MIWGAN-GP: MISSING DATA IMPUTATION USING WASSERSTEIN GENERATIVE ADVERSARIAL NETS WITH GRADIENT PENALTY

MIWGAN-GP: EKSİK VERİLERİN GRADYAN CEZALANDIRMALI WASSERSTEIN ÇEKIŞMELİ SİNİR AĞLARI İLE TAMAMLANMASI

Ebru UÇGUN ERGÜN

PROF. DR. SUAT ÖZDEMİR

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Master of Science

in Computer Engineering

June 2022

ABSTRACT

MIWGAN-GP: MISSING DATA IMPUTATION USING WASSERSTEIN GENERATIVE ADVERSARIAL NETS WITH GRADIENT PENALTY

Ebru UÇGUN ERGÜN

Master of Science, Computer Engineering

Supervisor: Prof. Dr. Suat ÖZDEMİR

June 2022, 68 pages

The success and dependability of IoT applications are heavily dependent on data quality. Due to hardware problems, synchronization challenges, inconsistent network connectivity, and manual system shutdown, produced data might be missing, erroneous, and noisy. These missing or erroneous values can also occur on health, military and surveillance data and result in errors can also cause important errors in mission systems. If the mission critical system is used in medical domain such missing data problems may affect human life. Hence, Missing values should be imputed appropriately to avoid erroneous judgments in IoT healthcare systems and other critical systems.

In addition, Naive Bayes, K-Nearest Neighbors, Decision Tree and XGboost algorithms are applied in the IoT health sector in this study to show in detail the effect of missing data on the outputs of machine learning algorithms. Following that, we compare different strategies for imputing missing data. The classification methods used were compared both for each defect percentage and with different imputation methods.

In this thesis, a new GAN-based approach is proposed to complete the missing data. The success of the proposed method is compared with classical imputation methods. Error

measurements are realized with four different error metrics. In addition, the success of the proposed GAN-based model is demonstrated by applying different classification methods on the data set filled with this method.

Keywords: Missing Data, Missing Data Imputation, Internet of Things, Deep Learning, Machine learning, Generative Adversarial Networks, GAN, Wasserstein GAN

ÖZET

MIWGAN-GP: EKSİK VERİLERİN GRADYAN CEZALANDIRMALI WASSERSTEIN ÇEKIŞMELİ SİNİR AĞLARI İLE TAMAMLANMASI

Ebru UÇGUN ERGÜN

Yüksek Lisans, Bilgisayar Mühendisliği

Danışman: Prof. Dr. Suat ÖZDEMİR

Mayıs 2022, 68 sayfa

IoT uygulamalarının başarısı ve güvenilirliği büyük ölçüde veri kalitesine bağlıdır. Donanım sorunları, senkronizasyon zorlukları, tutarsız ağ bağlantısı ve manuel sistem kapatma nedeniyle üretilen veriler eksik, hatalı ve gürültülü olabilir. Bu eksik veya hatalı değerler sağlık, askeri ve gözetleme verisetlerinde de oluşabilmekte ve bu verilerin kullanıldığı görev sistemlerinde de önemli hatalara neden olabilmektedir. Kritik görev sistemi; tıbbi alanda kullanılıyorsa, bu tür eksik veri sorunları insan hayatını etkileyebilir. Bu nedenle, IoT sağlık sistemlerinde ve diğer kritik sistemlerde hatalı yargılardan kaçınmak için Eksik veriler uygun şekilde doldurulmalıdır.

Bu çalışmada verilerin eksik olmasının makine öğrenmesi algoritmaları üzerindeki etkilerini göstermek için IoT sağlık verileri üzerinde Naive Bayes, K-Nearest Neighbors, Decision Tree ve XGboost algoritmaları uygulanmıştır. Bunu takiben, eksik verileri doldurmak için farklı stratejiler uygulanmıştır. Kullanılan sınıflandırma yöntemleri hem farklı eksiklik yüzdeleri hem de farklı atama yöntemleri ile karşılaştırılmıştır.

Bu tezde, eksik verileri tamamlamak için GAN tabanlı yeni bir yaklaşım önerilmiştir. Önerilen yöntemin başarısı klasik atama yöntemleri ile karşılaştırılmıştır. Hata değerleri

dört farklı hata metriği ile ölçülmüştür. Ayrıca önerilen GAN tabanlı modelin başarısı, bu yöntemle doldurulan veri seti üzerinde farklı sınıflandırma yöntemleri uygulanarak gösterilmektedir.

Anahtar Kelimeler: Eksik Veri, Eksik Veri Tamamlama, Nesnelerin İnterneti, Derin Öğrenme, Makine Öğrenmesi, Üretken Modeller, GAN, Wasserstein GAN

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Prof. Dr. Suat ÖZDEMİR for guiding me and teaching me more with her thoughtful comments and recommendations. He backed up my views during my thesis. Without his assistance, writing this thesis would be difficult; I am grateful for his support.

I would also like to thank Asst. Prof. Dr. İbrahim KÖK for sharing his extensive knowledge with me during my dissertation and providing me with ongoing support in terms of resources and methodologies. Also he always made time for me during difficult moments.

In addition, I would like to express my gratitude to my wonderful family for their unwavering support throughout my thesis and master's programs, as well as throughout my entire academic career.

