VALIDATION AND INTERPRETABLE MODEL EXPLANATIONS FOR SYNTHESIZED DATA IN HEALTHCARE

by

Barbara Mukami Maweu

APPROVED BY SUPERVISORY COMMITTEE:

Balakrishnan Prabhakaran, Chair

Gopal Gupta

Lawrence Chung

Vibhav Gogate

To Dad, Mom, Maweu, Ireri and Ndulu forever thankful for your faith in me.

VALIDATION AND INTERPRETABLE MODEL EXPLANATIONS FOR SYNTHESIZED DATA IN HEALTHCARE

by

BARBARA MUKAMI MAWEU, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

SOFTWARE ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

August 2021

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my supervising advisor, Dr. Balakrishnan Prabhakaran, for guiding my research with honest feedback and unwavering encouragement.

To my committee members: Dr. Gopal Gupta, Dr. Lawrence Chung and Dr. Vibhav Gogate, I say thank you for your guidance.

To my Multimedia Lab colleagues, I will hold dearly the memories of tireless brainstorming sessions, manuscript writing and some crazy fun times in the lab.

To the Computer Science Department administrative staff, Ms. Norma Richardson and Mr. Douglas Hyde and Women Who Compute (WWC) mentor Dr. Janell Straach, thank you for being patient with me and always encouraging me.

To Maweu, Ireri Maweu and Ndulu Maweu, my family, you have been my rock throughout this journey. I will forever be grateful to you.

June 2021

VALIDATION AND INTERPRETABLE MODEL EXPLANATIONS FOR SYNTHESIZED DATA IN HEALTHCARE

Barbara Mukami Maweu, PhD
The University of Texas at Dallas, 2021

Supervising Professor: Balakrishnan Prabhakaran, Chair

Recent advances in artificial intelligence (AI) based solutions for healthcare problems have led to the increased demands for quality accessible patient data and the functional understanding of the remarkable outcomes of AI decision support systems. Nonetheless, challenges persist from strict regulations that oversee patient privacy, small imbalanced datasets due to high costs of measurement and expert annotation, and the black-box nature of AI technology.

In this dissertation, we address foundational frameworks necessary for achieving quality and accessible synthesized healthcare time-series data and build the essential trust and confidence in outcomes of AI solutions through interpretable explanations for healthcare time series data.

In the first challenge, we propose validation approaches for synthesized healthcare time-series data and apply this quality synthesized data in training better performing healthcare decision support systems. Finally, we present a framework that generates and integrates modular interpretable explanations from varying deep learning models with model capacities achieved using synthesized data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Recent advances in the use of artificial intelligence (AI) for solving complex healthcare tasks have led to an increased demand for large volumes of data that is needed to implement AI based systems. AI innovations that induce specialized patient care and produce better health outcomes need large volumes of training data to produce good performance. Deep learning methods for example, are now being applied to model decision support systems that are used by healthcare providers to deliver timely and accurate diagnosis.

One of the most pressing challenge in healthcare is the limited availability of data due to patient privacy concerns, irregular data collection patterns, and high costs of data collection and annotation. Limited data availability is characterized by restricted data access, small datasets, and imbalanced class representation within the datasets. This challenge has provided a need for other means of accessing patterns and features in healthcare data. Currently, synthetic data is playing a critical role in bridging the gap of data availability and therefore, healthcare researchers are devising new techniques for synthetic data generation and validation. Synthetic data is artificial data that is generated using computer algorithms with the goal of capturing features that are present in real data. With synthetic data, researchers can access large volumes of data, balance training sets, employ machine learning and deep networks to train models, and test new tools before they are deployed for real-world use.

As new methods of generating synthetic data are discovered, there is a growing need to ensure that synthetic data is sufficient proxy for real data. Therefore, there is a need for

developing frameworks that analyze the quality of synthetic data and validates their effectiveness in application.

Additionally, as researchers leverage state-of-the-art (SOTA) AI to develop production tools, there is an increasing demand by users to understand the functional mechanism of black-box AI systems. For example, a physician diagnosing arrhythmia with the help of a decision support tool wants to know the electrocardiogram (ECG) features used by the tool to determine the diagnosis. Healthcare providers want to understand how AI systems learn and make outcome decisions therefore, interpretable explanations from AI systems are essential.

These challenges in healthcare have motivated us to:

- Develop a validation framework for synthetic healthcare time series data.
- Explore the effectiveness of healthcare time series data in (i) traditional machine learning, (ii) no-residual deep neural networks, and (iii) residual neural networks.
- Develop and implement a post-hoc interpretable explainability framework for convolutional neural networks (CNN) models trained to classify ECG signals.

1.2 Dissertation Objective

The focus of this research is in healthcare time series data and addresses the following key questions:

1. Can we develop foundational methods for validating synthetic healthcare time-series data?
    a. Implement a framework that accounts for statistical, visual and machine learning validation of synthesized healthcare time-series data

2. Can synthesized healthcare time-series data serve as effective proxy in training AI-based decision support systems?
    a. Comparative analysis of classification performance between synthetic and real healthcare time series datasets.
        i. *Electrocardiogram Study*: Features one dimensional time series with highly structured waveforms
        ii. *Electroencephalogram Study*: Features multi-dimensional time series
        iii. *Activity Recognition Multi-sensor Study*: Features multi-dimensional time series with structured patterns
3. Can we develop a framework that produces interpretable explanations of highly structured healthcare dataset mapping the explanation to respective classification outcomes?
    a. Develop a foundational framework to generate interpretable explanations for convolutional neural networks.
    b. Analyze model capacity of deep neural networks using synthetic data.

1.3 Contribution

- *Framework for validating synthesized data*: Develop the Virtual Patient Model (VPM) framework for statistical, visual and machine learning validation of synthetic healthcare time series data.

- *Application of validated synthesized data*: Perform a comprehensive analytical study to demonstrate the effectiveness of using synthetic healthcare time series data when training deep neural network for domain classification tasks.
- *Framework for AI Interpretable Explanations with synthesized data*: Develop the CNN Explainability Framework for ECG Signals (CEFEs), for explaining model capacity and producing statistical, visual, and feature interpretable explanations using synthetic time series data.

1.4 Organization of the Dissertation

Chapter 2: We introduce synthetic based measurements, challenges, and current trends in synthetic data generators for healthcare datasets.

Chapter 3: We develop and describe an explainability framework for generating CNN model explanations using feature statistics, feature visualization, and feature detection and mapping modules. These CNN interpretable explanations are generated from some models trained with real data only and others trained with real data augmented with different volumes of synthetic data to better inform on model capacity.

Chapter 4: We develop and describe an end-to-end modular framework for validating synthetic healthcare time series data. This framework accounts for statistical, visual and machine learning validation for any time-series data synthesizer.

Chapter 5: We implement the statistical analysis and visual validation schemes for healthcare time series data sets using the synthetic data validation framework in Chapter 4.

Chapter 6: We present experimental results from the application of validated synthetic time-series data in training traditional machine learning algorithms, non-residual and residual deep neural networks for healthcare classification tasks.

CHAPTER 2

SYNTHETIC MEASUREMENTS AND GENERATORS IN HEALTHCARE

Healthcare stakeholders are required to preserve privacy through patient protection laws during the use and dissemination of health information. The requirements such as the Health Insurance Portability and Accountability Act (HIPAA) impose substantial constraints on research innovations that require large volumes of patient data. To overcome the limited accessibility to patient data, considerable advancements have been achieved in the areas focused on development of synthetic generation systems. These systems simulate or generate healthcare patient data, treatment plans, and artifacts of human biological systems. Quality synthetic measurements are essential for training, testing, and evaluating deep learning healthcare systems because the expectation is they produce utility performance similar to that obtained from real-world data and in addition, offer unlimited access to representative large volumes of data.

Synthetic measurements must preserve the characteristics of real-world data to be effective in modeling deep learning tasks. Therefore, a synthetic data generation framework should encompass generation methods that accounts for the varying features within healthcare datasets and implement concrete validation schemes that evaluate the generated synthetic measurements. The goal of validation methods is to provide (i) a conclusive analysis on features present in the synthetic measurements when compared to real-world measurements and (ii) a measure of effectiveness by assessing behavior and capability of synthetic measurements when applied to real-world tasks. In this paper, we contribute a literature survey of synthetic data generation and the

commonly acceptable validation schemes used to evaluate synthetic measurements in the context of healthcare datasets

2.1 Introduction

Healthcare stakeholders are required to preserve privacy through patient protection laws during the use and dissemination of health information. The requirements such as the Health Insurance Portability and Accountability Act (HIPAA) are substantial constraints on research innovations that require large volumes of patient data. To overcome the limited accessibility to patient data, considerable advancements have been achieved in the areas focused on development of synthetic generation systems that simulate healthcare patient data, treatment plans, and artifacts of human biological systems.

Quality synthetic measurements are essential for training, testing, and evaluating deep learning healthcare systems because they offer unlimited access to large volumes of representative patient data. Synthetic measurements must preserve the characteristics of real-world data to be effective in their application in deep learning tasks. Therefore, a synthetic data generation framework should encompass generation methods that accounts for varying features within the data and implement concrete validation schemes that evaluate synthetic measurements.

The validation methods seeks to obtain (i) conclusive analysis on features present in the synthetic measurements when compared to real-world measurements and (ii) the measure of effectiveness by assessing behavior and capability of synthetic measurements when applied to real-world tasks. In this paper we contribute a literature survey of data generators and commonly

acceptable validation schemes. We discuss these approaches in the context of synthetic healthcare datasets.

2.2 Synthetic Data Generation in Healthcare

Patient data plays an important role in healthcare research and innovation. Patients continue to seek informed and better health outcomes yet this effort is masked by data privacy concerns and legal ramifications. These consequences discourage patient data access by those accountable for maintaining patient privacy. The data environment that supports innovation in solving medical problems requires access to large volumes of patient data yet healthcare data is intrinsically scarce.

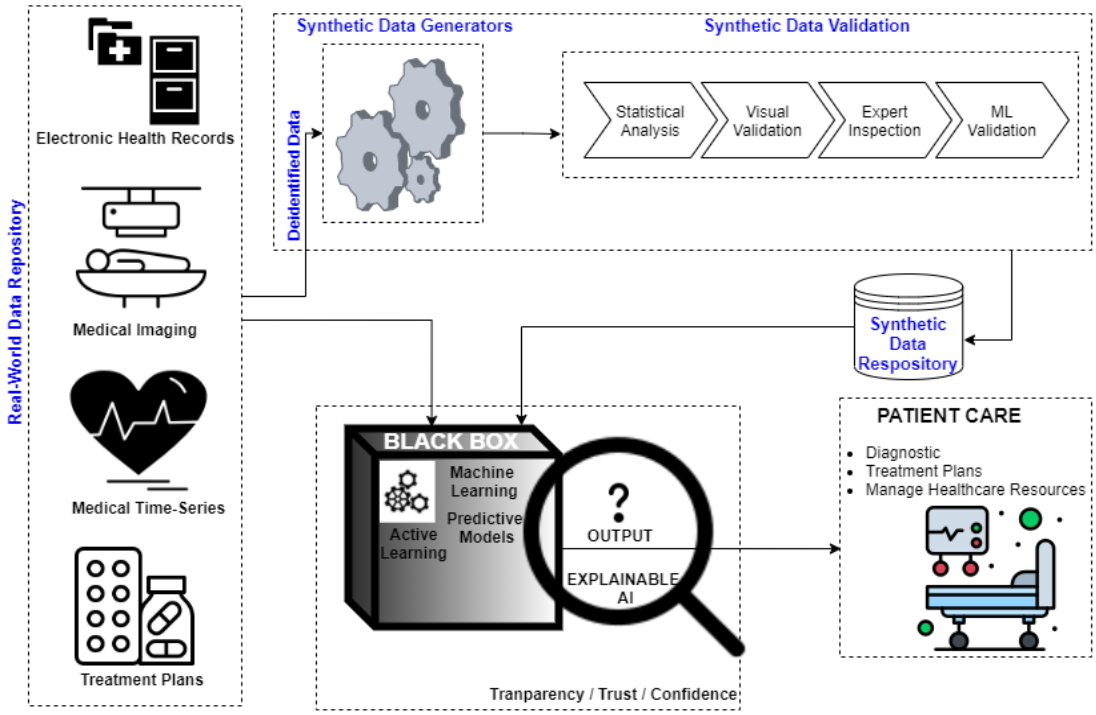


Figure 2.1. Synthetic Data Generation Process Flow in Healthcare

The challenges of data scarcity, small volumes of data, and high costs of annotation have promoted recent success in computational methods for generating cost-effective and realistic synthetic datasets. Synthetic generators gives researchers the ability to build synthetic data repositories thus unrestricted access to large volumes of data. One factor considered essential for the use of synthetic data in real-world tasks is ensuring that synthetic data encapsulates the characteristics of real data. Therefore, the widespread use of synthetic data highly depends on developing practical processes that qualitatively validate the data. The process of data synthesis in healthcare follows a standard flow as shown in Figure 2.1.

2.2.1 Real-World Data Repositories

Data synthesis in healthcare begins with a pool of patient-related data that is measured from different sources such as radiology, devices that monitor biological systems, electronic vitals measurements, and pharmaceutical dispensation records from medical treatment plans. Stored data formats range from medical images and bio-physical signals, treatment plans, and operational data from electronic health records. Data format is key to determining the appropriate synthetic data generator during the synthesis process.

2.2.2 Patient Data De-identification

The most important aspect that drives the need for synthetic data is preserving patient privacy. To guard patient privacy and protect organizational proprietary data aggregation methods, real-world data is first de-identified prior to making available for synthesis and public use. The de-identification process involves removing patient information that would link them to a particular health record or obscuring data fields that would expose proprietary business

methods. Government agencies (Health and Human Services) and healthcare providers (research collaborators) have published methods for patient de-identification and address the acceptable levels concerning identification risks.

2.2.3 Synthetic Data Generators

Synthetic measurements are generated using computer models and with the primary goal of capturing relevant statistical and morphological features present in real-world data. Synthetic measurements also referred to as synthetic data or artificial data are generated using mathematical models that require extensive domain knowledge of the underlying systems being modeled, stochastic processes that fit known or unknown distribution to real-world de-identified data, and hybrid methods that use machine learning techniques build models that learn the distribution of real-world data. These models are then sampled to generate synthetic data expected to characterize realistic representation of the real data.

2.2.4 Synthetic Data Validations

The utility of synthetic data in real-world healthcare tasks relies on the quality of data used to train computational models. Quality synthetic data refers to data that is effective in achieving similar outcomes to real data for a particular problem-solving task. Data quality is asserted by a set of evaluation procedures including statistical, visual, expert and machine learning analysis. Choice of validation tests that are used on synthetic data are both data and task specific. Therefore, the characteristics present in the data and the purpose for using this data must be clearly understood prior to validation. Synthetic data validation is particularly important

in the healthcare setting because utility of such data is applied mainly towards life-critical systems.

2.2.5 Synthetic Data Repositories

Once synthetic measurements are validated and deemed effective for use in training, testing, and evaluating healthcare models, the measurements are stored in synthetic data repositories. Synthetic data repositories are available to researchers for use in innovations that require large data volumes without the accessibility challenges prevalent in healthcare data.

2.2.6 Utility of Synthetic Data in Healthcare

The goal of healthcare innovations is to improve the quality of patient care. Healthcare innovations are in form of decision support systems modeled using artificial intelligence (AI) methods. During the research and development phase, these systems can be modeled, evaluated, and analyzed using quality synthetic measurements. Decision support systems aide healthcare providers when analyzing medical measurements and during subsequent diagnosis decision. They also enable medical providers to be efficient in their tasks, improve medical care which result in better healthcare outcomes for patients. The development of novel methods for synthesis and validation of healthcare synthetic measurements, opens more opportunities for researchers who leverage synthetic measurements for solving increasingly complex problems.

2.3 Trends in Healthcare Data Synthesis

We outline the evolving trends in process driven and data driven methods for synthesizing data. The goal of these methods is to generate synthetic data that is visually and

statically realistic in addition to being effective when applied for use in real-world healthcare tasks. Synthetic data is revolutionizing the way researchers approach AI in healthcare. Efforts are in place both in academia and industry settings to develop efficient synthetic data generators for all healthcare datasets (EHR, medical imaging, and time-series).

2.3.1 Process Driven Synthesizers

Whole Heart Modeling

Whole heart is a model of the electro-mechanical functions of the heart. It comprises of a torso model of the human body that produces electrocardiograms (ECG) surface potentials and provides an understanding of the complexity of interactions within the heart organ. Since the Whole-heart model was developed, several improvements have been proposed to improve the simulation of the cardiac function.

Whole-heart models use the Huygens' Principle of wave propagation to induce heart excitation and produce artifacts that are inherently synthetic measurements in the form of calculated surface ECG. Additionally, these models can simulate cardiac arrhythmia by using differing knowledge-based heart excitation sequence details (Wei, 1997). Whole-heart models require the use of bidomain representation of the cardiac tissue (Trayanova, 2011).

Monodomain / Bidomain

Bidomain is a mathematical models described by two coupled partial differential equations (PDE) that simulate the electrical properties of heart cell membranes and their kinetics. A classical bidomain uses the current flow of both intracellular and extracellular domain potentials to predict ECG. Different models have been developed from the classical bidomain including a 3D computer

model in (Harumi et al., 1989) which is based on simulation of the sequence of depolarization and on multiple action potentials taken in the appropriate time phase.

2.3.2 Data Driven Synthesizers

Data driven synthesizers leverage real-world patient data to generate similar synthetic patient measurements. In healthcare, biological systems are not completely understood therefore, one way of synthesizing healthcare data is by using stochastic methods. Stochastic models have been successfully used to model the characteristics of ECG wave forms.

These generative models use the following methods (i) fit an unknown distribution to real-world data, (ii) use domain knowledge of an underlying system to select and fit a known distribution, and (iii) use algorithms without making any assumptions about the data or the system that produced the data and learn the distribution present in the data. In the following sections we review and summarize the various peer-reviewed and commercial methods of generating different formats of synthetic measurements in healthcare.

Evolutionary Optimization Synthesizers

Evolutionary optimization algorithms are population-based searching heuristics that were first proposed by (Holland, 1975) and inspired by the abstraction of evolution in the theory of natural selection by Charles Darwin. Healthcare time-series synthesizers that are based on evolutionary optimization such as the genetic algorithm have been proposed in literature. We will review and discuss how evolutionary optimizers have been used to generate synthetic measurements in healthcare.

In the work of (Shamsuddin et al., 2018), the Virtual Patient Model framework implements the genetic algorithm to generate one-dimensional (1D) healthcare time-series data from statistical

features in real-world patient data. The optimization solver in (Shamsuddin et al., 2018), uses attributes of traditional GA such that the time series sequences are the chromosomes, and the algorithm operates on a selected initial random population. Here, the initial random population is sampled from random sine and cosine wave forms and once the initial population is defined, the optimizer is invoked, and the algorithm iterates through tournament selection, 100% crossover, and 5% mutation operations and a fitness function evaluation up to a pre-defined maximum iteration value.

The Guided Evolutionary Synthesizer proposed in (Maweu[2] et al., 2021) is a multi-dimensional healthcare time-series generator inspired by the theory of evolution. The synthesizer accepts real patient data as input and template for generating the synthetic data but does not account for preserving privacy in publicly available healthcare time-series datasets. GES approximates correlation between multi-dimensional template sequences and uses this approximation to initialize the generation process. The evolution concept is introduced when the algorithm alternates between an exploration and a guidance phase that evolves an individual data sample to a fit synthetic solution. Variations are induced in the synthetic sample through search space exploration constrained by the initializing correlation estimation and concept maps (trend, shape) to achieve convergence.

Statistical Matching Synthesizers

Statistical matching involves creating models that use statistical information from variables in real-world data. When designing these models, features in real-world data are statistically analyzed, extracted and in some cases combined to feature engineer (Kramer et al., 2001) inputs for the synthesis process.

The Synthetic Data Vault (SDV) proposed by (Patki et al., 2016) is a generative statistical model for relational databases that computes distributions and covariance of database objects. Their method uses model-based synthesis where numerical values and rows in a database table can be sampled, and database synthesis which generates an entire database. The SDV synthetic measurements are validated for predictive accuracy and subjective qualitative findings. SDV has been extended as an open-source synthetic data library that has incorporated *Single Table* modeling of tabular data in two generative models, CTGAN and TVAE as proposed by (Xu et al., 2019).

Another statistical matching synthesizer uses a linear programming solver proposed in (Bogle et al., 2016) and it generates synthetic healthcare data. The solver achieves statistical moments, similar variables, and data types of real-world data. The synthesizer is recorded as a macro function in the SAS© system and uses a sequence of events to specify the statistical moment order and parameters that control the synthesis algorithm. The input events enable the synthesizer to change the output size, determine the compute time and the quality of the generated synthetic measurements.

When validating the synthetic measurements, (Bogle et al., 2016) uses statistical analysis to compare the mean and standard deviation between the real-world and synthetic measurements. Data set variables interactions are also validated by computing covariance matrices and checking for similarities/dissimilarities in covariance magnitude. The final step in (Bogle et al., 2016) validation process is separately fitting a logistic regression model to real and synthetic training data, apply the trained models to real test data and evaluate classification performance, sensitivity, and specificity.

In the commercial space, MDClone (MDClone Ltd., Beer Sheva - Israel) and Synthea are big-data engine for generating patient data that is statistically similar to the real data and community patient populations. The MDClone engine is instantiated using real patient data obtained through collaboration efforts with hospitals and other patient data stores. The MDClone synthesis model estimates a kernel, fits a distribution, and samples synthetic patient data from the model. The model offers user options as queries which allow feature and attribute selection, and custom mathematical computations that implement desired de-identification of synthetic patient data.

Synthea is another commercially used open-source synthetic patient profile simulator proposed in (Walonoski et al., 2018). Synthea relies on publicly available datasets to generate synthetic Electronic Health Records (EHR). Implementation of Synthea focuses on simulating 10 most frequent reason for medical care and 10 chronic diseases with the highest morbidity in communities within the state of Massachusetts. The Synthea model simulation produces virtual patient disease progressions and treatment plans using two machine states, (i) a control state for module flow and (ii) clinical state for attributes i.e., symptoms and medication. The synthetic profiles from the Synthea model are evaluated against real-world patient profiles by comparing patient populations and the respective levels of statistical properties together with their probability distributions.

Deep Learning Synthesizers

We review generative models that use deep learning architectures such as convolutional neural networks (CNN) and recurrent neural networks (RNN). SenseGen is a generative model proposed by (Alzantot et al., 2017) which synthesizes sensory data. The promising results in

generative models as shown in SenseGen are a consequence of recent advancement in deep learning architectures. The SenseGen synthesizer is designed using Long-Short-Term Memory (LSTM) network architecture.

The LSTM network is a type of RNN that uses feedback connections. LSTM have hidden layers which contain hidden cells. Each hidden cell in an LSTM layer comprises of multiple hidden units which are composed of a cell, and memory gates that control the internal flow of information. SenseGen synthetic sensory measurements are validated using a discriminator model that is trained to distinguish between the real and synthetic measurements. The prediction performance of this discriminator model are used as the quality check for the synthetic measurements.

Generative Adversarial Network Synthesizers

Generative Adversarial Networks (GAN) have demonstrated huge success in generating synthetic images. The GAN architecture was first proposed by (Goodfellow et al., 2014) and the network consists of two separate competing (adversarial) neural networks that learn from each other. The two neural networks in the GAN model are a generator model and a discriminator model.

The generator model creates synthetic data which is evaluated by the discriminator model as either real or fake. The evaluation of synthetic data by the discriminator helps the generator create better and more realistic samples. As the generator improves, so does the discriminator which learns to better distinguish real and fake samples. We review how GAN models have been used successfully to synthesize medical images as proposed the works of (Xu et al., 2019), (Choi et al., 2017), (Torkzadehmahani et al., 2019), (Beers et al., 2018), (Nie et al., 2017), and (Guibas et al., 2017).

The CTGAN proposed by (Xu et al., 2019) which is also implemented as part of the SDV library, is a conditional GAN that uses the packing framework in (Lin et al., 2020). CTGAN mainly handles multi-modal distributions characteristics in tabular datasets. The TVAE model also by (Xu et al., 2019) implements two neural networks that ease the characteristic mixed data-types present in a single tabular dataset. The CTGAN and TVAE models learn tabular datasets and generate synthetic versions that match the structure and statistical properties of the real-world dataset. CTGAN and TVAE synthesized datasets are evaluated on the following analysis (i) statistical, (ii) likelihood fitness, (iii) machine learning discriminator, (iv) machine learning efficacy, and (v) adversarial attack.

Both CTGAN and TVAE models have been used to synthesize the same longitudinal EHR datasets in medGAN proposed by (Choi et al., 2017). EHR datasets are raw or aggregated digital versions of manual treatment entries in patient medical charts and contain a wide range of patient data types including medical imaging, pharmaceutical, time sequence records (medical history, vital signs, demographics), and discrete laboratory measurements.

In medGAN (Choi et al., 2017) combined a GAN and a Variational AutoEncoder (VAE) with mini-batch averaging to synthesize discrete multi-label high dimension EHRs. The medGAN synthesis is focused on PAMF, MIMIC II (Intensive Care Unit) and Sutter heart failure EHR datasets. Statistical, machine learning efficacy and expert review are the evaluations methods used to validate synthesized medGAN samples.

