distribution, say $p(y|x)$, by seeking an approximate distribution $q(y|x)$. However, this network approximation conducted by VAEs means that their generated data quality can be degraded compared to samples generated by GANs. GANs generate much more realistic data and, as such, have emerged as a front-runner in synthetic data generation. However, they are more complicated to optimise due to the proceeding challenges.

One of the significant challenges of GANs lies in their inherent instability, which makes it difficult to train. GAN models suffer from issues such as non-convergence, diminishing/vanishing gradients, and mode collapse. A non-converging model does not stabilise and continuously oscillates, causing it to diverge. Diminishing gradients prevents the generator from learning anything as the discriminator becomes too successful. Mode collapse is when the generator is trained without updates to the discriminator and the generator finds optimal data to fool the discriminator, producing only uniform samples with little-to-no variety.

The second challenge of GANs lies in its evaluation process. With image-based GANs, researchers have reached a loose consensus [22] surrounding the evaluation of the generated distribution estimated from the training data distribution. Unfortunately for time series GANs, due to the comparatively low numbers of papers published, there has not been an agreement reached on the generated data's evaluation metrics. There have been different approaches put forward, but none established as a front runner in the metrics space as of yet. Accordingly, we will propose solutions specifically useful to our tasks.

We define a time series as a sequence of vectors dependant on time (t) and can be represented as $x_t = x_1, ..., x_n$ for continuous/real-time and discrete-time. The time series' values can either be defined as continuous or discrete and, depending on the number of values recorded, are univariate or multivariate. In most cases, the time series will take either an integer value or a real value.

As Dorffner states, a time series can be viewed, from a practical perspective, as a value sampled at discrete steps in time [23]. This time-step can be as long as years to as short as milliseconds, for example. We define a continuous time series as a signal sampled from a continuous process, i.e. the function's domain is from an uncountable set. In contrast, a discrete time series has a countable domain.

In this chapter, given the relevance of GANs to subsequent solutions to deep signal processing for tracking human health and performance, we present the first complete review and taxonomy of time series GANs, namely discrete and continuous variants, their applications, architecture, loss functions and how they have improved on their predecessors in terms of variety and quality of their generated data. We also contribute by including experiments for the majority of time series GAN architectures applied to time series synthesis. A list of the reviewed GANs and datasets are available online [1].

2.2 Related Literature Reviews on GANs

There has been a handful of high-quality GAN review papers published in the past few years. For example, Wang *et al.* takes a taxonomic approach to GANs in Computer Vision [24]. The authors split GANs into architecture variants and loss variants. While they include applications of GANs and mention their applicability to sequential data generation, the work is heavily focused on media manipulation and generation. The authors in [25] breakdown GANs into their constituent parts. They begin by discussing the algorithms and architecture of various GANs and their evaluation metrics, then list their surrounding theory and problems such as mode collapse, amongst others. Finally, they discuss the applications of GANs and provide a very brief account of GANs used for sequential data. Gonog and Zhou [26] provide a short introduction to GANs, their theory and explores the variety of plausible models, again listing their applications in image and video manipulation with a mention of sequential data (NLP). In another review paper [27], the authors give an overview of GAN fundamentals, variants, and applications. Sequential data applications are

[1] `https://github.com/sheqi/GAN_Review/tree/master/GAN_TS`

mentioned in the form of music and speech synthesis.

As with most review papers, Yinka-Banjo and Ugot give an introduction and overview of Generative Adversarial Networks [28]. However, they also review GANs as adversarial detectors and discuss their limitations applied to cybersecurity. Yi, Walia, and Babyn [29] give a review of GANs and their applications in medical imaging, how they can be used in clinical research and potentially deployed to help practising clinicians. There is no mention of time series data use cases.

A recurring theme in these papers focuses on GAN variants which have mostly been applied to the computer vision domain. To the best of our knowledge, no review has been conducted with the main focus on time series GANs. While these reviews have mentioned the application of these GANs in generating sequential data, they have scratched the surface of what is becoming a growing body of research.

We contribute to lessening this gap by presenting our work which is concerned with presenting the latest up-to-date research around time series GANs, their architecture, loss functions, evaluation metrics, trade-offs and approaches to privacy preservation of their datasets.

2.3 Generative Adversarial Networks

GANs belong to the family of generative models and are an alternative method of generating synthetic data that do not require domain expertise. They were conceived in the paper by Goodfellow in 2014, where a multi-layer perceptron was used for both the discriminator and the generator [21]. Figure 2.2 shows a simple example of the GAN architecture and the game that the neural network models play. Radford *et al.* (2015) subsequently developed the deep convolutional generative adversarial network (DCGAN) to generate synthetic images [30]. Since then, researchers have continuously improved on the early GAN architectures, loss functions, and evaluation metrics while innovating on their potential contributions to real-world applications. To appreciate why there has been such concerted activity in the further development of GAN technologies it is important to understand the limitations of early architectures and the challenges these presented. We describe these next, and in so doing, prepare the reader for the particular manifestation of these challenges in the more specific context of time series.

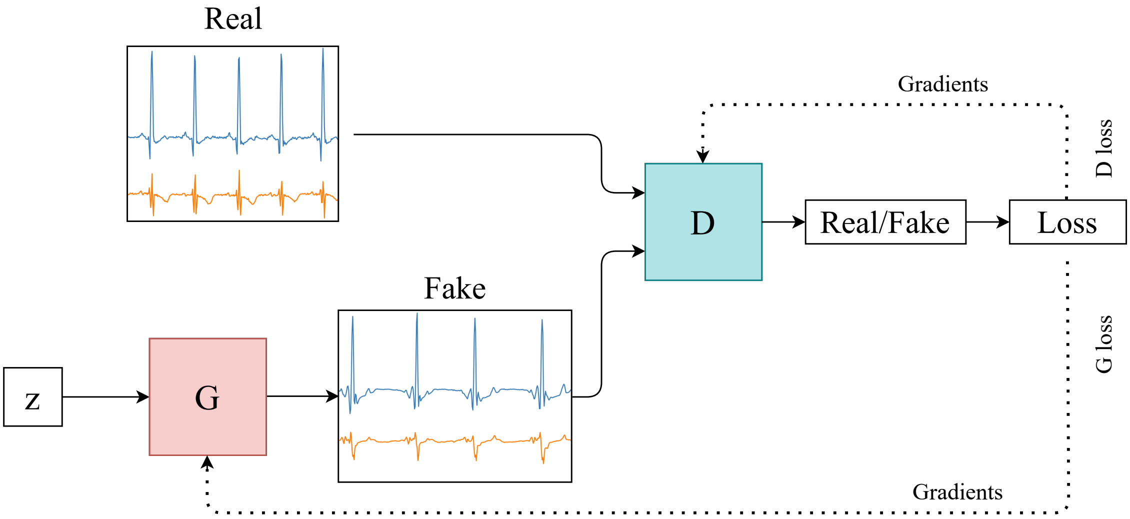

Figure 2.2: Generative Adversarial Network

2.3.1 Challenges

There are three main challenges in the area of time series GANs, i.e., training stability, evaluation and privacy risk associated with synthetic data created by GANs. We are going to explain these three challenges as follows.

Training stability. The original work [21] has already proved the global optimality and the convergence of GANs during training. It still highlights the instability problem that can arise when training a GAN. Two problems are well-studied in the literature 1. vanishing gradients and 2. mode collapse. The vanishing gradient is caused by directly optimising loss presented in equation (1.5). When D reaches the optimality, optimising the equation (1.5) for G can be converted to minimising the Jensen-Shannon (JS) divergence (details of derivation can refer to section 5 in [24]) between the real data distribution p_r and the generated data distribution p_g:

$$L_G = 2 \cdot JS(p_r \| p_g) - 2 \cdot \log 2 \tag{2.2}$$

L_G stays constant ($\log 2 = 0.693$) when there is no overlap between p_r and p_g. The gradient for G using this loss is near 0 in the situation where the distributions are far apart. Non-zero gradient for G only exists when p_r and p_g have substantial overlap. In practice, the possibility that p_r and p_g are not intersected or have negligible overlap is very high [31]. In order to get rid of the vanishing gradient problem for G, the original GAN work [21] highlights that the minimisation of

$$L_G = -\mathbb{E}_{\mathbf{x}\sim p_g}\mathrm{log}[D(\mathbf{x})] \tag{2.3}$$

for updating G. This strategy is able to avoid the vanishing gradient problem but leads to the mode collapse issue. A mode is simply an output that contains either real (p_r) or generated (p_g) data. A single mode means the generator only produces a single type of output. Optimising equation (2.3) can be converted to optimising the reverse Kullback–Leibler (KL) divergence. KL is given in equation (2.4), where p_r is the real data distribution, p_g is the generated data distribution and n is the number of samples in the data. When p_r contains multiple modes, p_g chooses to recover a single mode and ignores other modes when optimising the reverse KL divergence. Considering this case, G trained using equation (2.3) might be only able to generate few modes from real data. These problems can be amended by changing the architecture or the loss function, which are reviewed by Wang *et al.* [24] in detail.

$$KL(p_g||p_r) = \sum_{i=0}^{n} p_r log(\frac{p_r}{p_g}) \tag{2.4}$$

Evaluation. A wide range of evaluation metrics has been proposed to evaluate the performance of GANs [22], [32]–[34]. Current evaluations of GANs in computer vision are normally designed to consider two perspectives i.e., quality and quantity of generated data. The most representative qualitative metric is to use human annotation to determine the visual quality of the generated images. Quantitative metrics compare statistical properties between generated and real images i.e., two-sample tests such as maximum mean discrepancy (MMD) [35], Inception Score [36] and Fréchet Inception Distance (FID) [37]. Contrary to evaluating image-based GANs, it is difficult to evaluate time series data from human psycho-perceptual sense qualitatively. In terms of qualitatively evaluating time series based GANs, it normally conducts t-SNE [38] and PCA [39] analyses to visualize how well the generated distributions resemble the original distributions [40]. Quantitative evaluation for time series based GANs can be done by deploying two-sample tests similar to image-based GANs.

Privacy risk. Apart from evaluating the performance of GANs, a wide range of methods have been used to asses the privacy risk associated with synthetic data created by GANs. Choi *et al.* performed tests for presence disclosure and attribute disclosure. In contrast, others utilised a three-sample test on the training, test, and synthetic data to identify if the synthetic data has overfitted to the training data [41], [42]. It has been shown that common methods of de-identifying data do not prevent attackers from re-identifying individuals using additional data [43], [44]. Sensitive data is usually de-identified by removing personally identifiable information (PII). However, work is ongoing to create frameworks to link different sources of publicly available information together using alternative information to PII. Malin *et al.* developed a software program, REID, to connect individuals contained in publicly available hospital discharge data with their unique DNA records [44]. Culnane *et al.* re-identified individuals in a de-identified open dataset of Australian medical billing records using unencrypted parts of the records and known information about individuals from other sources [45]. Hejblum *et al.* developed a probabilistic method to link de-identified electronic health record (EHR) data of patients with rheumatoid arthritis [46]. The re-identification of individuals in publicly available datasets can lead to the exposure of their sensitive health information. Health data has been categorised as special personal data by GDPR and is subject to a higher level of protection under the Data Protection Act 2018 (Section36(2)) [10]. Consequently, concerned researchers must find alternative methods of protecting sensitive health data to minimise the risk of re-identification. This will be addressed in Section 2.7.

2.3.2 Popular Datasets

Unlike image-based datasets (CIFAR, MNIST, ImageNet [47]–[49]) there are no standardised or commonly used benchmarking datasets for time series generation. However, we have compiled a list of some of the more popular datasets implemented in the reviewed works, and they are listed in Table 2.1 along with their year of release/update, data type and how many instances and attributes they contain. What makes these datasets interesting/applicable to time series GANs is that they are signals made up of highly complex waveforms (physiological and audio) and contain important temporal dynamics crucial to preserve when generating new samples. Furthermore, these signals are the data type that have become highly regulated and can stand to benefit from being leveraged by GANs to generate further volumes of this kind of data.

There exist two repositories; the UCR Time Series Classification/Clustering database [50], and the UCI Machine Learning repository [51] that make available several time series datasets. Despite this, there is still no consensus on a standardised dataset used for benchmarking time series GANs, which may be due to the ‘continuous’ nature of the architecture dimensions. GANs designed for continuous time series generation often differ in the length of their input sequence due to either author preference or the constraints placed on their architecture for the generated data’s downstream tasks.

Table 2.1: Popular Datasets used in the reviewed works.

Name (Year)	**Data Type**	**Instances**	**Attributes**
Oxford-Man Institute "realised library" (Updated Daily)	Real Multivariate Time Series	>2.6M	5
EEG Motor Movement/Imagery Dataset (2004)	Real Multivariate Time Series	1,500	64
ECG 200 (2001)	Real Univariate Time Series	200	1
Epileptic Seizure Recognition Dataset (2001)	Real Multivariate Time Series	11,500	179
TwoLeadECG (2015)	Real Multivariate Time Series	1,162	2
MIMIC-III (2016)	Real, Integer & Categorical Multivariate Time Series	-	-
EPILEPSIAE project database (2012)	Real Multivariate Time Series	30	-
PhysioNet/CinC (2015)	Real Multivariate Time Series	750	4
Wrist PPG During Exercise (2017)	Real Multivariate Time Series	19	14
MIT-BIH Arrhythmia Database (2001)	Real Multivariate Time Series	201	2
PhysioNet/CinC (2012)	Real, Integer & Categorical Multivariate Time Series	12000	43
KDD Cup Dataset (2018)	Real, Integer & Categorical Multivariate Time Series	282	3
PeMS Database (Updated Daily)	Integer & Categorical Multivariate Time Series	-	8
Nottingham Music Database (2003)	Special Text Format Time Series	1000	-

2.4 Taxonomy of Time Series based GANs

We propose a taxonomy of the following time series based GANs based on two distinct variant types: **discrete variants** (discrete time series) and **continuous variants** (continuous time series). A discrete time series consists of data points separated by time intervals. This type of data might have: (1) a data-reporting interval that is infrequent (e.g., 1 point per minute) or irregular (e.g., whenever a user logs in) and (2) gaps where values are missing due to reporting interruptions (e.g., intermittent server or network downtime in a network traffic application). Discrete time series generation involves generating sequences that may have a temporal dependency but contain discrete tokens; these can be commonly found in electronic health records (International Classification of Diseases 9 codes) and text generation.

In the natural, analogue world, a continuous time series has a data value corresponding to every moment in time. However, we cannot have a continuous signal on a computer as it only works in discrete numbers. Digital electronics convert analogue signals to their digital equivalent via quantisation, introducing errors into the signal. To represent these analogue signals in the digital domain, they must be converted using an analogue-to-digital converter (ADC) which is underpinned by the Nyquist sampling theorem. Nyquist sampling specifies that a sinusoidal function in time or distance can be regenerated with no loss of information as long as it is sampled at a frequency greater than or equal to twice the highest frequency present in the signal [52]. If the sampling rate is too low, the original signal will be distorted, or show aliasing effects, when reproduced.

For simplicity, in this thesis, continuous data generation is concerned with generating a real-valued signal x with temporal dependencies where $x \in \mathbb{R}$. See Figure 2.3 for examples of discrete and continuous time series signals.

Challenges with discrete time series generation. GANs struggle with discrete data generation due to the zero gradient nearly everywhere, i.e., the distribution on discrete objects are not differentiable with respect to their parameters [53], [54]. This limitation makes the generator untrainable using backpropagation alone. The

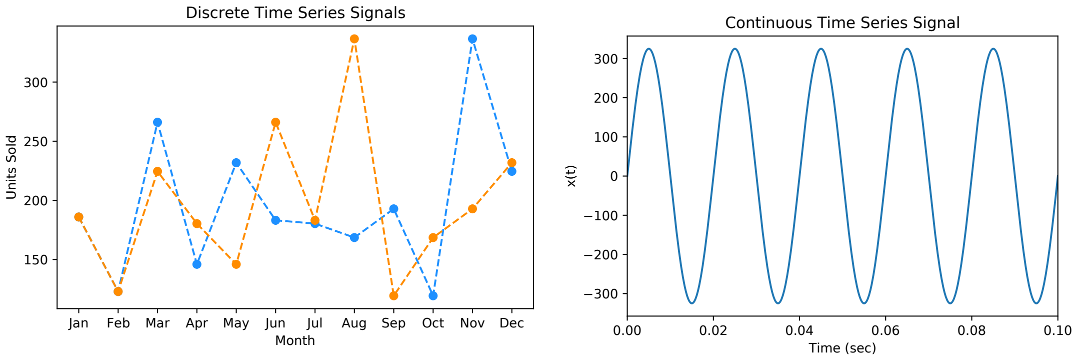

Figure 2.3: Example plots of discrete (left) and continuous time series (right).

generator starts with a random sampling and a deterministic transform guided via the gradient of the loss from the discriminator with respect to the output produced by G and the training dataset. This loss leads to a slight change in G's output, pushing it closer to the desired output. Making slight changes to continuous numbers make sense; adding 0.001 to a value of 10 in financial time series data will bring it to 10.001. However, a discrete token such as the word 'penguin' cannot simply undergo the addition of 0.001 as the sum 'penguin+0.001' makes no sense. What's important here is the impossibility for the generator to jump from one discrete token to the next because the small change gives the token a new value that does not correspond to any other token over that limited discrete space [55]. This is because there exists 0 probability in the space between these tokens, unlike with continuous data.

Challenges with continuous time series generation. Modelling continuous time series data presents a different problem for GANs, which are inherently designed to model continuous data, albeit most commonly in the form of images. The temporal nature of continuous data in time series presents an extra layer of difficulty. Complex correlations exist between the temporal features and their attributes, e.g., if using multichannel biometric/physiological data, the ECG characteristics will depend on the individual's age and/or health. Also, long-term correlations exist in the data, which are not necessarily fixed in dimension compared to image-based data under a fixed dimension. Transforming image dimensions may lead to a degradation in image quality, but it is a recognised practice. This operation becomes more difficult with continuous time series data as there is no standardised dimension used across time series GANs architectures, which means that benchmarking their performances becomes difficult.

Since their inception in 2014, GANs have shown great success in generating high-quality synthetic images indistinguishable from real images [56]–[58]. While the focus to date has been on developing GANs for improved media generation, there is a growing consensus that GANs can be used for more than image generation and manipulation, which has led to a movement towards generating time series data with GANs.

Recurrent neural networks (Figure 2.4, left), due to their loop-like structure, are perfect for sequential data applications but by themselves lack the ability to learn long-term dependencies that might be crucial in forecasting future values based on past. Long-short Term Memory networks (LSTM) (Figure 2.4, right) are a specific kind of RNN that have the ability to remember information for long periods of time and, in turn, learn these long-term dependencies that the standard RNN is not capable of doing. In most works reviewed in this chapter, the majority of the RNN based architectures are utilising the LSTM cell.

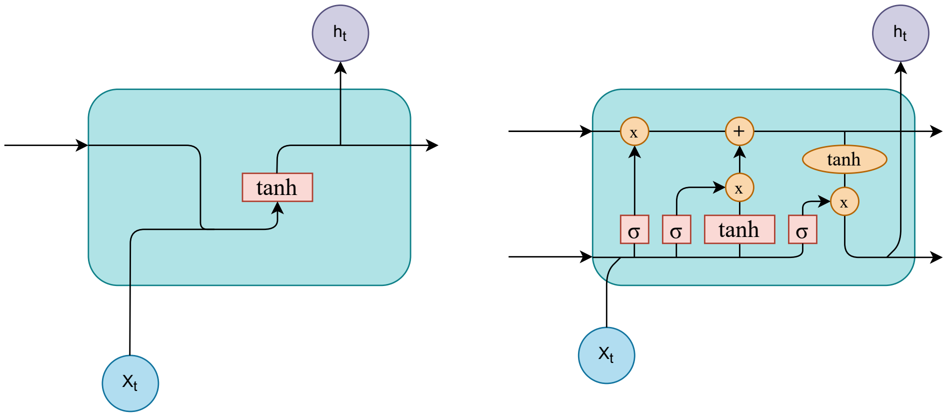

Figure 2.4: Block diagram of (left) a standard RNN and (right) LSTM cell. Here, x_t is the input signal at time t, h_t is the hidden state at time t, σ is the sigmoid operation and tanh is the hyperbolic tangent function.

RNNs can model sequential data such as financial data, medical data, text, and speech, and they have been the foundational architecture for time series GANs. A recurrent GAN (RGAN) was first proposed in 2016. The generator contained a recurrent feedback loop that used both the input and hidden states at each time step to generate the final output [59]. Recurrent GANs often utilise Long Short-Term Memory neural networks in their generative models to avoid the vanishing gradient problem associated with more traditional recurrent networks [60]. In the section that follows, we chronologically present time series GANs that have either contributed significantly to this space or have made some of the most recent novel advancements in addressing the challenges mentioned above.

2.4.1 Discrete-variant GANs

Sequence GAN (SeqGAN) (Sept. 2016)

Yu *et al.* proposed a sequential data generation framework [53] that could address the issues with generating discrete data as previously mentioned in 2.4. This approach outperformed previous methods for generative modelling on real-world tasks, including; a maximum likelihood estimation (MLE) trained LSTM, scheduled sampling [61], and Policy Gradient with bilingual evaluation understudy (PG-BLEU) [62]. SeqGAN's generative model comprises RNNs with LSTM cells, and its discriminative model is a convolutional neural network. Given a dataset of structured sequences the authors train G to produce a synthetic sequence $Y_{1:T} = (y_1..., y_t..., y_T), y_t \in \mathcal{Y}$ where $\mathcal{Y}$ is defined as the vocabulary of candidate tokens. G is updated by a policy gradient and Monte Carlo (MC) search on the expected reward from D, see Figure 2.5. The authors used two datasets for their experiments. A Chinese poem dataset [63] and a Barack Obama Speech dataset [64] with Adam optimisers and a batch size of 64. Their experiments are available online[2].

Although the purpose of SeqGAN is to generate discrete sequential data, it opened the door to other GANs in generating continuous sequential and time series data. The authors use a synthetic dataset whose distribution is generated from a randomly initialised LSTM following a normal distribution. They also compare the generated data to real-world examples of poems, speech-language and music. SeqGAN showed competitive performance in generating the sequences and contributed heavily towards

[2] SeqGAN GitHub: https://github.com/LantaoYu/SeqGAN/

the further development of the continuous sequential GANs.

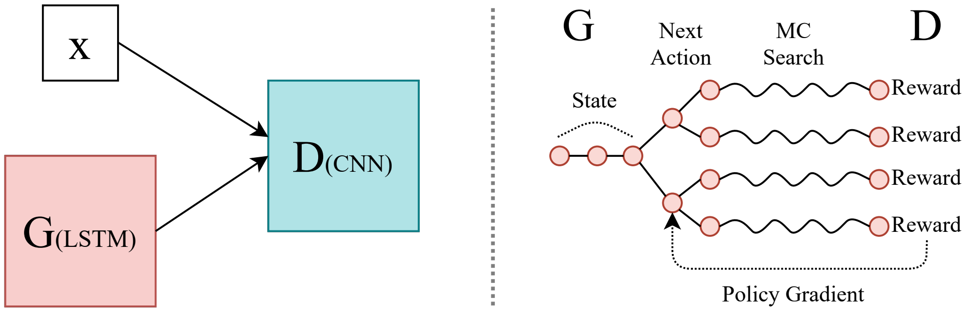

Figure 2.5: SeqGAN: D is trained over real and generated data (left), whereas G is trained by policy gradient where the final reward signal is provided by D and is passed back to the intermediate action value via Monte Carlo search (right).

Quant GAN (Jul. 2019)

Quant GAN is a data-driven model that aims to capture long-range dependencies in financial time series data such as volatility clusters. Both the generator and discriminator use Temporal Convolutional Networks (TCN) with skip connections [65] which are essentially dilated causal convolutional networks. They have the advantage of being suited to model long-range dependencies in continuous sequential data. The generator function is a novel stochastic volatility NN (SVNN) that consists of a volatility and drift TCN. Temporal blocks are the modules used in the TCN that consist of two dilated causal convolutions layers (Figure 2.6) and two Parametric Rectified Linear Units (PReLU) as activation functions. Data generated by G is passed to D to produce outputs, which can then be averaged to give an MC estimate of D's loss function. The authors used a dataset of daily spot-prices of the S&P 500 from May 2009 until December 2018.

The authors aim to capture long-range dependencies in financial time series; however, modelling the series in continuous time over these long time frames would blow up the models' computational complexity. Therefore, this method models the time series in discrete time. The authors report that this approach is capable of outperforming more conventional models from mathematical finance (Constrained SVNN and generalised autoregressive conditional heteroskedasticity (GARCH) [66])

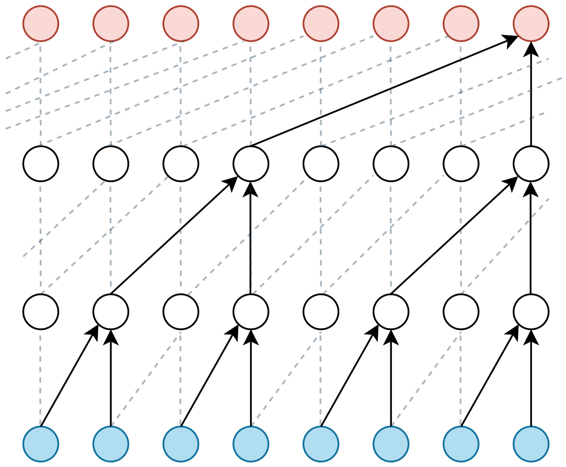

Figure 2.6: Dilated causal convolutional layer.

but state that there remain issues that need to be resolved for this approach to become widely adopted. One such issue concerns the need for a unified metric for quantifying the performance of these GANs, a point we discuss further in Section 2.6.

2.4.2 Continuous-variant GANs

Continuous RNN-GAN (C-RNN-GAN) (Nov. 2016)

In previous works, RNNs have been applied to modelling music but have generally used a symbolic representation to model this type of sequential data. Mogren proposed the C-RNN-GAN (Figure 2.7), one of the first examples of using GANs to generate continuous sequential data [67]. The generator is an RNN, and the discriminator a bidirectional RNN, which allows the discriminator to take the sequence context in both directions. The RNNs used in this work were two stacked LSTM layers, with each cell containing 350 hidden units. The loss functions can be seen in (2.5, 2.6), where $z^{(i)}$ is a sequence of uniform random vectors in $[0,1]^k$, and $x^{(i)}$ is a sequence from the training data. k is the dimensionality of the data in the random sequence and m is the number of data samples in the batch.

$$L_G = \frac{1}{m}\sum_{i=1}^{m}\log\Big(1 - D(G(z^{(i)}))\Big) \tag{2.5}$$

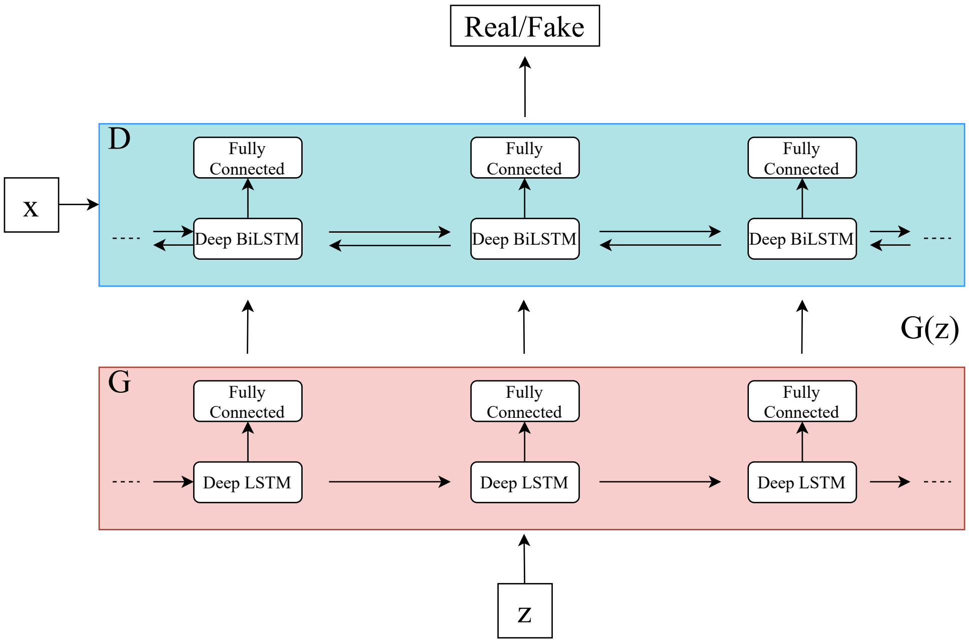

Figure 2.7: Structure of C-RNN-GAN's generator and discriminator.

$$L_D = \frac{1}{m}\sum_{i=1}^{m}\left[-\log\left(D(x^{(i)})\right) - \log\left(1 - D(G(z^{(i)}))\right)\right] \tag{2.6}$$

The C-RNN-GAN is trained with BPTT and mini-batch stochastic gradient descent with L2 regularisation on the weights of both G and D. Freezing was applied to both G and D when one network becomes too strong relative to the other. The dataset used was 3697 midi files from 160 different composers of classical music with a batch size of 20. Adam and Gradient Descent Optimisers were used during training; full implementation details are available online[3]. Overall the C-RNN-GAN was capable of learning the characteristics of continuous sequential data and, in turn, generate music. However, the author stated that their approach still needs work, particularly in rigorous evaluation of the generated data quality.

Recurrent Conditional GAN (RCGAN) (2017)

RCGAN for continuous data generation [42] differs architecturally from the C-RNN-GAN. Although the RNN LSTM is used, the discriminator is unidirectional, and the outputs of G are not fed back as inputs at the next time step. There is also additional

[3] C-RNN-GAN GitHub: https://github.com/olofmogren/c-rnn-gan/

information that the model is conditioned on, which makes for a conditional RGAN; see the layout of the model in Figure 2.8. The purpose of the RCGAN and RGAN in this work is to generate continuous time series with a focus on medical data intended for use in downstream tasks, and this was one of the first works in this area. The loss functions can be seen in Equations (2.7, 2.8) where CE is the average cross-entropy between two sequences. X_n are samples drawn from the training dataset. y_n is the adversarial ground truth; for real sequences, it is a vector of 1s, and conversely, for generated or synthetic sequences, it is a vector of 0s. Z_n is a sequence of points sampled from the latent space, and the valid adversarial ground truth is written here as $\mathbf{1}$.

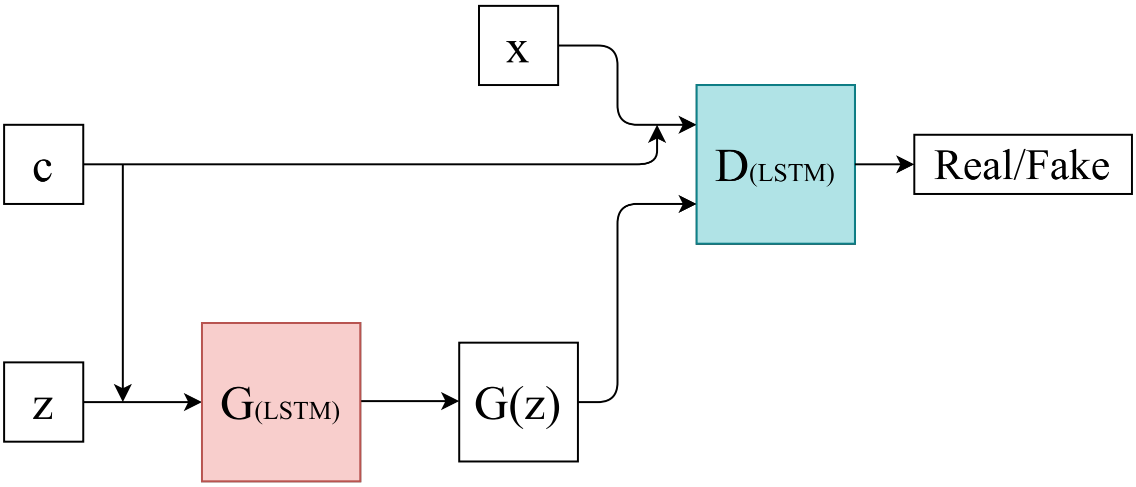

Figure 2.8: RCGAN architecture with conditional input **c**, input data **x** and latent variable **z**.

$$L_D(X_n, y_n) = -CE(D(X_n), y_n) \tag{2.7}$$

$$L_G(Z_n) = L_D(G(Z_n), \mathbf{1}) = -CE(D(G(Z_n)), \mathbf{1}) \tag{2.8}$$

In the conditional case, the inputs to D and G are concatenated with some conditional information c_n. This variant of an RNN-GAN facilitates the generation of a synthetic continuous time series dataset with associated labels. Experiments were carried out on generated sine waves, smooth functions sampled from a Gaussian process with a zero-valued mean function, MNIST dataset as a sequence, and the Philips eICU database [68]. A batch size of 28 with Adam and Gradient Descent Optimisers were used for training. The authors propose a novel method for evaluating their model, which is discussed further in Section 2.6. Full experimental details can be found online[4].

Sequentially Coupled GAN (SC-GAN) (Apr. 2019)

SC-GAN aims to generate patient-centric medical data to inform of a patient's current state and generate a recommended medication dosage based on the state [69]. It consists of two coupled generators tasked with producing two outcomes, one for the current state of an individual and the other for a recommended medication dosage based on the individual's state. The discriminator is a two-layer bidirectional LSTM, and the coupled generators are both two-layer unidirectional LSTMs. See Figure 2.9 for further details of the architecture.

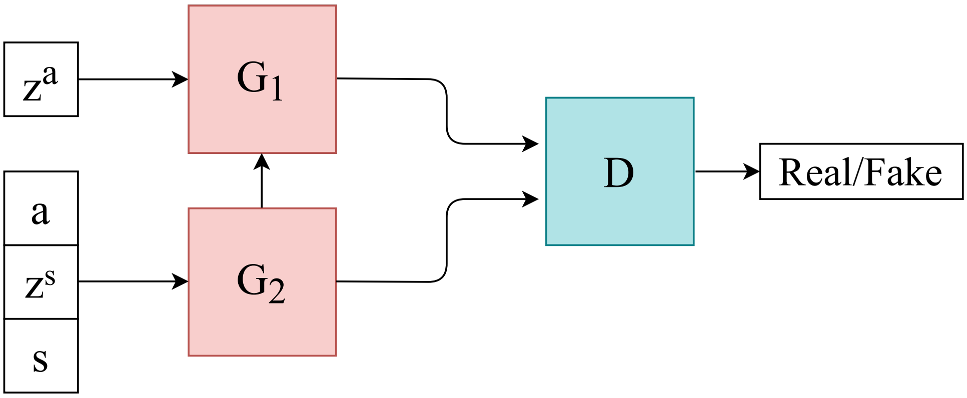

Figure 2.9: SC-GAN architecture. Here a is the medical dosage data, s is the continuous patient state data, z_t^a is the concatenation of s_t and $\hat{z}_t^a$ which is a random noise sequence. z_t^s is a concatenation of s_{t-1}, a_{t-1} and $\hat{z}_t^s$ that is a random noise sequence.

G_1 generates the recommended medication dosage data $(\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_T)$ with a combined input of the sequential continuous patient state data $(\mathbf{s}_0, \mathbf{s}_1, ..., \mathbf{s}_{T-1})$ and a random noise sequence $(\hat{\mathbf{z}}_0^a, \hat{\mathbf{z}}_1^a, ..., \hat{\mathbf{z}}_{T-1}^a)$ sampled from a uniform distribution. At each time step t the input $\mathbf{z}_t^a$ of G_1 is the concatenation of $\mathbf{s}_t$ and $\hat{\mathbf{z}}_t^a$.

[4] RCGAN GitHub: https://github.com/ratschlab/RGAN/

Conversely, G_2 is tasked with generating the patient state data $\mathbf{s}_t$ and at each time step the input $\mathbf{z}_t^s$ is the concatenation of $\mathbf{s}_{t-1}$, $\mathbf{a}_{t-1}$ and $\hat{\mathbf{z}}_t^s$. This means that the outputs from G_1 and G_2 are the inputs to one another. Combining the generators together leads to the following loss function:

$$L_G = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T}\log\Big(1 - D(G(\mathbf{z}_{i,t}))\Big) \tag{2.9}$$

$$G(\mathbf{z}_{i,t}) = \Big[G_1(\mathbf{z}_{i,t}^a); G_2(\mathbf{z}_{i,t}^s)\Big] \tag{2.10}$$

Where N is the number of patients and T is the time length of the patient record. The SC-GAN has a supervised pretraining step for the generators to avoid an excessively strong D that uses the least-squares loss.

The discriminator is tasked with classifying the sequential patient-centric records as real or synthetic, and the loss function is defined as:

$$L_D = -\frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T}\Bigg(\log\Big((D(\mathbf{x}_{i,t})\Big) + \log\Big((1 - D(G(\mathbf{z}_{i,t}))\Big)\Bigg) \tag{2.11}$$

where $\mathbf{x}_{i,t} = [\mathbf{s}_t; \mathbf{a}_t]$. This model contains novel coupled generators that coordinate to generate patient state and medication dosage data. It has performance close to real data for the treatment recommendation task. The dataset used in this experiment is MIMIC-III [70]. The authors benchmark their SC-GAN against variants of SeqGAN, C-RNN-GAN, and RCGAN and observe their model to be the best performing for this specific use case.