The goal of DP-CGAN (Differentially Private Synthetic Data and Label Generation) in (Torkzadehmahani et al., 2019) is to generate synthetic images and their corresponding label while preserving the privacy of the training data. (Torkzadehmahani et al., 2019) proposes use of a

privacy budget with a Renyi differential privacy accountant. The DP-CGAN approach illustrates differential privacy as replacing a specific individual with a random individual from the population.

In the context of individual replacement, the model should learn the same thing about the data in presence or absence of the replaced individual. (Torkzadehmahani et al., 2019) use gradient clipping and privatize model training by injecting random Gaussian noise in the discriminator optimization. The privacy budget is monitored and kept below a preset target throughout the training. Results from this approach are on the MNIST dataset which is used to train the DP-CGAN with 60k real samples and labels and it generates another 60k synthetic samples and labels. Synthetic image validation uses classification performances of logistic regression and multi-layer perceptron classifiers.

The focus of the PCGAN (Progressively Growing GAN) training methodology in (Beers et al., 2018) is to produce high-resolution synthesized biomedical images. Unlike traditional GAN training, (Beers et al., 2018) proposes to grow the generator and discriminator simultaneously by up-sampling and adding a convolution layer to each side for each phase of training. This progressively growing GAN architecture and training method produces significantly superior images as shown in (Beers et al., 2018). In their work, (Beers et al., 2018) present the results of eye fundus images for retinopathy and multi-modal glioma MRI. The synthesized images are validated for quality using a state-of-the-art vessel segmentation model and report an AUC of 97% on the generated data. To further bolster the results, the (Beers et al., 2018) demonstrate and confirm using the nearest-neighbors method that PGGAN generate vessel trees that lie outside the original training set.

3D images present healthcare providers better image resolution and more visualization angles of the target scan location. For patients, 3D images reduce duration of exposure to radiology imaging by producing fewer and more detailed images. (Nie et al., 2017) proposes a context-aware GAN that generates 3D computerized tomography scans from collected MRI images. The use of adversarial training strategy and a specialized image gradient difference loss function ensures that the model produces realistic images.

This context-aware GAN works on 3D volumetric patches using fully connected networks. During testing, the MRI patches of dimensions 32 × 32 × 32 are used to generate CT patches of dimensions 16×16×16. The predicted patches are merged by averaging over the intensities of the overlapping regions. An AutoContext model is used and allows leveraging probability map of previous classifier iteratively as an added context. This approach improves the receptive field of the model beyond the current patch. The synthetic 3D image scans are validated in terms of predication accuracy and (Nie et al., 2017) present results from both brain and pelvic images. Their results show significant improvement in model performance over traditional methods especially when AutoContext models are used.

(Guibas et al., 2017) proposes the DualGAN, a two-part pipeline with individual focus on geometric structure and realistic image generation. The DualGAN task is split into two individual GANs which show improved performance in generation of photorealistic vessel tree segmentation and corresponding synthesized retinal fundus images. The first stage of the proposed pipeline generates vessel tree segmentation masks using DCGAN (Deep Convolutional GAN).

The DCGAN uses convolution layers entirely without any pooling layers which are known for loss of spatial information. In the second stage of the pipeline, a CGAN (Conditional GAN) is

used to generate the photorealistic retina fundus images. During the training process, the CGAN accepts as input corresponding segmentation masks and retina fundus images. The CGAN then performs style transfer which outputs the photorealistic retina fundus image on the vessel tree segmentation input.

The synthesized retina fundus images are validated both qualitatively using side by side images of the real and synthesized images, and quantitatively by training a segmentation network with the synthesized images and testing the classification accuracy with real images. Additionally, (Guibas et al., 2017) computed the variance in the real and synthesized images using the Kullback-Leibler divergence score.

2.4 Summary

This study provides a literature review of synthetic generators and measurements in healthcare. The review demonstrates how advances in computer vision methods have influenced medical image synthesis and analysis processes. Synthesized images are being used to innovate and improve systems that require large volumes of inaccessible real patient data in medical specialized areas such as cardiology, endocrinology, pulmonary, neurology, orthopedic, dentistry, and dermatology.

Domain expert knowledge together with data collected from these medical areas continue to advance healthcare time-series and medical image synthesis as shown in Table 2.1. Realistic synthetic images in 2D and 3D formats are being synthesized from patient x-rays, magnetic resonance imaging (MRI), and computerized tomography (CT) scans.

The summary in Table 2.1, shows different computational approaches like mathematical modeling, heuristic optimization, stochastic process, and deep learning that highlight distinct features which uniquely enhance the synthesis process.

Table 2.1. Summary of Synthetic Data Generators in Healthcare

Synthesizers	References	Approach	Features
Process Driven	(Wei, 1997)	Whole-Heart Modeling	Anatomic modeling
	(Harumi et al., 1989)	Monodomain / Bidomain	Continuum model
Data Driven	(Shamsuddin et al., 2018)	Evolutionary Algorithms	1D Time-series data
	(Maweu[2] et al., 2021)		1D, 2D time-series data
	(Bogle et al., 2016)	Statistical Matching	EHR - Model central moments
	(Walonoski et al., 2018).		EHR - Simulate patient population
	(Alzantot et al., 2017)	Deep Learning	Sensory data
	(Torkzadehmahani et al., 2019)	Generative Adversarial Networks	Differential Privacy
	(Beers et al., 2018)		High resolution images
	(Nie et al., 2017)		Gradient loss and AutoContext modeling
	(Guibas et al., 2017)		Geometric structure modeling
	(Xu et al., 2019)		Mult-imodal distributions
	(Xu et al., 2019)	Variational AutoEncoder	Mixed-type datasets
	(Choi et al., 2020)		Discrete variables with mini-batch averaging

Furthermore, in Table 2.2, we show a representation the healthcare data synthesis approaches in terms of the different healthcare imaging, time-series and tabular data types. We observe that synthetic generators have successful in generating statistically, expert valid images physiological signals, and EHR datasets.

With images the main focus is to ensure that medical images are photo realistic and preserve or enhance image resolutions. Synthesis of physiological signals such as ECG and EEG ensures visually valid graphical representations and sensor correlation for high dimensional datasets. EHRs are collected during interactions with patients that seek medical care. EHR

datasets have unique challenges like mixed-type data types in a single patient record, unknown distributions, and missing values and the synthesis approaches in Table 2.2 aid in overcoming these challenges.

Table 2.2. Summary of Synthesized Healthcare Data-Types

Data-Type	Data Source	Real-World Data Collection Methods	Synthetic Generators	References
Images	Brain (Alzheimer)	MRI	Context-Aware GAN	(Nie et al., 2017)
	Brain (Glioma)	MRI	PCGAN	(Beers et al., 2018)
	Pelvic	MRI	Context-Aware GAN	(Nie et al., 2017)
	Diabetic Retinopathy	Fundus Camera	DCGAN	(Guibas et al., 2017)
	Retinopathy of Prematurity	Fundus Camera	PCGAN	(Beers et al., 2018)
Time-series	Heart	Electrocardiogram	GES	(Maweu[2] et al., 2021)
	Brain	Electroencephalogram	GA GES	(Shamsuddin et al., 2018) (Maweu[2] et al., 2021)
	Human Activity	Accelerometer	GA	(Shamsuddin et al., 2018)
Tabular / EHR	Longitudinal Health Records	Manual/Digital Collection	medGAN	(Choi et al., 2020) (Xu et al., 2019)
	Intensive Care Unit EHRs	Manual/Digital Collection	CTGAN	(Choi et al., 2020) (Xu et al., 2019)
	Heart Failure EHRs	Manual/Digital Collection	TVAE	(Choi et al., 2020) (Xu et al., 2019)

Beyond the synthesis process, this review provides insights into how data access challenges in healthcare motivate research in synthetic generators. Synthetic datasets are products of computer models not real patients therefore, healthcare data challenges are minimized and as such giving researchers unrestricted data access to leverage state-of-the-art machine learning algorithms for solving healthcare tasks.

Finally, this review shows that in most cases only user defined or task specific schemes are used during synthetic data validation. In literature, a disproportionate number of proposed synthetic data generation methods do not follow a streamlined or generally accepted framework

for validating healthcare synthetic outputs. Therefore, in Chapter 4, we present a foundational framework for validating synthetic data that employs (i) statistical analysis, (ii) visual validation, (iii) expert validation, and (iv) machine learning validation methods.

CHAPTER 3

CEFES: A CNN EXPLANABLE FRAMEWORK FOR ECG SIGNALS[1]

Healthcare systems built on artificial intelligence technologies have shown tremendous success in solving complex domain problems. Most of these AI systems are used as a support tool for healthcare providers during the delivery of patient care. The basic understanding of the internal mechanisms of these systems has eluded most users and developers. Knowledge about how AI systems solve problems is important to healthcare stakeholders who are wholly accountable for the decisions made by these systems. Stakeholders seek to gain trust and confidence (Miotto et al., 2018) when using AI in life-critical circumstances but the challenge is that AI systems remain opaque. AI decision support systems seemingly perform remarkably well and in some instances outperform human experts solving equivalent problems.

The performance measures attributed to AI systems are traditionally in terms of model accuracy, precision, recall, and non-functional measures such as speed and ease of use. These traditional performance metrics are limited in providing meaningful information which can be used to clearly and precisely address the "what", "how" and "why" questions about the inner workings of AI systems. Accuracy, precision, recall, area under the curve (AUC) inform about the degree to which a model accurately classifies its input. To answer these questions, researchers are investigating new and novel methods that consider the importance of interpretable and explainable models in the safety-critical healthcare domain.

---

[1] ©2021 AIIM. Portion reused, with permission from B.M. Maweu, S. Dakshit, R. Shamsuddin, and B. Prabhakaran "CEFEs: A CNN Explainable Framework for ECG Signals," in 2021 Artificial Intelligence in Medicine, 115, 102059.

Researchers have developed several methods of getting interpretable explanations from AI systems. Firstly, one method includes building self-explanatory models, which are models that integrate interpretability in their design and provide global or localized model explanations. Another method uses post hoc model analysis that evaluates model performance through input feature analysis. Input feature analysis explanations are derived from statistical and visual computations that assign feature importance scores to model inputs. In this work, we introduce a CNN Explainable Framework for ECG Signals (CEFEs) which provides post hoc analysis and interpretable explanations for CNN models trained on 1D ECG time-series datasets.

Using CEFEs we achieve significant transparency and functional understanding of how AI-based decision support systems map ECG input signals to an arrhythmia classification. CEFEs analysis flow provides interpretable explanations about non-linear transformation and layer-wise representations learned by a CNN model. Additionally, we use the CEFEs' interpretable explanations to justify model capacity by investigating three model performance conditions, model improvement, model degradation, and model no-change situation. In the context of CEFEs, model capacity is the ability of a model to correctly classify a range of model input cases.

The rigid structural characteristics of ECG signals, the small size of healthcare datasets (Shaikhina et al., 2017), the importance in the knowledge of how deep neural networks make decisions (Miotto et al., 2018), and limited research on knowledge encoded in 1D model inputs, motivated our choice of ECG signals for CEFEs implementation. In totality, the CEFEs framework achieves interpretable explanations through a functional understanding of the internal mechanism of CNN models trained on ECG signals thus addressing the trust gap found in these black-box systems.

3.1 Related Work

The terms interpretations and explanations are often used interchangeably in the context of computational modeling. We adopt the definition of these two terms from (Montavon et al., 2018) whereby *interpretation* is the idea of mapping from feature space (for example a predicted class) into a human comprehendible space and e*xplanations* is a set of features in the interpretable space that contribute towards class discrimination. To understand research trends in the area of explainable AI (XAI), we review research literature focused on interpretable models and on post-hoc model explanations.

3.1.1 Interpretable Models Explanations

Interpretable models integrate design features in the internal mechanism making it possible to extract interpretable explanations. Most research on interpretable models is concentrated on image data with limited work in other data types including time series. An automated method in (Zhang et al., 2018) is used to maps higher level CNN filters to an object-part (CNN semantics) rather than to the traditional image data patterns. This mapping technique is achieved by applying modification to the components of black-box deep learning models thus revealing model interpretable representations.

In (Strum et al., 2016) a score is assigned to inputs. The Layer wise Relevance Propagation (LRP) method analyzes healthcare time series data (EEG) and translates model decisions into heat maps that explain the relevance of each data point with respect to that decision. Unlike CEFEs, the decision explanations derived from (Strum et al., 2016) do not provide knowledge on temporal-spatial features, patterns and morphological properties of time series that are learned by a model.

3.1.2 Post-hoc Model Explanations

These explanations provide functional understanding of a trained model's internal mechanism in relation to the model inputs. CEFEs is a post-hoc model explanations framework. An important feature of model explanations discussed in (Ribeiro et al., 2016 and Lundberg et al., 2017) and shown in CEFEs is the flexibility of use in different deep learning models. Several post-hoc explanations methods have been proposed, including shapelet extraction (Ye et al., 2010), backward propagation (Simonyan, et al., 2014), data perturbation and Local Interpretable Model-agnostic Explanations (LIME) in (Ribeiro et al., 2016), shapely additive explanations (SHAP) (Lundberg et al., 2017), activation maximization (Montavon et al., 2018), Testing with Concept Activation Vectors (TCAV) in (Kim et al., 2018) and, visual explanations (Selvaraju et al., 2019).

Explanations especially for time series data are proposed in CEFEs and in (Ye et al., 2010 and Karlsson et al., 2018). The use of extracted time series subsequences known as shapelets to explain and discover the best representative pattern in time series target classes is proposed in (Ye et al., 2010). Time series tweaking is a method proposed in (Karlsson et al., 2018) that unlike CEFEs is not applied to deep networks models. In time series tweaking, the minimum number of changes needed in order to change an input classification outcome is computed for random forest type of classifier.

(Lundberg et al., 2017) describes a method that determines how input features contribute towards model outcomes. Similarly, (Sundararajan et al., 2017) improves on feature scoring explanation techniques by identifying two axioms (sensitivity and implementation invariance) that need to be satisfied to accurately attribute model inputs to task outcomes. LIME and SHAP are model-agnostic explanation methods. In (Ribeiro et al., 2016), LIME finds model behavior that is

local to the input being considered. This is achieved by perturbing the inputs around the neighborhood of a sample and then determining the behavior of the model. Similarly, SHAP (Lundberg et al., 2017) also computes local explanations but unlike LIME, it uses Shapely values found in game theory to explain how input features contribute to an outcome. Another feature importance based explanation is Concept Activation Vectors (CAV) by (Kim et al., 2018). This method quantifies the importance of input concepts in relation to model outcomes using directional derivatives. The goal of quantifying input feature importance is to subsequently draw attention to specific areas of an input that contain positive weight for a specific class.

(Montavon et al., 2018) describes activation maximization as a framework that searches for input patterns that maximize model response. A technique that employs activation maximization is Gradient-weighted Class Activation Mapping (Grad-CAM) proposed in (Selvaraju et al., 2019). Grad-Cam is a visualization based explanations method for CNN models that uses activation maximization to tags the discriminative parts of each input class. The tagged regions are then highlighted and output as heat maps after computing class gradients of the input in the last CNN layer.

Based on the research literature, few interpretation and explanations methods have been proposed for healthcare time series. The standard model performance metrics in deep learning classification tasks (accuracy, sensitivity, and selectivity) are insufficient in providing healthcare domain users with details of what features are learned by the model, which learned features contribute to model outcomes and whether the machine learned features can be mapped to the actual clinical features used in medical diagnosis. The ability to provide healthcare providers with human interpretable details from machine learning models foments trust, confidence and ease in

adoption of AI based decision support systems. The CEFEs framework proposed in this work describes how interpretable explanations are derived from CNN models trained on ECG time series data, and how this explanations can provide insights into model capacity.

3.2 CEFEs Framework

The goal of CEFEs is to provide transparency and functional understanding of the internal mechanism of CNN models trained on highly structure healthcare time series data by using a layer-wise method of interpreting relevant features learned by such a model. The proposed end-to-end framework (Figure 3.1) together with the detailed inset of the explanations module (Figure 3.2) is a post-hoc tri-modular evaluation configuration that produces local interpretations and explanations from CNN layers. Local interpretations and explanations of a model explain the "why" of the prediction of an input instance.

CEFEs modules provide users with the functional understanding of the CNN models in terms of data descriptive statistics, feature visualization, feature detection, and feature mapping. We identified three possible evaluation paradigms for CEFE: (a) model trained on different volumes of the same dataset; (b) different models trained on the same dataset; and (c) different models trained on different datasets.

We present CEFEs modular operations and detail of how the framework achieves interpretable explanations artifacts.

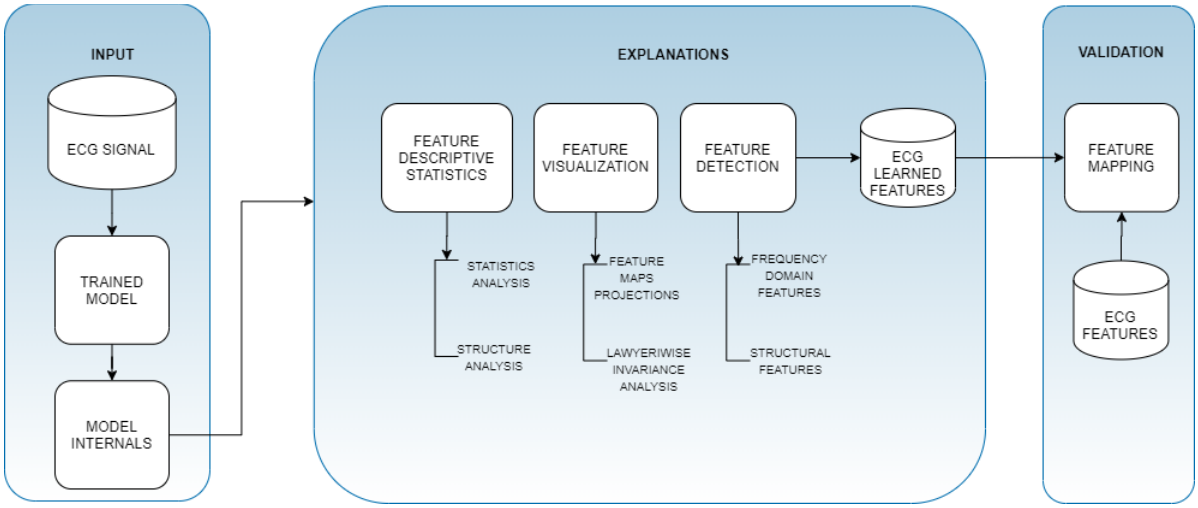


Figure 3.1. CEFEs: CNN Framework for ECG Signals in (Maweu[1] et al., 2021)

3.2.1 Input Module

CEFEs input module guides the flow that evaluates ECG data contained in a repository of real-world test data with a trained ECG classification model. Once a test data sample is evaluated on the model under consideration, the layer-wise model internals in form of learned feature maps are extracted and are subsequent inputs for the Explanation Module.

3.2.2 Explanations Module

Descriptive Statistics: These are summary statistical analysis representative of input data or machine learned features. This component uses task dependent statistical measures to analyze an input instance of ECG signal and the corresponding feature map extracted from a convolution layer of a trained CNN model. Statistical analysis is not limited to a specific convolution layer therefore a user has the choice of the layer under consideration. In CEFEs empirical study, we use the last convolution layer ($Conv_{final}$) because it combines both low and high level machine learned features

thus balancing spatial and semantics information that contribute to class discriminative component artifacts.

The descriptive statistics components computes data and task dependent statistical measures therefore we analyzed input ECG signals and learned features using the Dynamic Time Warping (DTW) algorithm. DTW computes alignment similarities between the input ECG signal and layer-wise extracted feature sequences. DTW is an effective measure for analyzing and to compare both visually and with distance metrics, the learned representation present in the highly structured ECG waveforms.

For better analysis, we organize computed DTW distance measures into *intra-model* and *inter-model* as shown in Equation 3.1 and Equation 3.2 respectively. The intra-model distance (d_{intra}) is the warped Euclidean distance between an instance of real ECG input and feature map projections. The value d_{intra} represents how well a model has learned ECG shape features and how well the learned features align to real ECG waveforms. Therefore, low d_{intra} values indicate that a model has adequately learned ECG shape features.

When d_{intra} values are computed from more than one model, we consider the value difference between models as inter-model distance (d_{inter}). The d_{inter} values is then used as a comparative similarity/dissimilarity measure of ECG shape features learned between two separate CNN models. High d_{inter} values effectively provide explanations of the differences in prediction outcomes between two models. DTW values provide interpretable explanations regarding a model's capacity to learn shape features and subsequently the inherent statistical and mechanical features of ECG signals. By approximating d_{inter} and d_{intra} we can better understand possible threshold values useful in explain model outcomes.

$$d_{intra} = \sqrt{\sum_{k=1}^{K}(x_{k,m} - y_{k,n}) * (x_{k,m} - y_{k,n})} \quad (3.1)$$

Where k represents the samples, m^{th} data point of one instance of input ECG sequence, n^{th} data point of other input sequence (Feature Map).

$$d_{inter} = |d_{intra}^{M_{y1}} - d_{intra}^{M_{y2}}| \quad (3.2)$$

Inter-model distance variables M_{y1}, M_{y2} represent the two separate models that are under model capacity comparative analysis.

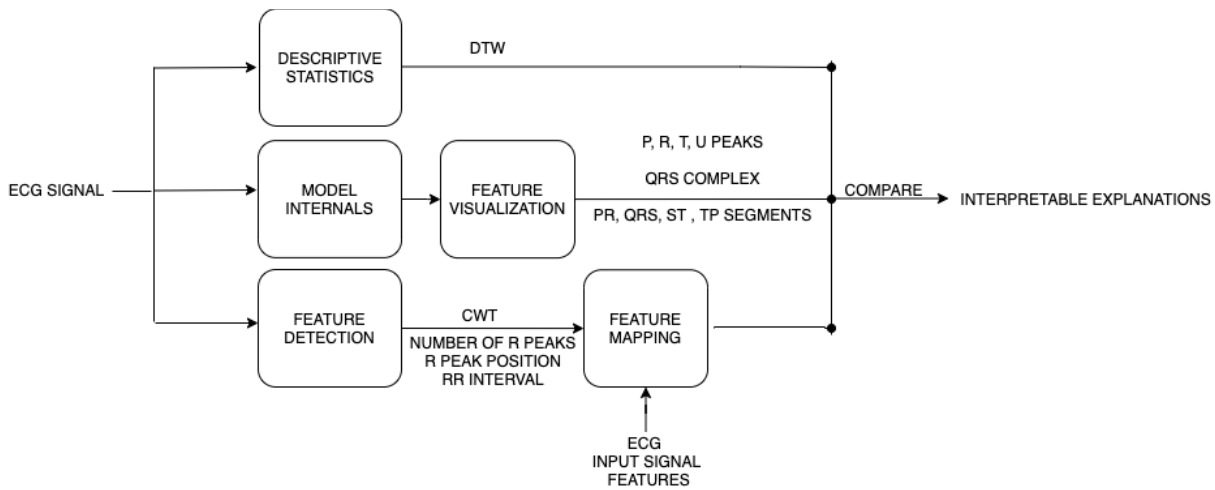


Figure 3.2. CEFEs Explainability Module in (Maweu[1] et al., 2021)

Feature Visualization: ECG signals are characterized by highly structured waveforms, segments and wave interval features. These visually discernable features in there structured form as seen in Figure 3.3 are essential diagnosis different cardiac conditions. Therefore, the ECG morphology is ideal for understanding the layer-wise feature transformations and the overall quality of features learned by black-box models.

This CEFEs component uses visualization techniques that accept an instance of real ECG signal and resultant feature maps as inputs, then produce overlay plots artifacts. Overlay plots are

an effective visual schema that CEFEs uses to evaluate the similarities/dissimilarities of ECG morphology between an input ECG signal and model learned features. These overlay plots are easily interpretable by domain experts because they enable comparative visual validation of machine learned features against the ECG features typically used in cardiac diagnosis (Medical Tests, 2008 and Read ECG, 2010).

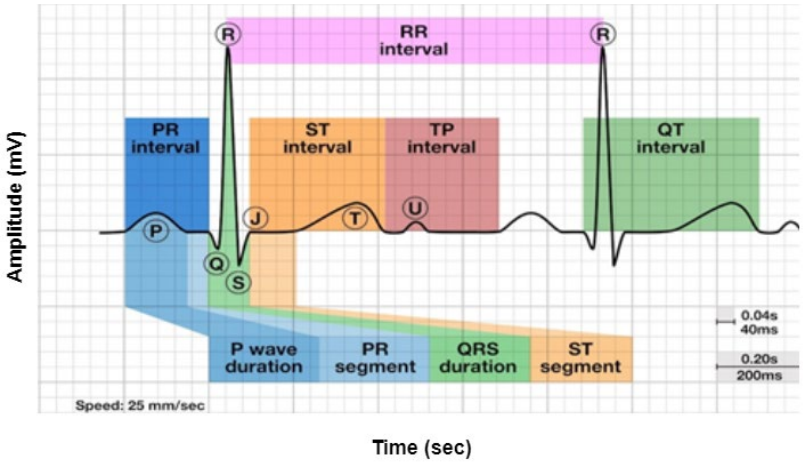


Figure 3.3. ECG Clinical Features in (Maweu[1] et al., 2021)

Feature Detection: CEFEs feature detection component accounts for both time and frequency domain features and applies feature detection algorithms to input ECG and feature maps sequences.

<u>*Time domain feature detection*</u>: The distinct ECG features are detected from these sequences, analyzed for structural similarities. These features including P-Q-R-S-T waveforms are characterized by quantifiable amplitudes, intervals and segments as shown in Figure 3.3. In addition to visual explanations of ECG waveforms, feature detection quantifies features present relative to the two input sequences. For example, given an ECG feature detection algorithms, one can compute the number of P-Q-R-S-T waveforms present in each sequence and then infer the

similarity/dissimilarity of the actual features in the input sequence and those detected in the layer-wise machine learned sequence. To detect R-Peaks in CEFEs experiments the Engzee ECG segmentation algorithm (Engelse et al., 1979) was applied. We used the algorithm to detect and extract ECG features while keeping track of the count, position and interval of R-Peak detected. Our attention was on R-R intervals due to constraints such as CNN layer positional invariant, temporal-spatial dependencies in ECG signals and semantics role of the interval length occurrence.

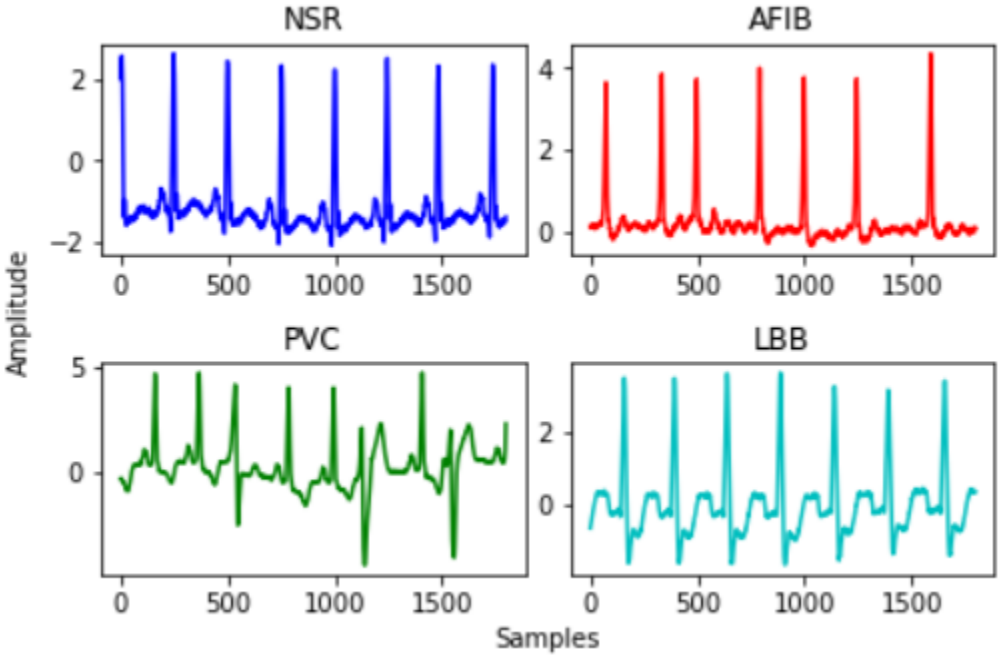


Figure 3.4. ECG Classes: Time domain features

Frequency domain feature detection: CEFEs uses Continuous Wavelet Transform (CWT) to transform input ECG and feature maps into the time-frequency domain for analysis. The goal is to detect frequency bands in the two sequences and CWT has successfully been used to extract such features for ECG feature engineering efforts (Addison, 2005) and (Gautam et al., 2012), by enhance small differences in continuous inputs. With the extracted CWT features of the two

sequences being compared we compute their Mean-Squared-Error (*MSE)* as shown in Figure 3.5. Here we are looking for any localized frequency variation constrained by a time window and any observed differences are quantified. CEFEs accounts for high *MSE* values as an indicator for model capacity and poorly machined learned frequency domain features.

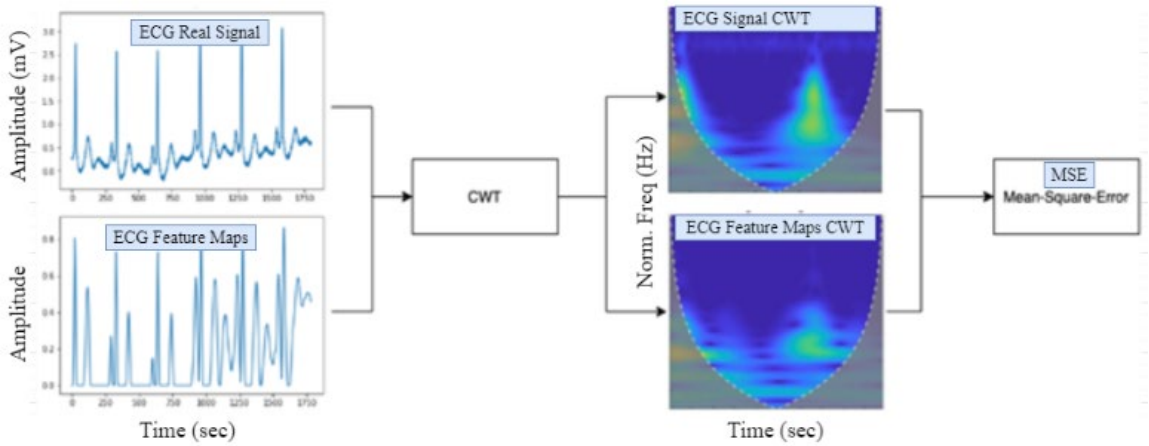


Figure 3.5. CEFEs Mean Squared Error (MSE) computation using continuous wavelet transform (CWT) features. (Yellow in the normalized frequency heat maps is high magnitude

3.2.3 Validation Module

Feature Mapping: The artifacts from the explanations module are used to map features from input ECG and machine learned features using the computed comparison measures i.e., DTW (intra and inter values), overlay plots, R-Peak count, R-Peak position, R-R interval, and MSE. Mapping these features derives interpretable explanations regarding similarities/dissimilarities thus providing insights into model capacity and understanding why and how a CNN model arrives at a prediction.

3.3 Methodology

To achieve an exhaustive evaluation of a CNN model using CEFEs, we identified three possible evaluation paradigms that can be applied:

- *Paradigm 1*: A model trained on different volumes of the same dataset;
- *Paradigm 2*: Different models trained on the same dataset;
- *Paradigm 3*: Different models trained on different datasets.

We note that models identified for Paradigm 2 require the use of different datasets which may introduce inconsistencies in the training process of a CNN model and by extension to the evaluation of the CEFEs model. Models for Paradigm 3 require multiple optimized deep learning architectures trained on datasets of different characteristics and biases, raising concerns of possible inconsistencies that could influence the evaluation of CEFEs.

In our experiments, we evaluate CEFEs on a model trained in accordance with Paradigm 1 and leave evaluations using Paradigm 2 and 3 for future work. We chose to evaluate CEFEs using one model architecture trained on varying quantities of training data. CEFEs explanations were derived from a custom one-dimensional convolution neural network (1D-CNN) model trained on ECG signals to classify 4 ECG rhythms.

Following model training, we performed post-hoc evaluation of the model using CEFEs' modular tests. CEFEs-based explanations derived from our 1D-CNN impart visual, metric, and feature-based analysis. CEFEs provides explanations for model improvement, model degradation, and no-change model in performance. Additionally, these model explanations provide insights into how varying quantity of training data affects model explanations.

3.3.1 Convolutional Neural Network Architecture

CNN models have demonstrated high accuracy in solving classification tasks the areas of computer vision and natural language processing. For that reason, we implemented a custom 14-

layer CNN architecture to train and classify four classes of record-based 1D ECG signals and as the baseline for CEFEs experiments. The architecture comprised of 11 1D-CNN layers with filters of width 18 and varying kernels of sizes 32, 64, 128, or 256 which are grouped with max-pooling, activation, and dropout layers. The last layer is a 4 output dense layer with a *softmax* activation. We selected this 14-layer deep CNN model to accommodate the structural nature of ECG signals to allow adequate feature learning for good classification accuracy.

3.3.2 Dataset Notation and Setup

Dataset: We used the ECG dataset described in Table 5.1 as the real ECG signals and also as seed data for generation synthetic ECG signals using the synthesizer proposed in (Shamsuddin et al., 2018). The four target classed in real ECG signals are the ***y*** input seed data (SD) into the synthesizer.

$$y = X_{SD}{}^{\dim(i)_{L_j}} \tag{3.3}$$

$X_{SD}{}^{\dim(i)}$ is the real 1D ECG signal with L_j number of samples which in these experiments is set to 1800 data samples representing 5-sec long observations. The synthesizer then outputs ***v*** as synthetic data (VPD).

$$v = X_{VPD}{}^{\dim(i)_{L_j}} \tag{3.4}$$

$X_{VPD}{}^{\dim(i)}$ is the synthetic 1D ECG signal with L_j number of same number of data samples and time duration as ***y***. The *dim(i)* is the dimensionality of *i*, where *i* is a data sample. The generation of synthetic data from the real ECG seed signals maintains consistency in quality without introducing unrealistic variations between the real and the synthetic signals.

Notation: We the following defined notation for our CNN trained model *M<param1, param2>* which describes proportion of real ECG signals (*param1*) and synthetic ECG signals (p*aram2*). To create the experimental training sets, we combined the training dataset such that $R_{(i)}$ (Real ECG signals) and $S_{(i)}$ (Synthetic ECG signals) are positive numbers, and elements of set X= {0,1, ...,100}selected in multiples of 20% that represent the proportion of real and synthetic data used to train model *M*, respectively. We define *c* as any positive number > 0 such that $R_{(i+c)}$ and $S_{(i+c)}$ are still elements of X. As an example, *M<R100, S0>* would indicate a CNN model (M) trained on a combination of 100% real ECG signals and 0% synthetic ECG signals.

Training Setup: We trained the custom 14-layer CNN model with different datasets generated by augmenting real ECG signals with synthetic ECGS signals. A model was trained and tested only on real ECG signals and the rest of the models were trained with varying combinations of real and synthetic ECG signals. The synthetic ECG signals were incrementally added to the real ECG signals training set and all models were tested on real ECG signals test set (Table 3.1).

Table 3.1. CNN model dataset setup (train/test) using real/synthetic ECG signals

	Real Train Data	**Uniform Testing Set**	**Real Train Data Augmented with Synthetic Data**				
ECG Class	$M<R_{100},S_0>$	$M<R_{100},S_0>$	$M<R_{100},S_{20}>$	$M<R_{100},S_{40}>$	$M<R_{100},S_{60}>$	$M<R_{100},S_{80}>$	$M<R_{100},S_{100}>$
NSR	199	83	199+40	199+80	199+119	199+159	199+199
AFIB	95	40	95+19	95+38	95+57	95+76	95+95
PVC	94	39	94+19	94+38	94+56	94+75	94+94
LBB	73	30	73+15	73+29	73+44	73+58	73+73
TOTAL	**461**	**192**	**554**	**645**	**737**	**829**	**922**

3.4 Results and Analysis

We conducted comprehensive experiments with the goal of gaining interpretable explanations of the learning behavior of the 1D-CNN layer trained on ECG signals. We exposed our trained

models to CEFEs modular workflow to extract explanation artifacts then analyzed them for interpretable explanations. We analyzed our custom 1D-CNN models that were trained on real ECG signals to classification of four ECG signal classes. We then did case-based classification tests of the signals on these models. Feature maps were then extracted and used as inputs into the CEFEs integrated modules, and we then generated interpretable explanations that showed the layer-wise learning behavior of a CNN model. Results of from one candidate test case obtained from each CEFEs module analysis are summarized and presented.

3.4.1 CNN Layer-Wise Machine Learning Behavior for ECG Signals

Descriptive Statistics

We computed DTW distanced measure d_{intra} and alignment of sequences from the candidate test case and feature maps from the first convolution layer ($Conv_{first}$) and the last convolution layer ($Conv_{final}$) as shown in Figure 3.6. We observed d_{intra} values of 293.6 units when comparing to $Conv_{first}$ and 316.44 units for $Conv_{final}$.

Analysis: The measure d_{intra} shows increased dissimilarities with $Conv_{final}$ and indicates that the model has inadequately learned features present in the candidate test case.

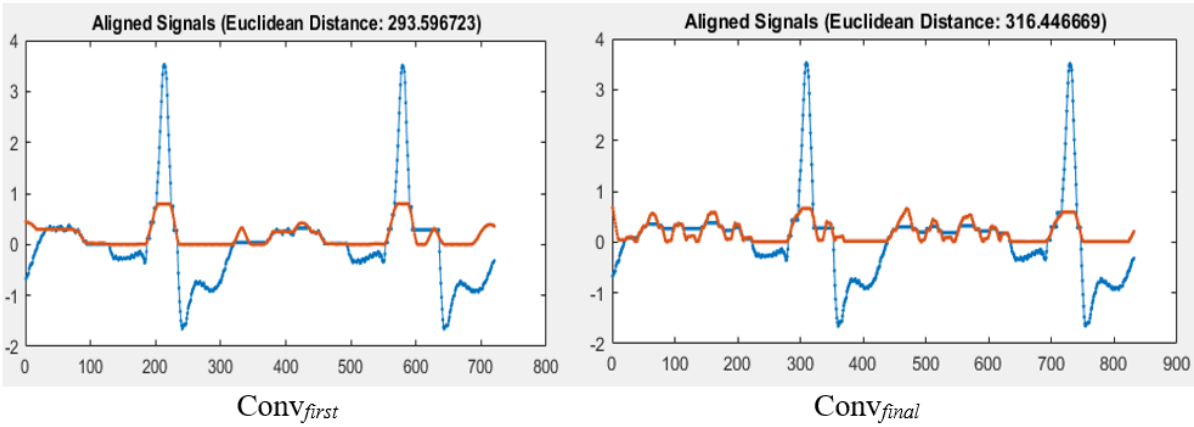
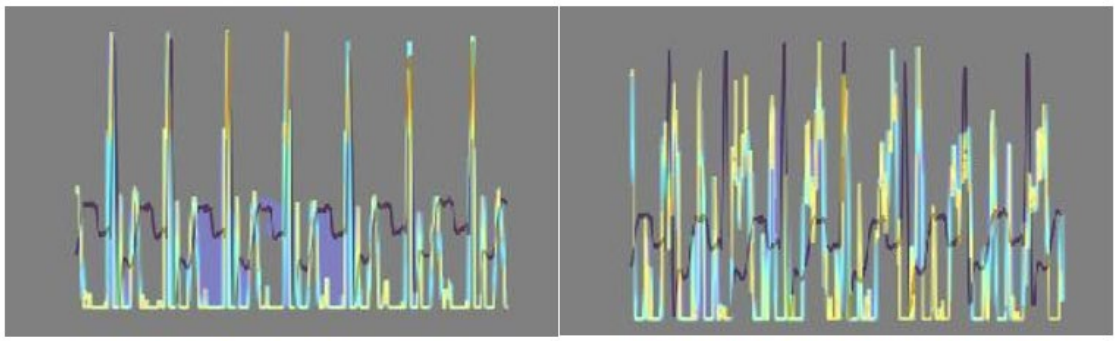


Figure 3.6. DTW alignment (amplitude/time) | Euclidean distance | warping (blue: real | orange: feature map)

Feature Visualization

We analyzed candidate test case and feature map sequences for the highly structure ECG waveform patterns (Figure 3.3) using CEFEs visualization scheme as shown in Figure 3.7 . The overlay plots enable visual and interpretable observations of ECG waveform features learned by the $Conv_{first}$ and $Conv_{final}$, layers of the model respectively. From these visual observations, a domain expert is able to identify learned ECG waveform representations present in the $Conv_{first}$ while $Conv_{final}$.

Observations from Figure 3.7 show the presence of P, R, T waveforms in $Conv_{first}$ while $Conv_{final}$ shows learned features too highly complex for interpretation. The complexity of learned features as seen in $Conv_{final}$ supports the incremental learning and interaction with high-level representations in deeper layers of deep neural networks.



$Conv_{first}$ $Conv_{final}$

Figure 3.7. Feature visualization (amplitude/time)

Feature Detection

We used Engzee ECG segmentation algorithm (Engelse et al., 1979) to detect the number of R-Peaks, R-R interval, and the position of R-Peaks within the ECG input signal and respective $Conv_{first}$ and $Conv_{final}$ feature map sequences. The feature detection algorithm detects 8 R-Peaks in the input candidate test case, 6 and 4 R-Peaks in the two convolution layers respectively. When

evaluating the R-Peak positions and R-R intervals in Table 3.2, we observe in $Conv_{first}$ similar positional properties as the input ECG more so than in $Conv_{final}$. From these observations we see that ECG features (Figure 3.3) are have been learned by the model. The detection of features at both low and high level parts of the model shows that the model has learned a comprehensive set of ECG features in Figure 3.3. Additionally, we performed experiments to detect and compare frequency domain features between the candidate test case and from $Conv_{first}$ and $Conv_{final}$.

Table 3.2. Feature detection | R-Peaks (count & position) | R-R interval (position)

Feature Detection Module Inputs	# of R-Peaks	R-Peak Position	R-R Interval Position
Candidate Test Case	8	[80,158,393,639,888,1140,1392,165	[78,235,246,249,252,252,264
$Conv_{first}$	6	[376,622,871,1124,1375,1638]	[246,249,253,251,263]
$Conv_{final}$	4	[76,157,240,325]	[81,83,85]

The results in Table 3.3 show comparable computed MSE values of 0.046 and 0.049 when the candidate test case is compared with features from $Conv_{first}$ and $Conv_{final}$ respectively. From these computed MSE estimations, we conclude that there is minimal variation in the learned frequency features at the two layers of the model.

Feature Mapping

A complete evaluation of results from the components of the CEFEs explanation module, we see that using the overlay plots we are able to visually map real ECG features to those learned by the model. Moreover, quantifiable measures from DTW Figure 3.7, ECG feature detection Table 3.2 and CWT Table 3.3 provide a basis for interpretable explanations and understanding of the learning behavior of CNN models trained on highly structured 1D signals. The interpretable explanations from these experiments play a role in providing insight to the internal mechanism of deep learning black-box models. The gained transparency creates gains in building trust and

confidence with healthcare providers as they increasingly adopt deep learning based decision support systems.

Table 3.3. Feature mapping | ECG Mean Squared Error of CWT Features (real /feature maps)

MSE **Candidate Tests Case and Conv*first***	**MSE** **Candidate Tests Case and Conv*final***
.046	.049

3.4.2 Interpretable Explanations for Model Capacity

We demonstrate how CEFEs interpretable explanations provide insights into model performance improvement, degradation, and in a no-change situation. With the limited access to healthcare data and the small size of available datasets, we use synthetic ECG signals to train models using the setup in Table 3.1. Synthetic ECG signals give us access to additional data which we incremental addition to the training set thus training several models with varying training data volumes.

These trained models enabled us to evaluate the model performance of different CNN models. The trained CNN models were evaluated by identifying real ECG signal test cases that the models (a) classified correctly; (b) misclassified and (c) test cases where classification did not change even with increased training data. We analyzed the classification performance of the models in Table Classification Results for (i) Performance Improvements, (ii) Performance Degradations, and (iii) No-Difference Situations. We use the CEFEs framework to evaluate a single candidate test case in each category of model performance.

Selecting the Experimental Candidate Test Cases:

The primary motivation of our experiments is to demonstrate CEFEs' ability to understand CNN model capacity for classifiers trained with highly structured ECG signals. Our experimental

design investigates three model behaviors that provide insights into a model capacity, (i) model improvement, (ii) model degradation, and (iii) model no change situation. We propose the use of synthetic data to understand, interpret and explain how CNN models trained with varying volumes of real training that are augmented with synthetic data learn.

We have discussed the problem of small datasets in healthcare therefore, using synthetic data in our experiments provides an opportunity to access more training data and training effective learning models. To best evaluate model performance outcomes, we identified test cases that were (a) correctly classified, (b) misclassified, and (c) test cases whose predictions did not change even with improvement in the model's performance metrics. With our models trained on multiple datasets (real and synthetic ECG signals) as describe in *Paradigm 1* in section 3.3, we analyzed performance of the 1D-CNN model based on whether there were: (i) *performance improvements*, (ii) *performance degradations*, and (iii) *No-Difference Situations*. For each model behavior, we present candidate test cases to demonstrate how CEFEs can contribute towards explaining/interpreting each behavior case.

Performance Improvements Test Case

For understanding model performance improvements, we consider a candidate test case that shows improved classification outcomes when a model is augmented with additional synthetic data. The attributes for the candidate test case for evaluation of model performance are: (a) it is misclassified in model $M < R_{100}, S_{i},>$ and, (b) it is correctly classified by models $M< R_{100}, S_{(i + c)}>$ (i.e., augmented with additional synthetic data).

Performance Degradation Test Case

Model performance degradation considers two candidate test cases that evaluate classification outcomes in twofold:

A. Classification outcomes change from correctly classified when a model is augmented with fewer synthetic data to misclassified when model is augmented with more synthetic data, i.e., a test case is correctly classified by models $M< R_{100},, S_i,>$ but misclassified by models $M< R_{100}, S_{(i+c)}>$.

B. Changes in classification outcomes where a test case is correctly classified when model is trained only on real ECG data and is misclassified in all models augmented with synthetic data, i.e., a test case is correctly classified in models $M< R_{100}, S_0>$ but misclassified in all models $M< R_{100}, S_{(i+c)}>$.

Performance No-Difference Situation Test Case

No-difference in performance situations are candidate test cases where we did not observe any performance change with respect to classification outcomes in all the models evaluated. The attributes for the candidate test case for evaluation of model no-difference situations are: (a) it is misclassified in model $M<R_{100}, S_0>$, and (b) misclassified in all models $M<R_{100}, S_{(i+c)}>$.

Classification Performance from CNN Trained Models

Both overall model performance and class-wise performance for all CNN models $M<R_{100}, S_i>$ are reported in Table 3.4 and Table 3.5 respectively. From Table 3.4, we observed that by augmenting the real ECG training set with synthetic ECG signals model performance increased and model loss decreased. Performance metrics for all the models are from when the models were evaluated on 192 real ECG test set samples shown in Table Training Setup.

Model sensitivity and specificity are important because they indicate to healthcare providers how well the model can classify a true arrhythmia (sensitivity) or rule out the presence of arrhythmia (specificity) in cases of healthy patients. While these traditional metrics show model performance, they fail to reason the model performance and behavior of model capacity. Therefore, CEFEs comprehensive experiments demonstrate how interpretable explanations summarize model improvement, degradation, and no change in model performance.

Table 3.4. CNN models classification performance for datasets incrementally augmented with synthetic ECG signals

Trained Models	**Test Acc. (%)**	**Test Loss (%)**
$M<R_{100}, S_0>$	78.12	0.5606
$M<R_0, S_{100}>$	75.52	0.5294
$M<R_{100}, S_{20}>$	75.52	0.6984
$M<R_{100}, S_{40}>$	79.69	0.5351
$M<R_{100}, S_{60}>$	80.21	0.5508
$M<R_{100}, S_{80}>$	85.94	0.4140
$M<R_{100}, S_{100}>$	91.15	0.3292

The ECG class-wise view in Table 3.5, details additional classification performance metrics; positive predictive value (PPV), negative predictive value (NPV), false-positive rate (FPR), false-negative rate (FNR), selectivity, and specificity. These additional metrics are effective indicators of possible learning bias present in the models and from them, we can surmise a model's ability to reduce false positives (FP) and false negatives (FN) samples.