Noise Reduction GAN (NR-GAN) (Oct. 2019)

NR-GAN is designed for noise reduction in continuous time series signals but more specifically has been implemented for noise reduction in mice electroencephalogram (EEG) signals [71]. This dataset was provided by the International Institute for Integrative Sleep Medicine (IIIS). EEG is the measure of the brain's electrical activity, and it commonly contains significant noise artefact. NR-GAN's core idea is to reduce or remove the noise present in the frequency domain representation of an EEG signal. The architecture of G is a two-layer 1-D CNN with a fully connected layer at the output. D contains almost the same two-layer 1-D CNN structure with the fully-connected layer replaced by a softmax layer to calculate the probability the input belongs to the training set. The loss functions are defined in (2.12, 2.13) as:

$$L_G = \sum_{x \in S_{ns}} \left[\log\left(1 - D(G(x))\right) + \alpha \left\| x - G(x) \right\|^2 \right] \tag{2.12}$$

$$L_D = \sum_{x \in S_{ns}} \left[\log\left(D(G(x)) \right) \right] + \sum_{y \in S_{cs}} \left[\log\left(1 - D(y)\right) \right] \tag{2.13}$$

where S_{ns} and S_{cs} are the noisy and clear EEG signals, respectively. α is a hyperparameter that essentially controls the aggressiveness of noise reduction; the authors chose a value of $\alpha = 0.0001$.

For this work, the generator does not sample from a latent space; rather, it attempts to generate the clear signal from the noisy EEG signal input, see Figure 2.10. The authors found that the NR-GAN is competitive with classical frequency filters in terms of noise reduction. They also state that the experimental conditions may favour the NR-GAN and list some limitations in terms of the amount of noise NR-GAN can handle and the influence of α. However, this is a novel method for noise reduction in continuous sequential data using GANs.

Time GAN (Dec. 2019)

TimeGAN provides a framework that utilises both the conventional unsupervised GAN training method and the more controllable supervised learning approach [40]. By combining an unsupervised GAN network with a supervised autoregressive model, the network aims to generate time series with preserved temporal dynamics. The architecture of the TimeGAN framework is illustrated in Figure 2.11. The input to the framework is considered to consist of two elements, a static feature and a

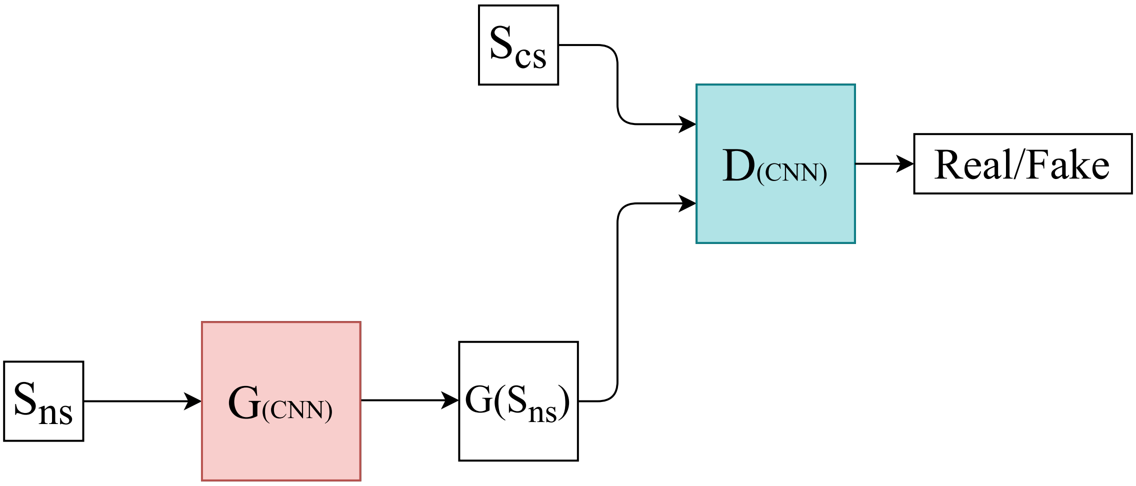

Figure 2.10: NR-GAN architecture with noisy EEG input S_{ns}, clean input data S_{cs}.

temporal feature. **s** represents a vector of static features and **x** of temporal features at the input to the encoder. The generator takes a tuple of static and temporal random feature vectors drawn from a known distribution. The real and synthetic latent codes **h** and $\hat{\mathbf{h}}$ are used to calculate the supervised loss element of this network. The discriminator receives the tuple of real and synthetic latent codes and classifies them as either real (y) or synthetic ($\hat{y}$), the $\tilde{}$ operator denotes the sample as either real or fake.

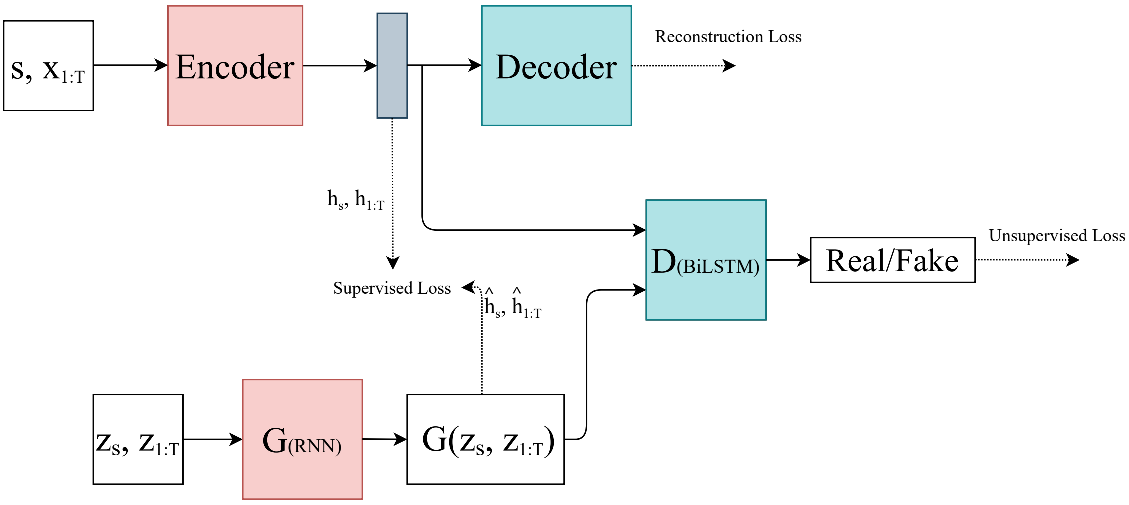

Figure 2.11: TimeGAN architecture.

The three losses used in TimeGAN are calculated as follows:

$$L_R = \mathbb{E}_{s,x_{1:T}\sim p}\left[\|s-\tilde{s}\|_2 + \sum_t \|x_t - \tilde{x}_t\|_2\right] \tag{2.14}$$

$$L_U = \mathbb{E}_{s,x_{1:T}\sim p}\left[\log(y_s)+\sum_t \log(y_t)\right]+\mathbb{E}_{s,x_{1:T}\sim \hat{p}}\left[\log(1-\hat{y}_s)+\sum_t \log(1-\hat{y}_t)\right] \quad (2.15)$$

$$L_S = \mathbb{E}_{s,x_{1:T}\sim p}\left[\sum_t \left\| h_t - g_X(h_S, h_{t-1}, z_t)\right\|_2\right] \quad (2.16)$$

Where L_R, L_U and L_S are the reconstruction, unsupervised and supervised losses respectively. The creators of TimeGAN conducted experiments on generating sine waves, stocks (daily historical Google stocks data from 2004 to 2019), energy (UCI Appliances energy prediction dataset) [51], and event (private lung cancer pathways) datasets. A batch size of 128 and Adam optimiser were used for training, implementation details are available online[5]. The authors demonstrated improvements over other state-of-the-art time series GANs such as RCGAN, C-RNN-GAN and WaveGAN.

Conditional Sig-Wasserstein GAN (SigCWGAN) (Jun. 2020)

A problem addressed by [72] is that long time series data streams can greatly increase the dimensionality requirements of generative modelling, which may render such approaches infeasible. To counter this problem, the authors develop a metric named Signature Wasserstein-1 (Sig-W_1) that captures time series models' temporal dependency and uses it as a discriminator in a time series GAN. It provides an abstract and universal description of complex data streams and does not require costly computation like the Wasserstein metric. A novel generator is also presented that is named conditional autoregressive feed-forward neural network (AR-FNN) that captures the auto-regressive nature of the time series. The generator is capable of mapping past series and noise into future series. For a rigorous mathematical description of their method, the interested reader should consult [72].

For the AR-FNN generator the idea is to obtain the step-q estimator $\hat{X}^{(t)}_{t+1:t+q}$.

[5] TimeGAN GitHub: https://github.com/jsyoon0823/TimeGAN

The loss function for the network is defined as:

$$L = \sum_t \left| \mathbb{E}_\mu \left[S_M(X_{t+1:t+q}) \middle| X_{t-p+1:t} \right] - \mathbb{E}_v \left[S_M(\hat{X}^{(t)}_{t+1:t+q}) \middle| X_{t-p+1:t} \right] \right| \tag{2.17}$$

Where v and μ are the conditional distributions induced by the real data and synthetic generator, respectively. $X_{t-p+1:t}$ is the true past path, $\hat{X}^{(t)}_{t+1:t+q}$ is the forecasted next path and $X_{t+1:t+q}$ is the true forecast value. S_M is the truncated signature of path X of degree M. Further details of the author's algorithm can be found in the appendix of the original paper. The authors state that SigCWGAN eliminates the problem of approximating a costly D and simplifies training. It is reported to achieve state-of-the-art results on synthetic and empirical datasets compared to TimeGAN, RCGAN and Generative Moment Matching Networks (GMMN) [73]. The empirical dataset consists of the S&P 500 index (SPX) and Dow Jones index (DJI) and their realized volatility, which is retrieved from the Oxford-Man Institute's "realised library" [74]. A batch size of 200 with the Adam optimiser was used for training[6].

Decision Aware Time series conditional GAN (DAT-CGAN) (Sept. 2020)

This framework is designed to provide support for end-users decision processes, specifically in financial portfolio choices. It uses a multi-Wasserstein loss on structured decision-related quantities [75]. The discriminator loss and generator loss are defined in Equations (2.18) and (2.19) respectively. For further details on the loss functions, see Section 3 of the original paper and equations (2.20) to (2.23) .

$$L_D = \inf_{\eta} \sup_{\gamma_k, \theta_{j,k}} \sum_{k=1}^{K} \omega_k \Big(\mathbb{E}^r_k - \mathbb{E}^{G_\eta}_k \Big) + \sum_{k=1}^{K} \sum_{j=1}^{J} \lambda_{j,k} (\mathbb{E}^{f,R}_{j,k} - \mathbb{E}^{f,G_\eta}_{j,k}) \tag{2.18}$$

$$L_G = \inf_{\eta} - \sum_k \omega_k \mathbb{E}^{G_\eta}_k - \sum_{k,j} \lambda_{j,k} \mathbb{E}^{f,G_\eta}_{j,k} \tag{2.19}$$

[6] SigCWGAN GitHub: https://github.com/SigCGANs/Conditional-Sig-Wasserstein-GANs/

$$\mathbb{E}_k^r = \mathbb{E}_{r_{t+k} \sim P(r_{t+k}|x_t)}[D_{\gamma k}(r_{t+k}, x_t)] \tag{2.20}$$

$$\mathbb{E}_k^{G_\eta} = \mathbb{E}_{z_{t,k} \sim P(z_{t,k})}[D_{\gamma k}(r'_{t,k}, x_t)] \tag{2.21}$$

$$\mathbb{E}_{j,k}^{f,R} = \mathbb{E}_{R_{t,k} \sim P(R_{t,k}|x_t)}[D_{\theta_{j,k}}(f_{j,k}(R_{t,k}, x_t), x_t)] \tag{2.22}$$

$$\mathbb{E}_{j,k}^{f,G_\eta} = \mathbb{E}_{Z_{t,k} \sim P(Z_{t,k})}[D_{\theta_{j,k}}(f_{j,k}(R'_{t,k}, x_t), x_t)] \tag{2.23}$$

We offer a full description of all terms used in equations (2.18) and (2.19). $D_{\gamma k}$ is the discriminator for the data at look ahead period k with respect to parameters γ. G_η is the generator with parameters η. As this is the conditional case, x_t is the conditioning variable containing relevant information up to time t. $r'_{t,k} = G_\eta(z_{t,k}, x_t)$ is defined as the synthetic data at look ahead point k where the noise is $z_{t,k}$. The discriminator for decision-related quantity j at look ahead period k with respect to parameters $\theta_{j,k}$ is defined as $D_{\theta_{j,k}}$. These decision related quantities may include mean and covariance for example. $f_{j,k}(R_{t,k}, x_t)$ represents the decision related quantity. Finally, ω_k and $\lambda_{j,k}$ are weights and inf and sup are the infimum and supremum or greatest lower bound and least upper bound of a non-empty subset, respectively.

The generator is a two-layer feed-forward neural network for each input which are assets in this case. G outputs asset returns that are used to compute decision-related quantities. These quantities are fed into D, which is also a two-layer feed-forward NN. Further details about the architecture can be found in the appendix of [75]. The dataset used is daily price data for each of four U.S. Exchange-traded fund (ETFs), i.e., Material (XLB), Energy (XLE), Financial (XLF) and Industrial (XLI) ETFs, from 1999 to 2016. The authors found this model capable of high-fidelity time series generation that supports decision processes by end-users due to incorporating a decision-aware loss function. However, this approach's limitation is that the computational complexity of this model is vast and requires one month of training time for a single generative model.

Synthetic biomedical Signals GAN (SynSigGAN) (Dec. 2020)

SynSigGAN is designed to generate different kinds of continuous physiological/biomedical signal data [76]. It is capable of generating electrocardiogram (ECG), electroencephalogram (EEG), electromyography (EMG), and photoplethysmography (PPG) from MIT-BIH Arrhythmia database [77], Siena Scalp EEG database [78] and BIDMC PPG and Respiration dataset [79]. A novel GAN architecture is proposed here that uses a bidirectional grid long short term memory (BiGridLSTM) for the generator (Figure 2.12) and a CNN for the discriminator. The BiGridLSTM is a combination of a double GridLSTM (a version of LSTM that can represent the LSTMs in a multidimensional grid) with two directions that can combat the gradient phenomenon from two dimensions and found to work well in time sequence problems. The authors used the value function defined previously in equation (1.5).

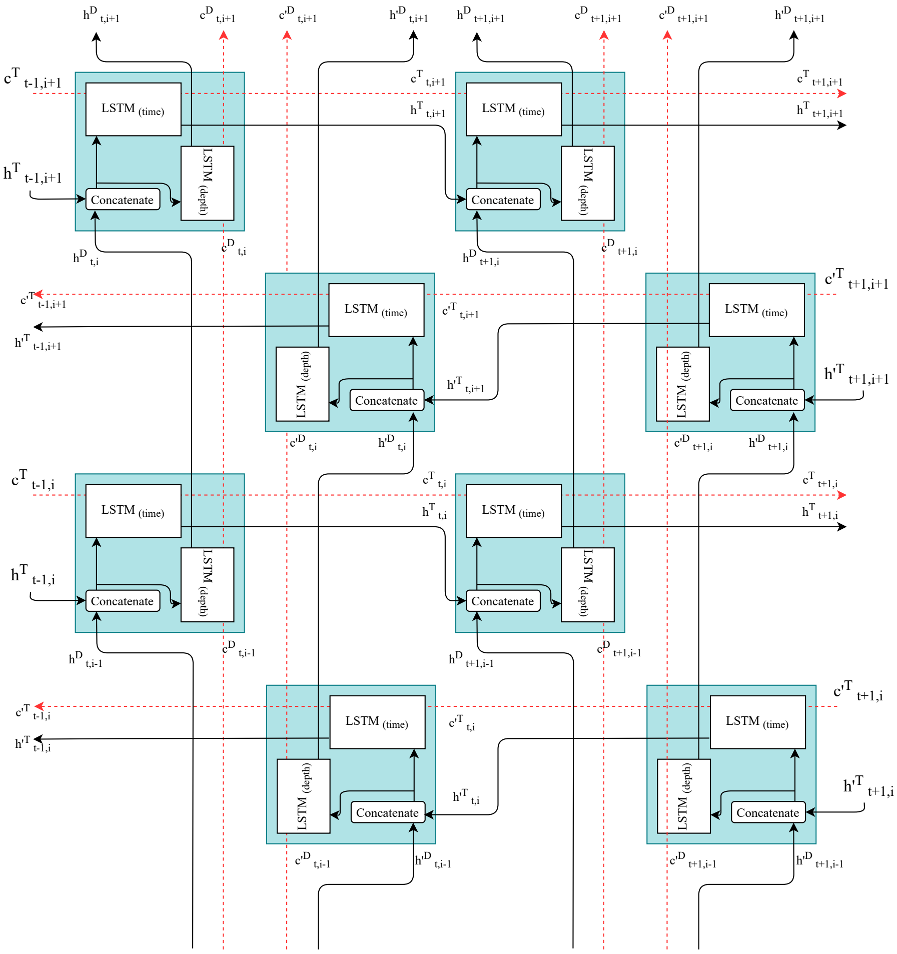

Figure 2.12: Architecture of BiGridLSTM with LSTM blocks for the time and depth dimension, ‘ symbol indicates reverse in the figure as in [80].

SynSigGAN is capable of capturing the different physiological characteristics associated with each of these signal types and has demonstrated an ability to generate biomedical time series data with a max sequence length of 191 data points. The authors also present a preprocessing stage to clean and refine the biomedical signals in this paper. They compare their architecture to several variants (BiLSTM-GRU, BiLSTM-CNN GAN, RNN-AE GAN, Bi-RNN, LSTM-AE, BiLSTM-MLP, LSTM-VAE GAN, and RNN-VAE GAN) and found the BiGrid-LSTM as the best performing model.

2.5 Applications

We have discussed the two classes of time series GANs and their contribution to solving the challenges presented in Section 2.3.1. Now we will list the wide ranging applications of time series GANs and the benefits of such to both research and industry.

2.5.1 Data Augmentation

It is common knowledge in the deep learning community that GANs are among the methods of choice when discussing data augmentation. Reasons for augmenting datasets range from increasing the size/variety of small and imbalanced datasets [81]–[84] to reproducing restricted datasets for dissemination.

A well-defined solution to the data shortage problem is transfer learning, and it works well in domain adaptation which has led to advancements in classification and recognition problems [85]. However, it has been found that augmenting datasets with GANs can lead to further improvements in certain classification and recognition tasks [86]. Data synthesised by a GAN can adhere to stricter privacy measures discussed in Section 2.7. This further demonstrates the advantages of augmenting your training dataset with GANs over implementing transfer learning with a pre-trained model from a different domain on a smaller dataset.

Many researchers find that accessing datasets for their deep learning research and models to be time-consuming, laborious work, particularly when the research is concerned with personal sensitive data. Often medical and clinical data are presented as continuous sequential data that can only be accessed by a small contingent of researchers who are not at liberty to disseminate their research openly. This, in turn, may lead to stagnation in the research progress in these domains.

Fortunately, we are beginning to see the uptake of GANs applied to time series with these types of medical and physiological data [42], [76], [87]–[89]. With [90] showing dependent multivariate continuous high-fidelity physiological signal generation is capable via GANs, demonstrating the impressive capability of these networks. See Figure 2.13 for an example of the real input and synthetic generated data.

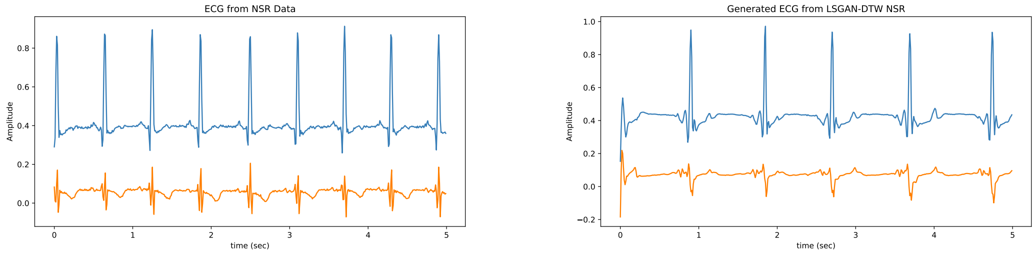

Figure 2.13: An example of dependent multichannel ECG data (left) and generated ECG from a multivariate GAN (right) [90]. NSR indicates the training dataset which is the Normal Sinus Rhythm. The generated data is produced by a GAN named by the authors as LSGAN-DTW.

Of course, this is not comprehensive coverage of the research using time series GANs for data synthesis and augmentation. GANs have been applied to time series data for a plethora of use cases.

Audio generation (both music and speech) and text-to-speech (TTS) [91] has been a popular area for researchers to explore with GANs. The C-RNN-GAN described in Section 2.4.2 was one of the seminal works to apply GANs to generating continuous sequential data in the form of music.

In the financial sector, GANs have been implemented to generate data and predict/forecast values. Wiese *et al.* implemented a GAN to approximate financial time series in discrete-time [65]. In [72], the authors designed a decision-aware GAN that generates synthetic data and supports decision processes to financial portfolio selection of end-users.

Other time series generation/prediction methods range from estimating soil temperature [92] to predicting medicine expenditure based on the current state of patients [93].

2.5.2 Imputation

In real-world datasets, missing or corrupt data is an all too common problem that leads to downstream problems. These issues manifest themselves in further analytics of the dataset and can induce biases in the data. Common methods of dealing with missing or corrupted data in the past have been the deletion of data streams containing the missing segments, statistical modelling of the data, or machine learning imputation approaches. Looking at the latter, we review the work in imputing these data using GANs. Guo *et al.* designed a GAN based approach for multivariate time series imputation [94], see Figure 2.14 for an example of imputed data from a toy experiment [89].

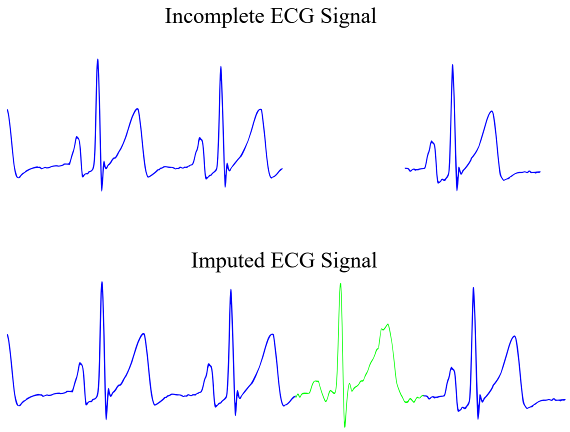

Figure 2.14: An example of the incomplete corrupted time series (top) and imputed signal (bottom).

2.5.3 Denoising

Artefacts induced in time series data often manifest themselves as noise in the signals. This has become an ever-present challenge in further processing and analytical applications. Corrupted data can cause biases in the datasets or lead to degradation in the performance of critical systems such as those used for health applications. Common methods for dealing with noise include the use of adaptive linear filtering. Another approach recently explored in [71] used GANs as a noise-reduction technique in EEG data. Their experiments showed that their proposed NR-GAN (Section 2.4.2) was capable of competitive noise reduction performance compared to more traditional frequency filters.

2.5.4 Anomaly Detection

Detecting outliers or anomalies in time series data is an important part of many real-world systems and sectors. Whether it is detecting unusual patterns in physiological data that may be a precursor to some more malicious condition or detecting irregular trading patterns on the stock exchange, anomaly detecting can be vital to keeping us informed on important information. Statistical measures of non-stationary time series signals may achieve good performance on the surface, but they might also miss some important outliers present in deeper features. They may also struggle in exploiting large unlabelled datasets; this is where the unsupervised deep learning approaches can outperform the conventional methods. Zhu *et al.* designed a GAN algorithm for anomaly detection in time series data (ECG and taxi dataset) with LSTMs and GANs, which achieved superior performance compared to conventional, more shallow approaches. Similar approaches have been applied to detect cardiovascular diseases [95], in cyber-physical systems to detect nefarious players [96] and even irregular behaviours such as stock manipulation on the stock markets [97].

2.5.5 Other Applications

Some works have utilised image-based GANs for time series and sequential data generation by first converting their sequences to images via some transformation function and training the GAN on these images. Once the GAN converges, similar images can be generated; then, a sequence can be retrieved using the inverse of the original transformation function. For example, this approach has been implemented in audio generation with waveforms [98]–[100], anomaly detection [101] and physiological time series generation [89].

2.6 Evaluation Metrics

As mentioned in Section 2.1 GANs can be difficult to evaluate, and researchers are yet to agree on what metrics reflect the GANs performance best. There have been plenty of metrics proposed in the literature [22] with most of them suited to the computer vision domain. Work is still ongoing to suitably evaluate time series GANs. We can break down evaluation metrics into two categories: qualitative and quantitative. Qualitative evaluation is another term for human visual assessment via the inspection of generated samples from the GAN. However, this cannot be deemed a full evaluation of GAN performance due to the lack of a suitable *objective* evaluation metric. The quantitative evaluation includes the use of metrics associated with statistical measures used for time series analytics and similarity measures such as; Pearson Correlation Coefficient (PCC), percent root mean square difference (PRD), (Root) Mean Squared Error (MSE and RMSE), Mean Relative Error (MRE) and Mean Absolute Error (MAE). These metrics are among the most commonly used for time series evaluation and, as such, used as a suitable GAN performance metric as they can reflect the stability between the training data and synthetic generated data and we show some of these common formulas in equations (2.24) to (2.27).

$$PCC = \frac{\sum_{i=1}^{N}(x_i - \tilde{x})(y_i - \tilde{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \tilde{x})^2 \sum_{i=1}^{N}(y_i - \tilde{y})^2}} \tag{2.24}$$

$$PRD = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{\sum_{i=1}^{N}(x_i)^2}} \tag{2.25}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \tag{2.26}$$

$$MRAE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{x_i - y_i}{x_i - f_i}\right| \tag{2.27}$$

Across these formulas x_i is the actual value of the time series x at time/sample i, y_i is the generated value of the time series y at time/sample i. $\tilde{x}$ and $\tilde{y}$ represents the mean values of x and y respectively. f_i is used in the MRAE calculation for the forecast value at time i of a chosen benchmark model. In general f_i can be chosen to be y_{i-1} for non-seasonal time series and y_{i-M} for seasonal time series, where M is the seasonal period of x.

Several metrics have become well-established choices in evaluating image-based GANs, and some of these have permeated through to the sequential and time series GANs such as Inception Score (IS) [36], Fréchet (Inception) Distance (FD and FID) [37]. Structural Similarity Index (SSIM) is a measure of similarity between two images. However, [102] use this with time series data as SSIM does not exclude itself from comparing aligned sequences of fixed length. Of course, some of these metrics are measures of similarities/dissimilarities between two probability distributions, suitable for many types of data, particularly the maximum mean discrepancy (MMD) [103]. In the real world we do not have access to the underlying distributions of data, therefore we show an empirical estimate of MMD in equation (2.28), which is a very suitable metric for this task across domains.

$$MMD[\mathcal{F}, X, Y] = \left[\frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i, x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i, y_i) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i, y_j)\right]^{\frac{1}{2}} \tag{2.28}$$

where $\mathcal{F}$ is a class F of smooth functions $f : \mathcal{X} \to \mathbb{R}$. Two observations $X :=$

$\{x_1, x_2, ..., x_n\}$ and $Y := \{y_1, y_2, ..., y_n\}$ are drawn from two distributions p and q with m points sampled from p and n from q. Lastly, k is the kernel function chosen by the user.

Another metric that generalises well to the sequential data case is the Wasserstein distance. The Wassterstein-1, or Earth Mover distance, shown in equation (2.29), describes the cost it takes to move one cumulative distribution function to another while preserving the shape of the functions, which is done by optimising the transport plan.

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{XxY} d^p(x, y) d\gamma(x, y) \right)^{\frac{1}{p}} \tag{2.29}$$

where $\Gamma(\mu, \nu)$ is the set of all transport plans, $d^p(x, y)$ is the distance function and $d\gamma(x, y)$ is the amount of 'mass' to be moved.

The data generated from GANs have been used in downstream classification tasks. Using the generated data together with the training data has lead to the Train on Synthetic, Test on Real (TSTR) and Train on Real, Test on Synthetic (TRTS) evaluation methods, first proposed by Esteban *et al.* [42]. In scoring downstream classification applications that use both real and generated data, studies have adopted the precision, recall, and F1 scores to determine the classifier's quality and, in turn, the quality of the generated data. Other accuracy measures of classifier performance include the weighted accuracy (WA) and unweighted average recall (UAR).

Often used distance and similarity measures in time series data are the Euclidean Distance (ED) and Dynamic Time Warping (DTW) algorithms. We introduce a new evaluation metric in the context of generative multivariate time series signals which is developed in detail in the next chapter, named Multivariate (in)dependent DTW (MVDTW). This metric can determine similarity measures across both dependent and independent multichannel time series signals [90].

Other metrics used across different applications include:

- **Financial Sector**; autocorrelation function (ACF) score, DY metric.

- **Temperature Estimation**; Nash-Sutcliffe model efficiency coefficient (NS), Willmott index of agreement (WI) and the Legates and McCabe index (LMI).
- **Audio Generation**; Normalised Source-to-Distortion Ratio (NSDR), Source-to-Interference Ratio (SIR), Source-to-artifact ratio (SAR) and t-SNE [38].

For a full list of GAN architectures reviewed in this work, their applications, evaluation metrics, and datasets used in their respective experiments, see Table 2.2. Results for the sine wave and ECG generation using variants of GAN architectures can be found in Tables 2.3 and 2.4, respectively.

Table 2.2: A list of GAN architectures, their applications, and datasets used in their experiments and evaluation metrics used to judge the quality of the respective GANs. For novel approaches, the GAN name is given as they have been covered already in Section 2.4.

Application	**GAN Architecture(s)**	**Dataset(s)**	**Evaluation Metrics**
Medical/Physiological Generation	LSTM-LSTM, [42], [81], [82] LSTM-CNN, [90] [87] BiLSTM-CNN, [88] BiGridLSTM-CNN, [76] CNN-CNN, [104], [105] AE-CNN, [106] FCNN [107]	EEG, ECG, EHRs, PPG, EMG, Speech, NAF, MNIST, Synthetic sets	TSTR, MMD, Reconstruction error, DTW, PCC, IS, FID, ED, S-WD, RMSE, MAE, FD, PRD, Averaging Samples, WA, UAR, MV-DTW

Continued on next page

Table 2.2 – continued from previous page

Application	**GAN Architecture(s)**	**Dataset(s)**	**Evaluation Metrics**
Financial time series generation/prediction	TimeGAN [40] SigCWGAN [72] DAT-GAN [75] QuantGAN [65]	S&P500 index (SPX), Dow Jones Index (DJI), ETFs	Marginal Distributions, Dependencies, TSTR, Wasserstein Distance, EM distance, DY Metric, ACF score, leverage effect score, discriminative score, predictive score
Time series Estimation/Prediction	LSTM-NN [92] LSTM-CNN [93] LSTM-MLP [93]	Meteorological data, Truven MarketScan dataset	RMSE, MAE, NS, WI, LMI
Audio Generation	C-RNN-GAN [67] TGAN(variant) [100] RNN-FCN [108] DCGAN(variant) [99] CNN-CNN [91]	Nottingham dataset, Midi music files, MIR-1K, TheSession, Speech	Human perception, Polyphony, Scale Consistency, Tone Span, Repetitions, NSDR, SIR, SAR, FD, t-SNE, Distribution of notes

Continued on next page

Table 2.2 – continued from previous page

Application	**GAN Architecture(s)**	**Dataset(s)**	**Evaluation Metrics**
Time series Imputation/Repairing	MTS-GAN [94] CNN-CNN [109] DCGAN(variant) [110] AE-GRUI [111] RGAN [112] FCN-FCN [113] GRUI-GRUI [114]	TEP, Point Machine, Wind Turbin data, PeMS, PhysioNet Challenge 2012, KDD CUP 2018, Parking lot data,	Visually, MMD, MAE, MSE, RMSE, MRE, Spatial Similarity, AUC score
Anomaly Detection	LSTM-LSTM [97] LSTM-(LSTM&CNN) [115] LSTM-LSTM (MAD-GAN) [96]	SET50, NYC Taxi data, ECG, SWaT, WADI	Manipulated data used as a test set, ROC Curve, Precision, Recall, F1, Accuracy
Other time series generation	VAE-CNN [102]	Fixed length time series 'vehicle and engine speed'	DTW, SSIM

2.7 Privacy

As well as evaluating the quality of the data, a wide range of methods have been used to evaluate and mitigate the privacy risk associated with synthetic data created by GANs.

Table 2.3: Experimental results comparing the performance of time series GANs for sinewave generation

Architecture	Loss Function	Toy Sine Dataset		
		MMD	DTW	MSE
LSTM-LSTM	BCE	0.9527	91.1071	0.2308
	MSE	0.0078	54.1644	**0.1480**
BiLSTM-LSTM	BCE	0.1215	428.4310	3.0700
	MSE	0.9515	79.5607	0.2362
LSTM-CNN	BCE	0.006	55.3620	0.3154
	MSE	0.5757	86.7357	0.5643
BiLSTM-CNN	BCE	**1.129E-05**	129.9257	0.9193
	MSE	0.4891	43.2694	0.1869
GRU-CNN	BCE	0.0244	**37.1630**	0.2303
	MSE	0.3727	42.7348	0.22823
FC-CNN	BCE	0.0039	58.3565	0.3048
	MSE	0.0117	43.3611	0.2972

2.7.1 Differential Privacy

The goal of Differential Privacy (DP) is to preserve the underlying privacy of a database. An algorithm or, more specifically, a GAN achieves differential privacy if, by looking at the generated samples, we cannot identify whether the samples were included in the training set. As GANs attempts to model the training dataset, the problem of privacy lies in capturing and generating useful information about the training set population without the possibility of linkage from generated sample to an individual's data [116].

As we have previously addressed, one of the uses of GANs is to augment existing under-resourced datasets for use in further downstream applications such as upskilling of clinicians where healthcare data is involved. These personal sensitive data must contain privacy guarantees, and the rigorous mathematical definition of DP [117] offers this assurance.

Work is ongoing to develop machine learning methods with privacy-preserving mechanisms such as differential privacy. Abadi *et al.* demonstrate the ability to train deep neural networks with DP and implement a mechanism for tracking privacy loss [118]. Xie *et al.* proposed a differentially private GAN (DPGAN) that achieved

Table 2.4: Experimental results comparing the performance of time series GANs for ECG generation on MIT-BIH Dataset

Architecture	Loss Function	MIT-BIH Arrhythmia Dataset		
		MMD	DTW	MSE
LSTM-LSTM	BCE	0.9931	30.1816	0.0867
	MSE	0.8842	44.4553	0.1389
BiLSTM-LSTM	BCE	0.9916	22.8634	0.0699
	MSE	0.9737	23.5533	0.0806
LSTM-CNN	BCE	0.5519	**13.0158**	**0.0151**
	MSE	**0.0005**	24.7306	0.0457
BiLSTM-CNN	BCE	0.9246	117.3994	0.2272
	MSE	0.0687	22.6740	0.0586
GRU-CNN	BCE	0.0055	20.4845	0.0335
	MSE	0.7704	108.4124	0.1948
FC-CNN	BCE	0.2068	23.9910	0.0309
	MSE	0.3082	18.2340	0.0212

differential privacy by adding noise gradients to the optimiser during the training phase [119].

2.7.2 Decentralised/Federated Learning

Distributed or decentralised learning is another method for limiting the privacy risk associated with personal and personal sensitive data in machine learning. Standard approaches to machine learning require that all training data be kept on one server. Decentralised/distributed approaches to GAN algorithms require large communication bandwidth to ensure convergence [120], [121] and are also subject to strict privacy constraints. A new method that enables communication efficient collaborative learning on a shared model while keeping all the training data decentralised is known as *Federated learning* [122]. Rasouli *et al.* applied federated learning algorithm to a GAN for communication efficient distributed learning and proved the convergence of their federated learning GAN (FedGAN) [123]. However, it should be noted that they did not experiment with differential privacy in this study but note it as an avenue of future work.

Combining the above techniques of federated learning and differential privacy in developing new GAN algorithms would lead to a fully decentralised private GAN capable of generating data without leakage of private information to the source data. This is clearly an open research avenue for the community.

2.7.3 Assessment of Privacy Preservation

We can also assess how well the generative model was able to protect our privacy through tests known as attribute and presence disclosure [41]. The latter test is more commonly known in the machine learning space as a membership inference attack. This has become a quantitative assessment of how machine learning models leak information about the individual data records on which they were trained [124]. Membership inference attacks attempt to detect the data that was used to train a target model without the attacker having access to the model's parameters. A nefarious actor creates random records for a target machine learning model. The attacker then feeds each record into the model. The model will return a confidence score and based on this score the records will be fine tuned until a higher confidence score is returned. This process will continue until the model returns a very high score and at this stage the record will be nearly identical to one of the examples used in the training dataset. These steps will be repeated until enough dataset examples are generated. The fake records will then be used to train an ensemble of models to predict whether a data record was used in the training set of the target model.

Hayes *et al.* carried out membership inference attacks on synthetic images and concluded that for acceptable levels of privacy in the GAN, the quality of the data generated is sacrificed [125]. Conversely, others have followed this approach and found that DP networks can successfully generate data that adheres to differential privacy and resists membership inference attacks without too much degradation in the quality of the generated data [42], [87], [90].