Prevalence | Positive Predictive Value (PPV): $= \frac{TP}{TP+FP}$ (3.5)

Negative Predictive Value (NPV): $= \frac{TN}{TN+FN}$ (3.6)

False Positive Rate (FPR): $= \frac{FP}{FP+TN}$ (3.7)

False Negative Rate (FNR): $= \frac{FN}{FN+TP}$ (3.8)

Table 3.5. Class-wise classification performance for ECG signals

Model	Sensitivity	Specificity	PPV	NPV	FNR	FPR
M<R$_{100}$, S$_{0}$>						
NSR	71.08	88.99	83.10	80.17	28.12	11.01
AFIB	92.50	96.05	86.05	97.99	7.50	3.95
PVC	69.23	84.97	54.00	91.55	30.77	15.03
LBB	90.00	99.38	96.43	98.17	10.00	0.62
M<R$_{0}$, S$_{100}$>						
NSR	80.72	74.31	70.53	83.51	19.28	25.69
AFIB	82.50	93.42	76.74	95.30	17.50	6.58
PVC	43.59	94.77	68.00	86.83	56.41	5.23
LBB	93.33	99.38	96.55	98.77	6.67	0.62
M<R$_{100}$, S$_{20}$>						
NSR	83.13	78.90	75.00	86.00	16.88	21.10
AFIB	80.00	88.82	65.31	94.41	20.00	11.18
PVC	48.72	97.39	82.61	88.17	51.28	2.61
LBB	83.33	98.15	89.29	96.95	16.67	1.85
M<R$_{100}$, S$_{40}$>						
NSR	84.34	82.57	78.65	87.38	15.66	17.43
AFIB	97.50	91.45	75.00	99.29	2.50	8.55
PVC	41.03	96.73	76.19	86.55	58.97	6.67
LBB	93.33	98.77	93.33	98.77	3.27	1.23
M<R$_{100}$, S$_{60}$>						
NSR	85.54	84.40	80.68	88.46	14.46	15.60
AFIB	95.00	89.47	70.37	98.55	5.00	10.53
PVC	53.85	97.39	84.00	89.22	46.15	2.61
LBB	80.00	99.38	96.00	96.41	20.00	0.62
M<R$_{100}$, S$_{80}$>						
NSR	86.75	86.24	82.76	89.52	13.25	13.76
AFIB	97.50	98.68	95.12	99.34	2.50	1.32
PVC	64.10	94.77	75.76	91.19	35.90	5.23
LBB	96.67	98.77	93.55	99.38	3.33	1.23
M<R$_{100}$, S$_{100}$>						
NSR	98.80	86.24	84.54	98.95	1.20	13.76
AFIB	100.00	99.34	97.56	100.00	0.00	0.66
PVC	61.54	99.35	96.00	91.02	38.46	0.65
LBB	96.67	100.00	100.00	99.39	3.33	0.00

Empirical CEFEs Thresholds

Through an empirical study of CEFEs modular artifacts, we derived threshold values for the modular test DTW and MSE. The threshold values were computed by testing the framework with real ECG signals as follows:

1. Obtain CEFEs modular test values from the 192 real data uniform testing set in Table 3.1.
2. Average the obtained test values from $M<R_{100}, S_0>$ over the whole test set to get a threshold value for each test.

Assumptions:

3. We reasoned that for real test samples correctly classified by model $M<R_{100}, S_{(i+c)}>$, the modular test values would be lower than the threshold value and vice versa for the misclassified samples. To better illustrate our reasoning, we explain finding using d_{intra} as follows: for $M<R_{100}, S_{(i+c)}>$ models that correctly classified a test sample, the d_{intra} value was less than 300 units and greater than 300 units for misclassified test samples.

The same procedure was followed for d_{inter}, and MSE threshold values.

- Threshold values from CEFEs descriptive statistics module were placed on both d_{intra} and d_{inter} values. Test cases with
    - d_{intra} value > 300 units showed evidence of inadequately learned features present in the input ECG signal,
    - d_{inter} values > 50 units showed evidence of differences in classification outcomes between models.
- Threshold values from CEFEs feature detection and mapping module, developed the *MSE* threshold.

- $MSE > 0.100$ was evidence of misclassified test cases and inadequately frequency domain learned features.

CEFEs evaluations hereon use these threshold values to interpret and explain model capacity and quality.

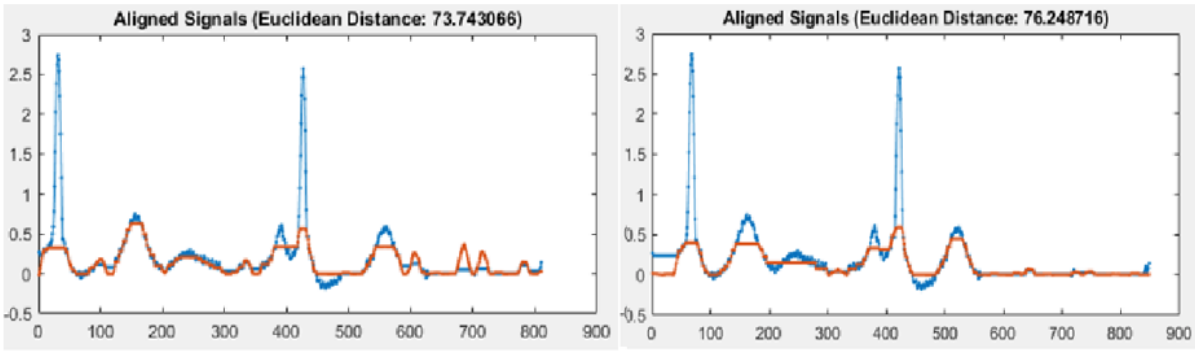
Model Performance Improvements

CNN models $M<R_{100}, S_{60}>$ and $M<R_{100}, S_{80}>$ were evaluated for understanding model performance improvement. The candidate test case was randomly selected from real ECG signals that had classified following the model performance improvement criteria. The input real ECG signal and its feature maps were from class NSR.

***Descriptive Statistics*:** We separately computed DTW values from feature maps in $Conv_{final}$ of models $M<R_{100}, S_{60}>$ and $M<R_{100}, S_{80}>$ warped with the candidate test case as shown in Figure 3.8. The d_{intra} values for model $M<R_{100}, S_{80}>$ were 73.743 units and 76.248 units for model $M<R_{100}, S_{60}>$. The distance between the $Conv_{final}$ the two models was d_{inter} of 2.505 units. The lower order of d_{intra} values represents adequately learned ECG features by both the models and a low d_{inter} shows minimal difference in the ECG learned features between the CNN models $M<R_{100}, S_{60}>$ and $M<R_{100}, S_{80}>$ although the candidate test case resulted in a different classification by each model.

***Feature Visualization*:** CEFEs generated overlay plots for the candidate test case and feature maps from $Conv_{first}$ and $Conv_{final}$ of CNN models $M<R_{100}, S_{60}>$ and $M<R_{100}, S_{80}>$. Figure 3.9 shows distinct P, T peaks, and QRS complex ECG features from $Conv_{first}$ of $M<R_{100}, S_{60}>$ yet the same features in $Conv_{final}$ are highly complex for visual interpretation. We recorded P, T peaks, and QRS

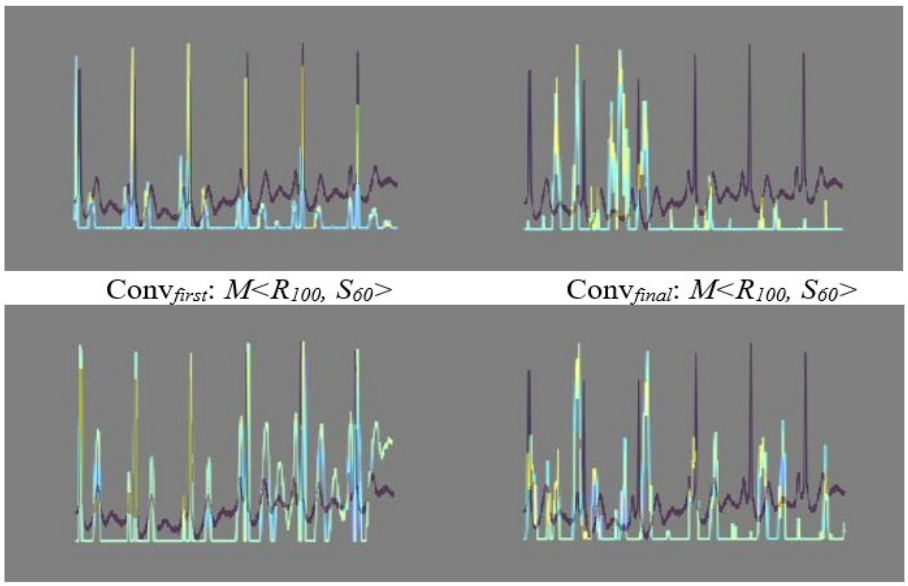
complex ECG features for $Conv_{first}$ in $M<R_{100}, S_{80}>$ the same as previously recorded for model $M<R_{100}, S_{60}>$.



$Conv_{final}$: $M<R_{100}, S_{80}>$ $Conv_{final}$: $M<R_{100}, S_{60}>$

Figure 3.8. Model Improvement: DTW alignment (amplitude/time) | Euclidean distance | warping (blue: real | orange: feature map)

However, the learned features of $M<R_{100}, S_{80}>$ displayed higher position accuracy of learned R-Peaks features in relation to R-Peak positions of the candidate input ECG signal. The $Conv_{final}$ in $M<R_{100}, S_{80}>$ displayed highly complex learned feature set of P-R, S-T, QRS segments and U-Peaks.



$Conv_{first}$: $M<R_{100}, S_{60}>$ $Conv_{final}$: $M<R_{100}, S_{60}>$

$Conv_{first}$: $M<R_{100}, S_{80}>$ $Conv_{final}$: $M<R_{100}, S_{80}>$

Figure 3.9. Model Improvement: Feature visualization (amplitude/time)

***Feature Detection and Mapping*:** In the case of model improvement we see in Table 3.6 that the $M<R_{100}, S_{60}>$ learned similar number of R-Peaks and R-R interval position information as the test case. $M<R_{100}, S_{80}>$ learned comparable features to the real data with one less detected R-Peak and (R-Peak, R-R interval) positions. We observed that $M<R_{100}, S_{80}>$ had R-Peak positions closely aligned to the real data than $M<R_{100}, S_{60}>$ and the MSE values in both models were comparable.

Table 3.6. Model Improvement: Feature detection | R-Peaks (count & position) | R-R interval (position)

Feature Detection Module Inputs	# of R-Peaks	R-Peak Position	R-R Interval Position
Candidate Test Case	5	[298,644,964,1275,1575]	[346, 320, 311, 300]
$M<R_{100}, S_{80}>$	4	[293,641,962,1274]	[348, 321, 312]
$M<R_{100}, S_{60}>$	5	[85,279,628,949,1259]	[194, 349, 321, 310]

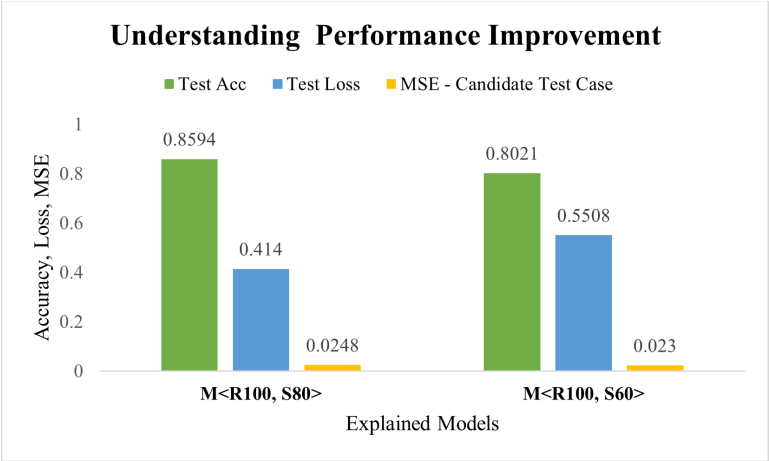


Figure 3.10. Case Model Improvement: performance accuracy (%), loss and MSE

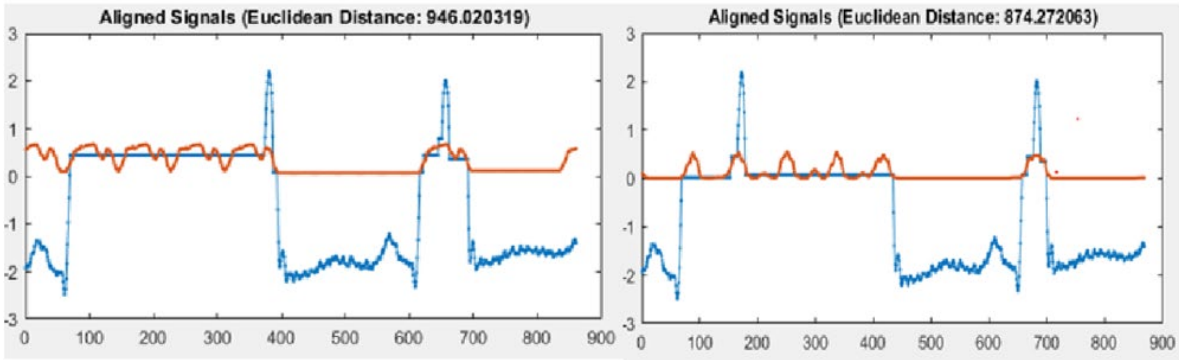
Model Performance Degradation

We approached model performance degradation by using CEFEs modular evaluations separately on two sets of CNN models:

A. Models $M<R_{100}, S_{40}>$ and $M<R_{100}, S_{60}>$ both trained with datasets augmented with synthetic data were used for candidate test case for performance degradation A.

CEFEs evaluations were applied to trained CNN models $M<R_{100}, S_{40}>$ and $M<R_{100}, S_{60}>$ using input ECG signal and feature maps from the NSR target class candidate test case. The test case was correctly classified in $M<R_{100}, S_{40}>$ but was misclassified in $M<R_{100}, S_{60}>$.

Descriptive Statistics: We computed d_{inter} and d_{intra} values for $Conv_{final}$ of each CNN model evaluated (Figure 3.11). DTW alignments of CEFEs module inputs for models $M<R_{100}, S_{40}>$ and $M<R_{100}, S_{60}>$ had d_{intra} values of 946.020 units and 874.27 units, respectively. The higher d_{intra} values represent inadequate learned ECG features in both models while the higher d_{inter} value of 71.75 computed for $M<R_{100}, S_{60}>$ and $M<R_{100}, S_{40}>$ accounts for the variation in classification outcomes between these models. We note that the candidate test case was misclassified by $M<R_{100}, S_{60}>$ although this model resulted in a lower d_{intra} value.



$Conv_{final}$: $M<R_{100}, S_{40}>$ $Conv_{final}$: $M<R_{100}, S_{60}>$

Figure 3.11. Model Degradation (A): DTW alignment (amplitude/time) | Euclidean distance | warping (blue: real | orange: feature map)

***Feature Visualization*:** CEFEs generated overlay plots of $Conv_{first}$ and $Conv_{final}$ and from $M<R_{100}, S_{40}>$ and $M<R_{100}, S_{60}>$. In Figure 3.12 we observed R-Peaks in $M<R_{100}, S_{60}>$ but failed to capture majority ECG features. Compared the previous observation, $M<R_{100}, S_{40}>$ showed more

representative ECG features and this accounts for the correct classification of the candidate test case by $M<R_{100}, S_{40}>$.

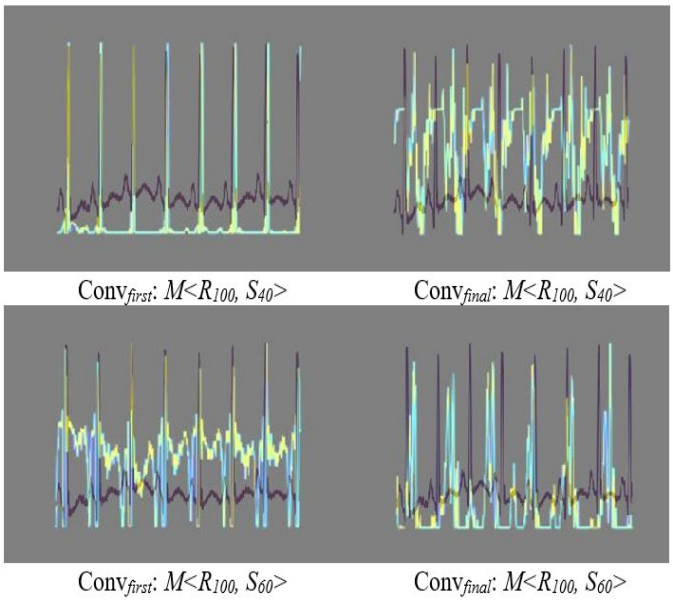


Figure 3.12. Model Degradation (A): Feature visualization (amplitude/time)

***Feature Detection and Mapping*:** Model degradation (A) features in Table 3.7 show that the $M<R_{100}, S_{60}>$ learned comparable number of R-Peaks and R-R interval position information as the test case. Learned features in $M<R_{100}, S_{40}>$ in terms of R-Peak and (R-Peak, R-R interval) positions are less aligned to those present in the test case.

In model performance degradation of CNN models augmented with high quantities of synthetic data, we observed that $M<R_{100}, S_{60}>$ had a lower *MSE* value although it misclassified the candidate test case. $M<R_{100}, S_{40}>$ instead correctly classified the candidate test case with a higher *MSE* value. $M<R_{100}, S_{40}>$ had model accuracy and loss slightly lower than $M<R_{100}, S_{60}>$.

In Figure 3.14 the gradient signal in Conv$_{final}$ for $M<R_{100}, S_{40}>$ shows more intense activation and highlights the discriminative signal region used by the CNN model to correctly

classify the test case. Likewise, the lower loss in Figure 3.13 indicates the distance from the candidate test case in $M<R_{100}, S_{40}>$ was less than in $M<R_{100}, S_{60}>$ and therefore supports the correct classification by $M<R_{100}, S_{40}>$.

Table 3.7. Model Degradation (A): Feature detection | R-Peaks (count & position) | R-R interval (position)

Feature Detection Module Inputs	# of R-Peaks	R-Peak Position	R-R Interval Position
Candidate Test Case	8	[76, 14,558,805,1058,1303,1544,1780]	[238, 244, 247, 253, 245, 241,
$M<R_{100}, S_{60}>$	7	[77, 315, 559, 808, 1059,1304, 1545]	[238, 244, 249, 251, 245, 241]
$M<R_{100}, S_{40}>$	6	[249, 493, 741, 997, 1238, 1479]	[244, 248, 256, 241]

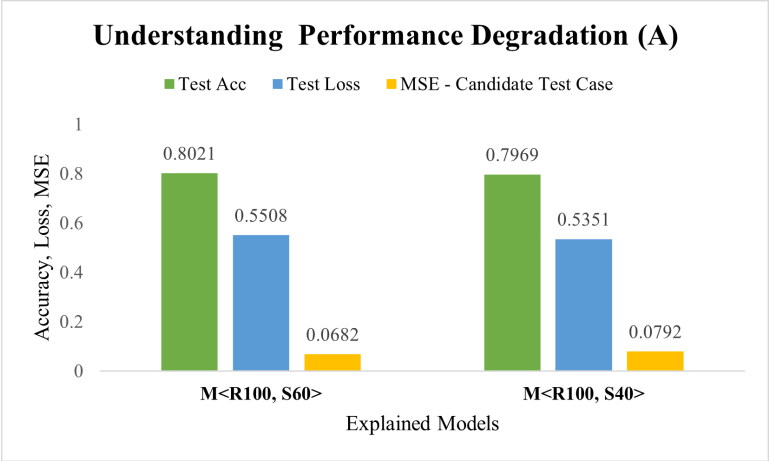


Figure 3.13. Case Model Degradation (A): performance accuracy (%), loss and MSE

B. Models $M<R_{100}, S_0>$ and $M<R_{100}, S_{100}>$ with one model trained on real ECG data only while the other was trained with both real and synthetic data were used for candidate test case for performance degradation B.

Descriptive Statistics: We computed d_{inter} and d_{intra} values for $Conv_{final}$ of each CNN model evaluated (Figure 3.15). DTW alignments of CEFEs module inputs for models $M<R_{100}, S_0>$ and $M<R_{100}, S_{100}>$ had d_{intra} values of 317.32 units and 167.68 units, respectively. We observed a

higher d_{intra} value in $M<R_{100}, S_0>$ which is an indicator of inadequately learned ECG features, yet this model correctly classified the candidate test case.

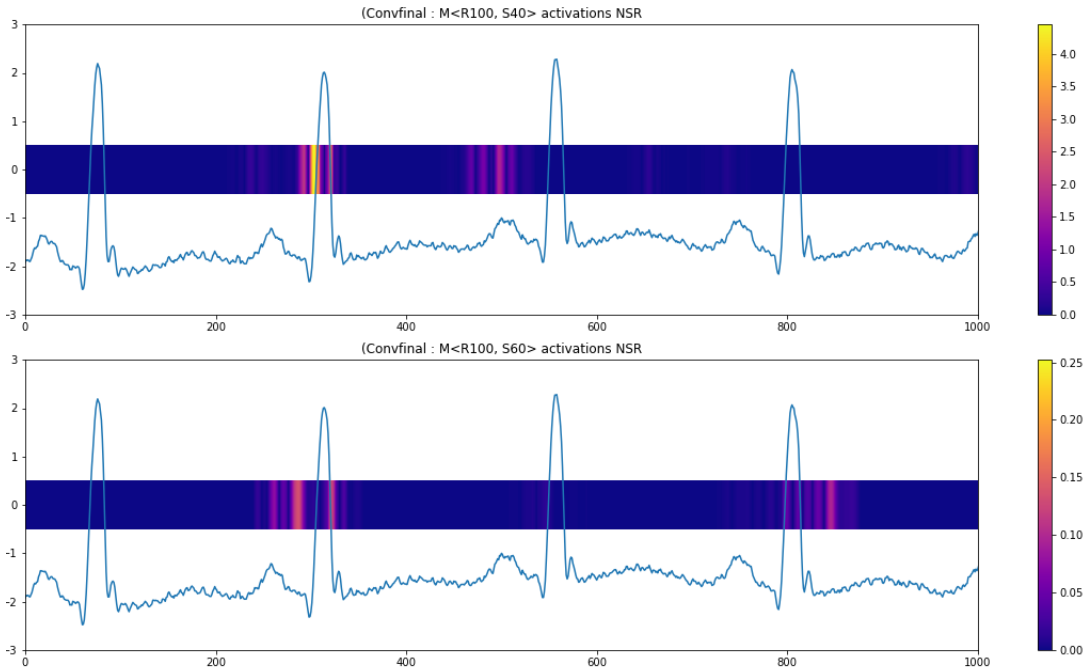
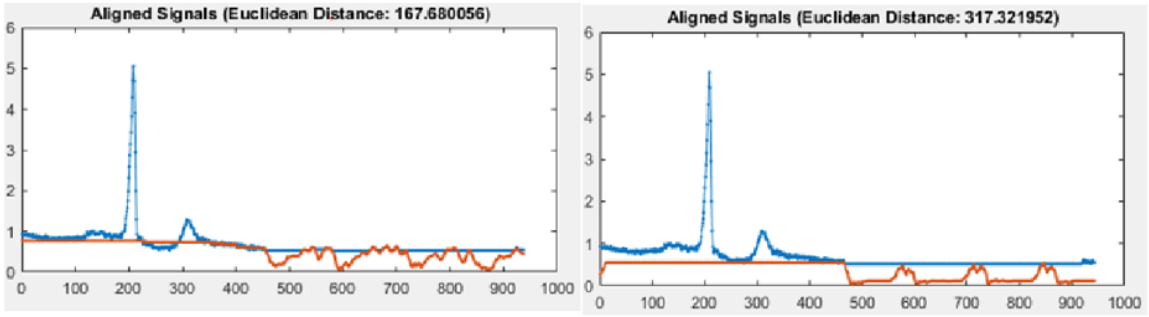


Figure 3.14. Case Model Degradation (A): Visualizing Conv*final* Activations

The d_{inter} value was recorded at 149.64 units, a high difference that accounts for change in classification outcome but not the correct classification by $M<R_{100}, S_0>$.



Conv$_{final}$: $M<R_{100}, S_{100}>$ Conv$_{final}$: $M<R_{100}, S_0>$

Figure 3.15. Model Degradation (B): DTW alignment (amplitude/time) | Euclidean distance | warping (blue: real | orange: feature map)

Feature Visualization: CEFEs generated overlay plots of $Conv_{first}$ and $Conv_{final}$ from $M<R_{100}, S_0>$ and $M<R_{100}, S_{100}>$. In Figure 3.16 we observed R-Peaks in $M<R_{100}, S_0>$ but failed to capture majority ECG features while observations of $M<R_{100}, S_{100}>$ showed complex ECG features. The $Conv_{final}$ of both models were not interpretable and therefore the learned ECG features are not detailed.

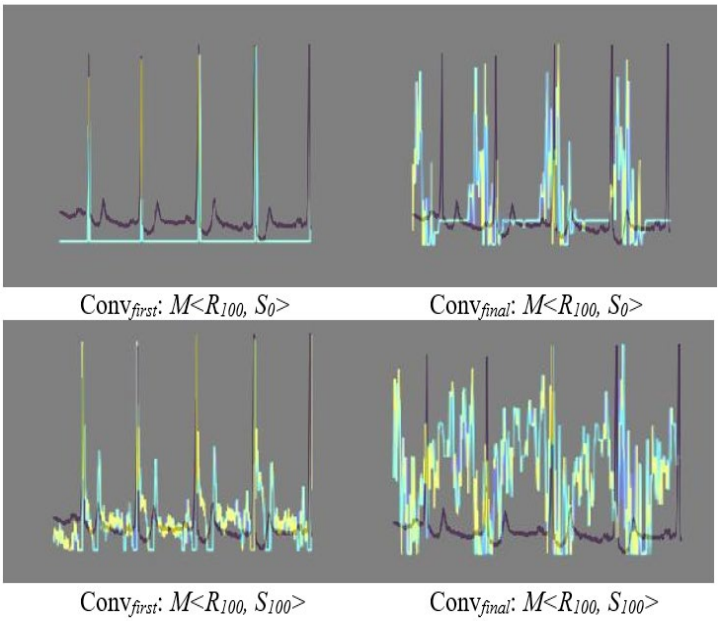


Figure 3.16. Model Degradation (B): Feature visualization (amplitude/time)

***Feature Detection and Mapping*:** We observed model performance degradation in $M<R_{100}, S_{100}>$ with was augmented with the same amount of synthetic data as the full real training set in $M<R_{100}, S_0>$. $M<R_{100}, S_{100}>$ with higher accuracy, lower d_{intra}, *MSE* value and training loss misclassified the candidate test case. The correct classification by $M<R_{100}, S_0>$ may indicate induced bias by the synthetic data.

Table 3.8. Model Degradation (B): Feature detection | R-Peaks (count & position) | R-R interval (position)

Feature Detection Module Inputs	# of R-Peaks	R-Peak Position	R-R Interval Position
Candidate Test Case	5	[208,581,991,1393,1784]	[373, 410, 402, 391]
$M<R_{100}, S_0>$	4	[204,579,989,1390]	[375, 410, 401]
$M<R_{100}, S_{100}>$	8	[83,154,256,471,580,1282,1400,1665]	[71, 102, 215, 110, 702, 118,

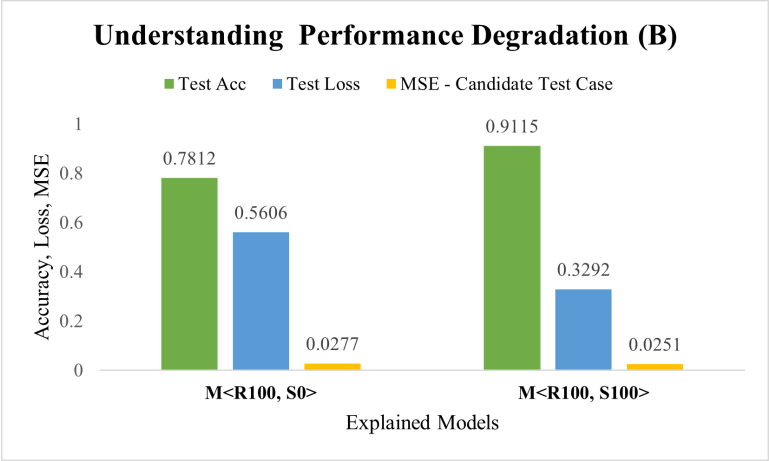


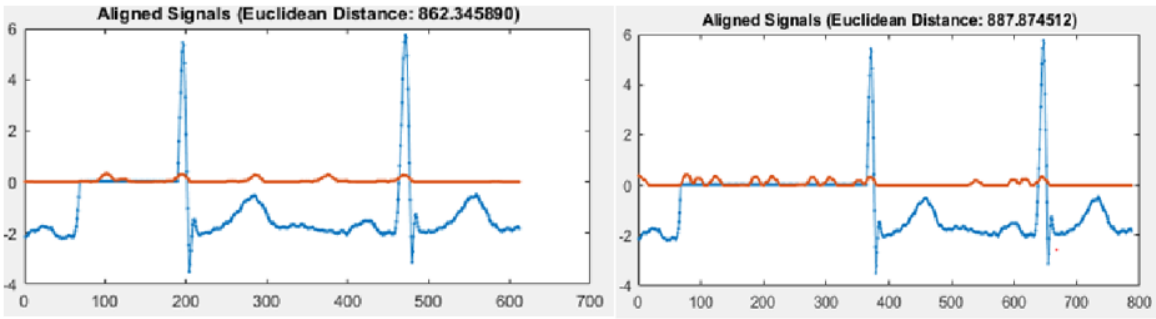
Figure 3.17. Case Model Degradation (B): performance accuracy (%), loss and MSE

Model No-Difference Performance

Evaluations in this section consider candidate test cases that did not show changes in classification outcomes regardless of model training data configuration. CEFEs modular evaluations were applied and results recorded for trained CNN models $M<R_{100}, S_0>$ and $M<R_{100}, S_i>$ using input ECG signal and feature maps from AFIB target class candidate test case.

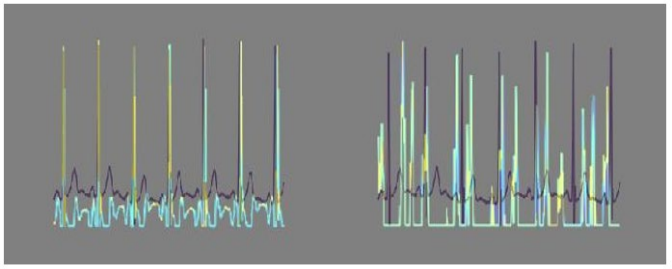
Descriptive Statistics: DTW values of models $M<R_{100}, S_{100}>$ and $M<R_{100}, S_0>$ were computed from feature maps of $Conv_{final}$ and the alignment with the candidate test case (Figure 3.18). We recorded d_{intra} values from $Conv_{final}$ of 862.34 units and 887.87 units in $M<R_{100}, S_{100}>$ and $M<R_{100}, S_0>$, respectively. The high d_{intra} values which is an indicator of inadequately learned ECG features, is evidence for candidate test case misclassification by both models.

Feature Visualization: Figure 3.19 illustrates CEFEs generated overlay plots for $Conv_{first}$ and $Conv_{final}$ of $M<R_{100}, S_0>$ and $M<R_{100}, S_{100}>$. In $Conv_{first}$ of $M<R_{100}, S_{100}>$ we observed learned ECG features including P and T peaks while in $M<R100, S_0>$ we observed more interpretable P, T, and U peaks. ECG learned features from $Conv_{final}$ of $M<R_{100}, S_{100}>$ and $M<R_{100}, S_0>$ were highly complex for visual interpretations.

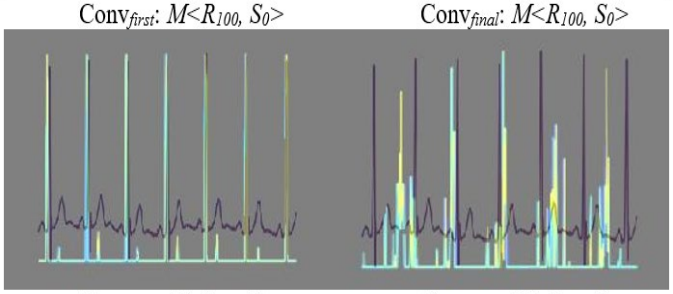


$Conv_{final}$: $M<R_{100}, S_{100}>$ $Conv_{final}$: $M<R_{100}, S_0>$

Figure 3.18. Model No-Change: DTW alignment (amplitude/time) | Euclidean distance | warping (blue: real | orange: feature map)



$Conv_{first}$: $M<R_{100}, S_0>$ $Conv_{final}$: $M<R_{100}, S_0>$



$Conv_{first}$: $M<R_{100}, S_{100}>$ $Conv_{final}$: $M<R_{100}, S_{100}>$

Figure 3.19. Model No-Change: Feature visualization (amplitude/time)

Feature Detection and Mapping

In the case where there were no differences in classification outcome between a model trained with only real data and a model trained with data augmented with synthetic data, we see in Table 3.9 that the $M<R_{100}, S_0>$ and $M<R_{100}, S_{100}>$ learned similar number of R-Peaks and R-R interval position information as the test case. When we visualize the learned features in Figure 3.19, we observe comparable waveforms in both models and additionally, the MSE values in Figure 3.20 were comparable to the test case despite the model classification accuracy and loss in $M<R_{100}, S_{100}>$ being considerably higher and lower respectively.

Table 3.9. Model No-Change: Feature detection | R-Peaks (count & position) | R-R interval (position)

Feature Detection Module Inputs	# of R-Peaks	R-Peak Position	R-R Interval Position
Candidate Test Case	7	[74, 349, 623, 898,1171, 1451, 1733]	[275, 274, 275, 273, 280, 282]
$M<R_{100}, S_0>$	7	[72, 348, 621, 896, 1170,1451, 1731]	[276, 273, 275, 274, 281, 280]
$M<R_{100}, S_{100}>$	6	[332, 607, 882, 1154, 1435, 1716]	[275, 275, 272, 281, 281]

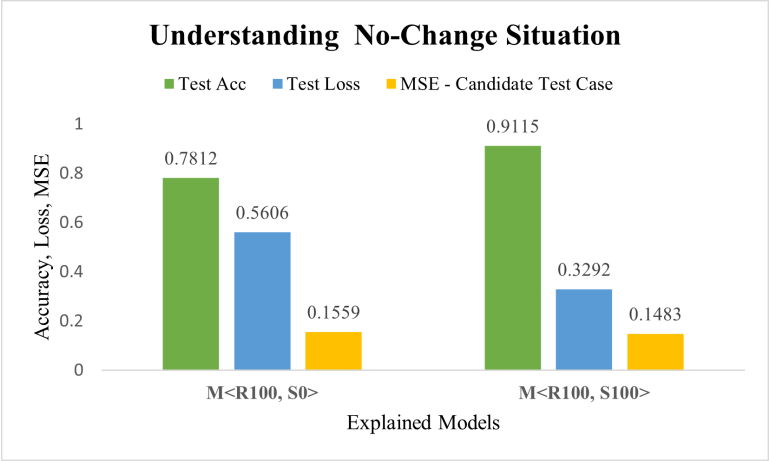


Figure 3.20. Case Model No-Change: performance accuracy (%), loss and MSE

3.5 Summary

The challenges posed by a lack of concrete understanding of how artificial intelligence systems make decisions hinder the mainstream adoption of such systems in the healthcare domain. Furthermore, data scarcity due to privacy concerns and small imbalanced datasets in healthcare impede current efforts in healthcare systems research. The current state-of-the-art AI technologies require a large dataset to learn the representation of real-world data and train effective life-critical decision support models.

This study on CEFEs is at the crossroads of these varying challenges, the need for interpretable explanations for AI systems, and the application of synthetic data as a means of additional data to facilitate the understanding of the prediction performance behaviors of deep learning model. CEFEs framework evaluates the internals of CNN models and produces modular interpretable explainable artifacts. CEFEs can be implemented on a single CNN model to understanding model capacity or with multiple CNN models for comparative evaluations that address model capacity or behavior in the lens of improvement, degradation, and no-change situations. Comparative analysis of features present in real-world data and representations learned by a deep learning algorithm provides intuitive, common-place knowledge that addresses the trust gap found in healthcare artificial intelligence-based decision systems.

CHAPTER 4

VPM: THE VIRTUAL PATIENT MODEL[2]

4.1 The VPM Framework

The Virtual Patient Model (VPM) is a framework consisting of six modules namely Seed Data Bank, Data Synthesizer, Statistical Analyzer, Machine Learning (ML) Validator, Visual Validator, and Virtual Patient Data (VPD). VPM is motivated by small and imbalanced datasets that are present as challenges in healthcare implementation of artificial intelligence methods. Trends in synthetic data generation methods (section 2.3) show unstructured validation strategies that do not cover fully account for quantitative and qualitative analysis necessary for validating synthetic data. VPM is a blueprint that boosts an end-to-end strategic approach to both quantitative and qualitative validation of synthetic healthcare datasets.

VPM quantitative analysis of synthetic data is mainly implemented in the Statistical Analyzer and ML Validator modules. Statistical tests and learning algorithms are efficiently applied to real-world data and synthetic data. Responses from quantitative analysis (descriptive statistics, sample tests, machine learning performances) are used in comparative analysis of real and synthetic data.

Qualitative analysis in VPM is in form of comparative visual inspections of real and synthetic data, expert analysis when available in the validation process and in performance

---

[2] © 2018 IEEE. Portion reused, with permission from R. Shamsuddin, B.M. Maweu, and B. Prabhakaran "Virtual Patient Model: An Approach for Generating Synthetic Healthcare Time Series Data," in 2018 IEEE International Conference on Healthcare Informatics (ICHI) IEEE, 2018, pp 208-218

outcomes of the models trained with data augmented with synthetic data. Figure 4.1 is a diagrammatic view of VPM modules and validation flow of generated synthetic data.

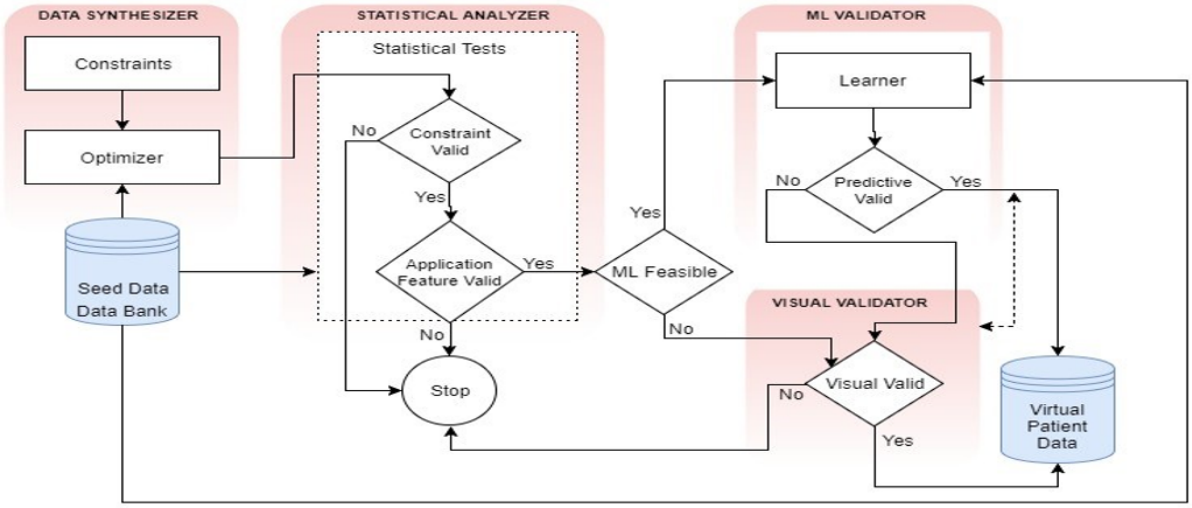


Figure 4.1. The Virtual Patient Model Framework in (Shamsuddin et al., 2018)

4.1.1 Seed Data Banks

Healthcare data banks contain medical measurement obtained from patient using various medical devices during physician or hospital visits. These repositories of real patient data adhere to strict privacy laws that control the flow of the data while protecting the health information. The VPM framework integrates a *data bank* module as a collection and source point of real patient data. The data bank is characterized by various data formats including, medical scans, images, time series, and unstructured data. The module output is referred to as seed data and assumes it to bed in a format that can be operated on by subsequent VPM modules. Seed data summarizes all required characteristics necessary for analysis.

4.1.2 Data Synthesizer

The VPM Data Synthesizer module is accountable for (i) optimizing constraints features and application features, (ii) the optimization algorithm, and (iii) output synthetic samples. The Data Synthesizer accepts any known model such that when given some optimizing constraints, it will output synthetic data that preserves the application features. Some examples of synthetic data generators that can utilize the VPM framework are Genetic Algorithm (GA) based time series synthesizer proposed in (Shamsuddin et al., 2018) and the Guided Evolutionary Synthesizer (GES) described in (Maweu[2] et al., 2021).

4.1.3 Statistical Analyzer

The Statistical Analyzer is one of a series of VPM synthetic data validation modules that apply relevant validation tests to the synthetic outputs of the Data Synthesizer. The goal of the Statistical Analyzer is to determine whether the synthetic data realizes the underlying distribution and statistical properties of the real data. Statistical tests in this module would incorporate functions that evaluate the distribution, test for moments, quantiles, confidence intervals, and tests for hypothesis significance. Synthetic data that is successfully validated in this module is propagated to the next VPM module.

4.1.4 Machine Learning Validator

The ML Validator module uses machine learning algorithms that examine whether the synthetic data is *Predictive Valid*. There is no limitation to the type of machine learning algorithms used in this module therefore, synthetic data validations can use traditional machine learning (Decision Trees, k-Nearest Neighbor, Linear Regression), non-residual deep learning (plain

networks architectures without residual blocks), and residual deep learning (ResNet). A user achieves ML validation by implementing a task-specific model training schema that utilizes synthetic data in the training process. A training schema of this sort would incorporate a model trained on real data only and compare it against models trained on data augmented with synthetic data.

4.1.5 Visual Validator

The Visual Analyzer provides an additional platform in form of visual plots, overlay plots, and graphs which help with an easy and fast understanding of the real and synthetic data. The visual analysis gives insights into how well the structural patterns and trends of the real data are captured by the synthetic data. Data visualization provides easy real-time answers to information questions about the synthetic data without tedious algorithmic analysis. When a domain expert uses VPM Visual Analyzer to (i) inspect the data for outliers and similarities/dissimilarities, (ii) validate the visible features, and (iii) confirm the quality of data, we consider the synthetic data *Expert Valid*. A data sample that fails predictive validity is still acceptable for use and data storage synthetic with expert validation.

4.1.6 Virtual Patient Data

Virtual Patient Data (VPD) is a synthetic data repository whose content has been validated successfully by three of the four VPM validation modules (Statistical Analyzer, ML Validator, Visual Validator, and Expert Validation). VPD repositories are an effective tool that progresses research and innovations geared to deliver quality patient care.

CHAPTER 5

VPM STATISTICAL ANALYSIS AND VISUAL VALIDATION IMPLEMENTATION

We investigate methods that provide knowledge and improve classification performances of healthcare decision support systems. Increasingly, healthcare providers are using automated systems that aid in disease diagnosis. These automatic systems are built mainly upon AI backbone systems that have shown superhuman performances. While these decision support systems are effective at what they do, it is a healthcare data challenge to train them due to data scarcity, patient privacy requirements, irregular measurement collection patterns, and costly annotation of patient data. These healthcare data challenges motivate this implementation of the VPM synthetic data validation blueprint. We make the following VPM implementation assumptions:

- The seed data for synthetic data generation is the experimental data sets described in Table 5.1.
- The constraint optimizer in the VPM Data Synthesis module is the Evolutionary Synthesis (GES) proposed in (Maweu[2] et al., 2021).

Our experiments will demonstrate statistical tests, visual analysis, and non-residual and residual machine learning synthetic data validation on one dimensional (1D) and two dimensional (2D) publically available healthcare time-series datasets.

5.1 Related Work

Several studies have been done on validating synthetic data therefore, we review the proposed validation techniques. In the Synthetic Data Vault (SDV) (Patki et al., 2016) the generator is validated for predictive accuracy and subjective qualitative findings. SDV validation

for qualitative findings contains feedback from SDV users with the purpose of gauging any confusion experienced by users while using SDV data. A predictive accuracy validation in SDV is similarly accounted for in our study for the VPM framework ML Validation module. SDV feature scripts are computed on both real and synthetic data then measure the predictive accuracy of each group. T-test statistics are then performed on the accuracy groups returning a test decision for their null or alternative hypothesis.

(Bogle et al., 2016) implement a synthetic data generator using a linear programming solver that achieves statistical moments, similar variables, and data types of the real data. This generator uses a macro that specifies the moment order and uses parameters that control the algorithm to vary the output size, determine the compute time and the quality of the synthetic data. Similar to the expected tests in the VPM statistical analysis module, (Bogle et al., 2016) compare the mean and standard deviation between the real and synthetic data. They also validate the data set variable interactions by computing covariance matrices and checking for similarities/dissimilarities in covariance magnitude. Finally, (Bogle et al., 2016) separately fit a logistic regression model to real and synthetic training data, apply the trained models to real test data and evaluate classification performance, sensitivity, and specificity.

Recent research shows advancing synthetic data generation that uses deep learning algorithm like in (Alzantot et al., 2017) who proposed SenseGen, a sensory data generator that used deep learning architecture built on a Long-Short-Term Memory (LSTM) network. SenseGen synthetic data was validated using a discriminator model that was trained to distinguish between the real and synthetic samples. Unlike SenseGen validation strategy, VPM ML Validation module uses predictive performance measures to evaluate the quality of synthetic data.

5.2 Synthetic Data Generator

The experimental synthetic time-series datasets are generated using the GES generator described in (Maweu[2] et al., 2021). Two types of synthetic data are used where each set is synthesized with a fitness function with or without the following regularization terms:

NoReg: A regularization term is not added to GES fitness function.

Reg1: Uses a regularization term with the weight set to 1 and that minimizes the *variance* between the synthetic sample and a randomly selected real data that is not the generation template but is from the same target class.

Reg2: Uses a regularization term with the weight set to 1 and that minimizes the *mean* difference between the synthetic sample and a randomly selected real data that is not the generation template but is from the same target class.

5.3 Experimental Datasets

We consider three publicly available datasets with contrasting purpose and complexity in healthcare. The datasets include measurements from electrocardiogram, electroencephalogram, and human activity recognition body sensors.

Table 5.1. Definitions for the experimental datasets

Dataset	Shape	Size	Samples/ Instance	Total Instances	Structural Complexity
EEG	2D	(256,64)	16384	300	No
ECG	1D	(3600,1)	3600	654	P, T Waves, QRS Complex

5.3.1 Electrocardiogram (ECG)

ECG recording captures changes in the electrical activity of the heart muscle over time. These recordings present as unique morphological patterns of P-QRS-T waveforms. Healthcare

providers use ECG as a basis of diagnosing heart conditions where disease manifests as deviations from the waveforms of a normal sinus rhythm. The ECG used in subsequent experiments were recorded at 360Hz from the MLII lead and a conversion factor of 200 ADU / mV. These are 10 seconds long signals (record-based ECG) provided by (Plawiak 2017) which were derived from the PhysioNet MIT-BIH Arrhythmia dataset (Goldberger et al., 2000).

The ECG signals were collected from forty-five patients: 19 females and 23 male subjects between the ages of 23 to 89 years old. While the full dataset contained 17 different heart rhythms, four rhythms were chosen to conduct the experiments: Normal Sinus Rhythm (NSR) the benchmark rhythm expected from a healthy patient and three abnormal rhythms, Atrial Fibrillation (AFIB), Premature Ventricular Contractions (PVC), and Left Bundle Branch Block (LBB). A total of 654 real patient records were used, 283 (NSR), 135 (AFIB), 133 (PVC), and 103 (LBB).

5.3.2 Electroencephalogram (EEG)

The electroencephalogram detects and measures active electrical impulses of the brain using varying number of electrodes placed on the scalp of a patient. The number of electrodes used to collect EEG data determine the overall dimensional complexity of a single dataset. Analysis of EEG for changes in brain activity are used to diagnose disorders such as epilepsy, strokes, tumors, and alcohol related conditions. The EEG dataset from (Begleiter 1975) was collected for a study seeking to identify factors associated with genetic predisposition to alcohol dependency.

The participants were presented with varying image stimuli and readings were collected at 256Hz for a duration of one second using 64 electrodes placed on 120 participants with each participant completing 120 trials of the study. The dataset was divided into a an alcoholic and

control set, the experiments presented in this study performed around participants presented with a one image stimulus as EEG Class A and those presented with two identical or distinct image stimuli as EEG Class B.

5.4 Synthetic Data Validation

The goal of synthetic data validation is to examine whether real data properties are captured during synthesis. Validation include testing for statistical properties (mean, variance, inter quartile range), correlation to validate that strength between sensors is maintained, and structural elements preservation using distance, upper/lower envelope measures.

5.4.1 Statistical Analysis

Here, we consider real and synthetic healthcare time-series data as two different (treatment) groups and compute respective descriptive statistics and the Wilcoxon Rank-Sum test. We use descriptive statistics to compare the statistical features between the groups. We then use a rank-sum test to check whether these treatment groups are from the same population distribution for a given feature statistics. Table 5.2 shows that the synthetic data group has similar statistical features as the real-world data.

The computed p-values for each feature statistic and the results from the rank-sum test at a 5% significance level did not reject our assumption of equal feature statistics among these groups. The statistical analysis results suggest that the synthetic group preserved class-specific distributions and that class labels transferred from the seed template to the synthetic sample are valid.

Table 5.2. Descriptive statistics for real and synthetic datasets

	Mean	Variance	Median	IQR	Min	Max
EEG Classes						
Class A (Real)	-0.72	5.48	-0.74	2.42	-11.90	18.80
Class A (Synthetic NoReg)	-0.34	5.01	-0.38	2.29	-9.55	19.44
Class B (Real)	-2.97	8.30	-2.56	3.76	-17.63	11.40
Class B (Synthetic NoReg)	-2.59	7.60	-2.17	3.56	-16.77	11.87
ECG Classes						
NSR (Real)	964.19	139.35	967	22	943	981
NSR (Synthetic NoReg)	963.84	155.52	967.08	24.49	942.76	987.33
AFIB (Real)	972.97	94.73	972	17	959	993
AFIB (Synthetic NoReg)	972.69	87.37	970.49	14.64	956.67	992.73
PVC (Real)	993.86	3943.75	975	24	926	1255
PVC (Synthetic NoReg)	996.08	3731.53	982.55	31.12	926	1238.26
LBB (Real)	1020.74	1157.56	1004	72	975	1072
LBB (Synthetic NoReg)	1019.86	1264.85	995.09	68.78	979.62	1089.31

5.4.2 Statistical Analysis of Highly Structured ECG

The goal of this section is to test whether synthetic ECG data preserve class boundaries, and whether the highly structured element were preserved with statistical significance. To do so, the real data and the synthetic data are treated as two groups for the Wilcoxon Rank-sum hypothesis test. The null hypothesis for this test is that both groups have the same *intra-class* variation distribution. If the null hypothesis is accepted (p-value > 0.05), then we accept that variations seen in the synthetic data falls within the class boundaries as defined by the real data.