2.8 Discussion

We have presented a survey of time series GAN-variants that have made significant progress in addressing the primary challenges identified in Section 2.3.1. These GANs introduced the idea of both discrete and continuous sequential data generation and have made incremental improvements over one another via an architecture variant or a modified objective function capable of capturing the spatio-temporal dependencies present in these data types. The loss functions implemented in these works for some architectures will not necessarily generalise to others; hence they become architecture-specific. The architecture choices of the time series GANs affect both the quality and diversity of the data. However, there remain open problems in terms of the practical implementation of the generated data and GANs in real-world applications, particularly in health applications where the performance of these models can directly affect patients' quality of care/treatment.

The 'best' GAN architecture and objective function is still yet to be determined. This is because humans have manually designed most architectures. As a result, there is growing interest in automated neural architecture search (NAS) methods [126] whereby automating the architecture engineering aspect of machine learning. It is a growing branch of automatic machine learning (AutoML) and automatic deep learning (AutoDL) that seeks to optimise the processes around machine learning. Work has been done in the image domain space with neural architecture search and GANs [127]. This method, named AutoGAN, achieved highly competitive performance compared to state-of-the-art human-engineered GANs. This is a promising area for time series GANs; to our knowledge, is yet unexplored.

As it stands, GANs tend to be application-specific; they perform well for their intended purpose but do not generalise well beyond their original domain. Furthermore, a major limitation of time series GANs is the restrictions placed on the length of the sequence specified that the architecture can manage; documented experiments validating how well a time series GAN can adapt to varying data lengths are notably absent at the time of writing. However, glimpses of work in the NLP literature in the form of Transformers [128] have demonstrated some applicability to dealing with varying sequence lengths that may prove beneficial in addressing this issue and might emerge in time series generation given time.

2.9 Conclusion

This chapter reviews a niche but growing use of Generative Adversarial Networks for time series data based mainly around architectural evolution and loss function variants. We see that each GAN provides application-specific performance and doesn't necessarily generalise well to other applications, e.g., a GAN for generating high-quality physiological time series may not produce high-fidelity audio due to some limitation imposed by the architecture or loss function. A detailed review of the applications of time series GANs to real-world problems has been provided, along with their datasets and the evaluation metrics used for each domain. As stated in [24], GAN-related research for time series lags that of computer vision both in terms of performance and defined rules for generalisation of models. In conclusion, this review has highlighted the open challenges in this area and offers directions for future work and technological innovation, particularly for those GAN aspects related to evaluation, privacy, and decentralised learning. We will carry this technology through the thesis for its capability in generating clinically relevant multivariate physiological time series data that adheres to differential privacy. This creates the opportunity for disseminating sensitive medical data for secondary use in further upskilling of relevant clinical specialists. We also explore GANs for further signal processing such as denoising physiological time series that seek to make biomarkers from a noisy signal simpler to uncover.

Chapter 3

Generating Physiological Time Series Data using Generative Adversarial Networks

3.1 Introduction

In this chapter, we demonstrate the ability of Generative Adversarial Networks to generate synthetic and, more significantly, multichannel physiological signals. The motivation behind this is as a generative model that samples from a latent space, GANs theoretically preserve privacy and generate synthetic data similar to the training dataset. The privacy-preserving nature of GANs is important to Research Question 1 as sharing and using inherently sensitive medical data is becoming increasingly complex, with tightening restrictions that lead to significant challenges in clinical research and development. A result of these regulations means traditional modes of data sharing have become hindered, and the ML and DL communities are making efforts to overcome these restrictions in ways that respect privacy sensitivities. This is a significant challenge because the development of effective models requires access to extensive datasets. Such data privacy concerns present researchers and clinicians with an additional set of obstacles in their pursuit of AI-enhanced innovations.

Addressing these obstacles raises ethical issues, especially regarding how data can be used while ensuring privacy is protected, and public trust is maintained. This poses policy and regulatory challenges for lawmakers and regulators. They must balance safeguarding personal data while not retarding vital innovation and research to improve patient outcomes.

Practitioners with access to sought after data often find themselves working through complex data privacy frameworks and discover that sharing and publishing the information available via the data is highly challenging. For example, personal sensitive data such as medical data intended for secondary purposes like clinical training or research requires anonymisation following its approval for dissemination.

Common methods for the de-identification of data are generalisation, randomisation, or pseudonymisation [129]. However, it has been shown that the de-identification of medical data does not guarantee privacy protection of all individuals in the dataset, and it is possible to re-identify individuals by linkage of data from other sources or residual information [43]. This may result in the inability to share data with further research or clinical institutions. In addition, there is often a shortage of available training data for clinicians and researchers alike, significantly impeding scientific progress, particularly in developing countries.

The generation of synthetic data is one such solution to the presented problem. The goal here lies in producing synthetic physiological data representative of real data gathered during the data collection experiment. However, as stated previously, it is important to note that substantial amounts of data are required to successfully train deep learning models for this purpose. Furthermore, protecting the privacy of the underlying real dataset must also be observed [130], [131].

If these problems can be addressed and overcome, the generated data can be published without breaching privacy and used in further training and research. Increasing access to this type of data will encourage scientific studies and facilitate the upskilling of clinicians, which will in turn aid in preventing or limiting chronic illnesses. This can contribute to a shift in the treatment paradigm from reactive to preventative healthcare.

Capitalising on recent advancements in machine learning and, in particular, deep learning could pave the way for the future of sharing data and disseminating research. The work described in this chapter is part of a larger-scoped effort to develop artificial intelligence for use in clinical training and upskilling of medical professionals.

In this chapter, we demonstrate the contributions of our method in generating realistic, dependent, multivariate physiological signals while maintaining sufficient levels of privacy in the training dataset. Using the Multivariate GAN (MVGAN) architecture developed in our preliminary work [90], we explore novel loss functions and their effects on generated data quality. We demonstrate our novel GAN, objective function and evaluation metrics capable of improved multivariate time series data generation for the first time. Finally, we benchmark our generative model against other classical generative models.

3.2 Related Work

A variety of methods have been used in the past to generate synthetic data. In the medical domain, research has mainly focused on the generation of synthetic Electronic Health Records [130], [132]. Of particular relevance for our research are those methods which generate synthetic time series data. Previous approaches include the creation of dynamical models to produce synthetic electrocardiogram signals [131]. These models consist of three coupled ordinary differential equations with the user required to specify the characteristics of the heart rate signals to be generated. Many early methods require expert domain knowledge to generate synthetic data. More recent developments in the machine learning space remove this dependency. For example, WaveNet implemented an auto-regressive neural network that successfully generated synthetic music and speech [133]. In other research, Dahmen and Cook (2019) developed SynSys to produce realistic home sensor data using hidden Markov models and regression models [134].

A significant breakthrough in synthetic data generation was facilitated by the introduction of Generative Adversarial Networks. GANs do not require input from domain experts and they can be designed to preserve privacy of the training datasets. They were first proposed in the seminal paper by Goodfellow in 2014. A multi-layer perceptron was used for both the discriminator and the generator [21]. Radford *et al.* (2015) subsequently developed the deep convolutional generative adversarial network (DCGAN) to generate synthetic images [30]. A recurrent GAN (RGAN) was first proposed in 2016. The generator contained a recurrent feedback loop that used both the input and hidden states at each time step to generate the final output [59]. Recurrent GANs often utilise Long Short-Term Memory neural networks (LSTMs) in their generative models to avoid the vanishing gradient problem associated with more traditional recurrent networks [60]. Since their inception in 2014, GANs have shown great success in generating high-quality synthetic images which are indistinguishable from the actual images [56]–[58].

While the focus has been on developing GANs for improved image generation, there has been a movement towards using GANs for time series and sequence generation [135]. In our exploratory work with GANs for time series generation we aimed to synthesise time series data by leveraging well established imaged-based GAN architectures for physiological data generation [89]. Our method segmented the time series data into suitable windows of fixed length (4096 data points). We then sampled the amplitude of the signal and mapped the amplitude to an RGB grayscale value. The array of RGB values were rasterised into an image that can be used to train an imaged-based GAN. The trained GAN is then capable of synthesising new rasterised images which can then be transformed back to time series using our transform, see Figure 3.1 for our developed pipeline.

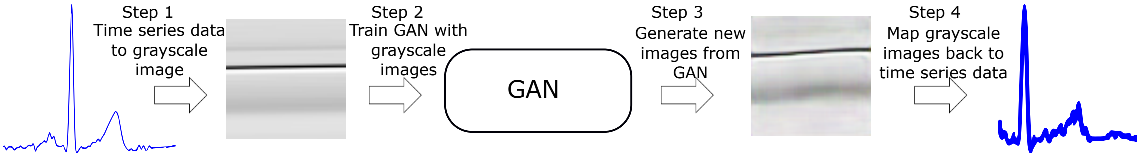
Step 1
Time series data to grayscale image
Step 2
Train GAN with grayscale images
GAN
Step 3
Generate new images from GAN
Step 4
Map grayscale images back to time series data

Figure 3.1: Pipeline of GAN model.

In Figure 3.1 step one involves segmenting the raw time series data into suitable windows of fixed length, and sampling the amplitude of the signal to map on to an RGB grayscale value. The RGB grayscale values are rasterised into an image. Step two involves training the GAN using the rasterised images. Step three uses the trained GAN to synthesise new images. Step four involves transforming the images back to their corresponding time series array.

Sequence-based neural networks have also generated synthetic electronic health records (EHR). While not continuous time series physiological data, it still serves as an essential foundational block in using GANs to develop a multimodal view of human health. In our preliminary work, we demonstrate the ability of a GAN with a fully connected architecture to generate tabular ICD-9 codes that classify diseases of patients in hospitals. The data was collected from the MIMIC-III database [136] available on PhysioNet [137]. We demonstrated that the GAN could generate statistically significant ICD-9 codes that were representative of the patients observed in the database. Furthermore, the generated data was evaluated by a group of medical professionals. They found that there was no significant difference between the three groups of patients regarding how plausible the medical professionals viewed them, confirming that the synthetic patients closely matched the actual patients, Figure 3.2. We also implemented Federated Learning for this experiment and found no significant degradation in performance of the GAN, see Chapter 5 for further details on Federated Learning.

Novel recurrent neural networks have since been developed for time series and sequence generation. One such implementation involved the generation of polyphonic music as real-valued continuous sequential data using an LSTM in both the generator and discriminator [67]. In contrast, Yu *et al.* (2017) generated synthetic music by representing 88 distinct pitches with discrete tokens [53]. This GAN, known as SeqGAN, contained an LSTM in the generator with a CNN in the discriminator and outperformed alternative approaches for generating data sequences. GANs were also used to generate single-channel electroencephalogram (EEG) data for motor

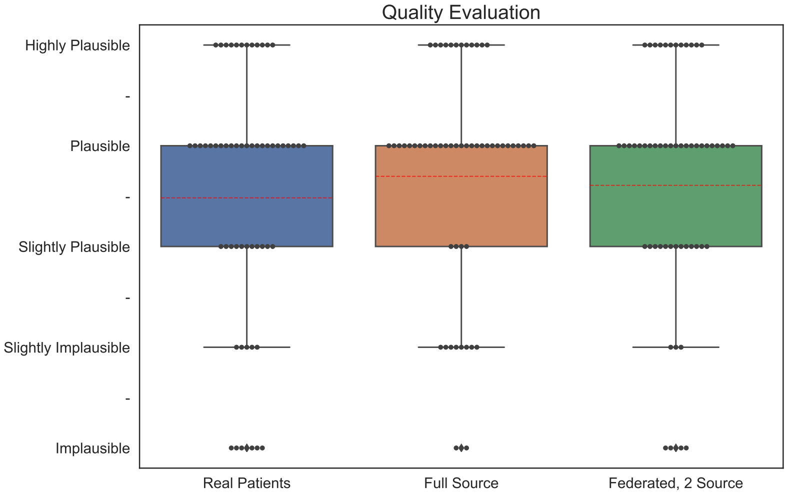

Figure 3.2: Medical professionals' evaluation of the synthetic EHRs.

movement in both the left and right-hand [104]. We are aware of one work that implements both a GAN and a conditional GAN (CGAN) to generate real-valued medical time series data [42]. A CGAN provides additional information to the generator and the discriminator to aid the creation of synthetic data [138]. More recent attempts to generate synthetic ECG used bidirectional LSTMs in the generator and convolutional neural networks in the discriminator [88].

By focusing on deep generative models for single channel physiological data, we demonstrated that realistic synthetic physiological signals could be generated from a dataset of real signals [87]. This was one of the earliest examples in literature that generated continuous physiological time series signals using GANs. An example of the single channel synthetic data generated is visualised in Figure 3.3.

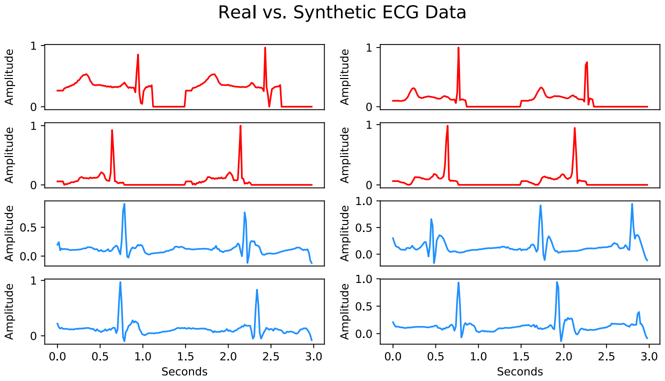

Figure 3.3: Early examples of real (red) ECG and synthetic (blue) physiological time series signals generated by a GAN.

However, our approach above was limited to single channel time series. Therefore, we extend this by exploring the possibility of generating multivariate, strongly-coupled physiological time series and investigating using appropriate evaluation metrics to obtain characteristics in the output present in the training dataset. This is an essential step as multivariate medical time series is not simply a collection of independent time series, each of which can be synthesised independently. An extensive deep coupling between the signals exists and is exemplified in multi-lead electrocardiography (an electrical measure of cardiac activity), also known as ECG. A multi-lead ECG involves measuring the heart's electrical activity via several projections over the body's surface via differential bipolar electrode sets. This produces a tightly coupled time series set that can reconstruct an approximation of the dipole dynamics associated with current flow in the beating heart. Furthermore, we improve the quality of our generated multichannel ECG through the development of novel loss functions. We compare them with other common loss functions that have been previously explored in the time series GAN literature [135]. This chapter focuses on the challenge of synthesising such data using our novel objective functions.

3.3 Multivariate Dynamic Time Warping

For a GAN to be considered successful, not only should it converge during training, but it should also learn the distribution of the training data. Dynamic Time Warping is used to measure the similarity or distance between two time series sequences and can be implemented as a univariate sequential data classifier. The single-dimensional DTW cumulative distance function defined in (3.1) is used to find the path that minimises the warping cost. Here $d(q_i, c_j)$ is the squared Euclidean distance between the i^{th} data point of the univariate time series Q and j^{th} data point of the univariate time series C. D(i,j) represents the n-by-n matrix constructed by the squared Euclidean distance between points q_i and c_j where n is the length of the sequence.

$$D(i,j) = d(q_i, c_j) + \min\Big\{D(i-1, j-1), D(i-1, j), D(i, j-1)\Big\} \tag{3.1}$$

To adapt to the multivariate Dynamic Time Warping (MVDTW) case we redefine $d(q_i, c_j)$ as the cumulative squared Euclidean distances of M data points as in [139]. M is defined as the number of time series that make up the multi-dimensional time series; for this work, the number of individual time series is two (M=2). Q and C are two separate multivariate time series, both with M=2. $q_{i,m}$ is the i^{th} data point in the m^{th} dimension of one multivariate time series Q and $c_{j,m}$ is the j^{th} data point in the m^{th} dimension of the other multivariate time series C, $d(q_i, c_j)$ now becomes:

$$d(q_i, c_j) = \sum_{m=1}^{M} (q_{i,m} - c_{j,m})^2 \tag{3.2}$$

Therefore we can now define the cumulative distance for MVDTW as in equation (3.3). This allows us to find the distance that minimises the warping path and calculate MVDTW. In turn, we can calculate the similarity between our generated data and training data. This equation can now be used as our novel evaluation metric for the generated multichannel physiological data.

$$D(i,j) = \sum_{m=1}^{M} (q_{i,m} - c_{j,m})^2 + \min\Big\{D(i-1, j-1), D(i-1, j), D(i, j-1)\Big\} \tag{3.3}$$

3.4 Model Design

This section presents our MVGAN model for generating synthetic, dependent, multivariate physiological time series data. Structurally, our model builds on the architectures of our previous preliminary work. We increase the limited sequence length of 187 in [87], [90] to a more realistic length of 500 sample points. This yields a time series more representative of digitised ECG for the time windows considered (5 seconds at a realistic sampling rate). This length is arbitrary and can be varied through the discriminator to produce data sequences of differing sizes. In terms of generating multichannel data, we increase the number of features available at the input and output of our model. This enables the model to generate realistic, coupled multivariate time series data; this has not been done in previous work. Extending on the earlier models, we also implement 2-dimensional convolution-pooling layers and include a minibatch discrimination layer in the discriminator to prevent mode collapse. The optimiser also has noise introduced to its gradients to create a differentially private GAN model (GAN-DP). Implementation details of the experiments are available online [1].

3.4.1 Generator

The generator is a two-layer stacked LSTM with 50 hidden units in each layer and a fully connected layer at the output. With the extra expected features at the input of the *torch.nn.LSTM* class, this architecture facilitates generation of multivariate time series data and can scale to more channels as needed.

3.4.2 Discriminator

The discriminator is a four-layer convolutional neural network, a minibatch discrimination layer, a fully connected layer and a sigmoid activation function. For differential privacy noise was added to the gradient of the optimiser (GAN-DP model). Figure 3.4 is a block diagram of the discriminator and Table 3.1 is an example of the model parameters.

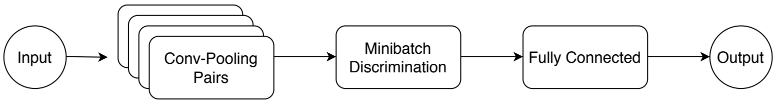

Figure 3.4: Diagram of discriminator architecture

[1] https://github.com/Brophy-E/Multivariate-timeseries-GANs

Table 3.1: Parameters for the convolution-pooling layers in the discriminator.

Layer	Input Size	Feature Maps	Filter	Stride	Output Size
C1	2×187	3×2	3	1	$3 \times 2 \times 185$
P1	$3 \times 2 \times 185$	3×2	3	1	$3 \times 2 \times 183$
C2	$3 \times 2 \times 183$	5×2	3	1	$5 \times 2 \times 181$
P2	$5 \times 2 \times 181$	5×2	3	2	$5 \times 2 \times 90$
C3	$5 \times 2 \times 90$	8×2	3	2	$8 \times 2 \times 44$
P3	$10 \times 2 \times 44$	8×2	3	2	$8 \times 2 \times 21$
C4	$10 \times 2 \times 21$	10×2	5	2	$10 \times 2 \times 9$
P4	$10 \times 2 \times 9$	10×2	5	2	$10 \times 2 \times 3$

3.5 Loss Functions

Keeping with the same architecture for the MVGAN, we explore novel loss functions by implementing the Loss Sensitive GAN's (LS-GAN) objective function [140] and tailoring it to our multivariate time series generation problem. When the distance between a generated and real multivariate sample becomes small, the GAN will stop increasing the difference $L(G(\boldsymbol{z})) - L(\boldsymbol{x})$ between their losses. The LS-GAN optimizes L and G alternately to minimise (3.4).

$$S = \mathbb{E}_{x \sim p_{data}(x)} \Big[L(x) + \lambda \Big] + \mathbb{E}_{x \sim p_{data}(x), z \sim p_{\mathbf{z}}(z)} \Big[\Delta(x, G(z)) + L(x) - L(G(z)) \Big] \quad (3.4)$$

where L is a given *loss function*, λ is a balancing parameter and $\Delta(x, G(z))$ is the difference between the real sample x and generated sample $G(z)$.

In exploring other loss functions, we investigate the LSGAN with (3.5) and without (3.6) an additional penalisation term in the discriminator. This term is the MVDTW, and it penalises the generator if the distance between the multivariate real and generated samples is large. This loss term holds if $1 \leq MVDTW(x, G(z))$. The generator's objective function remains unchanged (3.7). Here, a is the label for the generated samples, b is the label for the real samples, and c is the hyperparameter that G wants D to recognise the generated samples as real samples.

$$\min_{D} V_{LSGAN}(D) = 1/2 \times \mathbb{E}_{x \sim p_{data}(x)}\Big[(D(\mathbf{x}) - b)^2\Big] \\ +1/2 \times \mathbb{E}_{z \sim p_{\mathbf{z}}(z)}\Big[(D(G(\mathbf{z})) - a)^2\Big] \tag{3.5}$$

$$\min_{D} V_{LSGAN-DTW}(D) = 1/2 \times \mathbb{E}_{x \sim p_{data}(x)}\Big[(D(\mathbf{x}) - b)^2\Big] + \\ 1/2 \times \mathbb{E}_{z \sim p_{\mathbf{z}}(z)}\Big[(D(G(\mathbf{z})) - a)^2\Big] + \mathbb{E}_{x \sim p_{data}(x), z \sim p_{\mathbf{z}}(z)}\Big[1 - 1/\log(MVDTW(x, G(z))\Big] \tag{3.6}$$

$$\min_{G} V_{LSGAN}(G) = 1/2 \times \mathbb{E}_{z \sim p_{\mathbf{z}}(z)}\Big[(D(G(\mathbf{z})) - c)^2\Big] \tag{3.7}$$

The following objective function (3.8, 3.9) takes the MVDTW of the probability that a sample is either real or fake along with the adversarial ground truth. The adversarial ground truth is an array of either 0s or 1s. In this case, the MVDTW computes the distance between the probabilities and ground truth. In essence, this function computes the squared euclidean distance, and it is retained in this chapter as it produces both qualitatively plausible and quantitatively competitive samples.

$$\min_{D} V_{DTWGAN}(D) = 1/2 \times MVDTW_{x \sim p_{data}(x)}\Big[(D(\mathbf{x}), \text{valid})\Big] \\ +1/2 \times MVDTW_{z \sim p_{\mathbf{z}}(z)}\Big[(D(G(\mathbf{z})), \text{fake})\Big] \tag{3.8}$$

$$\min_{G} V_{DTWGAN}(G) = 1/2 \times DTW_{z \sim p_{\mathbf{z}}(z)}\Big[(D(G(\mathbf{z})), \text{valid})\Big] \tag{3.9}$$

Given these loss functions we now have 5 MVGAN variants as follows:

1. Least Squares GAN (LSGAN) (3.5), (3.7)

2. Least Squares GAN with DTW term (LSGAN-DTW) (3.6), (3.7)

3. GAN with DTW criterion (DTWGAN) (3.8), (3.9)

4. Loss Sensitive GAN (LS-GAN) (3.4)

5. Loss Sensitive GAN with DTW distance measure (LS-GAN-DTW) (3.4)

3.6 Materials and Methods

3.6.1 Datasets

Multichannel ECG records are signals from two or more leads simultaneously and are frequently used in place of single-channel ECG to give a more complete understanding of the cardiac state. To demonstrate that this MVGAN architecture effectively generates multichannel ECG, we have used two datasets in this work. The first openly available dataset is the MIT-BIH Normal Sinus Rhythm (NSR) Database, which includes 18 long-term ECG recordings of subjects found to have had no significant arrhythmia. Recordings were collected at Boston's Beth Israel Hospital and digitised at 128 Hz. Subjects include five men aged 26 to 45, and 13 women, aged 20 to 50.

The second dataset used is the publicly available MIT-BIH Arrhythmia (ARR) dataset [77]. This database contains 48 half-hour long recordings of two-channel ambulatory ECG. Both normal ECG and a range of uncommon but clinically significant ECG irregularities are included in this dataset. The authors of the data collection experiment digitised the recordings at 360 Hz. Each of the records was analysed by two cardiologists to provide reference annotations for every beat. For this dataset, a modified limb lead II (MLII) was used for recording one channel and a unipolar chest lead, also called precordial, or V lead, was used to measure the other channel. V1 was the most common chest lead used, but in some cases, V2, V4, or V5 was used.

In both cases, the datasets are open source and freely available on PhysioNet [137]. Figure 3.5 shows an example trace of a classic ECG expected from the datasets. The multichannel lead configuration illustrates the dependencies present in the signals that we are seeking to replicate.

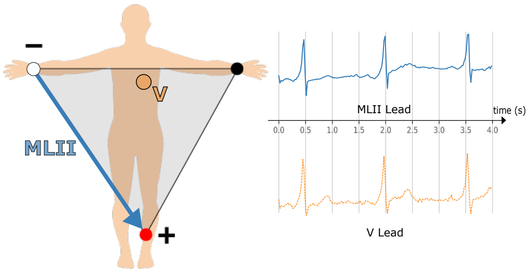

Figure 3.5: Lead configuration for the collection of the ARR dataset with corresponding ECG trace

3.6.2 Data Preprocessing

The datasets were pulled from PhysioNet and loaded using Python's *wfdb* library. Before training our GANs, the datasets required preprocessing in R-peak alignment, segmentation and downsampling. These steps are detailed in the following subsections.

R-peak alignment

Successfully generating dependent multivariate time series requires the training data to retain its inherent dependencies. Fortunately, the ECG channels are already concurrent before any preprocessing steps. An R-peak detector provided by *wfdb's* processing module was used on each of the ECG records. Aligning an R-peak in the centre of every training sample ensured a more effective training set as the QRS complexes occupy similar locations in the sequences. The QRS complex represents ventricular depolarisation and is a combination of the Q, R and S waves in the cardiac cycle.

Resampling

The signals were then resampled from their original sampling frequency of 128 Hz (NSR dataset) and 360 Hz (ARR dataset) to 100 Hz using SciPy's *signal.resample*.

Segmentation

Following resampling, we normalised and segmented the recordings into smaller samples, each consisting of 5 seconds of data for both leads. Naturally, these samples will not contain the same QRS complexes as the cardiac cycle has natural variability. The length of the data was varied from our previous works [87], [90] to demonstrate the scalability of our GAN architecture. An example of the multichannel input data is shown below in Figure 3.6 with an artificial offset on the lead two for visualisation purposes.

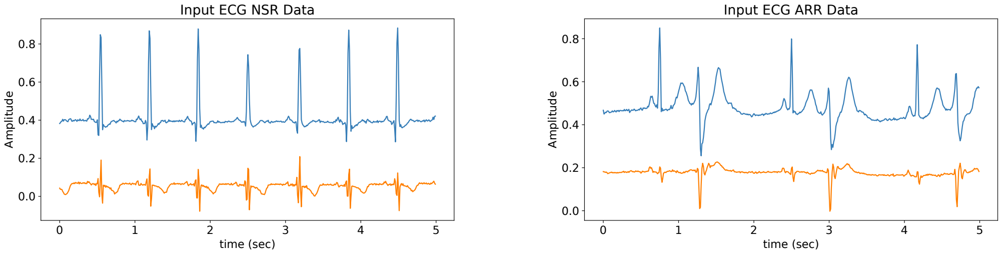

Figure 3.6: Example of preprocessed multichannel ECG from NSR dataset (left) and ARR dataset (right) used in training.

3.6.3 Training

For every loss function explored, the GAN was trained for 50 epochs. For each epoch, the entire training set was divided into batches of 50 multivariate samples. The RMSprop optimiser was used with a learning rate of $\alpha = 0.0002$ as it is computationally efficient and works well for this deep learning model. The GAN variants were trained without minibatch discrimination (MBD), and no mode collapse was observed. We have shown previously that the inclusion of MBD layers can be used with this architecture to prevent mode collapse [90]. In addition, noise was introduced into the gradients of the discriminator optimiser to ensure a differentially private network [119].

3.6.4 Evaluation

Quality

Maximum Mean Discrepancy (MMD) and multivariate Dynamic Time Warping were used to assess generated data quality. MMD is used here to reinforce the DTW results and to demonstrate that the GAN iteratively learns and generates data from a distribution more representative of the training data distribution. The MMD package used for these experiments is available on GitHub [2] [141].

Multivariate, dependent DTW was calculated to determine similarity measures across the dependent signals in the generated data against the training data. We have shown in the past that the MVDTW method can be used to evaluate generated data from time series GANs [90]. Generated data from the trained generator was compared against the complete training set for evaluation. The evaluation results were averaged over several runs of the model.

Privacy

Membership inference attacks observe the behaviour of our GAN and attempt to predict examples that were used to train it. A membership inference attack was run to assess presence disclosure. Presence disclosure occurs if it is possible to determine that a particular record was used to train a GAN by observing the generated samples. The sample size r was varied between [1000,10000] training records while the threshold ϵ ranged from [0.05,0.5] of the mean Euclidean distance between all samples. A synthetic dataset of 1000 generated samples was used for this test. This experiment was implemented in a Jupyter Notebook [3] and is also available as the python library package *mia* [142].

[2] `https://github.com/eugenium/MMD`

[3] `https://github.com/Brophy-E/Multivariate-timeseries-GANs/blob/main/Notebooks/membership_inference_attack.ipynb`

3.6.5 Benchmarking

To further evaluate and demonstrate the advantages of our GAN, we benchmark our results against current, well-known generative modelling methods. Using the same training dataset, we implemented a multivariate Variational Autoencoder (VAE) and LSTM as a means of generating the type of multichannel data that the proposed GAN is capable of generating. It is important to note that these methods are usually implemented with single-channel time series data. Here we adopt these methods from a single time series to the multichannel context for the first time to create a benchmark comparison.

To compare how closely the distribution and distance of the generated data match that of the training data, we implemented two time series classifiers alongside our MVDTW and MMD metrics. Support Vector Classification (SVC) and LSTM were the two classifiers of choice. We classify the generated and training data using these models; a classification rate closer to 0.5 demonstrates the classifiers have difficulty distinguishing actual data from the generated data. The poorer the performance of the classifier, the closer the generated data is to the training data. We also compare the data generated from our differentially private GAN to the GAN without DP.

Following this, we run a membership inference attack on the generated data for the GAN and GAN-DP to observe what difference, if any, the differential privacy offers. This series of experiments allows us to understand which model generates realistic multivariate time series signals and which models preserve the underlying privacy of the training data most effectively.

3.7 Results

In this section, we focus on the data generated by the GAN without differential privacy unless explicitly stated otherwise. Qualitative examples of high-quality generated ECG for each GAN can be found from Figure 3.7 to Figure 3.11. It becomes apparent that the LS-GAN-DTW generates the best qualitative results for the NSR and ARR datasets. The other variant models appear to have successfully generated realistic, multivariate and dependent ECG data. For visualisation purposes, an offset is again artificially introduced to lead II (orange).

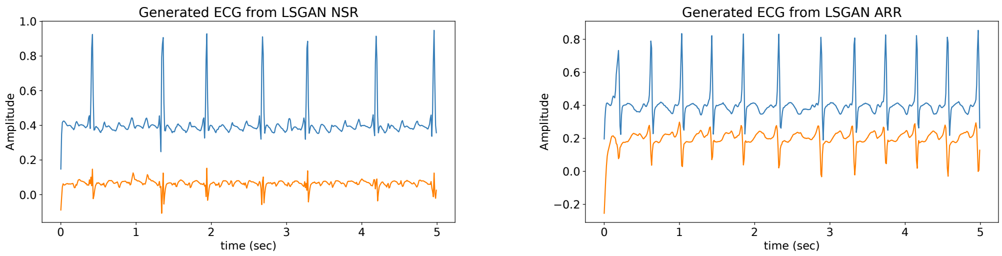

Figure 3.7: Examples of generated multichannel ECG from LSGAN for NSR dataset (left) and ARR dataset (right).

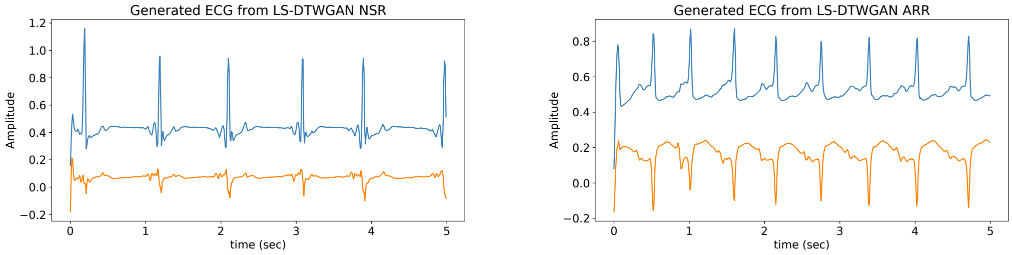

Figure 3.8: Examples of generated multichannel ECG from LSGAN-DTW for NSR dataset (left) and ARR dataset (right).

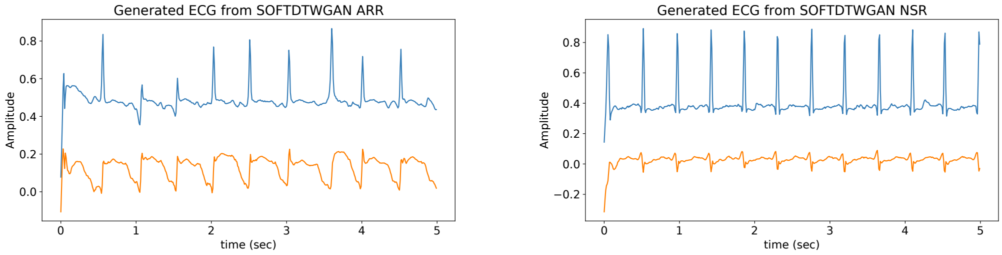

Figure 3.9: Examples of generated multichannel ECG from DTWGAN for NSR dataset (left) and ARR dataset (right).

The results shown in Figures 3.7 through 3.11 demonstrate that this architecture can successfully generate realistic ECG samples. Lead I is shown in blue and lead II in orange with an artificial offset introduced for visualisation purposes. It appears that for the ARR dataset the GAN models generate noisy ECG, but given the diverse nature of this dataset, the GANs generate good quality data, as is evident in the metrics that follow. However, a qualitative evaluation cannot be considered

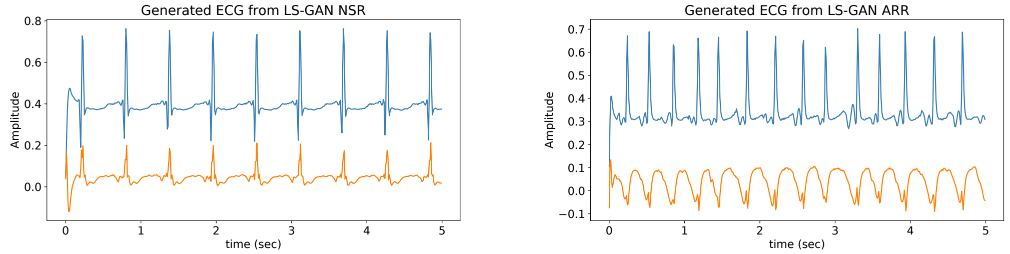

Figure 3.10: Examples of generated multichannel ECG from LS-GAN for NSR dataset (left) and ARR dataset (right).

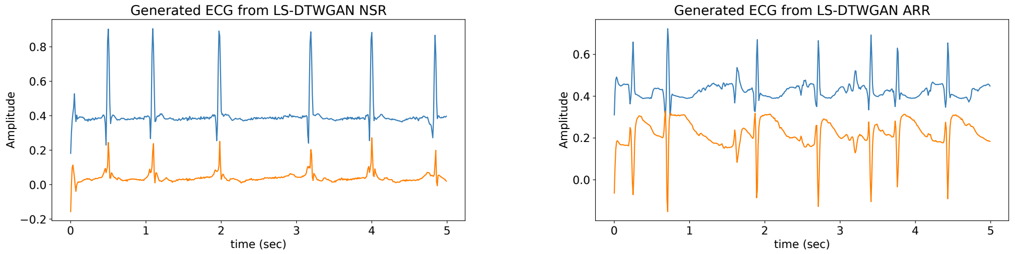

Figure 3.11: Examples of generated multichannel ECG from LS-GAN-DTW for NSR dataset (left) and ARR dataset (right).

a complete evaluation of GAN performance due to the lack of a suitable objective function to measure data quality. We address this challenge in the following section.

3.7.1 Evaluation

Quality

Visually, and therefore from a qualitative perspective, the multi-lead ECG synthesised is of high-quality; however, we augment this assessment through the development of suitable objective quantitative metrics. We demonstrate the results for these metrics in this section.

Maximum Mean Discrepancy results in Tables 3.2 and 3.3 demonstrate that as the GAN iterates through the training process, it is generating data from a distribution that is more representative of the training data distribution.

Results for DTW extended to multivariate time series can be seen in Tables 3.2 and 3.3. The distance measures between the dependent generated signals and the dependent training signals are reducing throughout the training process, indicating that the proposed GAN has successfully captured the multivariate training data distribution. Although the LSGAN appears to produce the best quantitative results for the NSR dataset according to the metrics used in this chapter, the DTWGAN produces an improved MMD for the ARR dataset. The best performing GAN is shown in Figure 3.12. Over both datasets, normalising DTW and MMD results, the best performing model is the LSGAN-DTW. The LSGAN-DTW shows a 4.9% improvement over the LSGAN and 4.5% improvement over the DTWGAN. As a result of the LSGAN-DTW being the overall best performing GAN, the results that follow will be reported for this variant unless explicitly stated otherwise.