We use DTW alignment as the variation similarity/dissimilarity measure. While DTW is not considered a metric, because it does not satisfy triangular inequality (Vidal et al., 1985), it provides a measure that signifies the degree of difference between two waveforms. To measure the intra-class variation, 50 pairs of signals were randomly sampled from a particular class for each test group. The DWT alignment value is calculated for each pair, and then the distribution of alignment values for the two groups are compared using the rank-sum test.

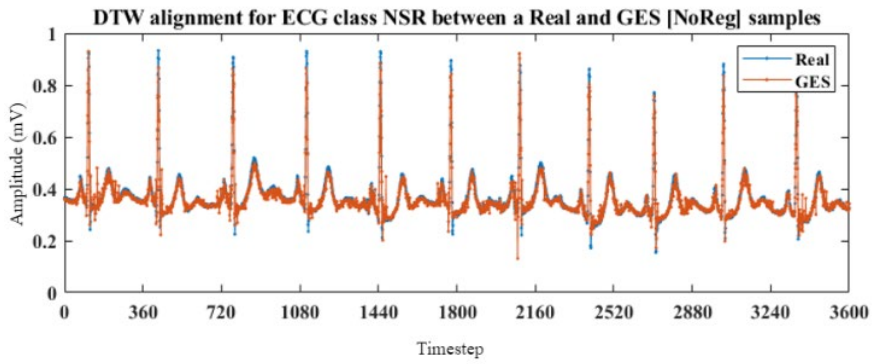


Figure 5.1. ECG – NSR overlay plot showing DTW alignment of real and synthetic data

Table 5.3 shows comparison of real intra-class variation with the intra-class variations from two synthetic groups: GES synthetic data; (b) data generated (referred to as Resampled in the table) using traditional perturbation techniques such as Noise Injection (Moreno-Barea et al., 2018) and Moving Average (MA). Moving average is a smoothing technique that computes average of the data over a fixed window size.

We find that none of the p-value is significant at alpha=0.05, and thus, we accept the null hypothesis and conclude that the variation within the synthetic datasets maintain class boundaries. We also present the minimum and maximum of the real-world data sample group, to further verify that the mean DTW value obtained from the synthetic groups lie within the empirical range.

Table 5.3. DTW, rank-sum and p-value for real and synthetic data (GES & Resampled)

ECG Classes	*p-value*	Min-Max DTW Real Test Group	Mean DTW Synthetic Test Group
		NSR: 20.70 - 426.0	
NSR (Real and Synthetic NoReg)	0.3610		117.6168
NSR Real and Synthetic Reg1)	0.0296		107.4933
NSR Real and Synthetic Reg2)	0.7174		128.5773
NSR Real and Synthetic Resampled)	0.4340		157.3552
		AFIB: 20.83 - 216.0	
AFIB (Real and Synthetic NoReg)	0.6666		71.4329
AFIB (Real and Synthetic Reg1)	0.2539		72.3785
AFIB (Real and Synthetic Reg2)	0.4928		72.9783
AFIB (Real and Synthetic Resampled)	0.5884		73.8502
		PVC: 24.00 - 287.5	
PVC (Real and Synthetic NoReg)	0.1891		96.2111
PVC (Real and Synthetic Reg1)	0.1338		92.4071
PVC (Real and Synthetic Reg2)	0.3574		101.2399
PVC (Real and Synthetic Resampled)	0.9588		114.806
		LBB: 28.03 – 250.0	
LBB (Real and Synthetic NoReg)	0.9753		94.4591
LBB (Real and Synthetic Reg1)	0.8067		92.0541
LBB (Real and Synthetic Reg2)	0.7854		91.9338
LBB (Real and Synthetic Resampled)	0.6566		98.2305

5.4.3 Visual Validation of Highly Structured ECG Signals

A distinct characteristic of ECG is its highly structured wave morphology. Therefore, to evaluate whether the highly structured elements were preserved or distorted by GES, we take Dynamic Time Warping (DTW) measure between the real and synthetic time series. DTW is a non-linear mapping that maximizes alignment; higher the alignment, lower the DTW measure. Figure 5.1 shows the DTW alignment between an NSR sample of real data and the corresponding GES generated synthetic data. We observe that the synthetic data preserved the highly structured ECG wave.

5.4.4 ECG and EEG Visual Analysis

We generate overlay plots of randomly selected real and GES synthetic data for EEG data and ECG signals. Patient ID 23 and 55, from EEG dataset and shown in Figure 5.2, are from target Class A and B, respectively. From the visual plots, we observe that changes over increasing time in GES generated synthetic data are similar but also, different from patterns in corresponding real (seed) data.

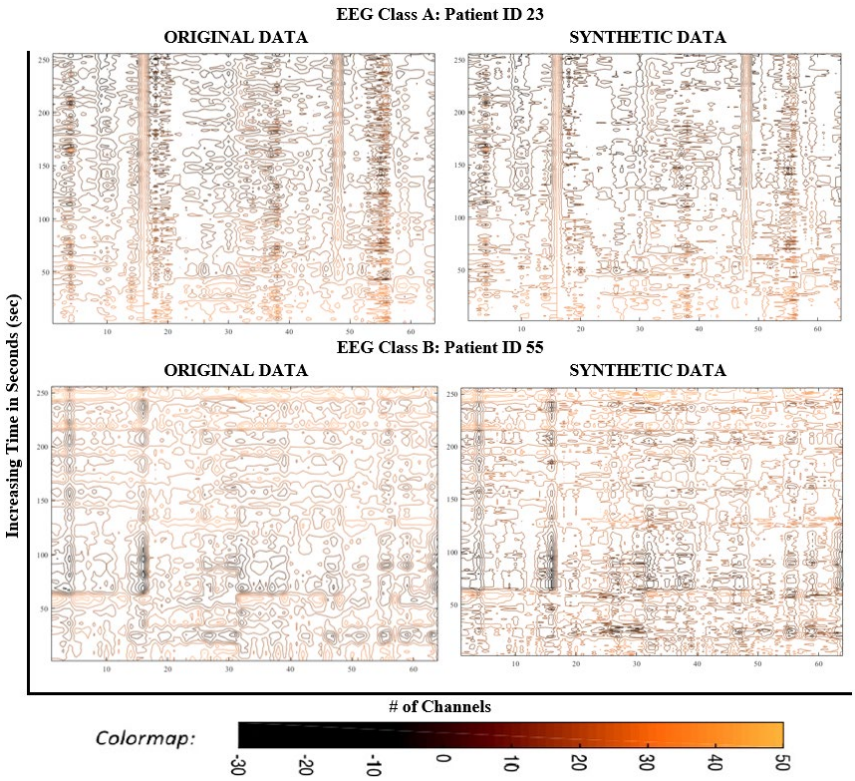


Figure 5.2. EEG contour plots for Class A (Patient ID 23) and Class B (Patient ID 55). We show variations between real and synthetic data for 64 channels/sensors 2D data.

We show in Figure 5.3, ECG class AFIB synthetic generated using the regularization terms in described in section 5.2.

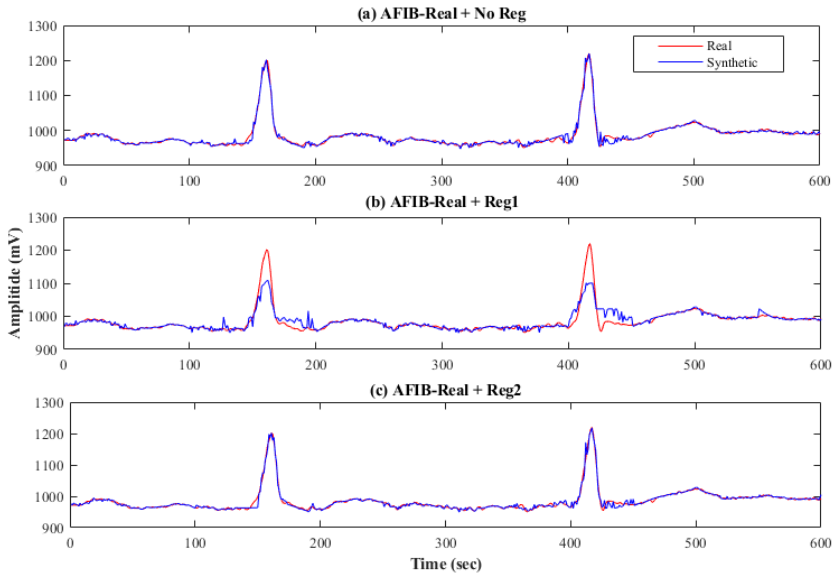


Figure 5.3. Overlay plot of ECG class AFIB showing different variations from regularization terms

5.4.5 Analyzing Sensor Correlation across EEG Instances and Datasets

An instance of EEG data is represented by 64 channels and therefore it is important to maintain channel correlation within an instance of GES generated synthetic data. To demonstrate that GES maintains channel correlation, we compute the minimum and maximum MSE values between pairs of real EEG data instances over the entire dataset. Next, we compute MSE between real and synthetic correlation matrices of EEG data from five randomly selected patient ID (66, 102, 123, 142, 149), as shown in Table 5.4. Evaluation of correlation from the randomly selected EEG patient data shows that their respective MSE values fall within the determined range of minimum (3.3365) and maximum (10.4094) MSE values. We therefore conclude that GES data generation process successfully preserves correlation between the channels (sensors).

On the other hand, while MSE values (between the upper triangular correlation matrices of the real and synthetic data) close to zero are ideal, we observe different behavior in EEG dataset.

EEG captures spontaneous brain activity with individual characteristics and correlation that will vary from patient to patient. This additional EEG complexity varies data profiles across patients and sensors and can make them indistinguishable from random noise even when the underlying biological system may not be random (Pijn et al., 1991). Thus, we expect to see a higher range of variation in the MSE values calculated between the upper triangular correlation matrices of two real EEG data instances (even if they belong to same class). This analysis shows GES's tendency to assume that any correlation present in real data originates from specific significant underlying biological events and not from individual characteristics.

Table 5.4. EEG min-max real data MSE and its correlation with synthetic data

EEG	MSE (Real & Synthetic Data)
Patient ID: 66	3.1124
Patient ID: 102	2.0777
Patient ID: 123	1.2532
Patient ID: 142	1.1924
Patient ID: 149	2.8603
Min Real Data MSE: **3.3365**	*Max Real Data MSE:* **10.4094**

5.5 Summary

We implemented the proposed VPM quantitative and qualitative validation strategies for synthetic data generated using the generator called the Genetic Evolutionary Synthesizer (GES) in (Maweu[2] et al., 2021). The validation of the synthesized data demonstrated time series data of varying complexities (including dimensions and dimension lengths) and accounted for statistical, structural and preserved correlation among the data channels.

We successfully demonstrated effectiveness of synthetic data in healthcare tasks through improvements in deep neural network performance in diverse and exhaustive experiments.

The central hypothesis guiding our experimental designs was that properly synthesized (synthetic data that retains distinguishable class-level features) and validated synthetic data can (a) augmented to deep learning training sets, or (b) serve as proxy training data for deep learning architectures. We showed in our experiments improved the classification accuracy, specificity, and sensitivity of state-of-the-art deep networks. The rationale for the hypothesis came from preliminary work with traditional machine learning algorithms and non-residual deep networks architectures, both of which showed promising results with synthesized data. Experimental results provided evidence of the following:

- Quality of synthetic data in terms of preserving class-distinguishable features;
- EEG diagnostic model that performed best was trained with lower volume of synthetic data training data when compared to other similar EEG models);
- Ability to obtain better ECG diagnostic models using synthetic data that better handle learning biases such as those observed in NSR-PVC trade-off;
- Showed that using synthetic data can address the challenge of class imbalances found in healthcare dataset. This was achieved without under-sampling the already small dataset considering that by under-sampling small datasets would result in overfitting training in deep networks.

CHAPTER 6

IMPROVED PERFORMANCE OF AI-BASED HEALTHCARE DECISION SUPPORT SYSTEMS WITH SYNTHETIC DATA[3]

Artificial intelligence continues to revolutionize the delivery of quality patient care. Information extracted and analyzed in the purview of healthcare is invaluable for enhanced personalized patient care, improved treatment plans for the overall patient health outcomes. Here, we propose synthetic healthcare datasets to boost the classification performance of deep neural network models and mitigate limited accessibility to data, the imbalanced nature of healthcare data. Once validated and analyzed for realism to real-world data, synthesized data presents a significant opportunity in overcoming privacy, ethical, and data collection concerns in addition to the cost of seeking medical experts for healthcare data annotation. Researchers gain access to publicly accessible, labeled, large, and balanced synthetic datasets for application to research initiatives.

We use validated synthetic data to demonstrate improvements in the analytical and diagnostic prediction outcomes of real-world data of deep learning models. In a series of experiments, we observe model performance improvements in individually unique healthcare time-series datasets described in Table 5.1. We generate synthetic ECG, EEG, and accelerometer (AReM, APPENDIX A) datasets using the GES generator (Maweu et al., 2021) and use the

---

[3] ©2021 IEEE. Portion reused, with permission, from B. Maweu, R. Shamsuddin, S, Dakshit, and B. Prabhakaran "Generating Healthcare Time Series Data for Improving Diagnostic Accuracy of Deep Neural Networks," in IEEE Transactions on Instrumentation and Measurement, doi: 10.1109/TIM.2021.3077049

validated synthesized data in multiple experiments to train traditional machine learning algorithms, non-residual and residual deep networks.

Model performance with synthesized data shows improvements specifically in better specificity and sensitivity when compared to models trained only on real-world data. We note that the reduction of false positives and false negatives values is significant when evaluating healthcare classifiers. Therefore, augmenting training sets with synthetic data for deep networks can result in better and more balanced classification models. Experimental results show promising results for building decision support systems that use deep network backbones for predictions and diagnosis in healthcare.

6.1 Related Work

SDV validation using qualitative findings contains feedback from SDV users with the purpose of gauging any confusion experienced by users while using SDV data. Predictive validation in SDV, (Patki et al., 2016) computes feature scripts on both real and synthetic data then measure the predictive accuracy of each group. T-test statistics are then performed on the accuracy groups returning a test decision for their null or alternative hypothesis. Statistical analysis in (Bogle et al., 2016) compare the mean and standard deviation between the real and synthetic data. They also validate the data set variable interactions by computing covariance matrices and checking for similarities/dissimilarities in covariance magnitude. Finally, (Bogle et al., 2016) separately fit a logistic regression model to real and synthetic training data, apply the trained models to real test data and evaluate classification performance, sensitivity, and specificity.

6.2 Methodology

We investigate the effect of synthetic data on the classification performances of (i) traditional machine learning algorithms, (ii) non-residual networks, and (iii) residual networks.

6.2.1 Dataset Description

The datasets used to train and test the models presented in the following experiments were derived using a train-test split of real ECG and EEG datasets (Table 5.1). The test split, which we refer to as a *uniform testing set,* has 31 EEG data samples with 13 Class A samples and 18 Class B samples. The ECG *uniform testing set* has 120 data samples with 30 data samples from each ECG class (NSR, AFIB, PVC, and LBB). All synthetic data used validation and performance analysis against a classification decision support system. The synthetic data was generated with the synthesizer GES (Maweu[2] et al., 2021) using the datasets in Table 5.1 as the seed data/template.

6.2.2 Traditional Machine Learning Algorithms

We instantiated the VPM framework in (Shamsuddin et al., 2018) and used the weighted K-Nearest Neighbor (KNN), Decision Tree (DT), and Ensemble of Bagged Tree (TB) machine learning algorithms to classify the ECG experimental dataset.

6.2.3 Non-Residual Network Architecture

We evaluated classification performance of ECG dataset on the 16-layer 1D-CNN classifier proposed for long ECG sequences in (Yildirim et al., 2018).

6.2.4 Residual Network Architecture

We grouped network layers into a convolution block and an identity block. The ResNet convolution block had three alternating one dimensional convolution neural network (1D-CNN) layers: a batch normalization layer (BN), an activation layer, and a 1D-CNN residual skip connection for dimensionality [17]. The identity block had three alternating layers: a 1D-CNN layer, a BN layer, and an activation layer. The filter setting for the alternating convolution and identity blocks are: (32 x 2 | 64 x 1), (64 x 2 | 128 x 1) and (128 x 2 | 256 x 1), and kernel size 3.

The ResNet model had an input layer specified by dataset dimensional length. The next layers were two blocks of 1D-CNN, BN, and Activation layers followed by three alternating calls to one convolution block and two identity blocks. The dense output layer was specific to the number of classes in each dataset. We set our learning rate (LR) scheduler to the dynamic function *ReduceLROnPlateau*. Based on experimental estimations, we found that 200 epochs with early stopping (to monitor validation loss), and LR lower bounded at 0.001 with a 0.5 reduction factor, allowed optimal tuning during the training process.

6.2.5 Metrics for Performance Measurement

To understand classification performance, we used classification confusion matrices to observe how well a model predicted respective target classes True positive (TP), false positive (FP), true negative (TN) and false negative (FN) values were used to calculate accuracy, sensitivity, and specificity (Galen et al., 1975) together with statistical metrics were computed from confusion matrices valued outputs e.g., samples in the uniform testing set is never used for training.

Specificity is the ability of a model to correctly identify true negative test samples, and sensitivity the ability for the model to correctly identify true positive test samples. The metrics used to evaluate our trained DNN models are shown in Equations 6.1 through 6.3.

Specificity | True Negative Rate (TNR): $= \frac{TN}{TN+FP}$ (6.1)

Sensitivity | True Positive Rate (TPR): $= \frac{TP}{TP+FN}$ (6.2)

Test Accuracy (Test Accuracy): $= \frac{TP+TN}{TOTAL}$ (6.3)

6.2.6 Trained Model Naming Convention

We adapt a naming convention for describing our trained models. In our experiments trained model are named as: *Model (dataset, data source, number samples, regularization*^{*optional}*)* where (i) *Dataset*: one of the experimental datasets EEG and ECG, (ii) *Data source*: set comprised of a combination of real, synthetic, and resampled data, (iii) *Number of samples*: is the number of training data samples from each *data source* used to train a model, and (iv) *Regularization*: is a regularization term (NoReg, Reg1, and Reg2) defined in section 5.2.

The regularization* term can be omitted if a regularization term is not used during synthesis. A sample usage of the naming convention is *Model (ECG, Real + Syn,100, Reg1)* that describes a model trained with a combination of 100 training samples equally from real data and synthetic ECG data generated using regularization term 1.

6.3 Results and Analysis

6.3.1 ECG Classification using Traditional Machine Learning

For training the machine learning models, we followed three strategies:

1. Model (ECG, Real, 200): Train and test on real data.
2. Model (ECG, Synthetic, 200, NoReg): Train, test on GES synthetic data.
3. Model (ECG, Real + Synthetic, 200, NoReg):
    a. Train and test on real combination of real and GES synthetic data
    b. Train on real and GES synthetic data; test on real data

For each strategy we report the machine learning algorithm with the highest classification accuracy. We observed equal classification accuracy of 80% with KNN for both Model (ECG, Real, 200) and Model (ECG, Synthetic, 200, NoReg), 82% with Bagged Tree for Model (ECG, Real + Synthetic, 200, NoReg), and 84% with Decision Tree for Model (ECG, Real + Synthetic, 200, NoReg). Decision Tree had higher accuracy when Model (ECG, Real + Synthetic, 200, NoReg), was tested with real data. However, the Bagged Tree performed better on the same model when tested with a combination of real and synthetic data.

Comparable classification accuracy between Strategy 1 and Strategy 2 demonstrates that validated synthetic data is sufficient for train learning models. Also, the classification accuracy observed in traditional machine learning models improved when trained with real data augmented with validated synthetic data.

6.3.2 ECG Classification using a No-Residual Network

The non-residual network architecture was maintained when training the following four models. The training sets for the models had 200 data samples made up of data samples from different *data source*:

1. Model (ECG, Real, 200)
2. Model (ECG, Synthetic, 200, NoReg)

3. Model (ECG, Real + Synthetic, 200, NoReg)
4. Model (ECG, Real + Resampled, 200)

The trained models were tested on the ECG *uniform testing set* and the classification performance obtained from the models showed the highest accuracy of 91.15%, from Model (ECG, Real + Synthetic, 200, NoReg), Model (ECG, Synthetic, 200, NoReg) with 89.58%, Model (ECG, Real, 200) with 88.54%, and Model (ECG, Real + Resampled, 200, NoReg) with 85.42%.

Classification performance from the trained models shows that of the four non-residual network classifiers, the one trained with real data augmented with synthetic data outperformed the others. We see also observed that using the VPM framework to analyze and validate the synthetic data before using it for training resulted in higher model performance and infers better quality than perturbed data samples. We infer this from the observed ~5.7% performance difference between Model (ECG, Real + Synthetic, 200, NoReg) and Model (ECG, Real + Resampled, 200, NoReg).

Overall, the results from non-residual network classifiers revealed that when models are trained real data and augmented with validated synthetic data can produce higher performance than a model trained with real data only or a model augmented with samples generated using traditional data perturbation methods.

6.3.3 ECG and EEG Classification using a Residual Network

Promising classification performance trends from traditional machine learning and non-residual networks inspired further investigation of the use of synthetic data in more complex and state-of-the-art DNNs. ResNets are well suited for solving vanishing gradient, overfitting, and

negative impacts of numerous variables, and are amenable to depth adjustments. These depth adjustments helped mitigate the different dimensional complexities of the experimental datasets.

EEG and ECG ResNet Baseline Models

In experiments *Imbalanced, Full Real Data Training Sets*, and *Balanced Real Data Training Sets*, we seek to obtain baseline models for the EEG and ECG datasets. We evaluate classification performance for the trained small, imbalanced, balanced, and complex models using the dataset's uniform testing set.

Table 6.1. ResNet classification results for obtaining baseline models

Trained Models	Baseline	Train Acc. (%)	Test Acc. (%)	Spec (%)	Sens (%)
		EEG			
Model (EEG, Real, 160)	✓	97.57	90.32	90.60	90.60
Model (EEG, Real, 260)		97.65	75.00	75.00	75.00
		ECG			
Model (ECG, Real, 100)	✓	95.13	94.13	96.94	90.83
Model (ECG, Real, 200)	✓	95.29	94.18	96.11	88.33
Model (ECG, Real, 514)		98.83	95.00	96.67	90.00

Imbalanced, Full Real Data Training Sets

With this experiment, we get baseline models trained with our small and imbalanced datasets. We investigate the classification performance of these baseline models as benchmark performance for the remaining experiments. The trained EEG model used 260 possible data samples and ECG with 514 data samples. The experimental training sets were imbalanced. The imbalanced training set had at least one class in each dataset overrepresented by twice as many data samples. The EEG Class A and B were trained on 80 and 180 training data samples,

respectively, whereas ECG classes NSR, AFIB, PVC, and LBB were trained 248, 100, 98, and 68 data samples, respectively.

In Table 6.1, we observe poor classification accuracy, specificity, and sensitivity for Model (EEG, Real, 260) at 75%; whereas Model (ECG, Real, 514) shows better overall accuracy (95%) but has a huge difference of 6.7% between specificity and sensitivity. We attribute these observed results for both models to the imbalanced training sets, a common challenge found in healthcare datasets 5.3.1 and 5.3.2. The specificity and sensitivity obtained from ECG can be deemed sufficient for screening target arrhythmias in the general population but may not be efficient for diagnosing at-risk cohorts (Maxim et al., 2014). Therefore, the experiment described in section *ResNet Model with Best Overall ECG Classification Performance* investigates the specificity and sensitivity outcomes of the ECG trained models. Classification performance of imbalanced training sets guides us to find better EEG and ECG baseline models with balanced classes in the next experiment.

Balanced Real Data Training Sets

We balanced EEG and ECG training sets by creating subsets of training sets from the imbalanced training sets. Since we used only real data, this was only achievable through under-sampling because Class A from the EEG dataset had the lowest number of samples. We randomly selected 80 data samples from class B to create a balanced training set for a total of 160 EEG samples. Two ECG training sets were balanced with an equal number of training samples for class NSR, AFIB, PVC, and LBB. One ECG training set had 100 data samples, and the other set had 200 data samples.

Even with improved classification accuracy, sensitivity, and specificity when compared to the imbalanced Model (EEG, Real, 260), the balanced Model (EEG, Real, 160) had a ~7% difference between training and classification accuracy (Table 6.1). The difference in training and classification accuracy indicates possible overfitting and loss of generalizability in the model. The classification accuracies of the balanced Model (ECG, Real, 100) and Model (ECG, Real, 200) were comparable but showed a ~1% drop when compared to the imbalanced Model (ECG, Real, 514). Disparities in the model specificity and sensitivity remained with a ~2% model sensitivity drop in Model (ECG, Real, 200). Class-wise classification analysis of ECG models showed that Model (ECG, Real, 200) misclassified NSR test samples as PVC. This misclassification contributed to the drop in model sensitivity compared to Model (ECG, Real, 514), in experiment *Imbalanced, Full Real Data Training Sets*.

Imbalanced Model (ECG, Real, 514) was trained on twice as many NSR training samples as PVC and may have just been predicting the majority class. We also visually compared the morphology of NSR and PVC and observed that their respective features were generally very similar. In a clinical context, PVC presents a single abnormal beat that occurs and disrupts the normal rhythm NSR. We further investigated these observations in section *ResNet Model with Best Overall ECG Classification Performance*.

Evaluating ResNet Classification Performance

Evaluating Quality of Validated Synthetic Data

Despite the biases and disparities observed in the models trained with real data, the baseline models were still the best models we obtained from the experiments described in section *EEG and ECG ResNet Baseline Models*. Therefore, we use the baseline models to investigate whether they

can distinguish the respective classes in the artificial dataset obtained from GES. Model performance evaluations used a synthetic testing dataset synthesized with the uniform testing set as the synthesis template.

We assume that if the classification accuracy of the baseline models is better than random chance (> 50%), we consider the synthetic data to be of good quality as that indicates the preservation of class-specific features within the GES synthetic dataset.

Classification accuracy, specificity, and sensitivity in Table 6.2 show a comparison of EEG and ECG baseline models tested on uniform testing sets and GES synthetic testing sets. Based on our goals to validate and use quality synthetic data, these results show that the synthetic testing set achieved better than random accuracy. These comparable classification outcomes show that the synthetic generation process captured and preserved class properties present in real data.

It is a favorable outcome, especially when considering the disparity between specificity and sensitivity values in Table 6.2, and the learning biases in the baseline models. Flexibility in a synthetic data generator allows researchers to apply constraints that control the variations induced during the generation process.

Table 6.2. Comparative analysis of ResNet baseline models tested on real and synthetic data

Baseline Models	**Tested on Real Data (%)**	**Tested on Synthetic Data (%)**
Model (EEG, Real, 160)	Acc.: 90.32 Spec: 90.60 Sens: 90.60	Acc.: 87.10 Spec: 86.75 Sens: 86.75
Model (ECG, Real, 100)	Acc.: 94.13 Spec: 96.94 Sens: 90.83	Acc.: 87.04 Spec: 91.27 Sens: 79.95

Evaluating Traditional Synthetic Data Perturbation

In this experiment, we investigate how GES validated synthetic data measures against synthetic data generated using traditional data perturbation techniques. Data perturbation data samples were from techniques that similar to GES, operate on data space. Real data was perturbed with noise injection and moving average methods. Following synthetic data generation, we trained two ResNet models, one with real data augmented with validated synthetic data (Real + Synthetic), and another with real data augmented with noise injection and moving average perturbed data (Real + Resampled). Both models were tested on a *uniform testing set* and classification performances were compared, as shown in Table 6.3.

We observe in Table 6.3 that EEG and ECG models augmented with GES synthetic data (Real + Synthetic) showed better classification performance than similar models augmented with synthetic data perturbation (Real + Resampled). Model (EEG, Real + Synthetic, 160, NoReg) had ~30% better classification accuracy than Model (EEG, Real + Resampled, 160) when evaluated on a uniform testing set. Similarly, the Model (ECG, Real + Synthetic, 200, NoReg) augmented with GES synthetic data outperformed Model (ECG, Real + Resampled, 200) in classification accuracy, specificity, and sensitivity.

Thus, the ResNet models trained with validated synthetic data performed better than similar models trained with data generated using traditional data perturbation methods. Because GES operates on the data space, its synthetic data is ideal for use in deep learning models that do not require feature engineering before use. As such, we did not consider over-sampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE) proposed in (Chawla et al., 2002) because it generates synthetic samples from a feature space.

Table 6.3. ResNet classification performance for datasets with GES regularization options and with data perturbation

	Test Acc. (%)	Spec (%)	Sens (%)
EEG Trained Models			
Model (EEG, Real + Synthetic, 160, NoReg)	**93.55**	**94.44**	**94.44**
Model (EEG, Real + Resampled, 160)	65.00	65.00	65.00
ECG Trained Models			
Model (ECG, Real + Synthetic, 200, NoReg)	**92.50**	**95.00**	**85.00**
Model (ECG, Real + Resampled,200)	85.83	90.56	71.67
Model (ECG, Real + Synthetic, 100, NoReg)	95.00	96.67	90.00
Model (ECG, Real + Synthetic, 100, Reg1)	**96.25**	**97.50**	**92.50**
Model (ECG, Real + Synthetic, 100, Reg2)	**96.25**	**97.50**	**92.50**

Effects of Using GES Regularization Feature on ResNet Models

GES provides several customizable features that give researchers the needed control over the data generation process. One such feature is the ability for the researcher to use regularization term(s) in the objective function that enables feature transfer between samples. Here we evaluate and discuss how the use of the regularization terms *NoReg*, *Reg1*, and *Reg2* (section 5.2) data affect the classification performance of ResNet models.

ECG and EEG Use Cases for No Regularization Term Data Synthesis

The default setting for the GES fitness function does not use a regularization term (NoReg in section 5.2). Therefore, we compare Model (EEG, Real + Synthetic, 160, NoReg) with Model (ECG, Real + Synthetic, 200, NoReg) from the experiment in section *Evaluating Traditional Synthetic Data Perturbation*, where both trained models used synthetic data generated with NoReg. The training set had an equal number of real and GES synthetic data training samples.

Our experimental results in Table 6.3 show that the Model (EEG, Real + Synthetic, 160, NoReg) outperformed the baseline Model (EEG, Real, 160) (Table 6.1) in classification accuracy,

sensitivity, and specificity metrics. We note that the 93.55% accuracy in Model (EEG, Real + Synthetic, 160, NoReg) was lower than the model specificity and sensitivity of 94.44%. Specificity and sensitivity values do not necessarily reflect the accuracy values (Zhu et al., 2010) because they describe different aspects of a testing set. We observe lower classification accuracy and specificity in Model (ECG, Real + Synthetic, 200, NoReg) when compared to baseline Model (ECG, Real, 200) (Table 6.1).

Upon visual inspection of the synthetic data, we concluded that the NoReg setting captured the rigid structures too tightly. The inadequacy of the NoReg option motivates further investigations into how variations from regularization terms influence ResNet classification performance.

ECG Use Case of Regularization Terms Reg1 and Reg2 for Synthesis

We use the flexibility of GES to customize how the objective function injects variations into the synthetic samples. We used regularization terms Reg1 and Reg2 (section 5.2) to generate ECG synthetic data. We then compared classification performances of the ECG baseline models, and models trained with synthetic data generated using NoReg, Reg1, and Reg2 GES options.

In Table 6.3, the classification results for Model (ECG, Real + Synthetic, 100, NoReg), Model (ECG, Real + Synthetic, 100, Reg1) and Model (ECG, Real + Synthetic, 100, Reg2) are reported. ECG baseline Model (ECG, Real, 100) reported in Table 6.1 had lower classification accuracy than models augmented and trained with GES NoReg, Reg1, and Reg2 synthetic data but showed better specificity and sensitivity than the NoReg model. All evaluated models had similar specificity and sensitivity.

Models Reg1 and Reg2 had the lowest specificity and sensitivity difference (~5%) and the best overall classification accuracy of 97.50%. They also showed equal performance outcomes which suggest that the regularization terms were not different in how they influenced variation in the synthetic data. GES regularization terms improved the classification performance when compared to the baseline model, even though the specificity and sensitivity disparity remained to a lower extent.

Therefore, it affirms that the quality validated GES synthetic data regardless of the regularization option achieves comparable performance to real data. Additionally, the varying classification outcomes observed in this experiment show the importance of having flexibility in the synthetic data generation process. A closer inspection at the specificity and sensitivity of each specific ECG class showed a trade-off of accuracy between PVC and NSR classes, and this observation motivates our next experiment. The trade-off between these classes is also observed in the work of (Hou et al., 2020).

ResNet Model with Best Overall ECG Classification Performance

We recall that in the experiment *Balanced Real Data Training Sets*, we visually compared obvious structural patterns in ECG classes NSR and PVC, which are the classes that influenced the performance disparities. To investigate the disparities, we trained and analyzed 33 ECG models to find a model that resulted in better specificity and sensitivity balance for NSR and PVC. These models were trained with varying compositions of real-world and GES synthetic data. These models were then ranked, based on the highest classification accuracy, sensitivity, and sensitivity for NSR and PVC separately.

Table 6.4. The top 5 ECG trained ResNet models sorted to show the best balance in specificity and sensitivity for class NSR and PVC

NSR Sorted Models		**NSR**		**AFIB**		**PVC**		**LBB**	
	Test Acc. (%)	**Spec (%)**	**Sens (%)**	**Spec (%)**	**Sens (%)**	**Spec (%)**	**Sens (%)**	**Spec (%)**	**Sens (%)**
Model (ECG, Synthetic, 822, [NoReg,Reg1, Reg2])*	97.08	**97.78**	**86.67**	97.78	100	96.67	90.00	100	100
Model (ECG, Synthetic, 411, NoReg)	96.67	96.67	86.67	100	96.67	94.44	90.00	100	100
Model (ECG, Synthetic, 2056, NoReg)	97.08	**93.33**	**96.67**	100	93.33	98.89	86.67	100	100
Model (ECG, Real + Synthetic, 500, [NoReg, Reg1, Reg2])	96.25	100	73.33	100	96.67	91.11	100	98.89	100
Model (ECG, Real + Synthetic, 100, Reg2)	96.25	100	73.33	100	96.67	91.11	100	98.89	100
PVC Sorted Models		**NSR**		**AFIB**		**PVC**		**LBB**	
	Test Acc. (%)	**Spec (%)**	**Sens (%)**	**Spec (%)**	**Sens (%)**	**Spec (%)**	**Sens (%)**	**Spec (%)**	**Sens (%)**
Model (ECG, Real,514)	95.00	98.89	70.00	100	90.00	97.78	100	90.00	100
Model (ECG, Synthetic, 2056, NoReg)	97.08	93.33	96.67	100	93.33	**98.89**	**86.67**	100	100
Model (ECG, Synthetic, 822, [NoReg, Reg1, Reg2])*	97.08	97.78	86.67	97.78	100	**96.67**	**90.00**	100	100
Model (ECG, Synthetic, 1234, NoReg)	95.00	95.56	76.67	97.78	96.67	96.67	86.67	96.67	100
Model (ECG, Real + Synthetic, 500, [NoReg, Reg1, Reg2])	96.25	100	73.33	100	96.67	91.11	100.00	98.89	100

The top 5 models for NSR and PVC are listed in Table 6.4. To obtain the most balanced model for ECG classification, the models that are common among the two lists are examined. In Table 6.4, we show the model that best balances the disparities in NSR, and PVC classification performance was Model (ECG, Synthetic, 822, [NoReg, Reg1, Reg2]). We trained this model entirely on a combination of equal volume (274 samples) of NoReg, Reg1, and Reg2 GES synthetic data. The model had overall classification accuracy of 97.08% and specificity, sensitivity difference of ~11% for NSR, and ~6.7% for PVC.

The average sensitivity and specificity for this model is 94% and 98% respectively, confirming that this model lowers the disparity to 4%, and is the best model among all the models we trained. Based on experimental results, models trained with real-world ECG data prioritized learning class PVC over NSR, whereas GES synthetic data in Model (ECG, Synthetic, 822, [NoReg, Reg1, Reg2]) generalized both classes thus better trade-off in performance metrics. Another model with similar results is the Model (ECG, Synthetic, 2056, NoReg), which too was trained on GES synthetic data.

Data Complexity and Training Volume

As mentioned earlier, HTSDs come with varying degrees of complexities and challenges. The two HTSDs we chose for this paper are EEG and ECG. Table 5.1 shows that while ECG is structurally complex, EEG is complex in terms of dimensionality. EEG has more units of information/samples per patient than the ECG data. Here, we investigate how these two differing complexities interact with synthetic training data. To do so, we compared all the concerning ECG and EEG datasets models listed in Table 6.1 through Table 6.4 and introduce two new models for the EEG dataset in in Table 6.5.

When the models are trained with real data only, we find that a lower number of training samples (in the 100 or 200 range) is enough to prevent overfitting for ECG data (Table 6.1). However, for the EEG data, training samples in the range of 93 (Table 6.5), 160, or even 260 (Table 6.1) are not sufficient for overcome overfitting. When validated GES synthetic data augments the training set of these models, we found that the performance of the ECG dataset improved to some extent while keeping the data sample size at 100.

However, improvement in the disparity of specificity and sensitivity of ECG models requires a much larger quantity of training data (in the 822 or 2056 range). For the EEG dataset (with dimensional complexity), however, the best model we found was for a data sample size of 186 (Table 6.5) with combined real and GES synthetic data. We note that even though we get a sufficiently good model with a 160 EEG training sample (real and GES synthetic combined), just increasing the number of training samples from 160 to 186 increases the classification accuracy by over 3%. Thus, we concluded that validated synthetic data, it is an effective option for augmenting small dataset for training machine learning models.

Table 6.5. ResNet performance for EEG models trained with real only and real + synthetic data

Trained Models	Train Acc. (%)	Test Acc. (%)	Spec (%)	Sens (%)
Model(EEG, Real, 93)	93.68	77.42	75.21	75.21
Model(EEG, Real + Synthetic, 186)	99.17	96.77	96.15	96.15

Comparing ResNet ECG Classification Performance with Related Works

Here, we compared ECG classification performance of published related work that reported record-based classification of ECG signals. Literature studies show that ECG classification tasks disproportionately employ beat-based classification over record-based classification. Record-based models are trained with multiple heartbeat recordings whereas beat-based models are trained with single heartbeats. For this reason, we compared our work with the work in (Hou et al., 2020), which proposed a deep learning method that integrates a Support Vector Machine (SVM) classifier with a Long Short-Term Memory (LSTM) based auto encoders (AE) for ECG record-based classification for similar ECG target classes in our work.

In Table 6.6, classification outcomes from Model (ECG, Real + Synthetic, 100, Reg2) demonstrated higher overall model accuracy of 96.25% for the four ECG target classes compared to 83.51% for LSTM-AE (Hou et al., 2020). In tandem with our work, we observed disparities in specificity and sensitivity performance in (Hou et al., 2020). Our model showed a ~4% difference in specificity and sensitivity compared to ~39% in LSTM-AE (Hou et al., 2020)

Table 6.6. ECG record-based comparative classification performance for related work and a GES ResNet model

	(Hou et al., 2020) LSTM-AE + SVM		**(Maweu[2] et al., 2021)** Model (ECG, Real + Synthetic, 100, Reg2)	
ECG Class	**Spec (%)**	**Sens (%)**	**Spec (%)**	**Sens (%)**
NSR	37.02	98.55	100	73.33
AFIB	99.58	1.70	100	96.67
PVC	90.89	71.50	91.11	100
LBB	99.71	0	98.89	100
	Acc. (%): 83.51 Spec (%): 81.80 Sens (%): 42.94		Acc. (%): **96.25** Spec (%): **97.50** Sens (%): **92.50**	

6.4 Summary

We successfully demonstrated the effectiveness of validated synthetic data through observed improvements in classification performances from deep neural networks. The central hypothesis that guided the presented experimental designs is if quality synthetic data, i.e., synthetic data validated and tested for retained distinguishable class-level features, then this synthetic data can be used to train learning models and should improve the performance accuracy, specificity, and sensitivity of state-of-the-art deep networks. The rationale for this hypothesis was motivated by the promising results from preliminary work with traditional machine learning models (6.3.1) and non-residual networks (6.3.2).

We presented a total of eight experiments using ResNets to classify EEG and ECG datasets, as detailed in section Table 5.1. These experiments provided evidence that supports (i) the importance of using quality of synthetic data in terms of preserving class-distinguishable features, (ii) that the EEG diagnostic model performed better when trained with synthetic data (even with a lower volume of training data compared to other similar models), (iii) that it is possible to obtain better ECG diagnostic models using validated synthetic data, and (iv) models trained with validated quality synthetic data handled the observed NSR-PVC trade-off better. Additionally, the experiments showed that users can better address class imbalances with validated and quality synthetic data rather than utilizing under-sampling, which tends to cause overfitting in deep network models. Future investigations on synthetic data would be advancing personalized healthcare with methods that build a personalized repository of privacy-aware synthetic data.

APPENDIX A

SUPPLEMENTAL WORK FOR CHAPTER 6

ANALYSIS OF HUMAN ACTIVITY DATASET

A.1. Activity Recognition Multisensors (AReM) Dataset

Human activity recognition datasets are sequences of motion collected using wearable accelerometers sensor. The AReM dataset from (Palumbo et al., 2016) is a real-life benchmark of seven types of human activities from a wireless sensor network. The human activities in the dataset are bending (2 types as B1 and B2), cycling (CY), lying (LY), siting (SI), standing (ST), and walking (WA). The dataset consists of 87 participants with each having 6 compressed 480 samples long sequences. Each sequence is a mean of the original data over 250ms.

Table A.1. AReM Dataset Description

Dataset	Shape	Size	Samples/ Instance	Total Instances	Structural Complexity
AReM	2D	(480,6)	2880	87	XYZ Correlation

A.2. AReM Statistical Analysis

Here, we consider the real data and synthetic data as two different (treatment) groups and compute (i) *Descriptive Statistics,* and (ii) *The Wilcoxon Rank-Sum Test* to gain insights of similarities and differences present in the groups. We use descriptive statistics to measure and compare central tendency and variability, and rank-sum to test whether the treatment groups are from the same population distribution for a given statistical feature of the time series data.

Descriptive Statistics: We observe similar descriptive statistics on central tendency (mean and median) for all the dataset. We see similar variability statistics (variance and IQR) for AReM.

Variance and IQR values of real data are higher than those of synthetic data which suggests variability in AReM treatment groups. To test whether these differences are significant, we run the rank sum test next. If the differences are significant then that means the synthetic data generator was not able to preserve the statistical distribution of the associated class and that the automatic transfer of labels from real-world data to synthetic is not valid.

Table A.2. AReM descriptive statistics for real and synthetic data

AReM Sensors	**p-value**	**Mean**	**Variance**	**Median**	**IQR**	**Min**	**Max**
Sensor 1 (Real)		18.03	252.32	15.00	32.25	0.00	51.25
Sensor 1(Synthetic)	0.7759	18.19	248.97	15.52	31.70	-3.97	51.25
Data Aug	0.7957	18.53	264.92	15.42	33.30	0.00	54.54
Sensor 2 (Real)		6.59	66.85	2.49	11.62	0.00	40.33
Sensor 2(Synthetic)	0.4712	6.81	66.28	3.05	12.13	-2.98	44.77
Data Aug	0.2113	6.48	61.80	2.71	12.46	0.00	38.80

We generate overlay plots of randomly selected real and synthetic data for AReM X-Y-Z sensor axes. Visual analysis of AReM in Figure A.1 shows patterns from sensors three axes for class Bending1 (B1) activity. Synthetic data (blue plot line) shows patterns that are consistent with patterns in the real data (red plot line) and additionally, we visually observe that the XYZ coordinate correlation is maintained in the synthetic data.

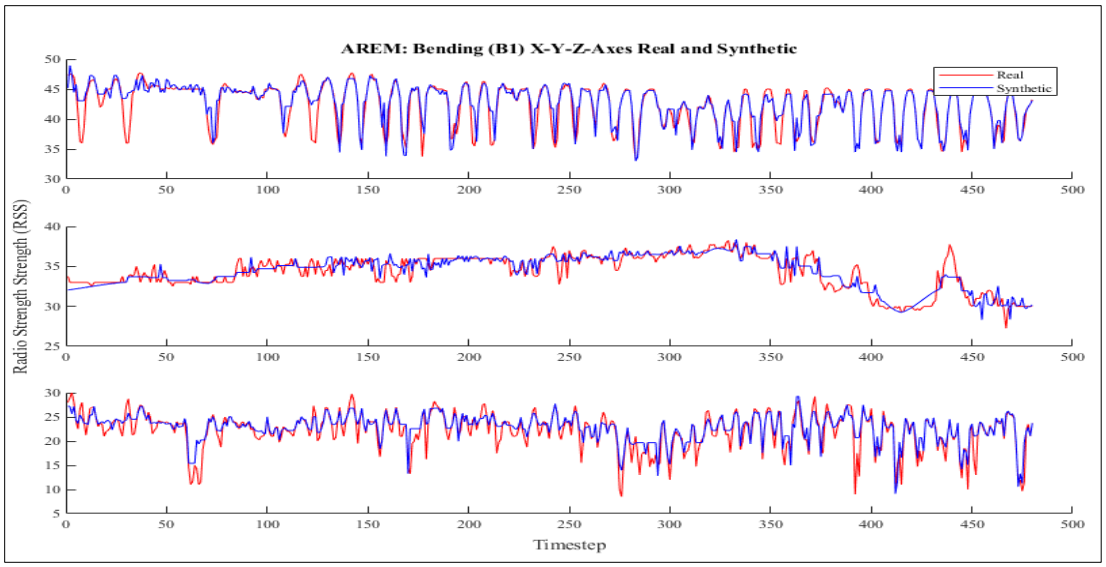


Figure A.1. AReM XYZ axes overlay plot for real and synthetic data

A.3. Comparing AReM Single Instance Correlation of Synthetic Data

In examining whether sensor correlations are preserved during the synthesis process, we demonstrate correlation relationship of AReM multi-dimensional variables (e.g. biosensors) for a single instance of AReM real world and the corresponding GES synthetic data in correlation matrices. Table A.3 shows same class variable correlation (on the diagonal) between AReM real and synthetic sensor axis data. We observe similar high correlation between real and corresponding synthetic variables for these samples. The MSE mentioned in the caption of Table A.4 denotes the mean squared error between the upper triangular correlation matrices of the real and synthetic data for the individual.

A.3.1. Comparing AReM Correlation across Instances and Dataset

While MSE values (between the upper triangular correlation matrices of the real and synthetic data) close to zero are ideal we observe different behavior in the AReM dataset. Table

A.4 highlights the minimum and maximum MSE values we obtained from pairs of real-world AReM data instances from the entire dataset.

Table A.3. AReM correlation across dataset

B1	1.00						
B2	0.883	1.00					
CY	0.965	0.854	1.00				
LY	0.935	0.755	0.938	1.00			
SI	0.980	0.858	0.971	0.954	1.00		
ST	0.955	0.791	0.956	0.972	0.972	1.00	
WA	0.948	0.830	0.945	0.929	0.955	0.943	1.00
REAL	**B1**	**B2**	**CY**	**LY**	**SI**	**ST**	**WA**

B1	1.00						
B2	0.885	1.00					
CY	0.967	0.856	1.00				
LY	0.936	0.757	0.940	1.00			
SI	0.981	0.859	0.972	0.955	1.00		
ST	0.955	0.791	0.957	0.973	0.973	1.00	
WA	0.950	0.832	0.947	0.931	0.956	0.944	1.00
SYNTHETIC	**B1**	**B2**	**CY**	**LY**	**SI**	**ST**	**WA**

These values are minimum and maximum MSE values of 5.8140×10^{-5} and 1.9914 respectively. MSE values for randomly selected AReM patients (Table A.4) are closer to the ideal because this dataset was preprocessed by (Palumbo et al., 2016) and therefore we expect low MSE values. Since these MSE values for AReM instances fall within the respective MSE range

(calculated from pairs of real-world data instance) computed over the dataset, we conclude that the synthetic data generator successfully preserved the correlation between the biosensors when synthesizing the data.

Table A.4. AReM min-max real data MSE with synthetic data

AReM	**MSE of (Real & Synthetic Data)**
Patient 7	0.1513
Patient 9	0.1661
Patient 44	0.1535
Patient 63	0.1585
Patient 64	0.1631
Min Real Data MSE: *$5.8140x10^{-5}$*	
Max Real Data MSE: *1.9914*	

A.4. Obtaining the AReM Deep Neural Network Baseline Model

Model (AReM, Real, 70) show good overall test accuracy but shows significant differences in specificity and sensitivity values. Model sensitivity in AReM is low at 88.10% compared to 96.59% test accuracy and 97.75% specificity. The scenario of unbalanced dataset is more evident in AReM (with seven target classes) and volume of available train data is as low as 3 instances in target classes B1 and B2. Since the limiting reference number for AReM dataset is 3 and we cannot go either above or below it with using just real data, we do not implement a balanced AReM model.

A.5. AReM Synthetic Data Compared to Traditional Perturbation Techniques

We investigate the quality of AReM generated synthetic data by transforming real-world, labeled time series data to help deep learning models predict better. We do so by comparing

synthetic data to data obtained through data resampling /perturbation techniques. To obtain the augmented dataset, we applied moving average and noise to real training set.

Table A.5. AReM ResNet classification results for baseline model

Trained Models	Baseline	(%) Train Acc.	(%) Test Acc.	(%) Spec	(%) Sens
Model(AReM, Real, 70)	AReM	97.74	96.59	97.75	88.10

The comparison medium is the performance of two deep learning models, one trained on real and synthetic data (Real + Synthetic), and the other trained on real and resampled/perturbed data (Real + Resampled). Both models are tested on the same testing dataset consisting of only real-world data. We observe from Table A.6 that models trained on real and synthetic data using traditional perturbation techniques perform significantly lower than comparable models trained on real and synthetic data. In Table A.6, AReM test accuracy, specificity and sensitivity are lower than Model (AReM, Real + Synthetic, 70) which when tested on the unseen test set had 98.32%, 99.05%, and 95.24% performance.