Table 3.2: Metrics NSR

Modelling Method	Evaluation	
	MVDTW	MMD
LSGAN	**3.0598**	**0.0057**
LSGAN-DTW	3.3069	0.0105
DTWGAN	3.5463	0.0181
LS-GAN	6.7879	0.0194
LS-GAN-DTW	3.3229	0.0086

Table 3.3: Metrics ARR

Modelling Method	Evaluation	
	MVDTW	MMD
LSGAN	**4.0101**	0.0568
LSGAN-DTW	4.3463	0.0195
DTWGAN	4.5918	**0.0157**
LS-GAN	7.4448	0.1917
LS-GAN-DTW	4.2674	0.0747

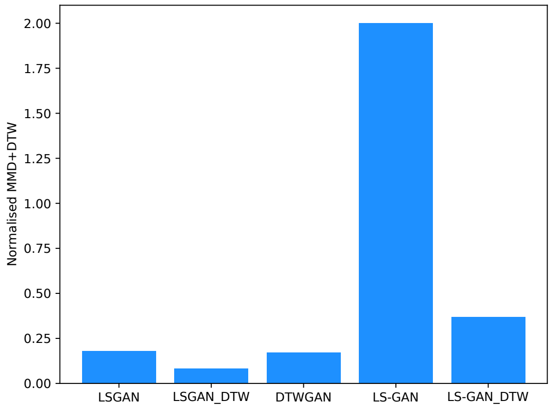

Figure 3.12: Combined, normalised MMD and MVDTW results demonstrate the best performing GAN across both datasets to be the LSGAN-DTW.

Privacy

In terms of privacy, Figure 3.13 and Figure 3.14 shows the presence disclosure (averaged over both datasets) for a membership inference attack on the LSGAN-DTW and LSGAN-DTW-DP, respectively. The number of training records identified is very low (Recall), with approximately 0% correctly identified for $\epsilon \leq 0.4 \times$ mean distance. However, as the threshold ϵ increases above this boundary, the number of records correctly identified as training records increases independently of the sample size r for the GAN without differential privacy. The GAN-DP preserves the training data privacy with no training records identified for $\epsilon \leq 0.5 \times$ mean distance. Precision is approximately 100% for all ϵ and r, which means once an attacker deems that a sample originates from the training set, it is almost always correctly attributed to the training set. Overall, this result tells us that for our generated data, an attacker will have a challenging time correctly identifying if a sample has originated from the training set. Therefore, this GAN architecture and loss function can generate data similar in distribution to the training set while maintaining sufficient privacy of the data.

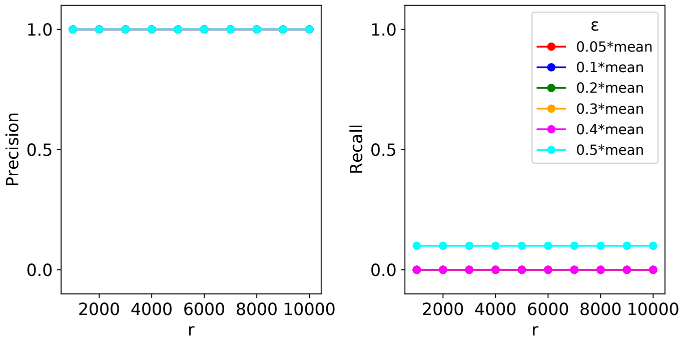

Figure 3.13: Presence disclosure of GAN following a membership inference attack with increasing ϵ.

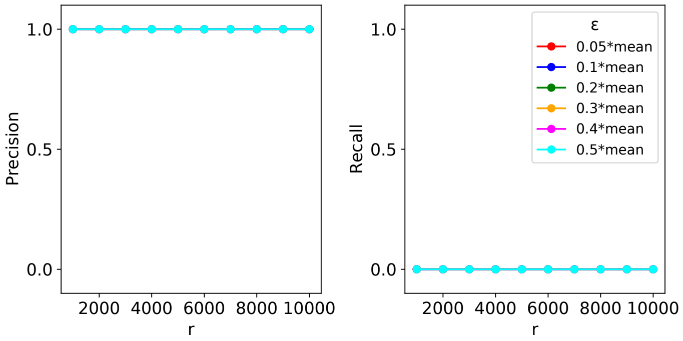

Figure 3.14: Presence disclosure of GAN-DP following a membership inference attack with increasing ϵ.

3.7.2 Benchmarking Results

These experiments offer a benchmark to compare our GAN to other generative models. Table 3.4 shows the classifier accuracy averaged over both datasets for each of the generative models introduced in Section 3.6.5. SVC and LSTM were the two time series classifiers used in these benchmarking tests. A lower classifier accuracy indicates the classifier had difficulty distinguishing the classes, which in this case were the real and generated data. The modelling method that generated the most similar data to the real data was the LSGAN-DTW.

Table 3.4: Classifier accuracy for the generative models averaged over both datasets

Modelling Method	Classifier Accuracy		
	SVC	LSTM	Average
VAE	0.9	0.68	0.79
LSTM	0.95	**0.62**	0.78
LSGAN-DTW	**0.4**	**0.62**	**0.51**
LSGAN-DTW-DP	0.56	0.86	0.71

To complement the results shown in Table 3.4, evaluation metrics were computed for each of the modelling methods. MVDTW and MMD were calculated as in Section 3.7.1 and the results for which are shown in Table 3.5 below, averaged over both datasets. Smaller distances for MVDTW reflect time series that are more similar to each other and for MMD indicate that the real and generated data distributions are closer. As can be seen, LSGAN-DTW has lower distances for MVDTW and MMD, followed by LSGAN-DTWGAN-DP. This quantitatively demonstrates that the data generated using the GANs are more representative of the real data compared with that of the other generative and time series modelling methods.

Table 3.5: Evaluation metrics for other generative models averaged over both datasets

Modelling Method	Evaluation Performance	
	MVDTW	MMD
VAE	47.14	0.96
LSTM	27.87	1.09
LSGAN-DTW	**3.8266**	**0.01505**
LSGAN-DTW-DP	13.36	0.29

3.8 Discussion & Conclusions

The multivariate GAN proposed in this work has demonstrated a capability for generating high-quality, dependent, multichannel ECG signals. Our introduction of the DTW penalisation term in the GAN objective function leads to a more robust design that avoids mode collapse without the need for MBD layers and results in the generation of diverse multichannel time series. We also introduced a new quantitative method for the assessment of output quality for multichannel time series GANs, namely MVDTW. These quantitative methods can complement qualitative evaluation and, in the context of this work, confirm the strong performance of the proposed GAN. Ideally, rather than solely relying on classical and novel metrics, we could enlist the help of a trained physician to classify samples of generated data to determine how accurate the signal traces are, as we have done in our previous work [143]. This also forms an avenue for our future work. Finally, given the nature of the data, it would be interesting to implement conditional models to generate normal and pathological data, which would enable future researchers to generate ECG based on their needs.

To address the growing privacy concerns with sensitive personal data such as physiological or medical data, we demonstrated the ability of the LSGAN-DTW, and in particular, the LSGAN-DTWGAN-DP, to sufficiently conserve the confidentiality of the underlying training data. Implementation of a membership inference attack demonstrated promising results for data privacy with these GANs; protecting and isolating the training set from the generated data ensures that a certain level of privacy is maintained. With the addition of a differentially private GAN architecture, we can generate data and ensure that the privacy of the training data is not violated.

We also presented benchmark experimental results for showcasing the advantages that the LSGAN-DTW holds over other generative time series modelling methods. Most of these well-known methods are tailored explicitly to univariate signals, whereas our methods can be scaled up to multivariate use cases, which include strong coupling between time series. Not only is the proposed method capable of generating multivariate medical time series data, but it also generates data from a closer distribution and distance to that of the training data in comparison to the other generative modelling methods utilised in this chapter.

Multivariate time series data presents an opportunity for the application of GANs in tackling the data shortage and sharing problem in medical research, put forward in Research Question 1. It should be stated here that we only generate ECG waveform data and no associated metadata. This type of data synthesis ensures that there is no risk of reidentifying subjects from our generated data, as discussed in detail in Chapter 1. In terms of our motivating challenge, the successful generation of both private and diverse samples of multichannel and dependent physiological data means we have the potential to use this technology for clinical training and research applications. With that goal in mind, we have shown, for the first time, a GAN design capable of generating high-quality dependent multichannel physiological time series with quality similar to that present in clinically relevant data repositories.

Chapter 4

Human Activity Recognition and Heart Rate Estimation from Single Sensor Devices

4.1 Introduction

From Chapter 2 to 3 we have seen the applicability of Generative Adversarial Networks for generating high-quality time series signals, successfully addressing Research Question 1. In the proceeding chapters, we switch our focus to DL architectures and techniques for processing human physiological signals with the idea of returning state-of-the-art biomarkers and further insights previously undiscovered in the data as laid out in Research Question 2.

Wrist-worn smart devices are providing increased insights into human health, behaviour and performance through sophisticated analytics. However, battery life, device cost and sensor performance in the face of movement-related artefact present challenges which must be further addressed to see effective applications and wider adoption through commoditisation of the technology. We address these challenges by demonstrating, through using a simple optical measurement, photoplethysmography (PPG) used conventionally for heart rate detection in wrist-worn sensors, that we can provide improved heart rate and human activity recognition (HAR) simultaneously at low sample rates, without an inertial measurement unit.

Photoplethysmography is an optical technique commonly employed in wearables and other medical devices to measure volume changes of blood in the microvascular tissue during the cardiac cycle. Light becomes reflected and absorbed at different rates during this cycle and the reflected light is read by a photo-sensor to detect these changes. The pulse rate is obtained from analysis of the small alternating component (which arises from the pulsatile nature of blood flow) superimposed on the larger base signal caused by the constant absorption of light.

Heart rate can be measured at multiple sites on the body using PPG including, but not limited to; ear, forehead, fingertip, ankle and wrist. In the context of personalised health and fitness monitoring using wearables, the wrist is the most frequently used location for photoplethysmographic heart rate monitoring. Accuracies of consumer-grade wearables, for the most part, are acceptable but are prone to errors during daily activities [144]. The difficulties associated with correctly estimating heart rate arise mostly in obtaining a strong physiological reading from the sensors. Often the signals read from the PPG modules are heavily corrupted with motion artefacts and the movement of the limbs is a major contributor to this introduced artefact. Retrieval of a clean PPG signal from a heavily corrupted signal can be achieved by applying filtering techniques including adaptive methods based on a measure of the artefact sourced from an accelerometer-based measurement [145].

Due to the ubiquitous nature of inertial and physiological sensors in phones and fitness trackers, human activity recognition studies have become more widespread [146]. The benefits of HAR include rehabilitation for recovering patients, activity monitoring of the elderly and vulnerable people, and advancements in human-centric applications [147]. In this chapter we show that HAR can also be performed on optical signals by taking advantage of the motion artefact present in the signal. For both heart rate estimation and activity recognition we sought to exploit the motion artefact to infer the physiological states from the PPG signals collected at the wrist.

Our hypothesis was that there is sufficient information in the disturbance induced in the source-detector path to distinguish different activities through the use of a machine learning approach.

The battery life of smartwatches and fitness trackers vary greatly depending on the features and functionality available on-board the wearable. The Apple Watch Series 5, which is more of a lifestyle and fitness tracker, can run for a period of up to 18 hours whereas the Fitbit Charge 3 fitness tracker can go for up to 7 days on a single charge. Continuous activity and heart rate monitoring speed up the depletion of the battery of most wearables. Gathering and processing of simultaneous sensor data can further increase the power consumption of the devices. Without explicitly stating the sampling frequency, Apple state that their heart rate monitor Light Emitting Diodes (LEDs) blink “hundreds of times per second” [148].

Capitalising on recent advancements in machine learning could pave the way for the simplification of wearables, allowing for a reduction in power requirements and subsequently smaller and lower-cost devices. The work described in this chapter is part of a larger-scoped effort to develop easily deployed artificial intelligence which can be used and interpreted by end-users who do not have deep levels of signal processing expertise.

We demonstrate the contributions of our pipeline, using a standalone optical sensor for both activity recognition and heart rate monitoring with significantly reduced sampling frequencies. This novel approach yields not only improved power efficiency but does so without significantly sacrificing accuracy thus advancing the development of simpler, more cost-effective wearables.

Although globally people are using hospitals more efficiently, public healthcare expenditure is rising. For example, in Ireland expenditure has risen from €14.9 billion in 2009 to an estimated €16.8 billion in 2018 with the increasing prevalence of chronic illness requiring long-term patient-provider engagement and management, accounting for roughly 80% of spending [149]–[151]. Frost & Sullivan in 2010 predicted, based on the then-current trends, that healthcare spending in Western economies would almost double (as a proportion of GDP) by 2050, reaching 20%–30% of GDP in some cases. The report also stated that per capita, healthcare spending is rising faster than per capita income in most countries [152].

As a response, globally there is a change in how healthcare is managed. For example, in Ireland, the Department of Health has signalled a major shift in the paradigm of treating people with illnesses. This signals a change in health policy from a reactive to proactive treatment-based models where the focus is increasingly on keeping people healthy [153]. Advancements in digital health technologies, including mHealth and MedTech, have the potential to contribute significantly to a transformation in healthcare delivery, e.g. enabling proactive care through the use of continuous monitoring devices and application of advanced data analytics that enable greater personalisation of treatments [151]. Thus the role of data gathered from wearables is important as part of such a shift in healthcare provision policies.

4.2 Related Work

CNNs have contributed tremendously to the success of machine learning since their introduction in the 1990s. They are an example of neuroscientific principles influencing deep learning [16], in that they are designed to mimic the processing of images in the visual cortex of the human brain [17]. Fully automatic learning of a CNN allows the neural network to extract features that are salient in the input data across different layers. Given the correct training, a CNN allows for the implementation of high accuracy classifiers without the need for signal processing or feature extraction knowledge. This had contributed to their success in practical applications, particularly with image classification.

The current state of the art in HAR systems are camera-based which allow for direct capture of the data but consequently requires significant computer processing to determine distinct activities. HAR studies are frequently carried out using data from inertial measurement units (IMU) which measure proper acceleration of a body or limb. Signal processing and feature extraction for these HAR studies are not trivial, including (but not limited to); singular value decomposition (SVD), support vector machine (SVM) and Random Forest (RF). High accuracy ranging from 80% to 99% can be achieved with such signal processing techniques but they often require a combination of sensor modalities [154] and using multiple IMUs located on various parts of the body which in turn gives rise to scalability and functionality issues in these studies.

Few studies have employed the use of a PPG sensor only for HAR as they are more commonly used with heart rate estimation [155], [156]. Biagetti *et al.* conducted a study on the same dataset used in this chapter for activity recognition [157]. Using the PPG data only for HAR they achieved 44.7% classification accuracy using their feature extraction algorithm. Later the authors combined the PPG data with accelerometer data and achieved 78.0% accuracy using their feature extraction technique. Mehrang *et al.* used a combination of PPG and accelerometer with feature extraction and classification techniques such as RF and SVM, achieving accuracy of $89.2 \pm 4.2\%$ and $85. \pm 6.8\%$ respectively [158].

It should be noted that leading, modern feature extraction and classification techniques using multiple IMUs can achieve 80% to 99% classification accuracy, which may require several sensors located throughout the body [154].

We have found few works that leverage CNN for heart rate estimation. Qui *et al.* computed heart rate estimation from facial videos using a CNN [159]. In [160], the authors proposed a method to estimate heart rate using a CNN trained on a sequence of facial images. Reiss *et al.* sought to solve a regression problem by estimating heart rate from PPG and accelerometer data by computing the Fast Fourier Transform (FFT) and z-normalisation on the 4 input channels to a CNN [161]. Extending on this, using a standalone PPG we develop a CNN regression architecture for heart rate estimation on a single channel time series without any preprocessing.

Junker, Lukowicz and Tröster [162] downsampled wearable accelerometers from 100 Hz in a wearable context recognition system. The authors found that they could achieve sufficient classification accuracy rates for sampling frequencies as low as

20 Hz. However, a significant drop in accuracy (below 60%) is observed when the sampling rate is reduced to 10 Hz.

Finally, in [163] the authors use a developed wrist-worn wearable consisting of a two-axis accelerometer, microphone, light and temperature sensors for context-aware wearable computing. They found that a sampling frequency of 6 Hz yields comparative accuracy compared to much high sampling rates using available time domain features with machine learning. Following this Krause *et al.* demonstrated that this decrease in sampling frequency from 20 Hz to 6 Hz increases the battery life of their constructed wearable by 85%.

4.3 Methodology

4.3.1 Dataset

A readily available wrist PPG exercise dataset collected by Jarchi and Casson [164] and publicly available on PhysioNet was used for the experiments in this work [137]. Data was collected during exercise by 8 healthy patients (5 male, 3 female) with a sampling frequency of 256 Hz. Data was gathered using a wrist-worn PPG sensor on board the Shimmer 3 GSR+ unit for an average recording time of 4 to 6 minutes with a maximum time of 10 minutes. Four exercises were performed; two on a stationary exercise bike and two on a treadmill. The exercises are broken down as follows; walk on a treadmill, run on a treadmill, high resistance exercise bike and low resistance exercise bike. Due to the data collection done during exercise, the PPG signals are heavily corrupted by motion artefacts. No further filtering is applied to the PPG data for the treadmill exercises other than what the Shimmer unit provides on board. For the exercise bike recordings, there was high-frequency noise present which was filtered in MATLAB using a second order IIR Butterworth filter with a 15 Hz cutoff frequency [164].

To accurately evaluate the unfiltered PPG heart rate performance, we compare it with a concurrent ECG that was collected by the authors of the data gathering experiment described above [164]; this provides a ground truth against which to assess our heart rate estimation. The dataset also contains recordings using a 3-axis low-noise accelerometer, a 3-axis wide-range accelerometer, and a 3-axis gyroscope; however, we only utilise the PPG and ECG signals in this chapter. Full implementation details for this chapter can be found online [1].

4.3.2 Downsampling and Segmentation

Prior to segmenting and plotting the PPG signal it was downsampled to a number of different sampling frequencies. The classifier was trained using the full 256 Hz sampling frequency, then retrained on the downsampled frequencies of 30 Hz, 15 Hz, 10 Hz, 5 Hz and 1 Hz respectively.

Once the signal had been downsampled it was then segmented into smaller chunks. A simple rectangular windowing function was used to capture 8 seconds worth of data and step through the data in increments of 1 second. We selected windows of 8 seconds to match the time windows used in [157].

4.3.3 Human Activity Recognition

We opt for an imaged-based approach for the HAR element as inspired by [165]. This methodology allows us to leverage a very deep pre-trained CNN without the need to retrain the whole model. A CNN based on the Inception-V3 architecture and pre-trained on ImageNet [166] was used as the classifier for the HAR experiments. The deep model was retrained leveraging the technique of transfer learning [167], the penultimate layer had its weights updated while all other layers remained the same. This allowed the use of smaller amounts of data to train a model with a large learning capacity that would normally require a lot of data and time to train from scratch. The retraining process can be fine-tuned through the optimisation of hyperparameters. The parameters were set as their defaults in this experiment except for the number of training steps which were changed from 10,000 to 4,000. This

[1] `https://github.com/Brophy-E/CNNs_HAR_and_HR`

helped minimise overfitting through sufficient convergence of the loss function (cross-entropy). See Figure 4.2 for a block diagram of the processing pipeline associated with our methodology.

As a machine vision approach is applied using this classifier, the temporal PPG signals are saved as images rather than time series vectors to be used as input. $Matplotlib$ was used to plot the PPG signal as images, which were saved using the $plt.savefig$ function as 299x299 JPEGs [168] because the default input image size of Inception-V3 is 299×299. All axis labels, legends, titles and grid ticks were removed. Python's $wfdb$ library was used to pull and load the data from PhysioNet [169].

To train the HAR classifier, a total of 6,653 images were stored in four sub-directories of the possible predicted classes (High, Low, Run and Walk). A train/test split validation approach was taken in this experiment. 80% of the data was used for training, 10% for validation and 10% for testing. See Figure 4.1 for examples of PPG data used during training of the classifier.

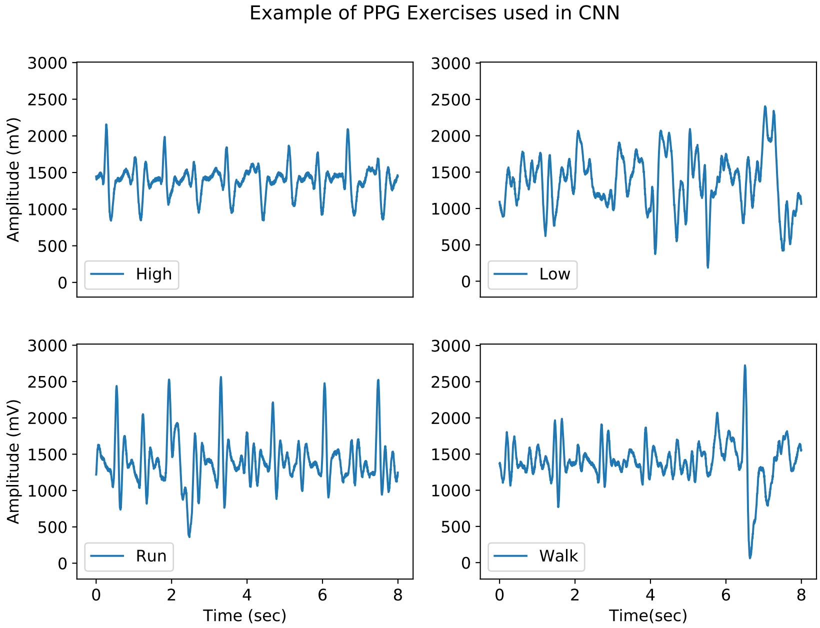

Figure 4.1: Example of PPG from each exercise used in CNN training

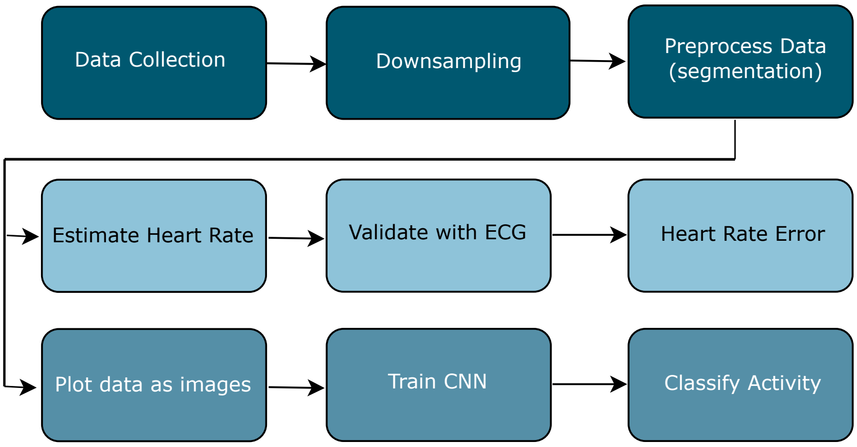

Figure 4.2: Block diagram of our processing approach

4.3.4 Estimation of Heart Rate

We designed a CNN with the output layer replaced by a regression layer. We refer to this model as CNNR (Convolutional Neural Network with Regression). It is a four-layer 1-D convolutional network with batch normalization and ReLU (Rectified Linear Units) followed by a fully connected and regression layer respectively. The model architecture can be seen in Figure 4.3. This model is used to estimate heart rate from the noisy PPG data. We used a train-test split of 90/10 with 5-fold cross validation for the CNNR.

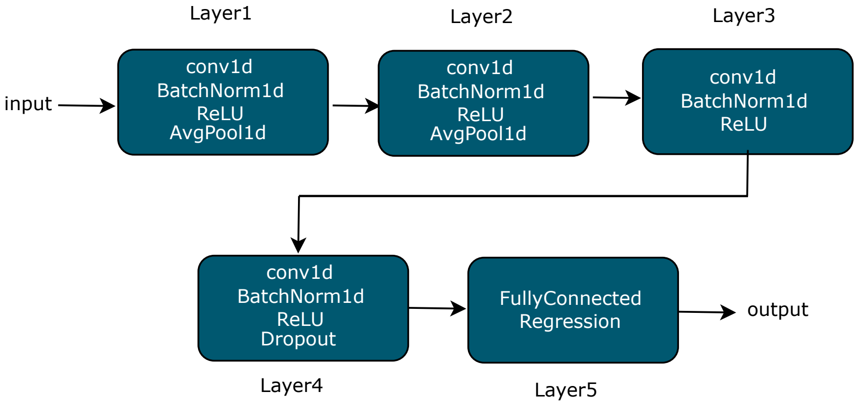

Figure 4.3: Architecture of CNNR

HeartPy, an open-source toolkit for estimating heart rate from the PPG data, was used in our work as a baseline reference to compare the performance of our CNNR approach [170]. The HeartPy toolkit is designed to handle clean and noisy PPG data collected from either PPG or camera sensors. In the case of both our

CNNR and HeartPy work, the PPG data used was the noisy, raw time series signal. The estimated heart rate value for a segment of the signal was then compared to its concurrent ECG time series. The QRS peaks from the ECG were annotated as part of the data collection experiment. An estimated heart rate obtained using the CNNR and PPG toolkit on noisy data was then compared to the ECG heart rate which acted as the ground truth.

4.4 Results

4.4.1 Human Activity Recognition

The results for the HAR experiment are shown in Table 4.1. As expected, the highest classification accuracy of 90.8% is achieved when the original sampling frequency of 256 Hz is used. However, we can still achieve a very competitive estimation performance even after downsampling the original sampling frequency to 5 Hz. Perhaps what is most surprising is the superior performance of our classifier when 10 Hz is chosen as the sampling frequency compared to the higher frequencies (15 Hz and 30 Hz) tested as part of this project. Due to the higher accuracy of 10 Hz we also tested 12 Hz, 11 Hz, 9 Hz and 8 Hz as the chosen sampling frequency but found no anomaly as the surrounding frequencies yield similar accuracies. To further investigate the 10 Hz performance, we low-pass filtered the PPG with a 4.5 Hz cut-off frequency to remove possible aliasing but this did not impact the classification accuracy.

As a sampling frequency of 10 Hz performed the best out of the sampling frequencies tested, we show the training results for this sampling frequency over the 4,000 epochs along with the cross-entropy loss function and confusion matrix for exercise classification in Figures 4.4, 4.5 and 4.6 respectively. We also show the relevant precision, recall and F1-scores in Table 4.2. F1-Score is often used over accuracy when there is an imbalance in the dataset classes. In this case, the dataset contains the following number of samples: 1,263 in High, 1,739 in Low, 1,461 in Run

Table 4.1: Sampling Frequency vs. Accuracy

Sample Frequency	Accuracy
256 Hz	90.8%
30 Hz	82.3%
15 Hz	81.6%
12 Hz	82.1%
11 Hz	81.6%
10 Hz	83.0%
9 Hz	81.2%
8 Hz	80.5%
5 Hz	80.2%
1 Hz	68.5%

and 2,190 in Walk. As a result of this imbalance, we show the F1-Score over the accuracy as the results can be distorted by predicting false negatives. We can also get an understanding of what classes the classifier had a more difficult time with, such as the Low exercise.

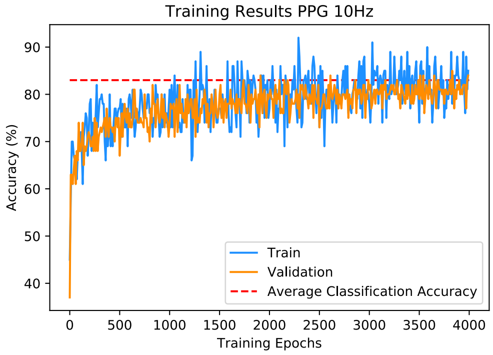

Figure 4.4: HAR training results for 10 Hz sampling frequency

Table 4.2: Precision, Recall and F1-score

Exercise	Precision	Recall	F1-Score
High	0.803	0.904	0.851
Low	0.846	0.666	0.745
Run	0.826	0.849	0.837
Walk	0.834	0.899	0.865

4.4.2 Estimation of Heart Rate

Results for estimating heart rate from the motion artefact (MA) corrupted PPG signal using HeartPy and our CNNR method are displayed below. Figure 4.7 and Figure 4.8 presents the average heart rate error across the various sampling frequencies for each exercise for the two methods. The Heart Rate Error (HRE) is defined here as the absolute difference between the estimated heart rate for a given PPG sample and the heart rate ground truth calculated from the concurrent ECG sample.

For the HeartPy method, exercise specific HRE is similar across all sampling rates except from the 10 Hz sampling frequency on the walk exercise. Other sampling frequencies return an error of between 46% and 55% whereas the 10 Hz sampling frequency reduces the error to 39%. The numerical results for the heart rate experiments is displayed in Table 4.3 where it can be clearly seen that 10 Hz sampling frequency performs best for estimating heart rate from the MA corrupted signal.

Table 4.3: Heart Rate Error using HeartPy

Sampling Frequency	Exercise			
	High	Low	Run	Walk
256 Hz	11.78	**8.14**	19.44	54.94
30 Hz	11.80	10.61	20.71	53.69
15 Hz	12.10	11.15	19.94	46.83
10 Hz	**10.46**	14.05	**17.82**	**39.28**
5 Hz	10.94	10.05	19.27	48.85

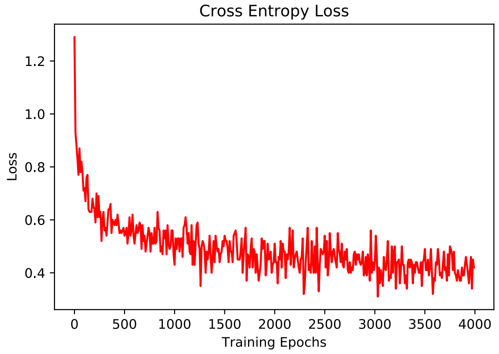

Figure 4.5: HAR cross entropy for 10 Hz sampling frequency

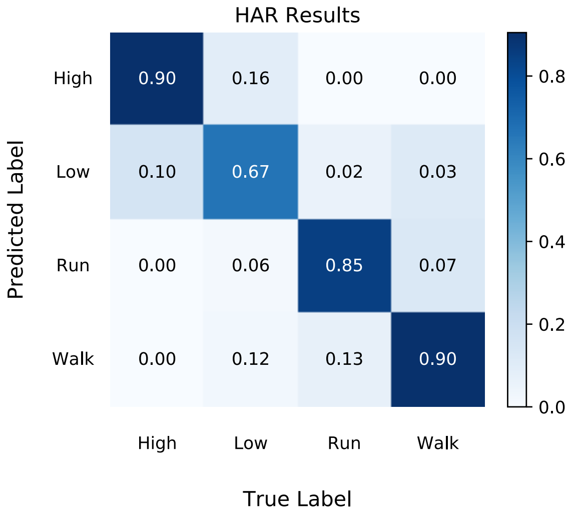

Figure 4.6: Confusion matrix of HAR classifier

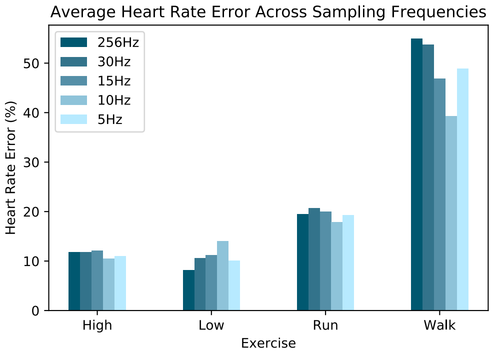

Figure 4.7: Average heart rate error across all exercises and sampling frequencies using HeartPy

Our CNNR results can be found in Table 4.4. It can be seen that the HRE is similar across all exercises and there is not a distinguishable loss in accuracy for any of the sampling frequencies. For the walk exercise there is a great improvement in accurately estimating heart rate compared to the HeartPy method. It should be noted that average HRE across all exercises and sampling frequencies has decreased using the CNNR method from 22.59% to 20.15%, an increase in over 2 percentage points.

4.4.3 Optimisation of CNNR

Following on from our results produced in [171], we wanted to further decrease the heart rate error. Computing a non-exhaustive grid search over some of the

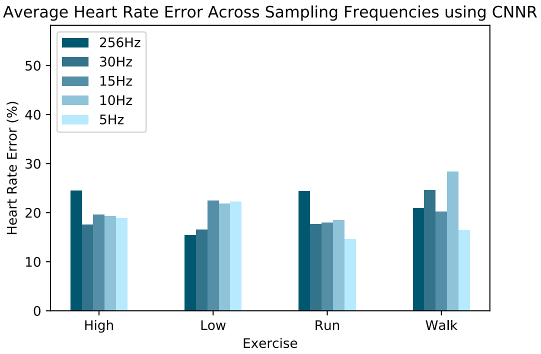

Figure 4.8: Average heart rate error across all exercises and sampling frequencies using our CNNR method

Table 4.4: Heart Rate Error using our CNNR method

Sampling Frequency	Exercise			
	High	Low	Run	Walk
256 Hz	24.55	**15.41**	24.38	20.94
30 Hz	**17.59**	16.58	17.68	24.56
15 Hz	19.62	22.47	18.00	20.21
10 Hz	19.30	21.87	18.49	28.41
5 Hz	18.90	22.26	**14.61**	**16.47**

hyperparameters for the CNNR returned an average HRE of 13.62%, a decrease of nearly 7 percentage points from that of the CNNR without optimisation. We chose, number of epochs, learning rate and the train-test split as some parameters to optimise. The results for the optimisation process have been graphically presented in Figure 4.9. To the authors knowledge, this is the best result using CNNs adapted for regression to estimate heart rate data from raw, noisy PPG sensor data.

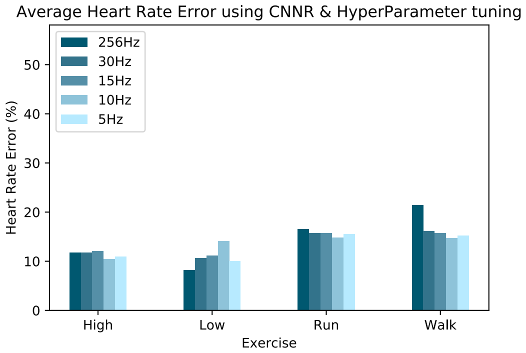

Figure 4.9: Heart rate error using our optimised CNNR

4.5 Conclusion

The approaches used in this chapter yield highly competitive results for HAR even though only the optical signal is used. This demonstrates that more cost and power-efficient wearables are possible through the exploitation of secondary information available from a simple optical sensor. This suggests single-sensor based wearables can achieve much of the functionality and capabilities of more complex multi-modal wearables.

The sampling rate did not have too much of an adverse effect on the performance of the algorithms. Interestingly, the CNN performed better at a 10 Hz sampling frequency compared to 15 Hz and 30 Hz. The reasons behind this have not been fully investigated, but do form avenues for future exploration.

Perhaps what was the most surprising from the results presented in this work was the heart rate estimation results. We demonstrate how a CNN regression approach is capable of robust heart rate estimation even during periods of high artefact. The performance, particularly during these high artefact scenarios, was superior to conventional signal processing approaches for such estimation as demonstrated by the relative performance of the open-source tool kit HeartPy which served as a baseline here. Furthermore, this heart rate estimation performance was sustained even at reduced sampling frequencies. Notably sampling the sensor at 5 samples per second performed just as well as all other sampling frequencies, including the original 256 Hz.

A pervasive computing approach to wearables is taken here. Using a low power wearable with a single optical sensor and a sampling frequency of 10 Hz, we can demonstrate compelling performance both in heart rate estimation and human activity recognition. This has the potential to reduce costs, improve battery performance and encourage wider adoption of digital technologies to a larger population and allow the transition to personalised, patient-centred preventative models of healthcare. Increasing access and affordability to these technologies will, in turn, lower costs and the strain on public healthcare expenditure and help improve overall wellness. We carry this theme through the following chapter; by exploiting single sensor devices using novel ML techniques, we seek to uncover biomarkers and state-of-the-art insights previously unobtainable from classic signal processing methods.

Chapter 5

Estimation of Continuous Blood Pressure from PPG via a Federated Learning Approach

5.1 Introduction

Our work in this chapter addresses Research Questions 1 and 2 in that we are using novel ML techniques and GANs for physiological data processing. We design a novel time series GAN architecture to learn mappings from one physiological signal to another. By applying a distributed learning framework we also adhere to privacy strategies mentioned previously. This work makes an effort to develop readily deployable artificial intelligence systems that non-expert consumers and downstream end-users can easily interpret. Capitalising on recent advancements in machine learning has the potential to simplify wearable devices, allowing for a reduction in power requirements and, subsequently, lower-cost devices, as our previous work also aims to achieve [172].

Chronic heart disease was the number one cause of death from 2000 to 2019, according to the World Health Organisation (WHO), and was responsible for 16% of the total worldwide deaths in 2019 [173]. Heart disease has also shown the most significant increase in deaths during this period. Obtaining unobtrusive, continuous measurements of the cardiac state has proven very difficult. The most commonly used indicator for measuring the state of the heart is blood pressure (BP), which is often gathered using a sphygmomanometer cuff, a finapres, or an arterial catheter. Sphygmomanometers provide spot measurements for BP over a very short time interval, and arterial catheters are an extremely invasive method of continuous BP measurement. The finapres is an alternative for continuous and unobtrusive BP measurement. However, these devices' size, shape, and price mean that they have not been commoditised for individuals seeking continuous home BP measurement devices. Regular monitoring of BP can prove vital for people suffering from cardiovascular diseases (CVDs) who are already vulnerable to BP fluctuations.

Methods for non-invasively measuring continuous arterial blood pressure (ABP) have been explored, using other physiological signals to infer ABP. One example uses the pulse transit time (PTT), the time interval taken for a pulse wave to travel between two arterial sites. PTT varies inversely to BP changes and has been demonstrated to be a valid and accepted measure of BP [174], [175]. PTT is formally defined as the time interval between the Q wave of the electrocardiogram signal and the pulse's arrival at a peripheral site. As such, this information should also be available from a PPG signal.

This chapter presents our novel framework for implementing continuous ABP measurement using a PPG sensor alone. Our methods use proven cutting-edge machine-learning techniques to capture the characteristics that correlate and link continuous PPG to continuous ABP measurements. For the first time, we demonstrate a decentralised learning approach to continuous ABP measurement that is capable of real-world implementation on a large scale and does not compromise patient privacy. This novel approach yields a more power-efficient learning framework, thus advancing the development of simpler, more cost-effective wearables without compromising the accuracy of ABP measurements and patient privacy.

5.2 Methodology

We designed a time-series-to-time-series generative adversarial network (T2T-GAN) (Figure 5.1) based on the popular CycleGAN that is capable of unpaired image-to-image translation [176]. The T2T-GAN can translate from one time series modality to another using cycle-consistency losses. More specifically, we implemented the T2T-GAN for capturing the complex characteristic relationship between ABP and PPG and trained this model to translate a PPG measurement into an accurate continuous ABP measurement. We opted for a decentralised learning approach here and implement federated learning in the interest of data privacy and protection and real-world implementation. With one central aggregate model and many decentralised models, we can implement our framework without handling individuals' personal sensitive data. Comprehensive details of our method can be found in the section that follows.

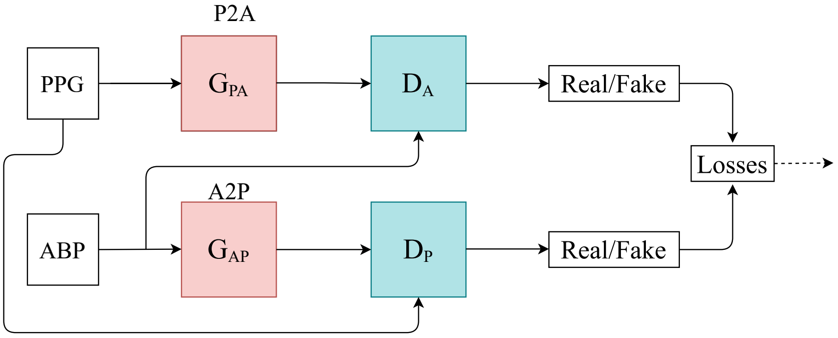

Figure 5.1: Architecture of the T2T-GAN. P2A represents the generator transform function from PPG to ABP. Conversely, A2P represents ABP to PPG.

5.2.1 Computing Platform

The experiments for this project were run on an Nvidia Titan Xp with PyTorch and Google Colaboratory in interest for making the project readily deployable. The code for these experiments are available online[1].

[1] GitHub Repository: https://github.com/Brophy-E/T2TGAN

5.2.2 Dataset

Two open-source datasets were used in this experiment. The first dataset, "Cuff-Less Blood Pressure Estimation", is freely available on Kaggle and UCI Machine Learning Repository. It contains preprocessed and validated ECG (electrocardiograms from channel II), PPG (fingertip) and ABP (invasive arterial blood pressure (mmHg)) signals all sampled at 125 Hz [177], [178]. The raw ECG, PPG, and ABP signals were originally collected from PhysioNet [137]. This dataset is split into multiple parts and consists of several records; for our work, we used the first 5 (part1.mat–part5.mat) records and segmented them into 8-second intervals, which yielded 144,000 training samples (320 h), and the last 2 ($part11.mat$–$part12.mat$) records into 55,000 validation samples (122 h). However, as there might be more than one record per patient (which is not possible to distinguish), we use a second unrelated dataset to test our framework and observe its generalisability. Therefore, we used a [144,000, 2, 1000] dimensional vector that constituted the training dataset for our framework. See Figure 5.2 for an example of the real data used in this work.

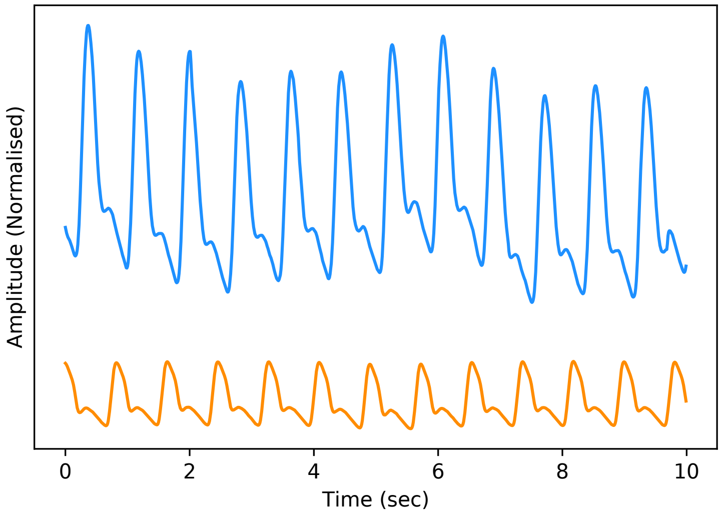

Figure 5.2: Example of Real PPG (top, blue) and ABP (bottom, orange). The signals are both normalised between 0 and 1 with an artificial offset on the ABP signal for visualisation purposes

The test dataset "University of Queensland vital signs dataset: development of an accessible repository of anaesthesia patient monitoring data for research" [179] provides a multitude of vital sign waveform data recorded from patients undergoing anaesthesia at the Royal Adelaide Hospital. The physical state of patients under anaesthesia contains marked changes to cardiovascular variables compared to ICU patients, presenting a further challenge to our framework. We are only concerned with the ABP and PPG measurements from this dataset; these are sampled at 100 Hz. We selected only one patient, namely Case 5, and segmented the data into 10-second intervals, which yields a [900, 2, 1000] dimensional vector (150 min) that constitutes the test dataset for our framework. We are only concerned with the PPG and ABP signals from these datasets.

5.2.3 Model

As previously mentioned, we adopted the learning framework of CycleGAN for time series data to translate from one time series modality to another. Here we will explicitly define the Discriminator and Generator architecture of our T2T-GAN. The Generators G_{PA} and G_{AP} are two-layer stacked LSTMs with 50 hidden units in each layer and a fully connected layer at the output, with an input size of 1000, see Figure 5.3. The Discriminators D_A and D_P are 4-layer 1-dimensional CNN with a fully connected layer and sigmoid activation function at the output, see Figure 5.3.

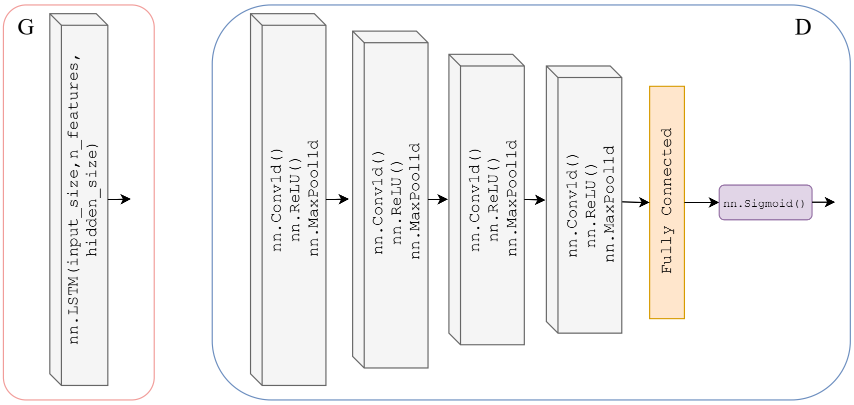

Figure 5.3: Architecture of Generators G_{PA} and G_{AP} (left) which are two-layer stacked LSTMs with 50 hidden units in each layer and a fully connected layer at the output, with an input size of 1000. Architecture of Discriminators D_A and D_P (right) which are 4-layer 1-dimensional CNNs (ReLU activation and max pooling functions) with a fully connected layer and sigmoid activation function at the output.

5.2.4 Federated Learning

To make the model perform closer to a real-world setting and to prevent data sharing to third-parties we implement the decentralised learning approach of Federated Learning. Our approach is limited to using one central server so to realise this learning method we split our dataset into N (where $N = 20$) equally sized random smaller data subsets and train N client-GANs on their own data with no cross-over from their respective subsets. The client-GANs are trained for e (where $e = 5$) epochs and their weights are then sent to a global-GAN that aggregates the received weights from the N clients-GANs. This global-GAN can then operate on unseen data or update the client-GANs with the aggregated global weights which eliminate the need for any data centralisation, see Figure 5.4 for a visual example of our method. Of course in a real-world training and testing environment, the training data will not come from a centralised repository. The data will instead be generated by the end-users. Consumers will generate their own PPG data from their smartwatch, in this case, that will be used to train a local-model and communicate weights to and from a global-model, see Figure 5.5 for a conceptual example. Each client GAN requires local training, meaning each GAN will need some form of blood pressure ground truth available during its training phase.

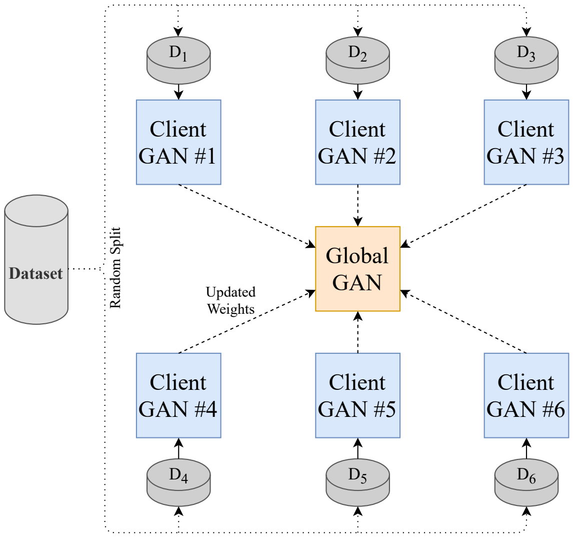

Figure 5.4: Federated Learning methodology employed in this chapter (Left). Each GAN is represented by the model shown previously in Figure 5.1. Federated Learning methodology that is implemented in the real-world (Right).

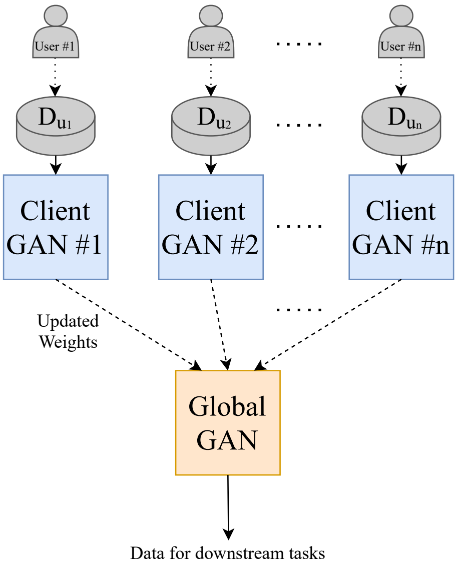

Figure 5.5: Federated Learning methodology that is implemented in the real-world (Right). Each GAN is represented by the model shown previously in Figure 5.1.

5.2.5 Training

We chose a total of 20 client models for training as demonstrated in Figure 5.4. Each used an equal proportion of the dataset. Six random clients were selected from the total client models in each communication round to be trained. There were ten communication rounds. Following each round of training on the client device, the aggregation of weights is computed on the global model. The total number of training rounds on each client was 5, with a batch size of 32. The total loss function of our T2T-GAN framework is calculated as in Equation (5.1).

$$\begin{aligned} L(P2A, A2P, D_P, D_A) = {} & L_{T2T-GAN}(P2A, D_A, PPG, ABP) \\ & + L_{T2T-GAN}(A2P, D_P, ABP, PPG) \\ & + \lambda_c L_{cyc}(P2A, A2P) + \lambda_i L_{identity}(P2A, A2P) \end{aligned} \tag{5.1}$$

where L_{cyc} and $L_{identity}$ are the cycle consistency loss and identity loss, respectively, and are defined by the L1-norm. $L_{T2T-GAN}$ is defined as the mean squared error loss. λ controls the relative importance of the two objectives, λ_c and λ_i were chosen as 10 and 5, respectively, as we want to emphasise the importance of cycling between the time series modalities (PPG to ABP). See Algorithm 1 for a full description of the training procedure.

Algorithm 1 FedT2TGAN Training Procedure.

```
Input: Training sample pairs of ABP and PPG $S_n = (X_{ABP1,PPG1} + ... + X_{ABPn,PPGn})$
Output: GAN Model T2TGAN
Initialise global-model
Synchronise client-models with global-model
for num clients do
    # Communication Round
    Select 6 random client-models to train
    for each client-model do
        # client-model training
        for num epochs do
            # Calculate identity, cycle and adversarial losses
            $L$ = GetGANLosses($X_{ABP,PPG}$, $\hat{X}_{A\hat{B}P,P\hat{P}G}$)
            # Update weights of client-model
            $W$ = UpdateClientWeights(L)
        end for
        Aggregate client-model's weights with global-model
    end for
end for
Generate ABP waveforms from unseen PPG using trained global-model
```

5.2.6 Evaluation

To successfully evaluate our model, we examine the mean arterial pressure (MAP) of generated samples. Using a completely independent test dataset from the training dataset grants us the freedom to implement a leave-one-out strategy and see how well our model generalises to other ABP-PPG datasets. We take the PPG measurements from the test dataset and pass them through our trained global deterministic function, P2A. This function converts our PPG signal to a corresponding ABP signal, and we then calculate the MAP from the generated signal and compare it with the true MAP measurements from the real ABP signal. MAP is considered a better indicator of perfusion to vital organs than systolic blood pressure (SBP) [180]. It is important to note that we can retrieve the systolic and diastolic blood pressure (DBP) from the P2A signal, which we use to calculate the MAP (5.2) rather than simply returning the mean of the continuous signal segment. We also present the Bland–Altman (BA) plots of the MAP error [181] that allow us to determine to what degree the generated ABP is a good substitute for the real ABP. The Association for the Advancement of

Medical Instrumentation (AAMI) standard requires BP measuring methods to have a mean error (μ) and standard deviation (σ) of less than 5 mmHg and 8 mmHg, respectively, [182]. Following this, we then select the entire 150-min period of the test data and perform calibration on these data for the first one-minute period only. This calibration is designed to remove user bias and provide more accurate results while mimicking a continuous BP measurement test that can be performed clinically. Bland–Altman plots are provided for the calibrated and uncalibrated measurements.

$$MAP = [SBP + (2 \times DBP)]/3 \tag{5.2}$$

In the interest of providing a comprehensive evaluation of our T2T-GAN, we implement the dynamic time warping, root-mean-squared error (RMSE) and Pearson Correlation Coefficient (PCC) algorithms as distance and similarity measures between the real and generated time series BP sequences for both the federated and unfederated approaches. These metrics allow us to quantify the similarities in the structure of the blood pressure waveforms. This is implemented for the entire test dataset (150 min, 900 samples at 10 s/sample) and a random sample of the validation dataset of equal size.

5.3 Results

As stated previously in Section 5.2.6, we evaluate our framework based on a qualitative and quantitative perspective. Visually, and therefore from a subjective qualitative perspective, we determine that our federated T2T-GAN framework has successfully modelled ABP from a single optical PPG signal alone. An example of real and generated data can be seen in Figure 5.6 below.

Observing the Bland–Altman plot in Figure 5.7 our framework achieved a mean MAP error of -4.02 mmHg and a standard deviation of 22.6 mmHg. We also present the Bland–Altman plots over the 149-min period with a 1-min calibration period that achieved a mean MAP error of 2.95 mmHg and a standard deviation of 19.33 mmHg. This calibration period can prove useful in bringing the mean error to within the AAMI standards. The BA plots show the 95% range from $\mu - 1.96\sigma$ to $\mu + 1.96\sigma$.

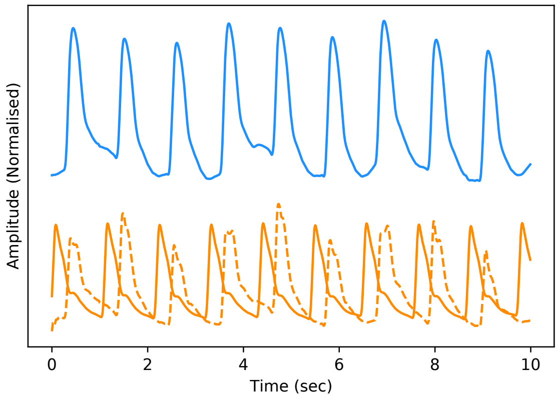

Figure 5.6: Example of Real PPG (top, blue) and the corresponding real ABP (bottom, dashed-orange) along with the fake ABP (bottom, orange) generated using the respective PPG. The signals are both normalised between 0 and 1 with an artificial offset on the ABP signals for visualisation purposes

The MAP range of [25.16 mmHg, −20.08 mmHg] in Figure 5.8 demonstrates that the one-minute calibration period was successful in reducing the overall bias in the mean error.

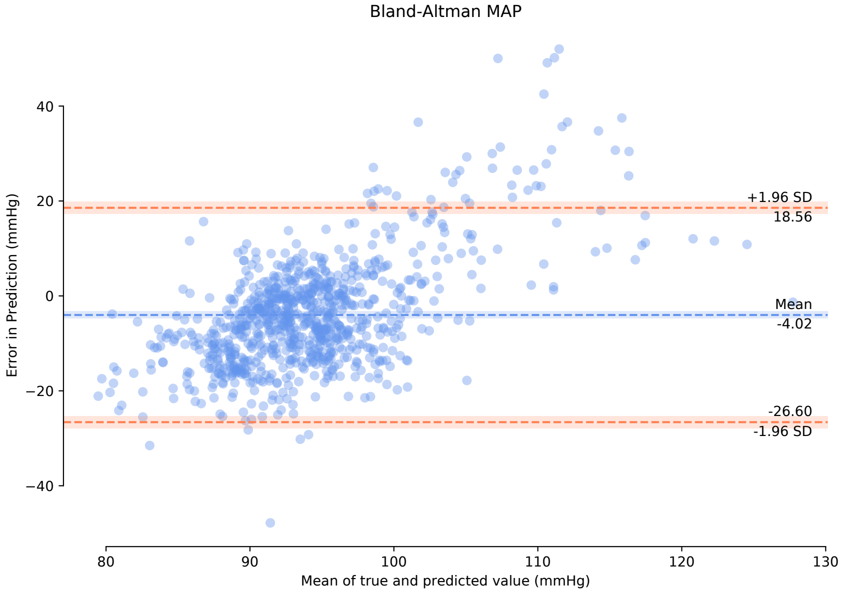

Figure 5.7: Bland–Altman plots of Mean Arterial Pressure on the unseen, unprocessed test data with a mean error of $-4.02\ mmHg$ standard deviation of $22.6\ mmHg$.

However, a qualitative evaluation cannot be considered a successful framework justification due to the lack of a suitable objective measure. Therefore, we compute DTW, RMSE error, and PCC of the real vs. generated continuous ABP signals from a quantitative perspective. The time series similarity results on the validation and test datasets for both the federated and un-federated frameworks are displayed in

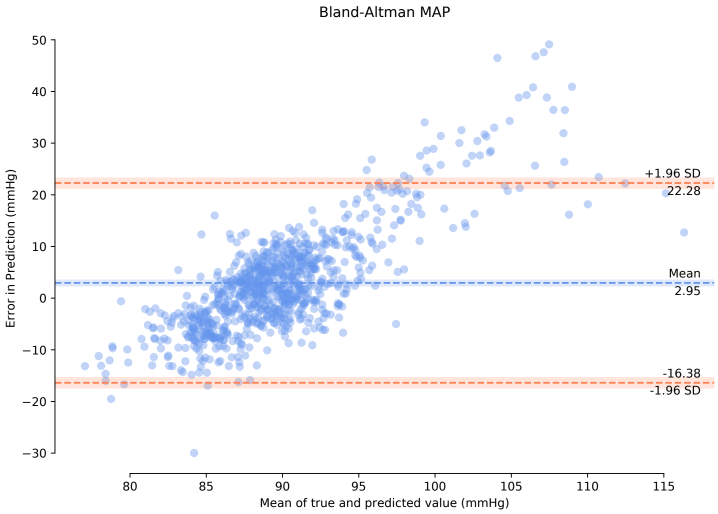

Figure 5.8: Bland–Altman plots of Mean Arterial Pressure on the unseen, unprocessed test data following a one-minute calibration period with a mean error of 2.95 mmHg standard deviation of 19.33 mmHg.

Table 5.1 below. It can be seen that, as expected, the federated results are degraded slightly compared to the non-federated results. However, in both cases, the models perform seemly equal on the validation dataset as they do on the test dataset.

Table 5.1: Time series similarity metrics

Federated Learning	Dataset	DTW	RMSE	PCC
No	Test Dataset	56.73	0.19	-0.11
No	Validation Dataset	55.18	0.23	-0.33
Yes	Test Dataset	62.55	0.24	-0.22
Yes	Validation Dataset	62.15	0.25	-0.34

We have implemented an explainable AI (XAI) approach known as t-Distributed Stochastic Neighbor Embedding (t-SNE) [38] in the interest of trustworthy AI. This is a well-known technique of dimensionality reduction that is suited well to the visualisation of high-dimensional datasets. We implement t-SNE on our real ABP and PPG test datasets, as well as the generated ABP data. Figure 5.9 (a) illustrates a clustering effect between the real and synthetically generated ABP that is distinctly different from the t-SNE embedding on the PPG data. This demonstrates that we can now effectively cycle the time series PPG data from its own modality (Figure 5.9

(b)) to a distribution that is much more representative of the ABP data.

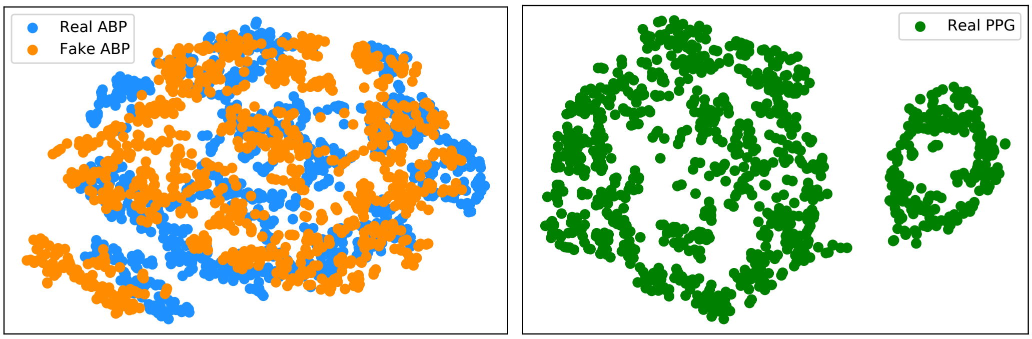

Figure 5.9: (a) t-SNE visualisation of real ABP (blue) and generated ABP (orange) dataset. (b) t-SNE visualisation of the real PPG dataset.

5.4 Discussion and Conclusion

Here, we have presented a novel decentralised learning framework for generating continuous ABP data and MAP estimates using a single optical sensor alone. Although our results of a mean error of 2.95 mmHg and a standard deviation of 19.33 mmHg do not meet the AAMI criterion, it must be stated that for our test dataset, we obtained a completely separate dataset and carried out no further processing on the retrieved data other than segmentation. Our framework performs deceptively well due to the real-world nature of the test dataset and the fact that, as stated before, the physical state of patients under anaesthesia contains marked changes in their cardiovascular variables (ABP and PPG in this case) in comparison to patients in the ICU (training dataset). With further work on cleaning and preprocessing the datasets, we might observe improved results, such as the results observed in [183]. However, we did not implement this as part of this work in keeping with noisy real-world data. In the case of using denoising methods for PPG signals to obtain clean training data, we can seek to apply the techniques listed in [184]. Furthermore, with the increasing environmental costs of machine-learning practises worldwide, we are concerned with model complexity and training time. Our model takes a total time of 5 h to train compared to the 11–12 days to train the models in [183]. We also add a layer of interpretability to our results through the use of t-SNE, which demonstrates that the

T2T-GAN can successfully cycle one time series modality to another. Although this work did not achieve the competitive performance of fully connected networks, it should be noted that this work is conceptually quite different from more conventional approaches and opens up new opportunities for consideration, particularly regarding federated learning and privacy.

Sustainable AI is an essential practice in the research community to continuously build quality machine learning systems while consuming smaller amounts of power. Furthermore, explainable AI is crucial to bridge the gap between human and computer understanding and build human trust in these AI systems. Overall our framework lays the foundation for continuous ABP measurements on a large scale for the first time by providing a sustainable, explainable, and private real-world example of how our models learn from small subsets of personal data and generalise well to previously unseen data. Achieving all this while using a sole PPG sensor will subsequently lead to lower-cost devices and the commoditisation of such. This may be one such solution for clinicians to remotely and accurately monitor patients' cardiovascular states and unlock further biomarkers into the human physiological state as we put forward in RQ2.

Chapter 6

Biosignal Denoising using Deep Learning Signal Processing

6.1 Introduction

The overall theme of Chapters 4 and 5 is centred around the measurement of human health in real-world settings utilising DL for optimal inference of physiological parameters. In this chapter we focus on denoising biological signals such as electroencephalography (EEG) and electrocardiography (ECG) signals using a combination of different ML and DL techniques.

The electrocardiogram is a non-invasive method to measure the heart's electrical activity and commonly used to diagnose heart disease. Chronic heart disease was the number one cause of death from 2000–2019 [173] and has shown the most significant increase in deaths during this period. Long-term ECG monitoring is currently the gold standard for diagnosing cardiovascular diseases. Unfortunately, obtaining reliable, long-term measurements of the cardiac state is a logistical challenge faced by health care professionals due to the time and resources involved. As a result, patients are frequently required to collect their ECG data remotely on a wearable device, indirectly leading to noise manifesting in the ECG signals.

EEG is a method of measuring the electrical activity of the brain. It is a non-invasive procedure that obtains measurements via several electrodes placed on the scalp of the patient. EEG has become an essential tool for practitioners in diagnosing abnormal brain activity and neurological conditions such as epilepsy. A recurring issue with EEG readings is that they can be heavily corrupted with artefacts induced from muscle movements, electrical interference or loose electrodes, to name a few. These artefacts make classification and, consequently, diagnosis of neurological conditions a bottleneck. As a result, denoising EEG has become an extensive area of research in the biomedical signal processing domain ([185], [186]).

In both the ECG and EEG case, contaminated signals can suppress essential pathological biomarkers and, in some cases, will render the biosignals completely unusable. A clean ECG signal of a typical, healthy patient is shown in Figure 6.1. The ECG signal contains essential information about the cardiac pathologies affecting the heart, characterised by five peaks known as fiducial points, represented by the P-wave, QRS complex, and T-wave. We also present an example of a typical single channel EEG signal in Figure 6.2.

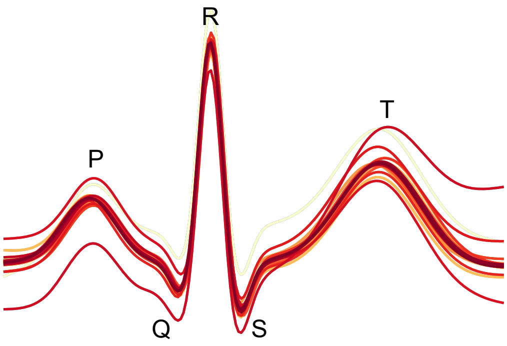

Figure 6.1: Typical ECG traces with fiducial points labelled, generated using *neurokit2* [199].

ECG signals can be contaminated by many types of noise such as: baseline wander, powerline interference, electromyographic noise, electrode motion artefact noise [200]. Baseline wander is a low-frequency artefact in electrocardiogram signal recordings that arises from breathing, electrically charged electrodes, or subject movement [201]. Muscle artefacts are generated due to skeletal muscle activity [202], and electrode motion is caused by changes in electrode-skin impedance and changes

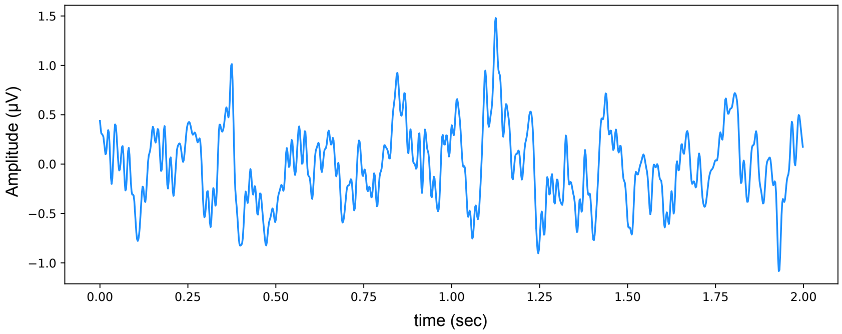

Figure 6.2: Typical example of clean EEG.

in skin potential [203].

Electrooculographic (EOG) and electromyographic (EMG) signals are among the most common sources of noise in the EEG. EOG originates from eye movements, such as blinking and rolling, whereas EMG originates from movements of the surrounding muscles. These artefacts are highly prevalent because it is next to impossible to prevent blinking of the eye and twitching of surrounding muscle groups. As a result of artefacts ubiquity in EEG signals, it becomes apparent that there is a need to estimate these noisy signals accurately and remove them to obtain a high-resolution EEG signal upon which fast and accurate diagnosis can be performed.

In the remote setting, where clinicians cannot readily inspect signal traces for artefacts, noise can become prevalent in the data, leading to a poor signal-to-noise ratio (SNR) and loss of vital information. This may conceal features that are important for diagnosis. In the ECG example, the interbeat interval (IBI), calculated using the R-wave, which is valuable for heart rate variability (HRV) measurements, can become difficult to estimate if the introduced noise heavily disfigures the R-wave. Therefore, denoising becomes fundamental to downstream ECG signal processing tasks. Likewise, with the EEG, the presence of artefacts makes downstream classification and diagnosis of neurological conditions much more challenging.

Identifying artefacts in the ECG signal can help improve patients diagnosis and treatment. As more recordings are being conducted in remote settings, this implies the manifestation of more artefacts in the ECG. Artefact detection approaches have been implemented in the literature to detect disturbances and help better estimate the quality of the recorded ECG signals [204]–[206]. Rather than taking an artefact detection approach, we seek to automatically suppress the artefact and improve the ECG signal structure essential for HRV analysis.

Similarly, several methods have been used in the past to denoise EEG signals. For example, Salis *et al.* implement a comparative study of Empirical Mode Decomposition (EMD), Discrete Wavelet Transform (DWT) and Kalman Filter (KF) in an attempt to remove EOG artefacts with different amplitudes from EEG [187]. Perhaps the most popular approach for denoising EEG signals is Independent Component Analysis (ICA) [188]. ICA is considered a powerful tool for extracting the EEG signal of interest, and it is used to filter out artefacts from the signal. The benefits of using ICA are the most apparent when a multi-channel signal is recorded [189]. However, more recent developments have focused on deep learning to denoise EEG signals, such as Zhang *et al.*, who denoise EOG and EMG artefacts using a novel deep learning-based architecture [190].

This chapter proposes a custom loss function capable of denoising electrode motion artefact in ECG data to a higher standard than other, more common loss functions. We implement our custom loss function with a convolutional neural network to return high-quality ECG, suitable for calculating key human health metrics from a previously unobtainable state.

We also present our EEG denoising pipeline based on a Generative Adversarial Network. We use two datasets to demonstrate the capability of our system. One dataset consists of EEG signals collected using the ANT Neuro eego sports. The other EEG data is the benchmark dataset EEGdenoiseNet presented in [190]. We show both the power spectral density (PSD) and SNR along with other classical time series similarity measures for quantitative metrics and compare our framework to the benchmark in the literature.

Conventional noise reduction methods focus on overall improvement in SNR but ignore the preservation of essential peaks. These peaks are necessary for heart rate, IBI, and HRV measurements for monitoring exercise, stress, and cardiovascular disease. In this chapter, we demonstrate a convolutional neural network and custom loss function for ECG signal denoising and R-wave preservation. This novel approach yields improved SNR and cardiovascular measures while allowing for less complex deep learning models, thus advancing the development of simpler, more reliable remote patient monitoring devices.

Furthermore, we showcase our competitive deep learning technique capable of denoising common artefacts induced in EEG data. Through SNR and other signal evaluation measures, we show the GAN is capable of high-quality denoising that outperforms the current deep learning benchmarks. Our experiments illustrate the potential for use in the brain-computer-interface (BCI) setting. A BCI is a device that allows a user to communicate with external devices by translating thought processes into physical signals. These physical signals can then be used to drive or inform external devices such as a computer. Developments in the BCIs space have the potential to offer portable, low-cost options for remote monitoring of the physiological state of the human brain.

6.2 Related Works

6.2.1 ECG Denoising

In the early stage of ECG denoising research, low-pass filters [207], adaptive filters [208] and filter banks [209] were utilised. Recently, there has been a move towards data-driven approaches for ECG signal denoising that are more suited to non-linear and non-stationary time series signals. Chatterjee *et al.* provide a review of techniques for noise removal in ECG signals [200]. The authors review six methods of ECG signal denoising, namely empirical mode decomposition, wavelet-based models, sparsity-based models, Bayesian-filter-based models, hybrid models and deep learning models based on autoencoders. In this work, we opt for a deep learning-based approach to denoising our ECG signals.

Corneliu *et al.* review deep learning-based models for removal of noise in ECG signals [210]. The authors mainly focus on Long short-term memory networks [60] and Convolution Neural Networks. They find that CNNs outperform LSTMs in the deep learning models. Given the correct training, a CNN allows for the implementation of high accuracy classifiers without the need for signal processing or feature extraction knowledge. This had contributed to their success in practical applications, particularly with image classification. Here, we implement a CNN-based architecture with a custom-loss function for denoising our ECG signals.

Loss functions are used in statistical models to define an objective function that evaluates the model's performance and enables the model to learn its parameters by minimising a said loss function. The Mean Squared Error is among the most popular loss functions used in machine learning problems. Mean squared error is calculated as the average of the squared differences between the predicted and actual values. Barton *et al.* introduce a non-standard loss function in Raman Spectra denoising by adding another term to the MSE loss to balance between overall signal denoising and excessive smoothing of spectral peaks [211]. The authors identified that traditional denoising algorithms, including CNNs with standard loss functions, successfully remove noise at the expense of smoothing or blurring the sharp spectral peaks, the heights of which are important in the context of Raman based diagnostics. Here, we identify the same problem for ECG signals, whereby traditional denoising methods can adversely distort the underlying heart signal. We extend the work of Barton *et al.* through the addition of multiple terms in the standard MSE loss, which helps improve signal denoising while maintaining the QRS complex structure, and in turn, the overall signal-to-noise ratio of the ECG.

6.2.2 EEG Denoising

GANs were initially developed for image generation and improved image synthesis. Although this has gained a lot of traction over recent years, there has been a movement towards implementing GANs for sequence and time series generation, imputation and augmentation [135]. In this work, we employ GANs to denoise common EEG artefacts experienced in BCIs.

GANs have been used for EEG generation, and augmentation, as in [104], [105], [191]–[194]. However, few works have explored GANs for denoising time series, particularly where EEG data is concerned.

Gandhi *et al.* [195] designed Asymmetric-GANs for denoising EEG time series data. Their model for denoising time series is trained using unpaired training corpora and does not need information about the noise source. Sumiya *et al.* [71] denoise mice EEG using adversarial training. Their training process requires a set of noisy signals and clear signals. Although these methods reduce the noise present in the EEG signals, they do not provide specific artefact removal nor solid quantitative evidence of the improvement in the SNR. We improve on this by showcasing GANs as a robust artefact removal/denoising tool via the benchmarking experiments and demonstrate both strong qualitative and, more importantly, quantitative evidence that our GAN is a competitive performer in improving the state-of-the-art denoising methods for EEG artefacts.

Other deep learning methods such as Convolutional Neural Networks and Variational Autoencoders have been used in the past to effectively denoise EEG signals [196], [197]. We demonstrate that the GAN developed in this work is competitive with the state-of-the-art deep learning methods. Many methods proposed in the literature deal with only one artefact type with each architecture. Our model is generalisable to each of the three artefacts explored in this work; in other words, the same architecture can be retrained to remove more than one artefact type effectively.

6.3 Methodology

6.3.1 ECG Datasets

We use two datasets in this work; both datasets are open source and freely available on PhysioNet [137]. The first and primary dataset we use is the MIT-BIH Arrhythmia

Database. This dataset contains 48 half-hour recordings of two-channel ambulatory ECG, including less common but clinically significant arrhythmias. The ECG recordings were digitised at 360 Hz [77].

The secondary dataset used in this work is the MIT-BIH Noise Stress Test Database that includes 12 half-hour ECG recordings and three half-hour recordings of noise typical in ambulatory ECG recordings [212]. We are only concerned with using the noise recordings from this dataset. The noise recordings were made using physically active volunteers and standard ECG recorders, leads, and electrodes; the electrodes were placed on the limbs in positions where the subjects' ECGs were not visible. We select the electrode motion artefact and artificially add it to the MIT-BIH Arrhythmia data. Electrode motion artefact is generally considered the most troublesome since it can mimic the appearance of ectopic beats and cannot be removed easily by simple filters, as can noise of other types [212].