Table A.6. AReM ResNet classification performance with data perturbation

Trained Models	(%) Test Acc.	(%) Spec	(%) Sens
Model (AReM, Real + Synthetic, 70)	**98.32**	**99.05**	**95.24**
Model (AReM, Real + Resampled, 70)	93.30	96.00	73.81

A.5. Investigating Whether Deep Models Trained on AReM Real-World Data can Properly Classify AReM Synthetic Data

If a model, which trained and learnt features from the real-world data, can also discriminate class features in the synthetic data that should provide evidence for success in the synthesis

process. However, since there were disparities in specificity and sensitivity, we do not expect very high testing accuracy. Instead, if the test accuracy is better than random and above the lowest observed test accuracy of ~80% from experiment in Table A.7, we say the synthesis process successfully synthesized the data.

Another goal here is to use the baseline models as the second-best alternative to having a medical expert in the team. Since most teams do not have the relevant medical expertise, all synthetic data generation are set to strict constraints that control variations in the synthetic data. For this reason, we experiment to see if the synthetic data exhibits real-world features and preserves distinguishing properties among the target classes.

With our goal for the experiment in mind, we see in Table A.7 that synthetic data achieved better than random test accuracy and higher than the lowest observed test accuracy of ~80%. These results show that class properties from real data are captured and maintained by during the synthesis process. Additional evidence of this comes from observation of pattern of specificity and sensitivity disparities seen in real data similarly present in the synthetic data performance metrics.

A.6. Effect of Using AReM Synthetic Data to Train Deep Model for Classification Task

AReM data does not present with dimensional or structural complexities but is a smaller dataset. Nonetheless, we see disparities in the specificity and sensitivity metrics in the AReM baseline model which is not a desirable outcome in healthcare. Therefore, we experiment with synthetic data and train deep models. In Table A.8 we organize class-wise specificity and sensitivity metrics of AReM deep models. AReM models demonstrates overall good predictive performance in all classes (except LY SI, and ST) and achieve 100% specificity and sensitivity.

Table A.7. AReM comparative analysis of ResNet baseline models tested on real and synthetic data

Baseline Model	(%) Tested on Real Data	(%) Tested on Synthetic Data
Model(AReM, Real,	Acc.: 96.59	Acc.: 97.07
	Spec: 97.75	Spec: 98.22
	Sens: 88.10	Sens: 87.03

In the real model, classes LY and SI have specificity of 91.67%, and 93.33% respectively and ST has sensitivity of 66.67%. We observe that models trained with synthetic data improved the specificity of LY sensitivity or SI while ST remained the same.

We also observe that by training on only on GES generated synthetic data, the model achieves improved specificity in LY, and SI sensitivity while the metrics from the remaining classes are unchanged. AReM models Real + Synthetic and Synthetic Only both had test accuracy of 98.32% therefore their individual performance across all target classes was equivalent and supports the positive influence of synthetic data in training deep model for improved prediction performance.

Table A.8. AReM comparing classification results for real, synthetic and real + synthetic models

Classes	Model(AReM, Real, 70)		Model(AReM, Synthetic, 70)		Model (AReM, Real + Synthetic, 70)	
	(%) Spec	(%) Sens	(%) Spec	(%) Sens	(%) Spec	(%) Sens
B1	100	100	100	100	100	100
B2	100	100	100	100	100	100
CY	100	100	100	100	100	100
LY	91.67	100	**100**	100	**100**	100
SI	93.33	50.00	93.33	**100**	93.33	**100**
ST	100	66.67	100	66.67	100	66.67
WA	100	100	100	100	100	100

APPENDIX B

DISCRIMINATOR MODEL FOR SYNTHETIC DATA VALIDATION

We experimented with two binary classifiers (non-residual and residual architectures) and trained them to discriminate between ECG synthetic data (Class 0) and real-world data (Class 1). We hypothesize that good synthetic samples are indistinguishable from real-world samples with a discriminator with better than random accuracy. We trained our binary classifiers with 460 samples each from Class 0 and Class 1.

B.1. CNN Model

We trained several CNN models with the architecture described in section 3.3.1, and the parameters in Table B.1. The best achieved model accuracy was 49.74% accuracy which was basically random for classifying the real and synthetic data. The model was tested on 192 samples of real and synthetic data each. The classification results with this model are shown in Figure B.1.

Table B.1. CNN architecture model parameters

Total params:	3,246,530
Trainable params:	3,243,530
Non-trainable params:	0

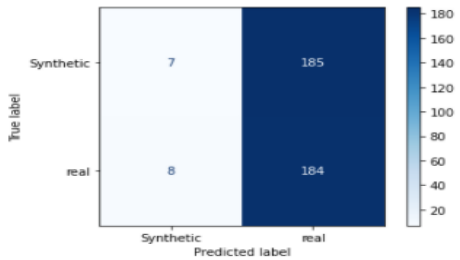


Figure B.1. CNN binary classification results for real and synthetic data

B.2. ResNet Model

The poor representation learning observed in the CNN model motivated training a ResNet model with the parameters in Table B.2 and the same training data described previously. With this model we achieved 99.74% accuracy with 0.0040 Kullback-Leibler Divergence loss. The results shows that the two classes were highly separable by this model.

Table B.2. ResNet architecture model parameters

Total params:	1,685,346
Trainable params:	1,681,154
Non-trainable params:	4,192

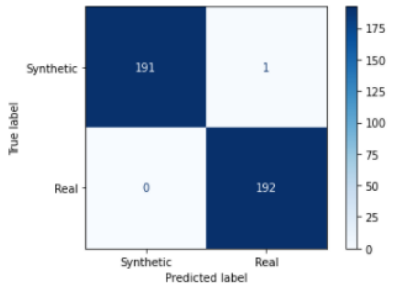


Figure B.2. ResNet binary classification results for real and synthetic data

We find that both the binary classifiers with the training parameters shown in Table B.1 and Table B.2, the model performance was not stable over the *k*-10 fold cross validation. The observed random accuracy in the non-residual network classifier was very different from the accuracy of the residual network. To understand our learning problem, we decided to position the discriminator model as a *Fine Grained Classification Problem* rather than *Binary Classification Problem.* Approaching our classification as a fine grained classification problem means we seek that the learning network is able to distinguish the real data and synthetic data as classes that share similar structure and primarily have subtle differences at a localized level.

APPENDIX C

GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) are generative models that uses two deep neural networks (a generator and a discriminator) to learn representations, discover patterns and generate new data instances from real-world data. The *generator* network accepts as input random noise or conditioned noise from some distribution and is responsible for creating new data samples. Meanwhile, the *discriminator* network is trained to discriminate between the real data and synthetic samples produced by the generator network.

C.1. DTW Vanilla GAN

We implemented a DTW Vanilla GAN network Figure C.1 for generating ECG signals. The model computed SoftDTW loss in the generator and binary cross entropy in the discriminator with the following model training configurations:

Configurations Settings

{"GPU_device": "0"
"model_type": "ecg_dtw_gan",
"discriminator_lr": 0.0002,
"valid_data": ". /valid.csv",
"minibatch_kernel_dims": 16,
"generator_lstm_hidden_units": 50,
"**feature_range": [-1, 1],**
"Epochs": 2000,
"discrimiantor_rounds_per_epoch": 3,
"simulation_directory": ". /sim01",
"generator_lr": 0.0002,
"train_data": ". /**NSR**.csv",
"minibatch_nb_kernels": 5,
"discriminator_final_activation": "sigmoid",
"generator_final_activation": "**tanh**",
"batch_size": 24,
"generator_rounds_per_epoch": 1,
"num_visualize_samples": 10}

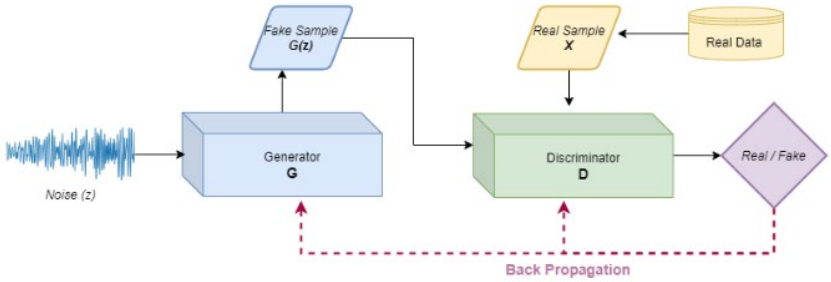


Figure C.1. DTW GAN model

We observed the following generated samples Figure C.2 which did not capture the structural characteristics of ECG signals.

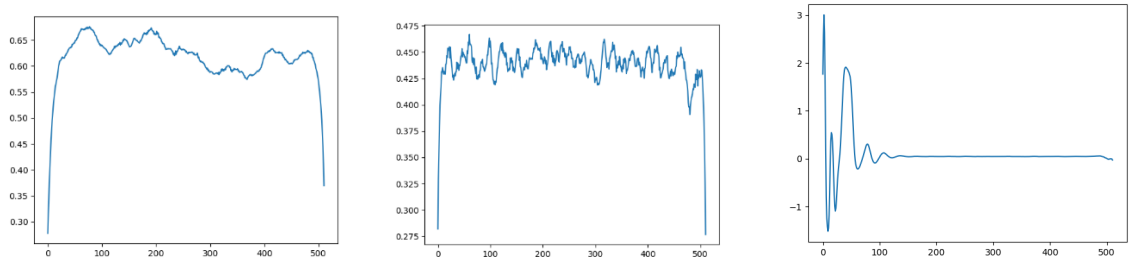


Figure C.2. DTW Vanilla GAN generated samples

C.2. Self-Attention GAN

In this experiment, we implemented a Self-Attention GAN Figure C.3 for ECG signals. We trained the generator using a discriminator frozen combined model and the discriminator loss was set to Wasserstein loss. Additionally, we computed the following metrics at each epoch because DTW and Maximum Mean Discrepancy (MMD) were costly to compute:

- dtw_metric = Fast DTW (Salvador et al., 2004)
- mmd_metric = Maximum Mean Discrepancy (MMD) Loss (Fortet et al., 1953)
- kld_metric = tensorflow metric – KLDivergence

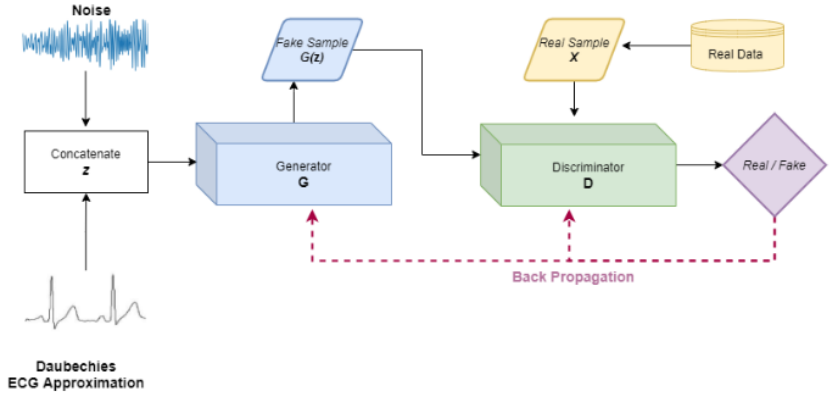


Figure C.3. GAN Conditioned on Daubechies ECG Approximation

The model was trained with the following training configurations:

{"GPU_device": "0"
"model_type": "ecg_self_attention_gan",
"discriminator_lr": 0.0002,
"valid_data": "./valid.csv",
"minibatch_kernel_dims": 16,
"generator_lstm_hidden_units": 50,
"feature_range": [0, 1],
"Epochs": 2000,
"discrimiantor_rounds_per_epoch": 3,
"start_epoch": 0,
"simulation_directory": "./sim02",
"generator_lr": 0.0002,
"train_data": "./**NSR**.csv",
"minibatch_nb_kernels": 5,
"discriminator_final_activation": "sigmoid",
"generator_final_activation": **null**,
"batch_size": 32,
"generator_rounds_per_epoch": 1,
"num_visualize_samples": 10,
"num_duplicates": 2}

We observed the following generated samples Figure C.4 which after about 2000 epochs showed the model progressively learning the cyclical peaks present in ECG signal but did not fully capture the well-defined waveforms.

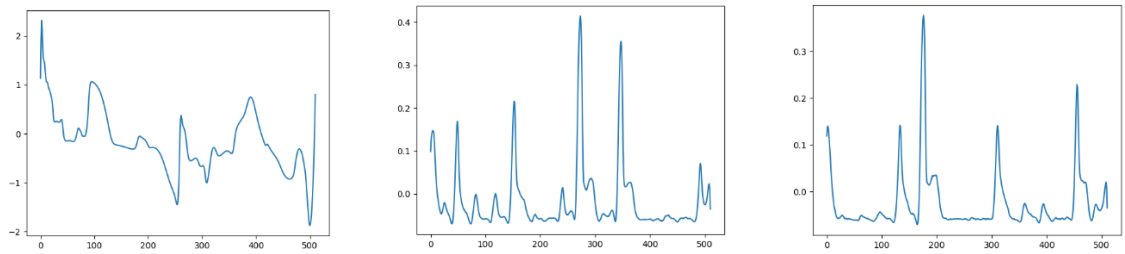


Figure C.4. Self-Attention GAN generated samples

REFERENCES

Alzantot, M., Chakraborty, S., & Srivastava, M. (2017). SenseGen: A deep learning architecture for synthetic sensor data generation. 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 188-193.

Beers, A., Brown, J., Chang, K., Campbell, J., Ostmo, S., Chiang, M., & Kalpathy-Cramer, J. (2018). High-resolution medical image synthesis using progressively grown generative adversarial networks. *ArXiv, abs/1805.03144*.

Begleiter, H. "EEG Database Dataset." *UCI Repo of ML DB* (1998).

Benaim, A.R., Almog, R., Gorelik, Y., Hochberg, I., Nassar, L., Mashiach, T., Khamaisi, M., Lurie, Y., Azzam, Z., Khoury, J., Kurnik, D., & Beyar, R. (2020). Analyzing Medical Research Results Based on Synthetic Data and Their Relation to Real Data Results: Systematic Comparison From Five Observational Studies. JMIR Medical Informatics, 8.

Bogle, B.M., & Mehrotra, S. (2016). A Moment Matching Approach for Generating Synthetic Data. Big data, 4 3, 160-78.

Belle, A., Thiagarajan, R., Soroushmehr, S., Navidi, F., Beard, D., & Najarian, K. (2015). Big Data Analytics in Healthcare. BioMed Research International, 2015.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res., 16*, 321-357. Addison, P. (2005). Wavelet transforms and the ECG: a review. *Physiological measurement, 26 5*, R155-99.

Chen, J., Chun, D., Patel, M., Chiang, E., & James, J. (2019). The validity of synthetic clinical data: a validation study of a leading synthetic data generator (Synthea) using clinical quality measures. BMC Medical Informatics and Decision Making, 19.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W., & Sun, J. (2017). Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. MLHC.

DTW: Dynamic Time Warping, https://www.mathworks.com/help/signal/ref/dtw.html; R2021a [accessed 12 May 2021].

Fortet, R., & Mourier, E. (1953). Convergence de la répartition empirique vers la répartition théorique. Annales Scientifiques De L Ecole Normale Superieure, 70, 267-285.

Galen, R., & Gambino, S. (1975). Beyond Normality: The Predictive Value and Efficiency of Medical Diagnoses.

Gautam, A., & Kaur, M. (2012). ECG Analysis using Continuous Wavelet Transform (CWT). *IOSR Journal of Engineering, 02*, 632-635.

Goldberger, A., Amaral, L., Glass, L., Hausdorff, J.M., Ivanov, P., Mark, R., Mietus, J., Moody, G., Peng, C., & Stanley, H. (2000). PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation, 101 23*, E215-20.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Nets. NIPS.

Guibas, J.T., Virdi, T.S., & Li, P. (2017). Synthetic Medical Images from Dual Generative Adversarial Networks. *ArXiv, abs/1709.01872*.

Harumi, K., Tsunakawa, H., Nishiyama, G., Wei, D., Yamada, G., Okamoto, Y., & Musha, T. (1989). Clinical application of electrocardiographic computer model. Journal of electrocardiology, 22 Suppl, 54-63. Addison, P. (2005). Wavelet transforms and the ECG: a review. *Physiological measurement, 26 5*, R155-99.

Holland, J. (1975). Adaptation in natural and artificial systems.

Hou, B., Yang, J., Wang, P., & Yan, R. (2020). LSTM-Based Auto-Encoder Model for ECG Arrhythmias Classification. *IEEE Transactions on Instrumentation and Measurement, 69*, 1232-1240.

Ilin, R., Watson, T.P., & Kozma, R. (2017). Abstraction hierarchy in deep learning neural networks. 2017 International Joint Conference on Neural Networks (IJCNN), 768-774.

Karlsson, I., Rebane, J., Papapetrou, P., & Gionis, A. (2018). Explainable Time Series Tweaking via Irreversible and Reversible Temporal Transformations. 2018 IEEE International Conference on Data Mining (ICDM), 207-216.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C.J., Wexler, J., Viégas, F.B., & Sayres, R. (2018). Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). ICML.

Kramer, S., Lavrac, N., & Flach, P.A. (2001). Propositionalization approaches to relational data mining.

Lin, Z., Khetan, A., Fanti, G., & Oh, S. (2020). PacGAN: The Power of Two Samples in Generative Adversarial Networks. IEEE Journal on Selected Areas in Information Theory, 1, 324-335.

Lundberg, S.M., & Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. NIPS.

Maweu[1], B.M., Dakshit, S., Shamsuddin, R., & Prabhakaran, B. (2021). CEFEs: A CNN Explainable Framework for ECG Signals. *Artificial Intelligence in Medicine, 115*, 102059.

Maweu[2], B.M., Shamsuddin, R., Dakshit, S., & Prabhakaran, B. (2021). Generating Healthcare Time Series Data for Improving Diagnostic Accuracy of Deep Neural Networks. *IEEE Transactions on Instrumentation and Measurement, 70*, 1-15.

Medical Tests – ECG, http://www.bloodpressureuk.org/BloodPressureandyou/Medicaltests/ECG/; 2008 [accessed 12 May 2021].

Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. (2018). Deep learning for healthcare: review, opportunities and challenges. Briefings in bioinformatics, 19 6, 1236-1246.

Montavon, G., Samek, W., & Müller, K. (2018). Methods for interpreting and understanding deep neural networks. Digit. Signal Process. 73, 1-15.

Moreno-Barea, F.J., Strazzera, F., Jerez, J.M., Urda, D., & Franco, L. (2018). Forward Noise Adjustment Scheme for Data Augmentation. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 728-734.

Nie, D., Trullo, R., Petitjean, C., Ruan, S., & Shen, D. (2017). Medical Image Synthesis with Context-Aware Generative Adversarial Networks. *Medical image computing and computer-assisted intervention: MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention, 10435*, 417-425.

Palumbo, F., Gallicchio, C., Pucci, R., & Micheli, A. (2016). Human activity recognition using multisensor data fusion based on Reservoir Computing. J. Ambient Intell. Smart Environ. 8, 87-107.

Patki, N., Wedge, R., & Veeramachaneni, K. (2016). The Synthetic Data Vault. 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 399-410.

Pijn, J.P., Neerven, J.V., Noest, A., & Silva, F.H. (1991). Chaos or noise in EEG signals; dependence on state and brain site. *Electroencephalography and clinical neurophysiology, 79 5*, 371-81.

Plawiak, P. (2017).“ECG signals (1000 fragments)”, Mendeley Data, v3 http://dx.doi.org/10.17632/7dybx7wyfn.3

Read ECG - How to read an Electrocardiogram, http://www.southsudanmedicaljournal.com/archive/may-2010/how-to-read-an-electrocardiogram-ecg.-part-one-basic-principles-of-the-ecg.-the-normal-ecg.html; 2010 [accessed 12 May 2021].

Ribeiro, M.T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Salvador, S., & Chan, P. (2004). FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space.

Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., & Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision, 128*, 336-359.

Shaikhina, T., & Khovanova, N. (2017). Handling limited datasets with neural networks in medical applications: A small-data approach. Artificial intelligence in medicine, 75, 51-63.

Shamsuddin, R., Maweu, B.M., Li, M., & Prabhakaran, B. (2018). Virtual Patient Model: An Approach for Generating Synthetic Healthcare Time Series Data. 2018 IEEE International Conference on Healthcare Informatics (ICHI), 208-218.

Shin, H., Tenenholtz, N.A., Rogers, J.K., Schwarz, C., Senjem, M., Gunter, J., Andriole, K., & Michalski, M.H. (2018). Medical Image Synthesis for Data Augmentation and Anonymization using Generative Adversarial Networks. *SASHIMI@MICCAI.*

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. CoRR, abs/1312.6034.

Sturm, I., Bach, S., Samek, W., & Müller, K. (2016). Interpretable deep neural networks for single-trial EEG classification. Journal of Neuroscience Methods, 274, 141-145.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. ArXiv, abs/1703.01365.

Torkzadehmahani, R., Kairouz, P., & Paten, B. (2019). DP-CGAN: Differentially Private Synthetic Data and Label Generation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 98-104.

Trayanova, N. (2011). Whole-heart modeling: applications to cardiac electrophysiology and electromechanics. *Circulation research, 108 1*, 113-28.

Vidal, E., Casacuberta, F., & Segovia, H. (1985). Is the DTW "distance" really a metric? An algorithm reducing the number of DTW comparisons in isolated word recognition. *Speech Commun., 4*, 333-344.

Walonoski, J.A., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., & McLachlan, S. (2018). Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. Journal of the American Medical Informatics Association: JAMIA, 25, 230 - 238.

Wei, D. (1997). Whole-heart modeling: progress, principles and applications. Progress in biophysics and molecular biology, 67 1, 17-66.

Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling Tabular data using Conditional GAN. NeurIPS.

Ye, L., & Keogh, E.J. (2010). Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Mining and Knowledge Discovery, 22, 149-182.

Yildirim, Ö., Plawiak, P., Tan, R., & Acharya, U. (2018). Arrhythmia detection using deep convolutional neural network with long duration ECG signals. *Computers in biology and medicine, 102*, 411-420.

Zhang, Q., Wu, Y., & Zhu, S. (2018). Interpretable Convolutional Neural Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8827-8836.

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2921-2929.

Zhu, W., Zeng, N.F., & Wang, N. (2010). 1 Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS.

BIOGRAPHICAL SKETCH

Barbara Maweu completed her PhD in Software Engineering at The University of Texas at Dallas with a research focus on artificial intelligence in healthcare. Her research focuses on investigating synthetic healthcare data and developing computational techniques that validate synthetic data for use in artificial intelligent systems.

Currently, she is working on developing human interpretable explainability methods in deep network models for healthcare decision support systems and applying new active learning algorithms to healthcare datasets.

Barbara is actively involved in issues relating to women and minorities in technology and she is a participating member of *Women Who Compute* (WWC) a University of Texas as Dallas club, was a student scholar at the Grace Hopper Conference and seeks organizations that promotes women in all levels of technology. She hopes to pursue a faculty positon and continue research and mentorship initiatives. In her free time, she volunteers at local coding event and enjoys traveling.

CURRICULUM VITAE

Barbara Maweu

Contact Information:

Department of Computer Science Email: barbara.maweu@utdallas.edu

The University of Texas at Dallas

800 W. Campbell Rd.

Richardson, Texas 75080-3021, U.S.A.

Educational History:

Ph.D. Candidate, Software Engineering, University of Texas at Dallas, 2021

M.S., Software Engineering, University of Texas at Dallas 2016

B.S., Software Engineering, University of Texas at Dallas, 2014

Validation and Explanations of Synthetic Healthcare Time Series Data for Improved Applications in Deep Based Decision Support Systems

Ph.D. Dissertation

Department of Computer Science, The University of Texas at Dallas

Advisor: Dr. Prabhakaran Balakrishnan

Employment History:

Teaching Assistant, The University of Texas at Dallas, August 2016 – Present

(Computer Graphics, Software Engineering, Software Evolution and Maintenance, Software Project Planning and Management, Digital Logic and Computer Design, Database Systems and Design, Programming Fundamentals and Lab).

Graduate Research Assistant, The University of Texas at Dallas, August 2018 – Present

Software Engineer, iHealth Technologies, March 2003 – October 2010

Publications:

B. M. Maweu, R. Shamsuddin, S. Dakshit and B. Prabhakaran, "*Generating Healthcare Time Series Data for Improving Diagnostic Accuracy of Deep Neural Networks*," 2021 IEEE Transactions on Instrumentation and Measurement, doi: 10.1109/TIM.2021.3077049.

B. M. Maweu, S. Dakshit, R. Shamsuddin, and B. Prabhakaran, "*CEFEs: A CNN Explainable Framework for ECG Signals*," 2021 Artificial Intelligence in Medicine, 115, 102059, doi: 10.1016/j.artmed.2021.102059.

R. Shamsuddin, B. M. Maweu, M. Li and B. Prabhakaran, "*Virtual Patient Model: An Approach for Generating Synthetic Healthcare Time Series Data*," 2018 IEEE International Conference on Healthcare Informatics (ICHI), 2018, pp. 208-218, doi: 10.1109/ICHI.2018.00031.