6.3.2 EEG Datasets

Two open-source datasets were used in this experiment. The first dataset of EEG signals was downloaded from PhysioNet [137],[198]. For this dataset, subjects performed different motor/imagery tasks while 64-channel EEG was recorded using the BCI2000 system [1] and sampled at 160 Hz. Each subject performed 14 experimental runs: two one-minute baseline runs (one with eyes open, one with eyes closed) and three two-minute runs of four motor movement and imagery tasks. We used the baseline eyes open recordings only from this dataset and artificially added mains noise at 50 Hz. The data preprocessing steps for this experiment are described in further detail in Section 6.3.4. We refer to this dataset as EEG-50 for the remainder of this work.

Our second dataset used is the EEGdenoiseNet, a benchmark EEG dataset designed to be implemented with deep learning-based denoising technologies. We use this dataset to act as a performance comparison of our GAN to the models tested

[1] `http://www.bci2000.org`

in the data collection paper, and we further benchmark our GAN model against other deep learning EEG denoising architectures previously used in the literature. EEGdenoiseNet contains a total of 13512 physiological signal segments. Of that, 4514 records are clean EEG, 3400 are ocular artefact records, and 5598 are muscular artefact records. This allows the dataset user to synthesise artificial EOG and EMG artefacts into the clean EEG records, resulting in contaminated EEG segments with the ground truth clean EEG. The EOG data was sampled at 256 Hz, and the EMG data was sampled at 512 Hz. The data preprocessing steps are described in detail in Section 6.3.4. Examples of both EOG and EMG artefacts can be seen in Figure 6.3.

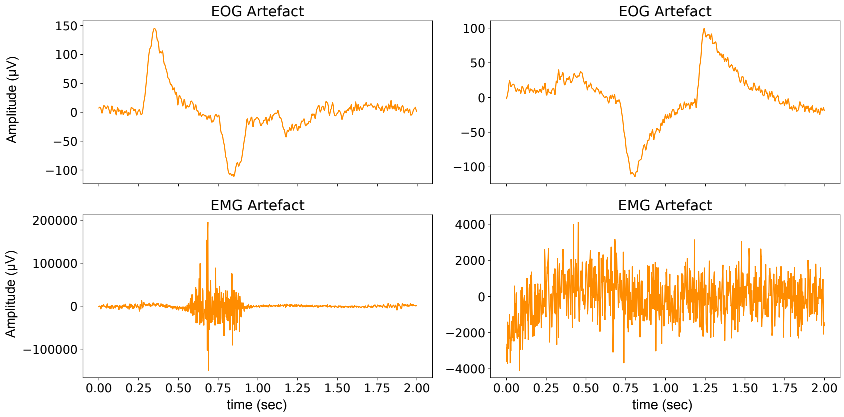

Figure 6.3: Examples of both EOG and EMG artefacts that have been artificially added to the dataset.

6.3.3 ECG Preprocessing

For each of the 48 signals in the arrhythmia dataset, the electrode motion noise signal mixed linearly with the clean ECG record as defined by equation (6.1).

$$y = x + \lambda \times n \tag{6.1}$$

Here x is the clean ECG signals, n is the electrode motion artefact, and λ is a hyperparameter that controls the SNR levels of the noisy EEG signal y. In addition, the dataset is divided into 80% of the data for training and 20% testing. We chose the first 38 records for the training and validation sets the last ten records for testing, ensuring we have a ‘leave n-subjects’ out approach. Following this, we chose a sliding window of 3-seconds over the data with an overlap of 0.5 seconds. Further details are available online [2].

6.3.4 EEG Preprocessing

Toy Data Processing

This dataset uses only the resting, eyes open EEG records from the eegmmidb database available on PhysioNet. The dataset is digitised initially at 160 Hz. We segment each EEG record into four second long intervals with an overlapping sliding window of two seconds. This yields *12200* EEG records which then have a noisy sinewave of varying amplitudes centred around 50 Hz added to the clean EEG signal. As a result of this, we then have the corresponding clean and noisy signal pairs. The dataset is then normalised before training.

EEGdenoiseNet Data Processing

The EEGdenoiseNet datasets use the same training setup as described in the original paper [190]. First, the noisy EEG segments are created by linearly mixing the clean EEG segments with the EOG and EMG artefacts according to equation (6.1). In this case, x is the clean EEG signals, n is the artefact (either EOG or EMG), and λ is a hyperparameter that controls the SNR levels of the noisy EEG signal y.

The contaminated signals are from a combination of EEG segments and ocular or myogenic artefact segments, with 80% for generating the training set and 20% for generating the test set. Each set was generated by randomly linearly mixing EEG segments and EMG or EOG artifact segments according equation (6.1), with SNR ranging from ten different SNR levels (-14 dB, -12 dB, -10 dB, -8 dB, -6 dB, -4 dB, -2 dB, 0 dB, 2 dB, 4 dB) rather than the 10 levels of (-7 dB, -6 dB, -5 dB, -4 dB, -3 dB, -2 dB, -1 dB, 0 dB, 1 dB, 2 dB) in [190]. This procedure expanded each dataset to ten times its original size. The clean EEG records act as ground truth, and the corresponding mixed records are the noisy EEG. Full experimental

[2] `https://github.com/Brophy-E/ECG_Denoise_Custom-Loss`

details are available online [3].

6.3.5 ECG Denoising Model and Custom Loss Function

We designed a four-layer 1-D convolutional network with batch normalisation and ReLU (Rectified Linear Units) followed by a fully connected layer. The model architecture can be seen in Figure 6.4. We chose our CNN model as it has successfully estimated heart rate in previous works [171], [172]. In addition, CNNs are pervasive in embedded devices and have achieved high performance in many real-world problems. However, their implementation often requires high-performance hardware [213]. Therefore, designing a CNN model also allows us to demonstrate the benefits of our custom loss function that can reduce the complexity of such systems.

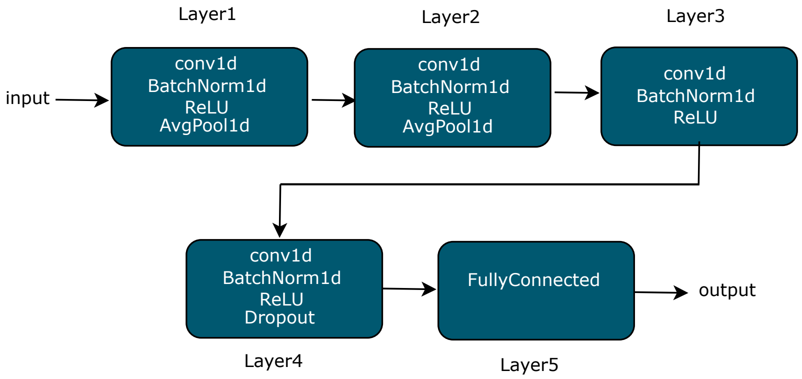
Layer1
Layer2
Layer3
input
conv1d BatchNorm1d ReLU AvgPool1d
conv1d BatchNorm1d ReLU AvgPool1d
conv1d BatchNorm1d ReLU
conv1d BatchNorm1d ReLU Dropout
FullyConnected
output
Layer4
Layer5

Figure 6.4: Architecture of CNN

The input size of the network is 1080 samples (3 s long signal). The noisy data is input to the model, and the denoised data is the output.

ECG Custom Loss Function

We design a custom loss function to prioritise overall signal improvement and preservation of important signal peaks. The loss function plays a critical role in training deep learning models. The Mean Square Error is commonly implemented as the loss function in signal denoising tasks. However, one problem that may occur when using it is that equal priority is given to areas of limited signal information.

[3] `https://github.com/Brophy-E/DenoiseEEG-GAN`

While removing noise throughout the signal is essential, it is important that signal features should not be mistaken for noise and smoothed. Hence, we utilised a more practical solution by designing a loss function with two MSE components; the first is the global MSE, which is the typical loss function, and the second is the MSE pertaining only to regions where the QRS wave features exist. In addition, a weighting parameter, α, is applied to the second MSE term, effectively enabling us to control the weighting given to important signal features, as opposed to 'flat' regions in the signal. Unlike previous studies, our loss function considers all QRS complexes in the signal; see equation (6.2). The average QRS complex range is 100 ms in healthy individuals; therefore, we select 100 ms on either side of the R-peak, which is easily identifiable by an automated routine based on an intense local maximum. This ensures we capture the regular QRS complexes and the complexes with clinically significant arrhythmias present. The value α is a hyperparameter that is used to determine the level of importance placed on the QRS complexes by the loss function relative to the ECG signal as a whole. The idea behind the loss function needing the location of QRS holds for training. When denoising the noisy test data, we use no labels, only the noisy ECG time series signal. So the intention here is that during training, the loss function allows the model parameters to learn to emphasise the QRS part of the signal.

$$L = MSE(y, x) + \alpha \times \sum_{i=1}^{n} MSE(R_{yi}, R_{xi}) \tag{6.2}$$

Where y represents the noisy signal, x is the clean ground truth signal, n is the total number of QRS complexes in the 3-second signal segment, R_{xi} is the i^{th} QRS complex in signal x.

6.3.6 EEG Denoising Model and Objective Function

The GAN model maps from a noisy time series to a denoised time series, and in this use case, we use it to learn the noise model of the artefact and denoise the EEG signal. We define the generators and discriminators of our GAN as follows. The generator is a two-layer stacked long short-term memory network with 50 hidden units in each layer and a fully connected layer at the output, see Figure 6.5 (left). The input size is 640 sample points for the EEG-50, 512 sample points for the EEG-EOG and 1024 sample points for the EEG-EMG datasets. The discriminator is a 4-layer 1-dimensional CNN with a fully connected layer and sigmoid activation function at the output, see Figure 6.5 (right).

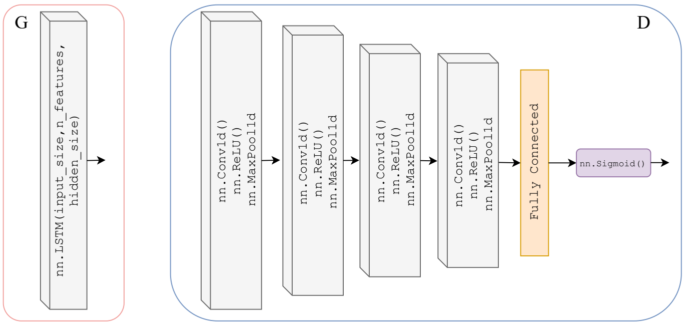

Figure 6.5: Detailed architecture of generator (left) which is a two-layer stacked LSTM with 50 hidden units in each layer and a fully connected layer at the output, with an input size varied to match the segment length for the chosen dataset. Architecture of discriminator (right) which is a 4-layer 1-dimensional CNN (ReLU activation and max pooling functions) with a fully connected layer and sigmoid activation function at the output.

EEG Objective Function

The loss function of our GAN framework is calculated as in equation (6.3) and (6.4). Here, a is the label for the generated samples, b is the label for the real samples, and c is the hyperparameter that G wants D to recognise the generated samples as real samples.

$$\min_{D} V_{LSGAN}(D) = 1/2 \times \mathbb{E}_{x \sim p_{data}(x)}[(D(\mathbf{x}) - b)^2] + 1/2 \times \mathbb{E}_{z \sim p_{\mathbf{z}}(z)}[(D(G(\mathbf{z})) - a)^2] \tag{6.3}$$

$$\min_{G} V_{LSGAN}(G) = 1/2 \times \mathbb{E}_{z \sim p_{\mathbf{z}}(z)}[(D(G(\mathbf{z})) - c)^2] \tag{6.4}$$

6.3.7 ECG Evaluation

To quantitatively evaluate our denoised data, we look at the SNR improvement in the ECG signal, heart rate error prediction, IBI and HRV of the denoised vs. noisy ECG signals. We also qualitatively evaluate our results through a visual inspection in both the time series domain of the ECG.

SNR is defined as the ratio of signal power to noise power, often expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise. The formula for SNR is given as in equation (6.5), again, where x is the ECG signal of interest, n is the artefact, and λ is the hyperparameter that controls the SNR.

$$SNR = 10 \times \log_{10}\left(\frac{RMS(x)}{RMS(\lambda \times n)}\right) \tag{6.5}$$

The Root Mean Square (RMS) of a signal is given in equation (6.6). N is defined as the number of samples in the ECG signal segment, and a_i denotes the i^{th} sample in the ECG signal. $N = 1080$.

$$RMS(x) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} a_i^2} \tag{6.6}$$

Interbeat interval (IBI)

Heart rate in physiological studies is mostly derived from measurements taken from the electrocardiogram. First, the number of R-waves per unit time, or the time between these waves (interbeat interval), is measured. This time can be translated to the rate of the heart for any collection of beats. Such detailed measurements permit how the heart reacts beat by beat to environmental and physiological stimuli. Unfortunately, while the interbeat interval is essential for clinical diagnosis, it is easily corrupted by noise. For the IBI, we calculate the location of the R-peaks using $scipy.signal.find\_peaks$ [214] and return the R-peak differences in milliseconds.

Heart rate variability (HRV)

Informative cardiac metrics rely not just on the heart rate but also on how the heart rate varies. Thus, another vital feature of measuring the cardiac state is heart rate variation. HRV is the temporal variation between consecutive heartbeats (RR intervals). A higher heart rate variability is associated with good health. On the other hand, a low HRV is associated with ill health — it becomes a significant predictor of mortality from several diseases. In this experiment, we will use the R peaks in calculating the HRV.

To reliably measure HRV and low-frequency cardiac components, long-term ECG records of at least 24 hours are necessary. However, short recordings can effectively capture the higher-frequency cardiac components. Recordings as short as 5 minutes are adequate for HRV [215]–[217]. Therefore, we compute HRV for the ECG signals over 1 hour broken down into 5 minute epochs. The HRV analysis is computed using the $neurokit2$ package [199].

6.3.8 EEG Evaluation

To quantitatively evaluate our denoised EEG data, we look at the SNR vs. relative root mean squared error (RRMSE), Pearson's correlation coefficient (CC) and the power ratios of the associated EEG bands across the signals. We also qualitatively evaluate our results through a visual inspection in both the time series domain and the frequency domain via the PSD of the EEG.

We use SNR to compare the level of the desired EEG signal to the level of noise/artefact present in the signal. The formula for SNR is given as in equation (6.5), again, where x is the EEG signal of interest, n is the artefact, and λ is the hyperparameter that controls the SNR.

The Root Mean Square (RMS) of a signal is given in equation (6.6). N is defined as the number of samples in the EEG signal segment a, and a_i denotes the i^{th} sample in the EEG signal. $N = 512$ and 1024 for the EOG and EMG signals respectively.

RRMSE is given in equation (6.7) for the temporal/time domain and in equation (6.8) for the frequency/spectral domain. $f(y)$ is the noisy signal passed through our model; in our case, it becomes the denoised signal generated by the GAN. We calculate the PSD using the FFT-length equal to the total length of the EEG input segment with a Hanning window.

$$RRMSE_{temporal} = \frac{RMS(f(y) - x)}{RMS(x)} \tag{6.7}$$

$$RRMSE_{spectral} = \frac{RMS(PSD(f(y)) - PSD(x))}{RMS(PSD(x))} \tag{6.8}$$

Pearson's correlation coefficient is shown in equation (6.9), where Cov is the covariance and Var is the variance of the signals $f(y)$ and x.

$$CC = \frac{Cov(f(y), x)}{\sqrt{Var(f(y))Var(x)}} \tag{6.9}$$

6.4 ECG Results

The following section details the denoising results of the electrode motion artefact on the ECG dataset at differing noise and α levels. Figures 6.6 to 6.8 illustrate the clean, ground-truth signal, the noisy ECG signal corrupted with electrode motion artefact and the denoised ECG signal. We show qualitative results for the same noisy input signal at 3 dB, 6 dB and 12 dB SNR levels. It becomes readily apparent that the model denoises the heavily corrupted ECG signals. We provide further results of the 6 dB ECG with differing noisy input signals in Appendix B, Section B.1.

We demonstrate our metrics for the 12, 6 and 3 dB noisy input signal at varying α levels to evaluate the model and custom loss function quantitatively. Firstly, we compare the SNR at the input to SNR at the output and compare the input and output signals' heart rate error (HRE). Again, as in Chapter 4, the HRE is defined as the absolute difference between the estimated heart rate for a given ECG sample and the heart rate calculated from its corresponding ground-truth ECG sample. The results for these metrics can be found in Table 6.1.

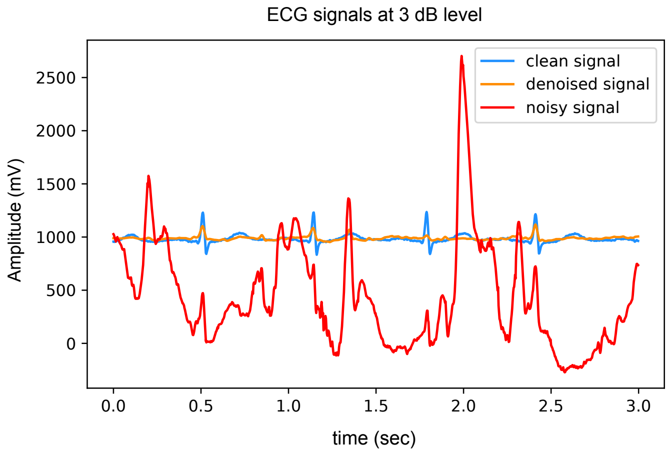

Figure 6.6: ECG signals before and after denoising with an artificial offset present in the noisy signal for visualisation purposes. The initial SNR of the noisy signal is 3 dB.

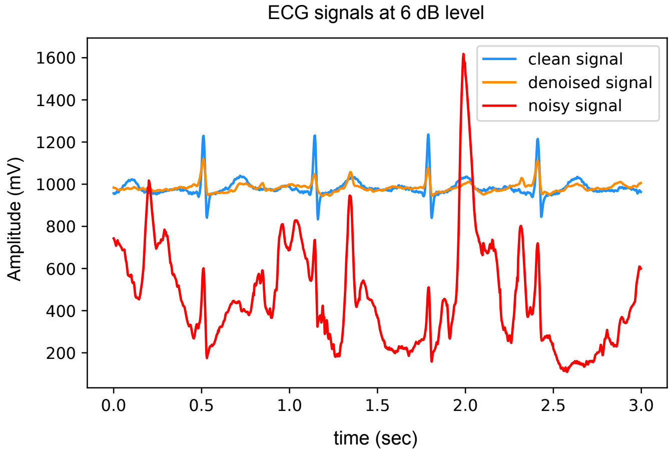

Figure 6.7: ECG signals before and after denoising with an artificial offset present in the noisy signal for visualisation purposes. The initial SNR of the noisy signal is 6 dB.

As can be seen in the SNR and HRE metrics in Table 6.1, the custom loss function outperforms the standard MSE loss function for medium-to-high noise levels. With increasing noise levels, the higher α values perform better. Furthermore, for the 6 dB signal, the custom loss function improves the SNR levels by 0.5 dB and the HRE by eight beats per minute or 25%. However, we consider this a good but not comprehensive evaluation of the models as the purpose of the custom loss function is to preserve the importance of QRS complexes. Therefore, we further analyse the IBI and HRV of the denoised ECG signals below as they are concerned with the R-peaks of the signals. The results of which can be found from Figures 6.9 to 6.11.

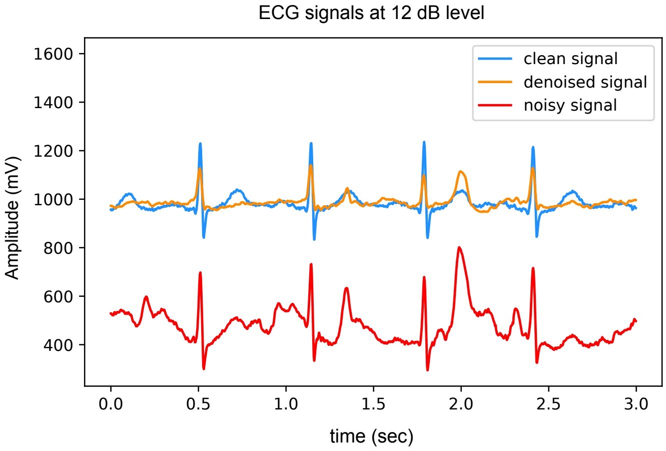

Figure 6.8: ECG signals before and after denoising with an artificial offset present in the noisy signal for visualisation purposes. The initial SNR of the noisy signal is 12 dB.

For the remainder of this section, we show results for the 6 dB signals at $\alpha = 2$. Figure 6.9 illustrates the R-R intervals in the ground-truth signal. Figure 6.10 illustrates the R-R intervals for the denoised signals and Figure 6.11 for the noisy signal. The denoised version shows an R-R variation similar to the clean ground-truth, whereas the noisy ECG signals lie far outside the ground-truth range. The HRV values for the ECG signals are as presented in Table 6.2. HRV Mean is the mean of the R-R intervals, and HRV SDNN is the standard deviation of the R-R intervals. From the denoised HRV, we can see that many outliers have been denoised. A clean, distinct HRV can provide considerable reference to practitioners for the diagnosis of patients. Overall, more accurate cardiac health can be deduced from the cleaner HRV/IBI information following the denoising of the ECG signals.

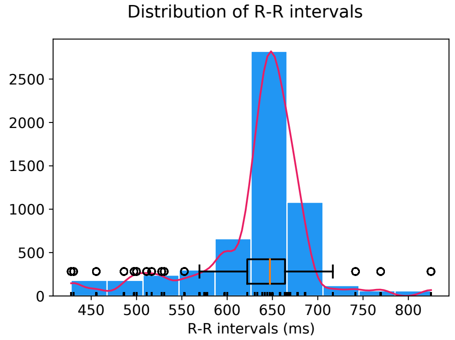

Figure 6.9: Distribution of R-R intervals in the clean (ground-truth) signal.

Table 6.1: SNR metrics of 12, 6 and 3 dB signals with custom loss function. $\alpha = 0$ indicates standard MSE loss function and std is the standard deviation of the HRE over the entire test set.

SNR Input	Alpha (α)	SNR Output	HRE (Noisy)	HRE (Denoised)	HRE-Denoised (std)
12 dB	0	**15.07 dB**	6	2.0	6.0
	2	13.47 dB	6	5.0	18.84
	20	14.83 dB	6	**1.0**	**4.35**
	50	13.87 dB	6	15.0	17.74
6 dB	0	14.6 dB	32	10.0	16.12
	2	**15.1 dB**	32	**2.0**	**6.0**
	20	14.7 dB	32	3.0	14.52
	50	14.5 dB	32	18.0	25.22
3 dB	0	13.80 dB	40	28.0	19.39
	2	14.06 dB	40	27.0	23.04
	20	14.09 dB	40	28.0	20.39
	50	**14.24 dB**	40	**19.0**	**14.79**

Table 6.2: HRV of the ECG signals for $\alpha = 2$.

	Ground-truth	Denoised	Noisy
HRV Mean	631 ms	645 ms	594 ms
HRV SDNN	62 ms	96 ms	195 ms

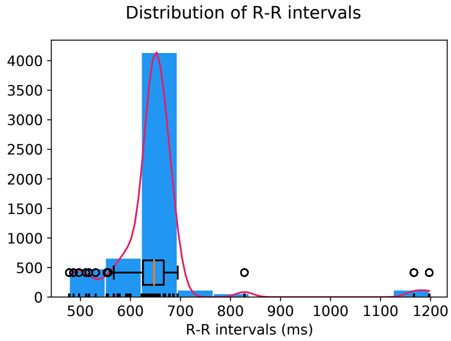

Figure 6.10: Distribution of R-R intervals in the denoised signal with some outliers present.

6.5 EEG Results

6.5.1 Benchmark EEGdenoiseNet Experiment

In this section, we showcase our model's performance on the EEGdenoiseNet dataset. We present both quantitative and qualitative evidence of our methods competitive

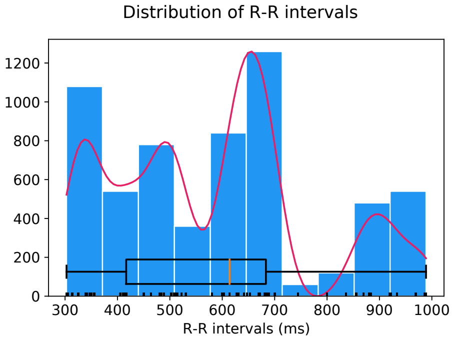

Figure 6.11: Distribution of R-R intervals in the noisy signal.

performance against the benchmark established in the original paper. A qualitative example of high-fidelity denoised EEG for our GAN model is presented in Figure 6.12. For visualisation purposes, an offset is artificially introduced to the ground truth and denoised EEG signals. Further examples of denoised EEG with the corresponding noisy EEG and ground truth can be found in Appendix A, Section A.1.

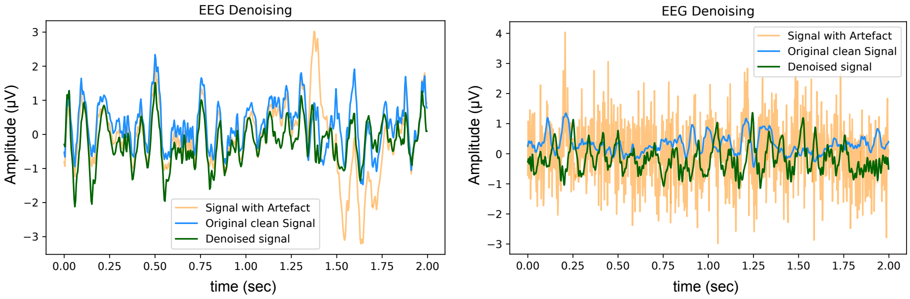

Figure 6.12: Example of denoised time series EEG corrupted with (left) EOG artefact and (right) EMG artefact. The signals contain an artificial offset for visualisation purposes.

In keeping with the benchmark evaluation metrics we present the $RRMSE_{temporal}$, $RRMSE_{spectral}$ and CC graphs at all our SNR levels. It should be noted that the performance of our model outperforms the other models in the benchmark experiment, we also provide results from deep learning models that have been implemented in the literature as a comparison to our GAN. For all SNR levels, our GAN performs extremely well, see Section 6.5.1 for further details. The graphs in Figure 6.13 correspond to the denoised EEG signal in Figure 6.12 (left). Similarly, the graphs in Figure 6.14 correspond to the denoised EEG signal in Figure 6.12 (right). For both EOG and EMG our model outperforms the benchmarks across $RRMSE_{temporal}$,

$RRMSE_{spectral}$ and CC. In general, the denoising capability of our model improves as the SNR improves. The CC for the EEG-EMG experiment does increase as the SNR improves, however, this is one of the few metrics that needs further experimentation on to improve.

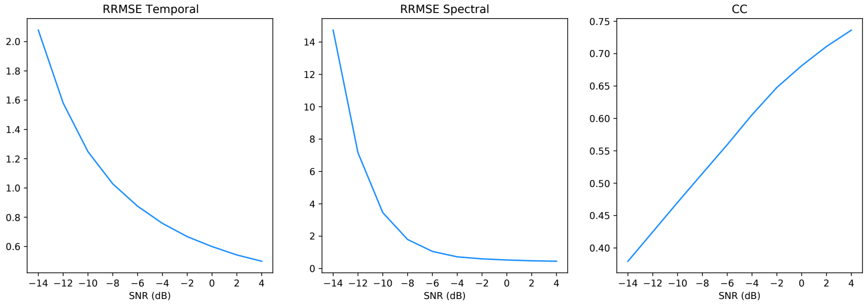

Figure 6.13: Metrics of the EEG-EOG signals shown in Figure 6.12 (left).

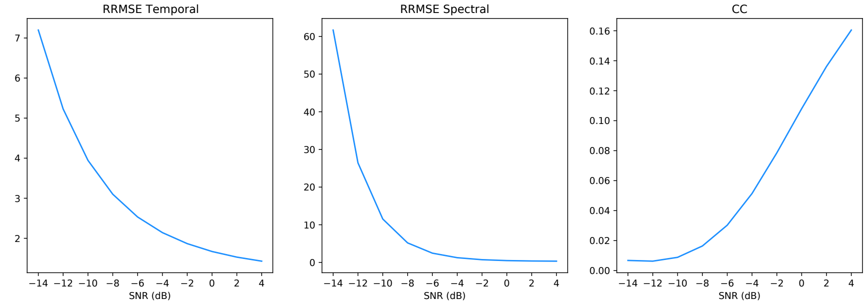

Figure 6.14: Metrics of the EEG-EMG signals shown in Figure 6.12 (right).

We present a final metric to evaluate our denoising model quantitatively, and it is the power present in the different EEG bands. Figure 6.15 and Table 6.3 are the corresponding PSD and power band ratios for the EEG signals shown in Figure 6.12 (left), respectively. It can be seen that the high power low-frequency components in the delta band are present in the EOG contaminated signal are removed from the denoised signal. We present results for the noisy EEG at -14 dB as this can be considered the worst-case scenario for the denoising GAN model. As can be observed, the model effectively removes the EOG artefacts in the contaminated data.

Likewise, Figure 6.16 and Table 6.4 are the PSD and power band ratios that corresponds to the EEG signals in Figure 6.12 (right). Again, it is apparent that the high-frequency noise in the beta and gamma bands present in the EMG contaminated

EEG is suppressed in the denoised signal. For both EOG and EMG datasets, the power across the denoised EEG frequency bands is recovered in the denoised signal.

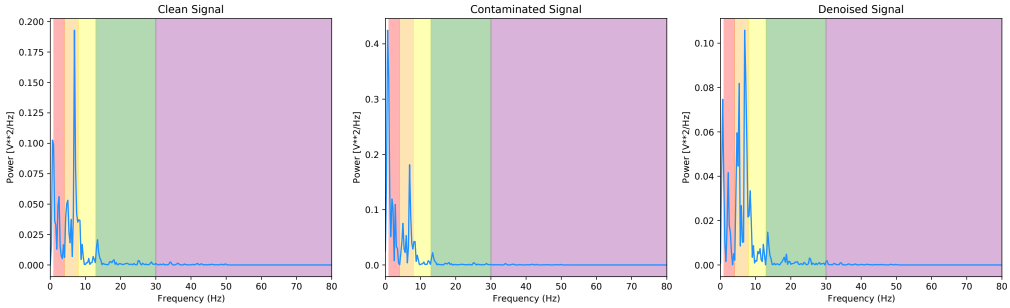

Figure 6.15: PSD of the EEG-EOG signals shown in Figure 6.12 (left) with corresponding EEG bands.

Table 6.3: Power ratios of different frequency bands before and after EOG artifact removal.

Denoising Method	Delta	Theta	Alpha	Beta	Gamma
GAN (-14 dB)	0.3020	0.4091	0.1647	0.1023	0.0217
Ground Truth	0.2769	0.4299	0.1349	0.1158	0.0424
Contaminated Signal (-14 dB)	0.7999	0.1280	0.0319	0.0289	0.0113

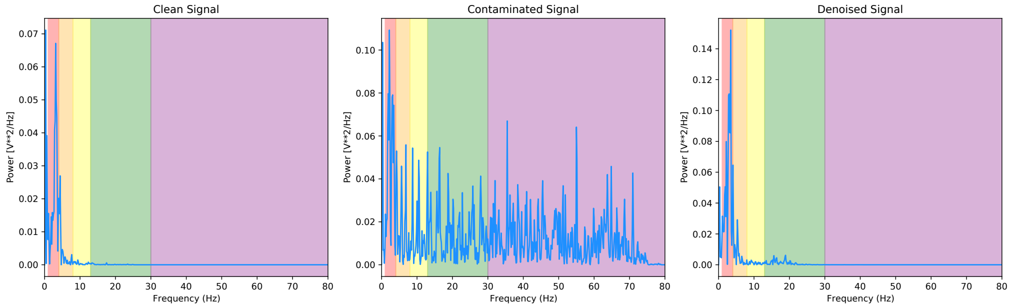

Figure 6.16: PSD of the EEG-EMG signals shown in Figure 6.12 (right) with corresponding EEG bands.

Table 6.4: Power ratios of different frequency bands before and after EMG artifact removal.

Denoising Method	Delta	Theta	Alpha	Beta	Gamma
GAN (-14 dB)	0.6528	0.2243	0.0299	0.0908	0.0023
Ground Truth	0.6458	0.2213	0.0658	0.0671	1.19e-10
Contaminated Signal (-14 dB)	0.0612	0.0471	0.0796	0.2981	0.5140

Benchmarking Deep Learning Models

It is important to frame our model in the context of other deep learning frameworks, and as such, we benchmark our model against neural networks that have been used successfully in the past to denoise EEG data. We implemented various CNNs, VAEs and Convolutional Autoencoders (CAEs) with the same training process as the GAN and 5-fold cross-validation. As GANs are not intuitive, the training and validation losses can be misleading. Still, it is worth monitoring the losses for convergence between the discriminator and generator as the original concept is a zero-sum game between the two NNs. Rather than observing a validation loss, it is better practice to quantitatively and qualitatively evaluate the data generated by the GAN.

We compare each NN models' denoised signals to the ground truth. The comparison we present is the ability of each model to preserve the power ratios across the various EEG frequency bands. We compute the cosine similarity of the power ratio across the frequency bands between the denoised and ground truth at the -14 dB level, as can be seen in Table 6.5.

Table 6.5: Cosine similarity score of the different frequency bands after artifact removal (to ground truth).

Denoising Model	EOG-Score	EMG-Score
GAN	**0.995**	**0.998**
SimpleCNN	0.985	0.9766
C-VAE	0.982	0.9916
CAE	0.819	0.9202
Novel-CNN	0.793	0.9914

Ranking the DL-models in terms of the EEG frequency preservation across bands shows that the GAN outperforms the other models.

To truly demonstrate the usefulness of deep learning models, it should be shown that the denoising method can improve downstream tasks. However, we cannot readily apply this to a classification task with this dataset. Instead, to demonstrate the effectiveness of the denoised data, we trained a classifier to distinguish between the original ground truth data and noisy data. Following this, we test the trained classifier on the ground truth vs. noisy data and then again on the denoised data vs.

noisy data. Finally, we compare the F1-score of both classifiers. The F1-score of the original ground truth data is 0.8987, with an accuracy of 88.75%. Whereas when using the denoised data, the F1-score reduces to 0.7799 and accuracy of 77.94%.

6.5.2 Toy Experiment

Here we present brief examples of the GAN's performance on the toy EEG-50 dataset. Further examples of the results from this experiment can be found in Appendix A, Section A.1. Examples of the denoised time series EEG signal can be seen in Figure 6.17.

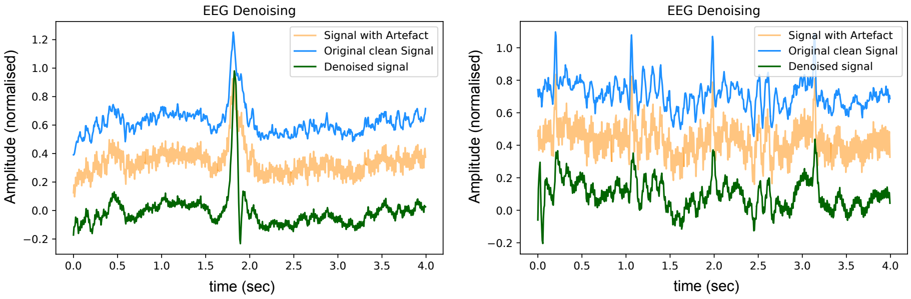

Figure 6.17: Example of denoised time series EEG corrupted with 50 Hz mains noise. The signals contain an artificial offset for visualisation purposes. Note the appearance of an ocular artefact in both examples.

Similar to the EEGdenoiseNet example, we demonstrate the performance of our model at removing the 50 Hz noise through the use of metrics. As this dataset was not divided into SNR levels we return one set of values for the metrics and they are as follows; $RRMSE_{temporal} = 0.05$, $RRMSE_{spectral} = 0.1$ and $CC = 0.89$. These metrics show that our model is more than capable of learning the noise model between our signal pairs.

Again, to quantitatively evaluate our denoising model, we illustrate the power present in the different EEG bands of our signals. Figure 6.18 and Table 6.6 are the corresponding PSD and power band ratios for the EEG signals shown in Figure 6.17 (left), respectively. It can be seen that the high power high-frequency components, centred around 50 Hz in the gamma band, is present in the contaminated signal and is heavily suppressed in the denoised signal. Once again, the model effectively reduces the mains noise artefacts in the contaminated data.

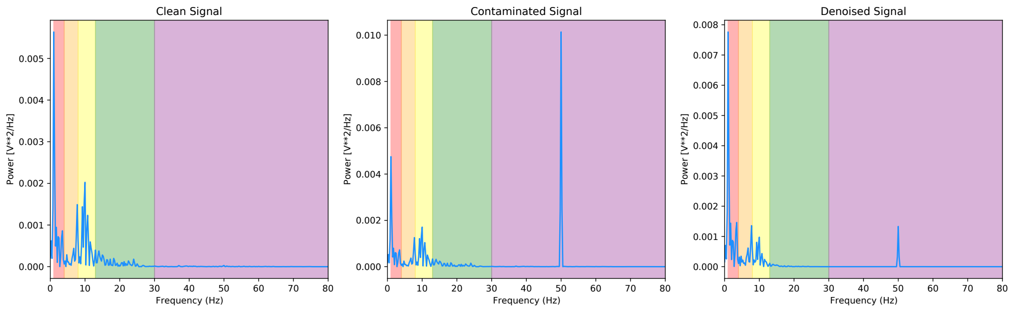

Figure 6.18: PSD of the EEG-50 signals shown in Figure 6.17 (right) with corresponding EEG bands.

Table 6.6: Power ratios of different frequency bands before and after 50 Hz noise removal.

Denoising Method	**Delta**	**Theta**	**Alpha**	**Beta**	**Gamma**
GAN	0.6528	0.2243	0.0299	0.0908	0.0023
Ground Truth	0.6046	0.2212	0.0658	0.0671	1.19e-10
Contaminated Signal	0.0612	0.0472	0.0796	0.2981	0.5140

6.6 Discussion

In this work, we have introduced novel deep learning frameworks capable of denoising and evaluating human physiological signal data.

For our ECG denoising, we have found that the proposed loss function outperforms the standard mean squared error loss function based on the experimental results. Specifically, the QRS complexes are better preserved by the custom loss function, leading to the preservation of important, relevant ECG information. This allows us to calculate IBI and HRV values more accurately as the QRS complexes now have more importance placed on them by the loss function.

Building on this, we also completed a short experiment to demonstrate that the proposed loss function further allows for less complex and faster models for denoising the ECG signals. Figures 6.19 and 6.20 presents qualitative evidence of the denoising capabilities of a simple CNN model without and with the custom loss function, respectively. The less computationally complex CNN model is made of 4 convolutional layers and one output layer. The original model presented in Figure 6.4 contains 2,322,180 trainable parameters, whereas the reduced complexity model discussed here contains only 1,562 trainable parameters. A reduction of almost 1,500 times the number of parameters, hence lower computational complexity is possible through the use of our proposed loss function.

It is clear that the reduced complexity CNN model using the standard MSE loss function does not converge nor learn any characteristics of the ECG. In contrast, the same model with the custom loss function learns to denoise the ECG adequately well and achieves better performance with the same volume of training data and training epochs.

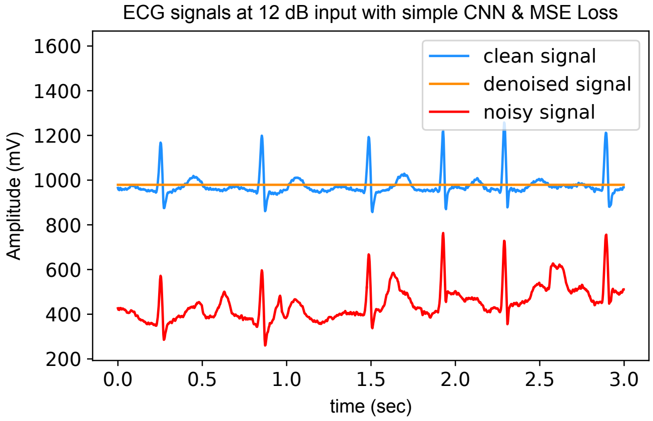

Figure 6.19: **Standard loss function** — ECG signals before and after denoising with a less-complex CNN and standard MSE loss. An artificial offset present in the noisy signal for visualisation purposes. The initial SNR of the noisy signal is 12 dB.

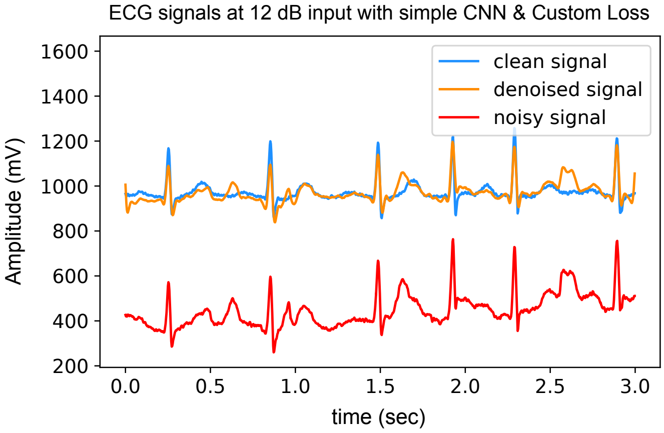

Figure 6.20: **Custom loss function** — ECG signals before and after denoising with a less-complex CNN and custom loss function. An artificial offset present in the noisy signal for visualisation purposes. The initial SNR of the noisy signal is 12 dB.

For our EEG denoising, we have presented our qualitative and quantitative analysis that demonstrated that our model outperforms the benchmarks on many of the metrics provided in the original paper [190]. Thus, we build on and contribute to the initial experiments and show that our model is currently state-of-the-art in this deep learning-based EEG denoising experiment.

To demonstrate the full capability of these models and that they have not overfitted to their respective datasets, we pass a signal with both 50 Hz noise and EOG artefact through our models. This signal is taken from the eegmidb dataset that has a natural ocular artefact introduced from the subject [198]. We then artificially introduce 50 Hz noise to the signal. This corrupted EEG signal is then denoised using the EEG-50 model, the output of which is then resampled and passed through the EEG-EOG model. Two examples of EEG signals at each stage of the denoising process are shown below in Figure 6.21.

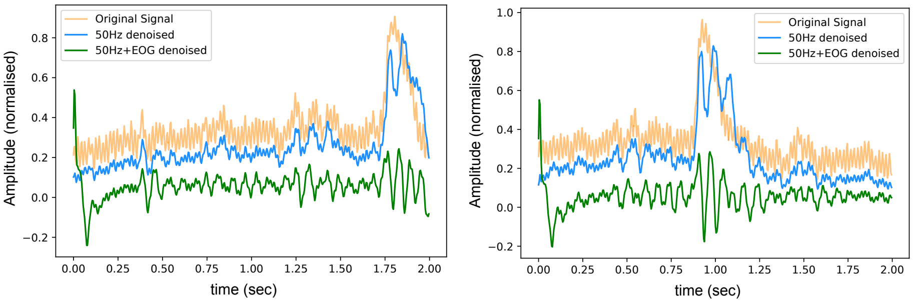

Figure 6.21: Denoised EEG signals following 50 Hz and EOG artefact removals.

6.7 Conclusion

This work proposes two DL frameworks for denoising human physiological signal data. Our first contribution is a convolution network with a novel loss function to more effectively preserve the QRS complex structure of ECG and improve the denoising of noisy ECG signals. The proposed model and custom loss function computes a weighted combination of global and local Mean Square Errors and improves the denoising performance of the ECG in terms of the SNR and heart rate. This demonstrates the capability of the algorithm to balance between denoising the signal and preserving the peaks effectively. Our second contribution is a novel GAN architecture that can effectively denoise single channel EEG data that is corrupted with a collection of various artefacts. This work is the first example of a GAN for denoising human physiological signals, and it demonstrates that GANs can be implemented as effective denoising architectures, making them ideal for portable physiological monitoring systems.

With high noise reduction and low signal distortion, the practicality and superiority of the proposed methods becomes evident and move DL architectures towards being more suitable for clinical prognosis. Observing our additional experiments, we show that our custom loss function can reduce the computational cost associated with CNNs. Furthermore, with the integration of AI systems into wearable technologies, our framework lays the foundation for continuous, portable and remote ECG and EEG monitoring devices with less noisy and more stable physiological signals. Using these methods to produce high fidelity and reliable physiological data may be a solution for clinicians to remotely and accurately monitor patients' heart and brain health states. This remote monitoring of patient state data will open the doors for improving treatment and diagnosis, contributing to the shift from reactive-based measures to proactive ones and further improving the lives of individuals experiencing issues with the current reactive, costly treatment model. Work such as this can assist in revolutionising this treatment paradigm.

Chapter 7

Conclusion

7.1 Summary

This thesis discussed both theoretical aspects and practical applications of machine learning and deep learning techniques for in the wild physiological signal processing. To prepare for our discussion, we presented the motivation and background knowledge in Chapter 1 from two related areas: (1) Human health and analytics, where we introduce the deep learning architectures that unlock further insights into human health and pathologies using previously unseen biomarkers and (2) Privacy-preserving machine learning methods to allow for rapid development and deployment of human-centric DL models.

Two research questions have been proposed related to the work presented in this thesis. The contents in Chapters 2 and 3 addressed Research Question 1: Can we provide a foundation for time series generation with Generative Adversarial Networks? Chapters 4 to 6 addressed Research Question 2: Can we successfully leverage novel deep learning-based models to process human physiological signals and return state-of-the-art insights from the data?

Our first contribution lies in forwarding the field in time series generation using recurrent Generative Adversarial Networks. In Chapter 3 we developed a custom architecture and loss function towards improved physiological data generation and, more significantly, provided continuous multichannel time series generation using

GANs for the first time. This advancement is important, and as stated previously, it means we can use this technology in further clinical training and research applications.

The second contribution put forward by our work explores novel deep learning technologies applied to continuous physiological time series data and unlocks further insights from these datasets. In Chapter 4 we used CNNs for human activity recognition and heart rate estimation using a sole PPG sensor that achieves state-of-the-art performance and competes with more energy-intense, complex multimodal devices. Using similar data in Chapter 5 we demonstrate that we can cycle time series modalities, inferring one physiological signal from another, which can give further understanding of the human physiological state. Finally, Chapter 6 presented our work on developing novel loss functions and deep learning architectures for denoising physiological data to a better standard than more classical methods in the literature. Once again, we developed novel applications of existing architectures and pioneering loss functions to improve model performance. Both approaches push the envelope for state-of-the-art deep learning methods for human physiological data processing.

7.2 Future Work

Improved Classification Metrics for Healthcare data

To complement the classical and novel metrics, we could enlist the help of a suitably qualified physician to assist with the validation and classification of generated data samples to determine how accurate the signal traces are, as we have done in our previous work [143] and briefly discussed in Chapter 3. This forms an avenue for our future work with time series GANs.

Sharing and Dissemination of Synthetic Healthcare data

We have shown in Chapter 3 that GANs are capable of generating synthetic human healthcare data that maintains differential privacy of the training dataset. Having a high-quality synthetic dataset that conforms to GDPR standards not only allows for easier dissemination of the data but will help in the upskilling of clinical professionals, particularly in less developed countries where access to this kind and quality of data can be problematic. Building a database of synthetic physiological data that contains important and clinically relevant biomarkers remains an area of future work.

Sustainable and Robust AI

In the interest of environmentally sustainable AI, we have presented methods in Chapter 4 and 5 that proposes low-power, low-cost wearables capable of successfully calculating human activity recognition, heart rate estimation and continuous blood pressure measurements, all from a single optical PPG sensor. Given the state-of-the-art performance of such methods, these devices hold the potential to become ubiquitous in the remote-sensing environment. As a result, these tools may take some of the burdens off the public healthcare systems.

As ML models in production are subject to concept drift, given the accumulation or streaming of new data, it is essential to keep models up-to-date and robust to changing times. Approaches such as CRISP-ML(Q) [218] developed state-of-the-art procedures for monitoring and maintenance of machine learning applications to mitigate the risk of model degradation. As we advance and AI-assisted devices become pervasive in our society, it is vital to establish a set of principles to ensure the quality and simplify the management process of ML models. These will allow us to automate the deployment of robust models in large-scale production environments.

References

[1] Central Statistics Office, *System of Health Accounts 2017*. [Online]. Available: https://www.cso.ie/en/releasesandpublications/er/sha/systemofhealthaccounts2017/ (visited on 11/26/2021).

[2] P. K. Health System Tracker, *How does health spending in the U.S. compare to other countries?* [Online]. Available: https://www.healthsystemtracker.org/chart-collection/health-spending-u-s-compare-countries (visited on 11/26/2021).

[3] Department of Health, "Statement of Strategy 2021-2023", 2021.

[4] S. A. Page, K. P. Manhas, and D. A. Muruve, "A survey of patient perspectives on the research use of health information and biospecimens", *BMC Medical Ethics*, vol. 17, no. 1, p. 48, 2016. DOI: 10.1186/s12910-016-0130-4.

[5] N. Menachemi and T. H. Collum, "Benefits and drawbacks of electronic health record systems", *Risk management and healthcare policy*, vol. 4, pp. 47–55, 2011, ISSN: 1179-1594 (Electronic); 1179-1594 (Linking). DOI: 10.2147/RMHP.S12985.

[6] M. Meingast, T. Roosta, and S. Sastry, "Security and Privacy Issues with Health Care Information Technology", in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 5453–5458. DOI: 10.1109/IEMBS.2006.260060.

[7] A. M. Arellano, W. Dai, S. Wang, X. Jiang, and L. Ohno-Machado, "Privacy Policy and Technology in Biomedical Data Science", *Annual Review of Biomed-*

ical Data Science, vol. 1, no. 1, pp. 115–129, 2018. DOI: 10.1146/annurev-biodatasci-080917-013416. eprint: https://doi.org/10.1146/annurev-biodatasci-080917-013416.

[8] HIPAA Journal, *HIPAA Privacy Laws*. [Online]. Available: https://www.hipaajournal.com/hipaa-privacy-laws/ (visited on 08/16/2022).

[9] Ropes & Gray LLP, *At a glance: data protection and management of health data in China*. [Online]. Available: https://www.lexology.com/library/detail.aspx?g=fd2bb402-33d5-4ba7-85a7-c5383cb11526 (visited on 08/16/2022).

[10] European Union, *Data Protection Act 2018 (Section36(2))*, http://www.irishstatutebook.ie/eli/2018/si/314/made/en/pdf, 2018.

[11] L. Rocher, J. M. Hendrickx, and Y.-A. de Montjoye, "Estimating the success of re-identifications in incomplete datasets using generative models", *Nature Communications*, vol. 10, no. 1, p. 3069, 2019. DOI: 10.1038/s41467-019-10933-3.

[12] A. Fratini, M. Sansone, P. Bifulco, and M. Cesarelli, "Individual identification via electrocardiogram analysis", *BioMedical Engineering OnLine*, vol. 14, no. 1, p. 78, 2015. DOI: 10.1186/s12938-015-0072-y.

[13] H. Kim, E. Bell, J. Kim, A. Sitapati, J. Ramsdell, C. Farcas, D. Friedman, S. F. Feupe, and L. Ohno-Machado, "iCONCUR: informed consent for clinical data and bio-sample use for research", *American Medical Informatics Association*, vol. 24, no. 2, pp. 380–387, Mar. 2017.

[14] M. A. Cauchy, "Méthode générale pour la résolution des systémes d'équations simultanées", *Comptes Rendus des séances de l'académie des sciences*, vol. 10, no. 383, pp. 536–538, 1847. [Online]. Available: https://cs.uwaterloo.ca/~y328yu/classics/cauchy-en.pdf.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors”, *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: 10.1038/323533a0. [Online]. Available: https://doi.org/10.1038/323533a0.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[17] S. Raschka and V. Mirjalili, *Python Machine Learning.* Packt Publishing, 2017, p. 622, ISBN: 9780874216561.

[18] E. W. Weisstein, *Convolution*, https://mathworld.wolfram.com/Convolution.html.

[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition”, *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. DOI: 10.1162/neco.1989.1.4.541.

[20] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks”, *Science*, vol. 313, no. 5786, pp. 504–507, 2006. DOI: 10.1126/science.1127647.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets”, in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, Montréal, Canada: Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[22] A. Borji, “Pros and Cons of GAN Evaluation Measures”, *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019, ISSN: 1090235X. DOI: 10.1016/j.cviu.2018.10.009. arXiv: arXiv:1802.03446v5.

[23] G. Dorffner, “Neural Networks for Time Series Processing”, *Neural Network World*, vol. 6, pp. 447–468, 1996.

[24] Z. Wang, Q. She, and T. E. Ward, "Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy", *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[25] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, *A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications*, 2020. arXiv: `2001.06937 [cs.LG]`.

[26] L. Gonog and Y. Zhou, "A Review: Generative Adversarial Networks", in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Xi'an, China: IEEE, Jun. 2019, pp. 505–510, ISBN: 978-1-5386-9490-9. DOI: `10.1109/ICIEA.2019.8833686`. [Online]. Available: `https://ieeexplore.ieee.org/document/8833686/` (visited on 01/06/2021).

[27] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of Generative Adversarial Networks (GANs): An Updated Review", *Archives of Computational Methods in Engineering*, vol. 28, pp. 525–552, Dec. 2019, ISSN: 1134-3060, 1886-1784. DOI: `10.1007/s11831-019-09388-y`. [Online]. Available: `http://link.springer.com/10.1007/s11831-019-09388-y` (visited on 10/05/2020).

[28] C. Yinka-Banjo and O.-A. Ugot, "A review of generative adversarial networks and its application in cybersecurity", *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1721–1736, Mar. 2020, ISSN: 0269-2821, 1573-7462. DOI: `10.1007/s10462-019-09717-4`. [Online]. Available: `http://link.springer.com/10.1007/s10462-019-09717-4` (visited on 01/06/2021).

[29] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review", *Medical Image Analysis*, vol. 58, p. 101 552, 2019, ISSN: 1361-8415. DOI: `https://doi.org/10.1016/j.media.2019.101552`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1361841518308430`.

[30] A. Radford, L. Metz, and S. Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, 2015. arXiv: 1511.06434. [Online]. Available: http://arxiv.org/abs/1511.06434.

[31] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein GAN*, 2017. DOI: 10.2507/daaam.scibook.2010.27. arXiv: 1701.07875. [Online]. Available: http://arxiv.org/abs/1701.07875.

[32] A. Borji, *Pros and Cons of GAN Evaluation Measures: New Developments*, 2021. arXiv: 2103.09396 [cs.LG].

[33] Z. Wang, Q. She, A. F. Smeaton, T. E. Ward, and G. Healy, "Synthetic-Neuroscore: Using A Neuro-AI Interface for Evaluating Generative Adversarial Networks", *Neurocomputing*, vol. 405, pp. 26–36, 2020.

[34] Z. Wang, G. Healy, A. F. Smeaton, and T. E. Ward, "Use of Neural Signals to Evaluate the Quality of Generative Adversarial Network Performance in Facial Image Generation", *Cognitive Computation*, vol. 12, no. 1, pp. 13–24, 2020.

[35] D. J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton, *Generative Models and Model Criticism via Optimized Maximum Mean Discrepancy*, 2016. arXiv: 1611.04488. [Online]. Available: http://arxiv.org/abs/1611.04488.

[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved Techniques for Training GANs", in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Barcelona, Spain: Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*, 2018. arXiv: 1706.08500 [cs.LG].

[38] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html.

[39] F. B. Bryant and P. R. Yarnold, "Principal-components analysis and exploratory and confirmatory factor analysis", in *Reading and understanding multivariate statistics.* Washington, DC, US: American Psychological Association, 1995, pp. 99–136, ISBN: 1-55798-273-2 (Paperback).

[40] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series Generative Adversarial Networks", in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Vancouver, Canada: Curran Associates, Inc., 2019, pp. 5508–5518. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

[41] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, *Generating Multi-label Discrete Patient Records using Generative Adversarial Networks*, 2017. arXiv: 1703.06490. [Online]. Available: http://arxiv.org/abs/1703.06490.

[42] C. Esteban, S. L. Hyland, and G. Rätsch, *Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs*, 2017. DOI: 10.1002/fut. arXiv: 1706.02633. [Online]. Available: http://arxiv.org/abs/1706.02633.

[43] K. El Emam, E. Jonker, L. Arbuckle, and B. Malin, "A Systematic Review of Re-Identification Attacks on Health Data", *PLoS ONE*, vol. 6, 2011, ISSN: 19326203. DOI: 10.1371/journal.pone.0028071.

[44] B. Malin and L. Sweeney, "Re-identification of DNA through an automated linkage process", *Proceedings. AMIA Symposium*, vol. 2001, pp. 423–427, 2001, Publisher: American Medical Informatics Association, ISSN: 1531-605X. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/11825223.

[45] C. Culnane, B. I. P. Rubinstein, and V. Teague, *Health Data in an Open World*, 2017. arXiv: 1712.05627. [Online]. Available: http://arxiv.org/abs/1712.05627.

[46] B. P. Hejblum, G. M. Weber, K. P. Liao, N. P. Palmer, S. Churchill, N. A. Shadick, P. Szolovits, S. N. Murphy, I. S. Kohane, and T. Cai, "Probabilistic record linkage of de-identified research datasets with discrepancies using diagnosis codes", *Scientific Data*, vol. 6, 2019, ISSN: 20524463. DOI: 10.1038/sdata.2018.298.

[47] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images", University of Toronto, Toronto, Ontario, Tech. Rep. 0, 2009.

[48] Y. LeCun and C. Cortes, *MNIST handwritten digit database*, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA: IEEE, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[50] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML, *The UCR Time Series Classification Archive*, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/, Oct. 2018.

[51] D. Dua and C. Graff, *UCI Machine Learning Repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[52] P. Colarusso, L. H. Kidder, I. W. Levin, and E. Neil Lewis, "Raman and Infrared Microspectroscopy", in *Encyclopedia of Spectroscopy and Spectrometry*, J. C. Lindon, Ed., Elsevier, 1999, pp. 1945–1954. DOI: https://doi.org/10.1006/rwsp.2000.0402.

[53] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient", in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17, San Francisco, California, USA: AAAI Press, 2017, pp. 2852–2858.

[54] R. D. Hjelm, A. P. Jacob, T. Che, A. Trischler, K. Cho, and Y. Bengio, *Boundary-Seeking Generative Adversarial Networks*, 2018. arXiv: `1702.08431 [stat.ML]`.

[55] I. Goodfellow, *Generative Adversarial Networks for Text*, 2016. [Online]. Available: `https://www.reddit.com/r/MachineLearning/comments/40ldq6/generative_adversarial_networks_for_text/`.

[56] J. T. Guibas, T. S. Virdi, and P. S. Li, *Synthetic Medical Images from Dual Generative Adversarial Networks*, 2017. arXiv: `1709.01872`. [Online]. Available: `http://arxiv.org/abs/1709.01872`.

[57] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 105–114, 2017. DOI: `10.1109/CVPR.2017.19`.

[58] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative Adversarial Text to Image Synthesis", *33rd International Conference on Machine Learning, ICML 2016*, vol. 3, pp. 1681–1690, 2016. arXiv: `arXiv:1605.05396v2`.

[59] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, *Generating images with recurrent adversarial networks*, 2016. arXiv: `1602.05110`. [Online]. Available: `http://arxiv.org/abs/1602.05110`.

[60] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, ISSN: 0899-7667. DOI: `10.1162/neco.1997.9.8.1735`. arXiv: `1206.2944`. [Online]. Available: `http://`

www7.informatik.tu-muenchen.de/%7B~%7Dhochreit%7B%5C%%7D0Ahttp://www.idsia.ch/%7B~%7Djuergen.

[61] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, *Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks*, 2015. arXiv: 1506.03099 [cs.LG].

[62] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a Method for Automatic Evaluation of Machine Translation”, in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. [Online]. Available: https://www.aclweb.org/anthology/P02-1040.

[63] M. Lapata, *EMNLP14*, http://homepages.inf.ed.ac.uk/mlap/Data/EMNLP14/, Accessed: 2021-04-30, 2015.

[64] S. Winiger, *Obama Political Speech generator - Recurrent Neural Network*, https://github.com/samim23/obama-rnn, Accessed: 2021-04-30, 2015.

[65] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, “Quant GANs: Deep Generation of Financial Time Series”, *Quantitative Finance*, vol. 20, no. 9, pp. 1419–1440, Sep. 2020, arXiv: 1907.06673, ISSN: 1469-7688, 1469-7696. DOI: 10.1080/14697688.2020.1730426. [Online]. Available: http://arxiv.org/abs/1907.06673 (visited on 01/08/2021).

[66] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity”, *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:econom:v:31:y:1986:i:3:p:307-327.

[67] O. Mogren, *C-RNN-GAN: Continuous recurrent neural networks with adversarial training*, 2016. arXiv: 1611.09904 [cs.AI].

[68] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eICU Collaborative Research Database, a freely available multi-center database for critical care research", *Scientific Data*, vol. 5, no. 1, p. 180 178, Sep. 2018, ISSN: 2052-4463. DOI: `10.1038/sdata.2018.178`. [Online]. Available: `https://doi.org/10.1038/sdata.2018.178`.

[69] L. Wang, W. Zhang, and X. He, "Continuous Patient-Centric Sequence Generation via Sequentially Coupled Adversarial Learning", in *Database Systems for Advanced Applications*, G. Li, J. Yang, J. Gama, J. Natwichai, and Y. Tong, Eds., Cham: Springer International Publishing, 2019, pp. 36–52, ISBN: 978-3-030-18579-4.

[70] A. E. Johnson, T. J. Pollard, L. Shen, L. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database", *Scientific data*, vol. 3, p. 160 035, 2016.

[71] Y. Sumiya, K. Horie, H. Shiokawa, and H. Kitagawa, "NR-GAN: Noise Reduction GAN for Mice Electroencephalogram Signals", in *Proceedings of the 2019 4th International Conference on Biomedical Imaging, Signal Processing*, Nagoya Japan: ACM, Oct. 2019, pp. 94–101, ISBN: 978-1-4503-7295-4. DOI: `10.1145/3366174.3366186`. [Online]. Available: `http://dl.acm.org/doi/10.1145/3366174.3366186` (visited on 01/08/2021).

[72] H. Ni, L. Szpruch, M. Wiese, S. Liao, and B. Xiao, *Conditional Sig-Wasserstein GANs for Time Series Generation*, 2020. arXiv: `2006.05421 [cs.LG]`.

[73] Y. Li, K. Swersky, and R. Zemel, *Generative Moment Matching Networks*, 2015. arXiv: `1502.02761 [cs.LG]`.

[74] Oxford-Man Institute, *Oxford-Man Institute of Quantitative Finance Realized Library*, `https://realized.oxford-man.ox.ac.uk`, Accessed: 2021-04-30, 2021.

[75] H. Sun, Z. Deng, H. Chen, and D. C. Parkes, *Decision-Aware Conditional GANs for Time Series Data*, 2020. arXiv: `2009.12682 [cs.LG]`.

[76] D. Hazra and Y.-C. Byun, "SynSigGAN: Generative Adversarial Networks for Synthetic Biomedical Signal Generation", *Biology*, vol. 9, no. 12, p. 441, Dec. 2020, ISSN: 2079-7737. DOI: 10.3390/biology9120441. [Online]. Available: https://www.mdpi.com/2079-7737/9/12/441 (visited on 01/06/2021).

[77] G. Moody and R. Mark, "The impact of the MIT-BIH Arrhythmia Database", *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001. DOI: 10.1109/51.932724.

[78] P. Detti, G. Vatti, and G. Zabalo Manrique de Lara, "EEG Synchronization Analysis for Seizure Prediction: A Study on Data of Noninvasive Recordings", *Processes*, vol. 8, no. 7, 2020, ISSN: 2227-9717. DOI: 10.3390/pr8070846. [Online]. Available: https://www.mdpi.com/2227-9717/8/7/846.

[79] M. A. F. Pimentel, A. E. W. Johnson, P. H. Charlton, D. Birrenkott, P. J. Watkinson, L. Tarassenko, and D. A. Clifton, "Toward a Robust Estimation of Respiratory Rate From Pulse Oximeters", *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, pp. 1914–1923, 2017. DOI: 10.1109/TBME.2016.2613124.

[80] H. Fei and F. Tan, "Bidirectional Grid Long Short-Term Memory (BiGridLSTM): A Method to Address Context-Sensitivity and Vanishing Gradient", *Algorithms*, vol. 11, no. 11, 2018, ISSN: 1999-4893. DOI: 10.3390/a11110172. [Online]. Available: https://www.mdpi.com/1999-4893/11/11/172.

[81] K. Nikolaidis, S. Kristiansen, V. Goebel, T. Plagemann, K. Liestøl, and M. Kankanhalli, *Augmenting Physiological Time Series Data: A Case Study for Sleep Apnea Detection*, 2019. arXiv: 1905.09068 [cs.LG].

[82] S. M. Abdelfattah, G. M. Abdelrahman, and M. Wang, "Augmenting The Size of EEG datasets Using Generative Adversarial Networks", in *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro: IEEE, Jul. 2018, pp. 1–6, ISBN: 978-1-5090-6014-6. DOI: 10.1109/IJCNN.2018.

8489727. [Online]. Available: https://ieeexplore.ieee.org/document/8489727/ (visited on 01/08/2021).

[83] S. Harada, H. Hayashi, and S. Uchida, "Biosignal Data Augmentation Based on Generative Adversarial Networks", in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Honolulu, HI: IEEE, Jul. 2018, pp. 368–371, ISBN: 978-1-5386-3646-6. DOI: 10.1109/EMBC.2018.8512396. [Online]. Available: https://ieeexplore.ieee.org/document/8512396/ (visited on 01/07/2021).

[84] D. Kiyasseh, G. A. Tadesse, L. N. T. Nhan, L. Van Tan, L. Thwaites, T. Zhu, and D. Clifton, "PlethAugment: GAN-Based PPG Augmentation for Medical Diagnosis in Low-Resource Settings", *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 11, pp. 3226–3235, Nov. 2020, ISSN: 2168-2194, 2168-2208. DOI: 10.1109/JBHI.2020.2979608. [Online]. Available: https://ieeexplore.ieee.org/document/9078801/ (visited on 01/07/2021).

[85] S. J. Pan and Q. Yang, "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.

[86] Z. Zhang, J. Han, K. Qian, C. Janott, Y. Guo, and B. Schuller, *Snore-GANs: Improving Automatic Snore Sound Classification with Synthesized Data*, 2019. arXiv: 1903.12422 [cs.LG].

[87] A. M. Delaney, E. Brophy, and T. E. Ward, *Synthesis of Realistic ECG using Generative Adversarial Networks*, 2019. arXiv: 1909.09150. [Online]. Available: https://arxiv.org/abs/1909.09150.

[88] F. Zhu, F. Ye, Y. Fu, Q. Liu, and B. Shen, "Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network", *Scientific Reports*, vol. 9, no. 1, 2019, ISSN: 20452322. DOI: 10.1038/s41598-019-42516-z.

[89] E. Brophy, Z. Wang, and T. E. Ward, *Quick and Easy Time Series Generation with Established Image-based GANs*, 2019. arXiv: 1902.05624 [cs.LG].

[90] E. Brophy, “Synthesis of Dependent Multichannel ECG Using Generative Adversarial Networks”, in *Proceedings of the 29th ACM International Conference on Information &; Knowledge Management*, ser. CIKM ’20, Virtual Event, Ireland: Association for Computing Machinery, 2020, pp. 3229–3232, ISBN: 9781450368599. DOI: 10.1145/3340531.3418509. [Online]. Available: https://doi.org/10.1145/3340531.3418509.

[91] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, “Waveform Generation for Text-to-speech Synthesis Using Pitch-synchronous Multi-scale Generative Adversarial Networks”, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom: IEEE, May 2019, pp. 6915–6919, ISBN: 978-1-4799-8131-1. DOI: 10.1109/ICASSP.2019.8683271. [Online]. Available: https://ieeexplore.ieee.org/document/8683271/ (visited on 01/08/2021).

[92] Q. Li, H. Hao, Y. Zhao, Q. Geng, G. Liu, Y. Zhang, and F. Yu, “GANs-LSTM Model for Soil Temperature Estimation From Meteorological: A New Approach”, *IEEE Access*, vol. 8, pp. 59 427–59 443, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2982996. [Online]. Available: https://ieeexplore.ieee.org/document/9045947/ (visited on 01/08/2021).

[93] S. Kaushik, A. Choudhury, S. Natarajan, L. A. Pickett, and V. Dutt, “Medicine Expenditure Prediction via a Variance-Based Generative Adversarial Network”, *IEEE Access*, vol. 8, pp. 110 947–110 958, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3002346. [Online]. Available: https://ieeexplore.ieee.org/document/9116991/ (visited on 01/08/2021).

[94] Z. Guo, Y. Wan, and H. Ye, “A data imputation method for multivariate time series based on generative adversarial network”, *Neurocomputing*, vol. 360, pp. 185–197, Sep. 2019, ISSN: 09252312. DOI: 10.1016/j.neucom.2019.06.007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231219308306 (visited on 01/08/2021).

[95] F. Luer, D. Mautz, and C. Bohm, “Anomaly Detection in Time Series using Generative Adversarial Networks”, in *2019 International Conference on Data Mining Workshops (ICDMW)*, Beijing, China: IEEE, Nov. 2019, pp. 1047–1048, ISBN: 978-1-72814-896-0. DOI: 10.1109/ICDMW.2019.00152. [Online]. Available: https://ieeexplore.ieee.org/document/8955499/ (visited on 01/07/2021).

[96] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, “MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks”, in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, I. V. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds., vol. 11730, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 703–716, ISBN: 978-3-030-30490-4. DOI: 10.1007/978-3-030-30490-4_56. [Online]. Available: http://link.springer.com/10.1007/978-3-030-30490-4_56 (visited on 01/08/2021).

[97] T. Leangarun, P. Tangamchit, and S. Thajchayapong, “Stock Price Manipulation Detection using Generative Adversarial Networks”, in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Bangalore, India: IEEE, Nov. 2018, pp. 2104–2111, ISBN: 978-1-5386-9276-9. DOI: 10.1109/SSCI.2018.8628777. [Online]. Available: https://ieeexplore.ieee.org/document/8628777/ (visited on 01/08/2021).

[98] C. Donahue, J. McAuley, and M. Puckette, *Adversarial Audio Synthesis*, 2019. arXiv: 1802.04208 [cs.SD].

[99] A. Kolokolova, M. Billard, R. Bishop, M. Elsisy, Z. Northcott, L. Graves, V. Nagisetty, and H. Patey, *GANs & Reels: Creating Irish Music using a Generative Adversarial Network*, 2020. arXiv: 2010.15772 [cs.SD].

[100] P.-S. Cheng, C.-Y. Lai, C.-C. Chang, S.-F. Chiou, and Y.-C. Yang, “A Variant Model of TGAN for Music Generation”, in *Proceedings of the 2020 Asia Service*

Sciences and Software Engineering Conference, Nagoya Japan: ACM, May 2020, pp. 40–45, ISBN: 978-1-4503-7710-2. DOI: 10.1145/3399871.3399888. [Online]. Available: https://dl.acm.org/doi/10.1145/3399871.3399888 (visited on 01/08/2021).

[101] Y. Choi, H. Lim, H. Choi, and I.-J. Kim, “GAN-Based Anomaly Detection and Localization of Multivariate Time Series Data for Power Plant”, in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Busan, Korea (South): IEEE, Feb. 2020, pp. 71–74, ISBN: 978-1-72816-034-4. DOI: 10.1109/BigComp48618.2020.00-97. [Online]. Available: https://ieeexplore.ieee.org/document/9070362/ (visited on 01/16/2021).

[102] D. Parthasarathy, K. Bäckström, J. Henriksson, and S. Einarsdóttir, *Controlled time series generation for automotive software-in-the-loop testing using GANs*, 2020. arXiv: 2002.06611 [cs.LG].

[103] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A Kernel Two-Sample Test”, *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 723–773, Mar. 2012, ISSN: 1532-4435.

[104] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, *EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals*, 2018. arXiv: 1806.01875. [Online]. Available: http://arxiv.org/abs/1806.01875.

[105] F. Fahimi, Z. Zhang, W. B. Goh, K. K. Ang, and C. Guan, “Towards EEG Generation Using GANs for BCI Applications”, in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, Chicago, IL, USA: IEEE, May 2019, pp. 1–4, ISBN: 978-1-72810-848-3. DOI: 10.1109/BHI.2019.8834503. [Online]. Available: https://ieeexplore.ieee.org/document/8834503/ (visited on 01/07/2021).

[106] D. Pascual, A. Amirshahi, A. Aminifar, D. Atienza, P. Ryvlin, and R. Wattenhofer, “EpilepsyGAN: Synthetic Epileptic Brain Activities with Privacy

Preservation", *IEEE Transactions on Biomedical Engineering*, vol. 67, pp. 1–1, 2020. DOI: 10.1109/TBME.2020.3042574.

[107] L. Yi and M. Mak, "Adversarial Data Augmentation Network for Speech Emotion Recognition", in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Lanzhou, China: IEEE, 2019, pp. 529–534. DOI: 10.1109/APSIPAASC47483.2019.9023347.

[108] H. Zhang, N. Xiao, P. Liu, Z. Wang, and R. Tang, "G-RNN-GAN for Singing Voice Separation", in *Proceedings of the 2020 5th International Conference on Multimedia Systems and Signal Processing*, Chengdu China: ACM, May 2020, pp. 69–73, ISBN: 978-1-4503-7748-5. DOI: 10.1145/3404716.3404718. [Online]. Available: https://dl.acm.org/doi/10.1145/3404716.3404718 (visited on 01/08/2021).

[109] F. Qu, J. Liu, Y. Ma, D. Zang, and M. Fu, "A novel wind turbine data imputation method with multiple optimizations based on GANs", *Mechanical Systems and Signal Processing*, vol. 139, p. 106 610, May 2020, ISSN: 08883270. DOI: 10.1016/j.ymssp.2019.106610. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0888327019308313 (visited on 01/08/2021).

[110] L. Han, K. Zheng, L. Zhao, X. Wang, and H. Wen, "Content-Aware Traffic Data Completion in ITS Based on Generative Adversarial Nets", *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 950–11 962, Oct. 2020, ISSN: 0018-9545, 1939-9359. DOI: 10.1109/TVT.2020.3007025. [Online]. Available: https://ieeexplore.ieee.org/document/9133309/ (visited on 01/08/2021).

[111] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E^{2}GAN: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation", in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China: International Joint Conferences on Artificial Intelli-

gence Organization, Aug. 2019, pp. 3094–3100, ISBN: 978-0-9992411-4-1. DOI: `10.24963/ijcai.2019/429`. [Online]. Available: `https://www.ijcai.org/proceedings/2019/429` (visited on 01/08/2021).

[112] Y. Sun, L. Peng, H. Li, and M. Sun, “Exploration on Spatiotemporal Data Repairing of Parking Lots Based on Recurrent GANs”, in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI: IEEE, Nov. 2018, pp. 467–472, ISBN: 978-1-72810-323-5. DOI: `10.1109/ITSC.2018.8569319`. [Online]. Available: `https://ieeexplore.ieee.org/document/8569319/` (visited on 01/08/2021).

[113] Y. Chen, Y. Lv, and F.-Y. Wang, “Traffic Flow Imputation Using Parallel Data and Generative Adversarial Networks”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020, ISSN: 1524-9050, 1558-0016. DOI: `10.1109/TITS.2019.2910295`. [Online]. Available: `https://ieeexplore.ieee.org/document/8699108/` (visited on 01/08/2021).

[114] Y. Luo, X. Cai, Y. ZHANG, J. Xu, and Y. xiaojie, “Multivariate Time Series Imputation with Generative Adversarial Networks”, in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Montréal, Canada: Curran Associates, Inc., 2018, pp. 1596–1607.

[115] G. Zhu, H. Zhao, H. Liu, and H. Sun, “A Novel LSTM-GAN Algorithm for Time Series Anomaly Detection”, in *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*, Qingdao, China: IEEE, Oct. 2019, pp. 1–6, ISBN: 978-1-72810-861-2. (visited on 01/07/2021).

[116] C. Dwork, “Differential Privacy”, in *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ser. ICALP’06, Venice, Italy: Springer-Verlag, 2006, pp. 1–12, ISBN: 3540359079.

DOI: 10.1007/11787006_1. [Online]. Available: https://doi.org/10.1007/11787006_1.

[117] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, Aug. 2014, ISSN: 1551-305X. DOI: 10.1561/0400000042. [Online]. Available: https://doi.org/10.1561/0400000042.

[118] M. Abadi, H. B. McMahan, A. Chu, I. Mironov, L. Zhang, I. Goodfellow, and K. Talwar, "Deep Learning with Differential Privacy", *Proceedings of the ACM Conference on Computer and Communications Security*, no. Ccs, pp. 308–318, 2016, ISSN: 15437221. DOI: 10.1145/2976749.2978318. arXiv: arXiv:1607.00133v2.

[119] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, *Differentially Private Generative Adversarial Network*, 2018. arXiv: 1802.06739. [Online]. Available: http://arxiv.org/abs/1802.06739.

[120] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, and B. A. y Arcas, *Generative Models for Effective ML on Private, Decentralized Datasets*, 2020. arXiv: 1911.06679 [cs.LG].

[121] C. Hardy, E. L. Merrer, and B. Sericola, *MD-GAN: Multi-Discriminator Generative Adversarial Networks for Distributed Datasets*, 2019. arXiv: 1811.03850 [cs.LG].

[122] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, *Communication-Efficient Learning of Deep Networks from Decentralized Data*, 2017. arXiv: 1602.05629 [cs.LG].

[123] M. Rasouli, T. Sun, and R. Rajagopal, *FedGAN: Federated Generative Adversarial Networks for Distributed Data*, 2020. arXiv: 2006.07228 [cs.LG].

[124] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, *Membership Inference Attacks against Machine Learning Models*, 2017. arXiv: 1610.05820 [cs.CR].

[125] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: Membership Inference Attacks Against Generative Models", *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 1, pp. 133–152, 2019. DOI: 10.2478/popets-2019-0008. arXiv: arXiv:1705.07663v4.

[126] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey", *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[127] X. Gong, S. Chang, Y. Jiang, and Z. Wang, *AutoGAN: Neural Architecture Search for Generative Adversarial Networks*, 2019. DOI: 10.48550/ARXIV.1908.03835. [Online]. Available: https://arxiv.org/abs/1908.03835.

[128] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need", in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017.

[129] K. E. Emam, S. Rodgers, and B. Malin, "Anonymising and sharing individual patient data", *BMJ (Online)*, vol. 350, 2015, ISSN: 17561833. DOI: 10.1136/bmj.h1139.

[130] S. McLachlan, K. Dube, and T. Gallagher, "Using the CareMap with Health Incidents Statistics for Generating the Realistic Synthetic Electronic Healthcare Record", *Proceedings - 2016 IEEE International Conference on Healthcare Informatics, ICHI 2016*, pp. 439–448, 2016. DOI: 10.1109/ICHI.2016.83.

[131] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals", *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 3, pp. 289–294, 2003, ISSN: 00189294. DOI: 10.1109/TBME.2003.808805.

[132] Y. Park and J. Ghosh, *Perturbed Gibbs Samplers for Synthetic Data Release*, 2013. DOI: `10.1109/ICHI.2013.76`. arXiv: `arXiv:1312.5370v1`. [Online]. Available: `http://arxiv.org/abs/1312.5370`.

[133] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, *WaveNet: A Generative Model for Raw Audio*, 2016. arXiv: `1609.03499`. [Online]. Available: `http://arxiv.org/abs/1609.03499`.

[134] J. Dahmen and D. Cook, "SynSys: A synthetic data generation system for healthcare applications", *Sensors (Switzerland)*, vol. 19, no. 5, pp. 1–11, 2019, ISSN: 14248220. DOI: `10.3390/s19051181`.

[135] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in time series: A survey and taxonomy", 2021. arXiv: `2107.11098 [cs.LG]`.

[136] A. Johnson, T. Pollard, and R. Mark, *MIMIC-III Clinical Database (version 1.4)*. PhysioNet, 2016. [Online]. Available: `https://doi.org/10.13026/C2XW26`.

[137] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals", *Circulation*, vol. 101, no. 23, e215–e220, 2000, Circulation Electronic Pages: `http://circ.ahajournals.org/content/101/23/e215`.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[138] M. Mirza and S. Osindero, *Conditional Generative Adversarial Nets*, 2014. arXiv: `1411.1784`. [Online]. Available: `http://arxiv.org/abs/1411.1784`.

[139] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. J. Keogh, "Generalizing DTW to the multi-dimensional case requires an adaptive approach", *Data Mining and Knowledge Discovery*, vol. 31, pp. 1–31, 2016.

[140] G.-J. Qi, *Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities*, 2018. arXiv: `1701.06264 [cs.CV]`.

[141] W. Bounliphone, E. Belilovsky, M. B. Blaschko, I. Antonoglou, and A. Gretton, *A Test of Relative Similarity For Model Selection in Generative Models*, 2015. DOI: 10.48550/ARXIV.1511.04581. [Online]. Available: https://arxiv.org/abs/1511.04581.

[142] B. Kulynych and M. Yaghini, *mia: A library for running membership inference attacks against ML models*, Sep. 2018. DOI: 10.5281/zenodo.1433744. [Online]. Available: https://doi.org/10.5281/zenodo.1433744.

[143] J. Weldon, T. Ward, and E. Brophy, *Generation of Synthetic Electronic Health Records Using a Federated GAN*, 2021. arXiv: 2109.02543 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2109.02543.

[144] B. W. Nelson and N. B. Allen, "Accuracy of Consumer Wearable Heart Rate Measurement During an Ecologically Valid 24-Hour Period: Intraindividual Validation Study", *JMIR Mhealth Uhealth*, vol. 7, no. 3, e10828, Mar. 2019, ISSN: 2291-5222. DOI: 10.2196/10828. [Online]. Available: https://mhealth.jmir.org/2019/3/e10828/.

[145] J. Allen, "Photoplethysmography and its application in clinical physiological measurement", *Physiological Measurement*, vol. 28, no. 3, 2007. DOI: 10.1088/0967-3334/28/3/R01.

[146] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013. DOI: 10.1109/SURV.2012.110112.00192.

[147] E. S. Sazonov, G. Fulk, N. Sazonova, and S. Schuckers, "Automatic recognition of postures and activities in stroke patients", *31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, pp. 2200–2203, 2009. DOI: 10.1109/IEMBS.2009.5334908.

[148] Apple Inc. "Your heart rate. What it means, and where on Apple Watch you'll find it". (2019), [Online]. Available: https://support.apple.com/en-ie/HT204666 (visited on 10/21/2019).

[149] Department of Health, *Health in Ireland: Key Trends 2018*, https://www.gov.ie/en/press-release/4f8096-health-in-ireland-key-trends-2018/?referrer=/wp-content/uploads/2018/12/key-health-trends-2018.pdf/, 2018.

[150] GP. Bullhound, *Digital-Healthcare: Local Challenges, Global Opportunities*, 2015.

[151] Deloitte, *2019 Global health care outlook*, https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Life-Sciences-Health-Care/gx-lshc-hc-outlook-2019.pdf, 2019.

[152] Frost & Sullivan, *Top 20 Global Mega Trends and Their Impact on Business, Cultures and Society*, 2010.

[153] Department of Health, *Future Health – A Strategic Framework for Reform of the Health Service 2012 – 2015*, https://www.gov.ie/en/publication/1dd5b7-future-health-a-strategic-framework-for-reform-of-the-health-service/, 2012.

[154] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers", *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010. DOI: 10.3390/s100201154.

[155] E. Brophy, Z. Wang, J. J. Dominguez Veiga, A. F. Smeaton, and T. Ward, "An Interpretable Machine Vision Approach to Human Activity Recognition using Photoplethysmograph Sensor Data", in *26th AIAI Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin: CEUR Workshop Proceedings, 2018.

[156] M. Boukhechba, L. Cai, C. Wu, and L. E. Barnes, “ActiPPG: Using deep neural networks for activity recognition from wrist-worn photoplethysmography (PPG) sensors”, *Smart Health*, vol. 14, p. 100 082, 2019, ISSN: 2352-6483. DOI: `https://doi.org/10.1016/j.smhl.2019.100082`.

[157] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, “Human Activity Recognition Using Accelerometer and Photoplethysmographic Signals”, in *Intelligent Decision Technologies 2017*, I. Czarnowski, R. J. Howlett, and L. C. Jain, Eds., Cham: Springer International Publishing, 2018, pp. 53–62, ISBN: 978-3-319-59424-8.

[158] S. Mehrang, J. Pietila, J. Tolonen, E. Helander, H. Jimison, M. Pavel, and I. Korhonen, “Human Activity Recognition Using A Single Optical Heart Rate Monitoring Wristband Equipped with Triaxial Accelerometer”, in *EMBEC & NBC 2017*, H. Eskola, O. Väisänen, J. Viik, and J. Hyttinen, Eds., Singapore: Springer Singapore, 2018, pp. 587–590, ISBN: 978-981-10-5122-7.

[159] Y. Qiu, Y. Liu, J. Arteaga-Falconi, H. Dong, and A. E. Saddik, “EVM-CNN: Real-Time Contactless Heart Rate Estimation From Facial Video”, *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1778–1787, Jul. 2019. DOI: `10.1109/TMM.2018.2883866`.

[160] R. Spetlík, V. Franc, J. Cech, and J. Matas, “Visual Heart Rate Estimation with Convolutional Neural Network”, in *BMVC*, 2018.

[161] A. Reiss, I. Indlekofer, P. Schmidt, and K. Van Laerhoven, “Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks”, *Sensors*, vol. 19, no. 14, p. 3079, Jul. 2019, ISSN: 1424-8220. DOI: `10.3390/s19143079`. [Online]. Available: `http://dx.doi.org/10.3390/s19143079`.

[162] H. Junker, P. Lukowicz, and G. Troster, “Sampling frequency, signal resolution and the accuracy of wearable context recognition systems”, in *Eighth International Symposium on Wearable Computers*, vol. 1, Oct. 2004, pp. 176–177. DOI: `10.1109/ISWC.2004.38`.

[163] A. Krause, M. Ihmig, E. Rankin, D. Leong, Smriti Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta, “Trading off prediction accuracy and power consumption for context-aware wearable computing”, in *Ninth IEEE International Symposium on Wearable Computers (ISWC’05)*, Oct. 2005, pp. 20–26. DOI: 10.1109/ISWC.2005.52.

[164] D. Jarchi and A. Casson, “Description of a Database Containing Wrist PPG Signals Recorded during Physical Exercise with Both Accelerometer and Gyroscope Measures of Motion”, *Data*, vol. 2, no. 1, p. 1, 2016. DOI: 10.3390/data2010001.

[165] J. J. Dominguez Veiga, M. O’Reilly, D. Whelan, B. Caulfield, and T. E. Ward, “Feature-Free Activity Classification of Inertial Sensor Data With Machine Vision Techniques: Method, Development, and Evaluation”, *JMIR mHealth and uHealth*, vol. 5, no. 8, e115, 2017. DOI: 10.2196/mhealth.7521.

[166] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, *Advances In Neural Information Processing Systems*, pp. 1–9, 2012. DOI: http://dx.doi.org/10.1016/j.protcy.2014.09.007. arXiv: 1102.0183.

[167] S. J. Pan and Q. Yang, “A Survey on Transfer Learning”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.

[168] J. D. Hunter, “Matplotlib: A 2D graphics environment”, *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[169] C. Xie, L. McCullum, A. Johnson, T. Pollard, B. Gow, and B. Moody, “Waveform Database Software Package (WFDB) for Python”, *PhysioNet*, 2022. DOI: https://doi.org/10.13026/mmpm-2v55.

[170] P. Van Gent, H. Farah, and V. Gent, “Heart Rate Analysis for Human Factors: Development and Validation of an Open Source Toolkit for Noisy Naturalistic

Heart Rate Data", in *Proceedings of The 6th HUMMANIST Conference*, 2018, pp. 170–175.

[171] E. Brophy, W. Muehlhausen, A. F. Smeaton, and T. E. Ward, "CNNs for Heart Rate Estimation and Human Activity Recognition in Wrist Worn Sensing Applications", in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6. DOI: `10.1109/PerComWorkshops48775.2020.9156120`.

[172] ——, *Optimised Convolutional Neural Networks for Heart Rate Estimation and Human Activity Recognition in Wrist Worn Sensing Applications*, 2020. arXiv: `2004.00505 [eess.SP]`.

[173] *The top 10 causes of death*, World Health Organisation, 2020. [Online]. Available: `https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death`.

[174] J. M. Padilla, E. J. Berjano, J. Saiz, L. Facila, P. Diaz, and S. Merce, "Assessment of relationships between blood pressure, pulse wave velocity and digital volume pulse", in *2006 Computers in Cardiology*, 2006, pp. 893–896.

[175] M. Y.-M. Wong, C. C.-Y. Poon, and Y.-T. Zhang, "An Evaluation of the Cuffless Blood Pressure Estimation Based on Pulse Transit Time Technique: A Half Year Study on Normotensive Subjects", *Cardiovascular Engineering*, vol. 9, no. 1, pp. 32–38, Mar. 2009, ISSN: 1567-8822, 1573-6806. DOI: `10.1007/s10558-009-9070-7`. [Online]. Available: `http://link.springer.com/10.1007/s10558-009-9070-7` (visited on 02/11/2021).

[176] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, 2020. arXiv: `1703.10593 [cs.CV]`.

[177] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time", in *2015 IEEE International Symposium on Circuits and Systems*

(ISCAS), Lisbon, Portugal: IEEE, May 2015, pp. 1006–1009, ISBN: 978-1-4799-8391-9. DOI: 10.1109/ISCAS.2015.7168806. [Online]. Available: http://ieeexplore.ieee.org/document/7168806/ (visited on 02/11/2021).

[178] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cuffless Blood Pressure Estimation Algorithms for Continuous Health-Care Monitoring", *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 4, pp. 859–869, 2017. DOI: 10.1109/TBME.2016.2580904.

[179] D. Liu, M. Görges, and S. A. Jenkins, "University of Queensland Vital Signs Dataset: Development of an Accessible Repository of Anesthesia Patient Monitoring Data for Research", *Anesthesia & Analgesia*, vol. 114, no. 3, pp. 584–589, Mar. 2012, ISSN: 0003-2999. DOI: 10.1213/ANE.0b013e318241f7c0. [Online]. Available: http://journals.lww.com/00000539-201203000-00015 (visited on 02/11/2021).

[180] L. Bonsall, *Calculating the mean arterial pressure (MAP)*, Lippincott NursingCenter, 2011. [Online]. Available: https://www.nursingcenter.com/ncblog/december-2011/calculating-the-map.

[181] D. Giavarina, "Understanding Bland Altman analysis", *Croatian Society of Medical Biochemistry and Laboratory Medicine*, vol. 25, pp. 141–51, 2 Jun. 2015, ISSN: 1330-0962. DOI: 10.11613/BM.2015.015.

[182] Association for the Advancement of Medical Instrumentation, "American National Standards for Electronic or Automated Sphygmomanometers", *ANSI/AAMI SP 10-1987*, 1987. [Online]. Available: https://ci.nii.ac.jp/naid/10024828510/en/.

[183] N. Ibtehaz and M. S. Rahman, "PPG2ABP: Translating Photoplethysmogram (PPG) Signals to Arterial Blood Pressure (ABP) Waveforms using Fully Convolutional Neural Networks", *arXiv:2005.01669 [cs, eess]*, May 2020, arXiv: 2005.01669. [Online]. Available: http://arxiv.org/abs/2005.01669 (visited on 02/05/2021).

[184] B. Mishra and N. S. Nirala, “A Survey on Denoising Techniques of PPG Signal”, in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 2020, pp. 1–8. DOI: 10.1109/INOCON50539.2020.9298358.

[185] P. Anderer, S. Roberts, A. Schlögl, G. Gruber, G. Klösch, W. Herrmann, P. Rappelsberger, O. Filz, M. Barbanoj, G. Dorffner, and B. Saletu, “Artifact processing in computerized analysis of sleep EEG - a review”, *Neuropsychobiology*, vol. 40, pp. 150–157, 3 Sep. 1999, ISSN: 1084-6654. DOI: 10.1159/000026613..

[186] X. Jiang, G.-B. Bian, and Z. Tian, “Removal of Artifacts from EEG Signals: A Review”, *Sensors*, vol. 19, no. 5, 2019, ISSN: 1424-8220. DOI: 10.3390/s19050987. [Online]. Available: https://www.mdpi.com/1424-8220/19/5/987.

[187] C. I. Salis, A. E. Malissovas, P. A. Bizopoulos, A. T. Tzallas, P. A. Angelidis, and D. G. Tsalikakis, “Denoising simulated EEG signals: A comparative study of EMD, wavelet transform and Kalman filter”, in *13th IEEE International Conference on BioInformatics and BioEngineering*, 2013, pp. 1–4. DOI: 10.1109/BIBE.2013.6701613.

[188] L. Albera, A. Kachenoura, P. Comon, A. Karfoul, F. Wendling, L. Senhadji, and I. Merlet, “ICA-based EEG denoising: a comparative analysis of fifteen methods”, *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 60, no. 3 Special issue on Data Mining in Bioengineering, pp. 407–418, 2012.

[189] I. Rejer and P. Górski, “Benefits of ICA in the Case of a Few Channel EEG”, *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 7434–7437, 2015.

[190] H. Zhang, M. Zhao, C. Wei, D. Mantini, Z. Li, and Q. Liu, “EEGdenoiseNet: A benchmark dataset for deep learning solutions of EEG denoising”, vol. 18, no. 5, p. 056 057, Oct. 2021. DOI: 10.1088/1741-2552/ac2bf8. [Online]. Available: https://doi.org/10.1088/1741-2552/ac2bf8.

[191] Y. Luo and B. L. Lu, "EEG Data Augmentation for Emotion Recognition Using a Conditional Wasserstein GAN", *Annu Int Conf IEEE Eng Med Biol Soc*, vol. 2018, pp. 2535–2538, Jul. 2018.

[192] F. Fahimi, S. Dosen, K. K. Ang, N. Mrachacz-Kersting, and C. Guan, "Generative Adversarial Networks-Based Data Augmentation for Brain–Computer Interface", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4039–4051, 2021. DOI: 10.1109/TNNLS.2020.3016666.

[193] I. A. Corley and Y. Huang, "Deep EEG super-resolution: Upsampling EEG spatial resolution with Generative Adversarial Networks", in *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, 2018, pp. 100–103. DOI: 10.1109/BHI.2018.8333379.

[194] S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano, and M. Shah, "Generative Adversarial Networks Conditioned by Brain Signals", in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3430–3438. DOI: 10.1109/ICCV.2017.369.

[195] S. Gandhi, T. Oates, T. Mohsenin, and D. Hairston, "Denoising Time Series Data Using Asymmetric Generative Adversarial Networks", in *Advances in Knowledge Discovery and Data Mining*, D. Phung, V. S. Tseng, G. I. Webb, B. Ho, M. Ganji, and L. Rashidi, Eds., Cham: Springer International Publishing, 2018, pp. 285–296, ISBN: 978-3-319-93040-4.

[196] H. Zhang, C. Wei, M. Zhao, Q. Liu, and H. Wu, "A Novel Convolutional Neural Network Model to Remove Muscle Artifacts from EEG", in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1265–1269. DOI: 10.1109/ICASSP39728.2021.9414228.

[197] J. F. Hwaidi and T. M. Chen, "A Noise Removal Approach from EEG Recordings Based on Variational Autoencoders", in *2021 13th International*

Conference on Computer and Automation Engineering (ICCAE), 2021, pp. 19–23. DOI: 10.1109/ICCAE51876.2021.9426150.

[198] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, "BCI2000: a general-purpose brain-computer interface (BCI) system", *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1034–1043, 2004. DOI: 10.1109/TBME.2004.827072.

[199] D. Makowski, T. Pham, Z. J. Lau, J. C. Brammer, F. Lespinasse, H. Pham, C. Schölzel, and S. H. A. Chen, "NeuroKit2: A Python toolbox for neurophysiological signal processing", *Behavior Research Methods*, vol. 53, no. 4, pp. 1689–1696, Feb. 2021. DOI: 10.3758/s13428-020-01516-y. [Online]. Available: https://doi.org/10.3758%5C%2Fs13428-020-01516-y.

[200] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, and D. K. Raghuvanshi, "Review of noise removal techniques in ECG signals", *IET Signal Processing*, vol. 14, no. 9, pp. 569–590, 2020. DOI: https://doi.org/10.1049/iet-spr.2020.0104.

[201] M. Blanco-Velasco, B. Weng, and K. E. Barner, "ECG signal denoising and baseline wander correction based on the empirical mode decomposition", *Computers in Biology and Medicine*, vol. 38, no. 1, pp. 1–13, 2008, ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2007.06.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482507001114.

[202] V. de Pinto, "Filters for the reduction of baseline wander and muscle artifact in the ECG", *Journal of Electrocardiology*, vol. 25, pp. 40–48, 1992, Research and Applications in Computerized Electrocardiology, ISSN: 0022-0736. DOI: https://doi.org/10.1016/0022-0736(92)90060-D. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002207369290060D.

[203] H. W. Tam and J. G. Webster, “Minimizing Electrode Motion Artifact by Skin Abrasion”, *IEEE Transactions on Biomedical Engineering*, vol. BME-24, no. 2, pp. 134–139, 1977. DOI: 10.1109/TBME.1977.326117.

[204] N. Seeuws, M. De Vos, and A. Bertrand, “Electrocardiogram Quality Assessment using Unsupervised Deep Learning”, *IEEE Transactions on Biomedical Engineering*, pp. 1–1, 2021. DOI: 10.1109/TBME.2021.3108621.

[205] J. Moeyersons, E. Smets, J. Morales, A. Villa, W. De Raedt, D. Testelmans, B. Buyse, C. Van Hoof, R. Willems, S. Van Huffel, and C. Varon, “Artefact detection and quality assessment of ambulatory ECG signals”, *Computer Methods and Programs in Biomedicine*, vol. 182, p. 105 050, 2019, ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2019.105050. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260719312817.

[206] J. Moeyersons, J. Morales, A. Villa, I. Castro, D. Testelmans, B. Buyse, C. Van Hoof, R. Willems, S. Van Huffel, and C. Varon, “Supervised SVM Transfer Learning for Modality-Specific Artefact Detection in ECG”, *Sensors*, vol. 21, no. 2, 2021, ISSN: 1424-8220. DOI: 10.3390/s21020662. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/662.

[207] T.-Y. Slonim, M. Slonim, and E. Ovsyscher, “The use of simple FIR filters for filtering of ECG signals and a new method for post-filter signal reconstruction”, in *Proceedings of Computers in Cardiology Conference*, 1993, pp. 871–873. DOI: 10.1109/CIC.1993.378347.

[208] N. Thakor and Y.-S. Zhu, “Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection”, *IEEE Transactions on Biomedical Engineering*, vol. 38, no. 8, pp. 785–794, 1991. DOI: 10.1109/10.83591.

[209] V. Afonso, W. Tompkins, T. Nguyen, S. Trautmann, and S. Luo, “Filter bank-based processing of the stress ECG”, in *Proceedings of 17th International*

Conference of the Engineering in Medicine and Biology Society, vol. 2, 1995, 887–888 vol.2. DOI: 10.1109/IEMBS.1995.579254.

[210] C. T. Arsene, R. Hankins, and H. Yin, "Deep Learning Models for Denoising ECG Signals", in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5. DOI: 10.23919/EUSIPCO.2019.8902833.

[211] S. Barton, S. Alakkari, K. O'Dwyer, T. Ward, and B. Hennelly, "Convolution Network with Custom Loss Function for the Denoising of Low SNR Raman Spectra", *Sensors*, vol. 21, no. 14, 2021, ISSN: 1424-8220. DOI: 10.3390/s21144623. [Online]. Available: https://www.mdpi.com/1424-8220/21/14/4623.

[212] G. B. Moody, R. G. Mark, and M. W.E., "A noise stress test for arrhythmia detectors", in *Computers in Cardiology, 1984*, vol. 11, 2005, pp. 381–384.

[213] M. A. Neggaz, I. Alouani, P. R. Lorenzo, and S. Niar, "A Reliability Study on CNNs for Critical Embedded Systems", in *2018 IEEE 36th International Conference on Computer Design (ICCD)*, 2018, pp. 476–479. DOI: 10.1109/ICCD.2018.00077.

[214] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[215] T. Pham, Z. J. Lau, S. H. A. Chen, and D. Makowski, "Heart Rate Variability in Psychology: A Review of HRV Indices and an Analysis Tutorial", *Sensors*,

vol. 21, no. 12, 2021, ISSN: 1424-8220. DOI: 10.3390/s21123998. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/3998.

[216] F. Shaffer and J. P. Ginsberg, "An Overview of Heart Rate Variability Metrics and Norms", *Frontiers in Public Health*, vol. 5, p. 258, 2017, ISSN: 2296-2565. DOI: 10.3389/fpubh.2017.00258.

[217] N. Bourdillon, L. Schmitt, S. Yazdani, J.-M. Vesin, and G. P. Millet, "Minimal Window Duration for Accurate HRV Recording in Athletes", *Frontiers in Neuroscience*, vol. 11, 2017. DOI: 10.3389/fnins.2017.00456.

[218] S. Studer, T. B. Bui, C. Drescher, A. Hanuschkin, L. Winkler, S. Peters, and K.-R. Mueller, *Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology*, 2021. arXiv: 2003.05155 [cs.LG].

Appendix A

Denoising EEG Signals

A.1 Supplementary Material EEG Denoising

A.1.1 Figures

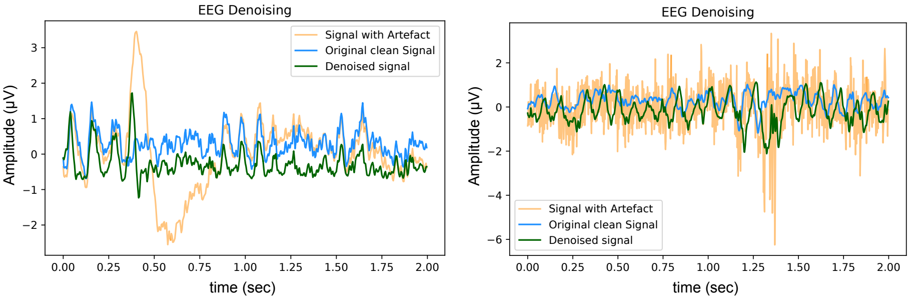

Figure A.1: Example of denoised time series EEG corrupted with (left) EOG artefact and (right) EMG artefact. The signals contain an artificial offset for visualisation purposes.

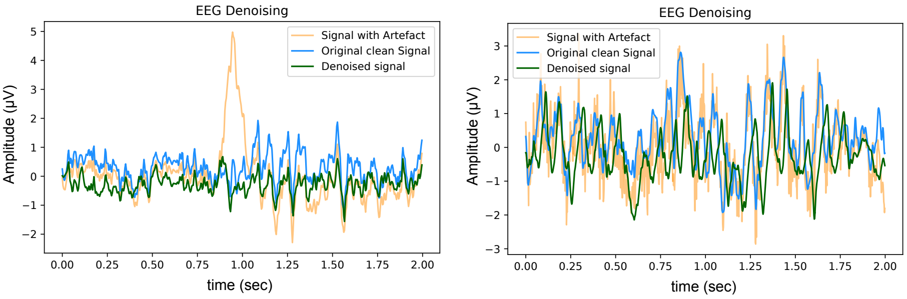

Figure A.2: Example of denoised time series EEG corrupted with (left) EOG artefact and (right) EMG artefact. The signals contain an artificial offset for visualisation purposes.

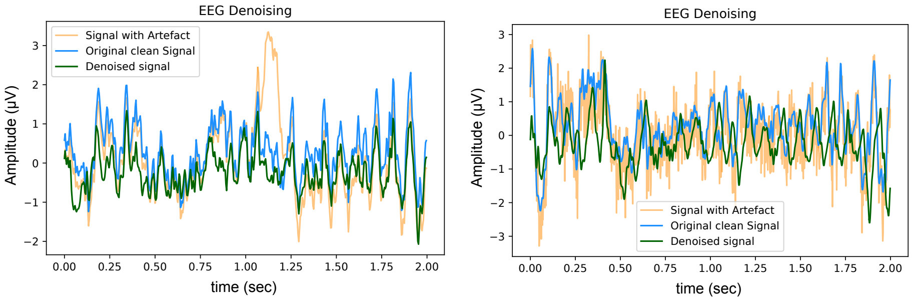

Figure A.3: Example of denoised time series EEG corrupted with (left) EOG artefact and (right) EMG artefact. The signals contain an artificial offset for visualisation purposes.

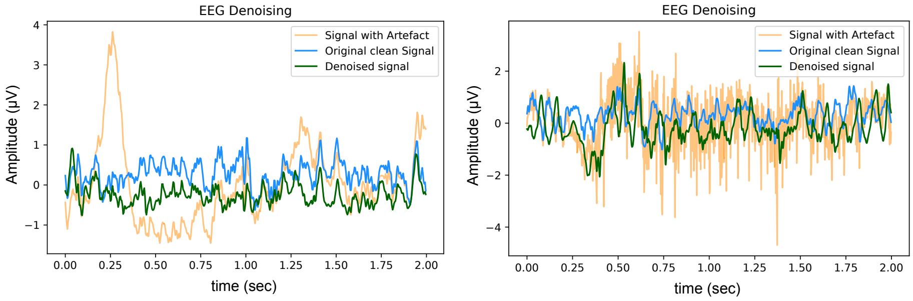

Figure A.4: Example of denoised time series EEG corrupted with (left) EOG artefact and (right) EMG artefact. The signals contain an artificial offset for visualisation purposes.

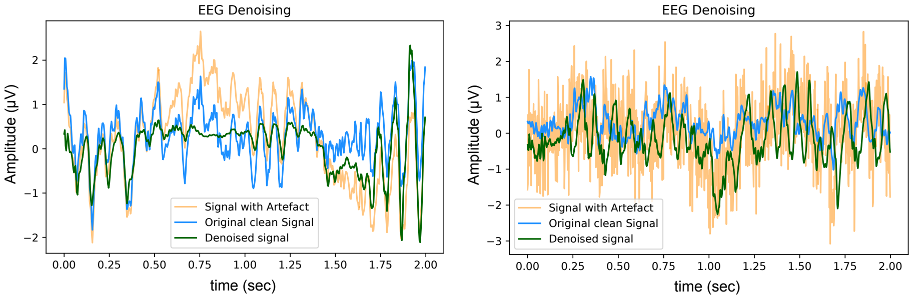

Figure A.5: Example of denoised time series EEG corrupted with (left) EOG artefact and (right) EMG artefact. The signals contain an artificial offset for visualisation purposes.

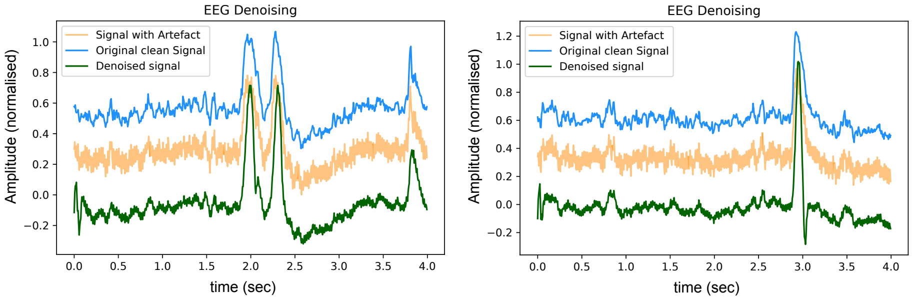

Figure A.6: Example of denoised time series EEG corrupted with 50 Hz mains noise. The signals contain an artificial offset for visualisation purposes.

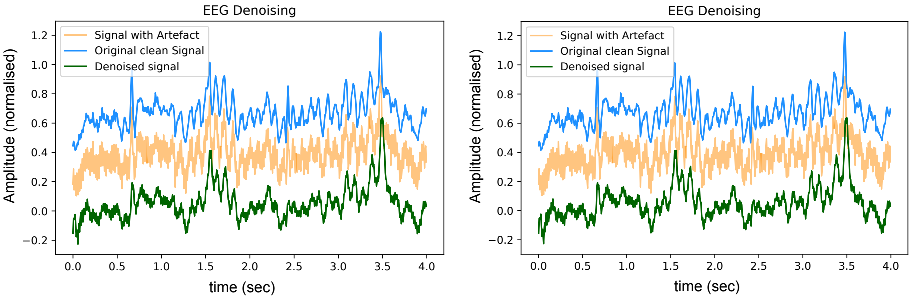

Figure A.7: Example of denoised time series EEG corrupted with 50 Hz mains noise. The signals contain an artificial offset for visualisation purposes.

Appendix B

Denoising ECG Signals

B.1 Supplementary Material ECG Denoising

B.1.1 Figures

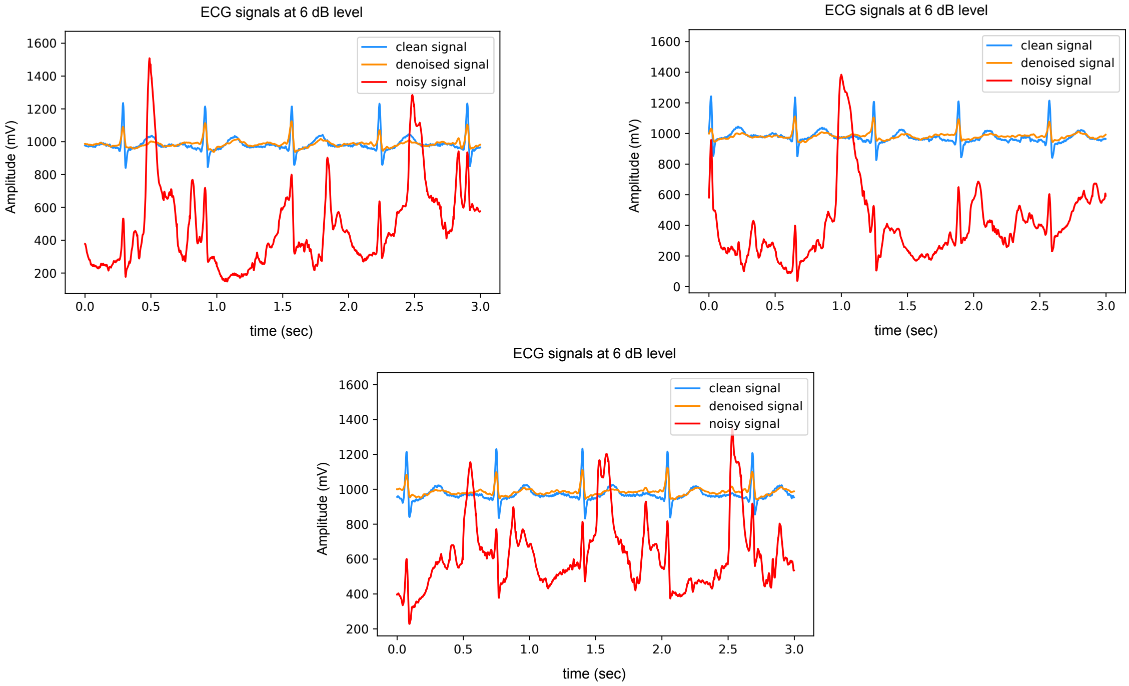

Figure B.1: Further results of ECG denoising with differing noisy input signals. Signals are shown before and after denoising with CNN and custom loss function. An artificial offset is present in the noisy signal for visualisation purposes. The initial SNR of the noisy signal is 6 dB.