I would like to express my deepest gratitude to my beloved husband, who sheds light on me and always supports me with his motivation, knowledge and experience at every point I get stuck. He always motivated me to work when my energy was depleted and my desire to work decreased. I am also deeply indebted to him.

Finally, I would also like to thank my dear friend Canan SEVGİLİ. She always motivated and encouraged me throughout my master education and thesis writing period.

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

IOT	:	**I**nternet **O**f **T**hings
MAR	:	**M**issing **A**t **R**andom
MCAR	:	**M**issing **C**ompletely **A**t **R**andom
MNAR	:	**M**issing **N**ot **A**t **R**andom
MSE	:	**M**ean **S**quare **E**rror
RMSE	:	**R**oot **M**ean **S**quare
MAE	:	**M**ean **A**bsolute **E**rror
MAPE	:	**M**ean **A**bsolute **P**ercentage **E**rror
ECG	:	**E**lectro **C**ardio **G**ram
GRU	:	**G**ated **R**ecurrent **U**nit
GRUI	:	**G**ated **R**ecurrent **U**nit for data **I**mputation
GAIN	:	**G**enerative **A**dversarial **I**mputation **N**ets
IWAE	:	**I**mportance **W**eighted **A**uto **E**ncoder

1. INTRODUCTION

The Internet of Things (IoT) is a worldwide network of physical objects equipped with sensors, applications, as well as other technologies that connect and exchange information between devices and systems over the Internet. IoT allows for the smooth connection of sensors, actuators, and communications equipment, paving the way for new application development in a variety of fields including health, industrial, automotive, transportation, and the environment. The quantity of gadgets linked to the Internet grows progressively as the number of created apps grows rapidly. IoT provides for the seamless integration of sensors, controllers, and communications equipment in real-time applications. Smart technologies based on the IoT are starting to be used in autos, homes, and other infrastructure systems.

All connected devices generate massive amounts of data from their connected sensors. The generated data must be gathered, evaluated, interpreted, and supplied to the end-user rapidly in order for the applications to perform effectively and efficiently in accordance with their development aims [1]. In data analysis operations, the quantity and quality of the obtained data is critical [2]. This is especially important in applications that involve people's lives, require quick responses, and demand great service quality. Due to the nature of IoT, gathered data may be incorrect, missing, or noisy for a variety of causes, including collision, unreliable network connectivity, malfunctioning devices, and manual system closure [3].

IoT devices are used quite often in examining people's activities and health status. From the health data collected with the assistance of IoT devices, the critical conditions and treatments of people can be monitored. But It is not always possible to get a significant number of datasets that are free of missing data in the health monitoring field. Signals received from sensors are disrupted by hardware or software defects, resulting in failures. Simultaneously, algorithms that rely on motion tracking data often rely on full and labeled datasets, emphasizing the need for label and dataset integrity. In health monitoring, on the other hand, it is seen to be advantageous to limit the number of sensors that gather mobility data information.

Missing data imputation challenges have been solved in a variety of ways. One of them is the classic imputation methods. It can be used to effectively fill in the gaps in the dataset. Another filling method is the use of machine learning-based algorithms. In this thesis, missing data are filled with a Generative Adversarial (GAN) based imputation method. It is shown through many different error metrics that this method is more successful than the classical methods. In addition, Naive Bayes, K-Nearest Neighbors, Decision Tree and XGboost classification methods are applied to show how effective the proposed method is to fill in missing data in the data processing step. Compared to other classical imputation methods, the proposed methods showed superior success in imputing in missing data. In addition, the overall success of the model has been demonstrated because it was studied on two different data sets.

1.1. Scope Of The Thesis

This thesis mainly focuses on missing data problem in IoT data. To solve this problem, a new GAN based model is proposed in this thesis scope.

1.2. Contributions

In this research, we explain that missing data in datasets collected from IoT devices is an significant problem that needs to be addressed in the preprocessing step, and we propose a new GAN-based model, which is more successful than classical imputation methods in the manner of filling missing parts. The main contributions of this paper can be summarized as follows:

- We explain the implication of dataset deficiencies in data processing and the significance of filling in these absence during the preprocessing stage.
- We explain and provide examples of how missing data in datasets can be resolved using traditional methods.
- We develop and present a method that fills in the missing data created based on the GAN method.
- We compared the success of our method with the classical imputation methods using four different error metrics.
- In addition, classification successes were compared using classical imputation methods and six different classification algorithms on the data set imputed with our method which we call MiWGAN-GP.

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 explains our motivation, purpose, contributions and the scope of the thesis.
- Chapter 2 briefly summarizes the basic concepts associated with the scope of thesis.
- Chapter 3 documents the literature review.
- Chapter 4 explains the details of the developed method and gives formulations of the proposed MIWGAN-GP method.
- Chapter 5 presents the results of 2 different datasets implemented with the proposed method and analyzes these results.
- Chapter 6 summarizes the thesis and provides research directions.

2. BACKGROUND OVERVIEW

2.1. Internet of Things (IoT)

Kevin Ashton of Procter & Gamble [4] used the phrase ”Internet of Things” for the first time in 1999. IoT is a network of items embedded with sensors, software, and other associated technologies that are primarily used to connect, exchange, and transfer data with other devices and systems through the internet. Simple domestic things to complicated corporate and industrial equipment and technology are all possible [5].

Hundreds of billions of devices, items, and devices are now connected to the web due to the IoT. Data collection and sharing, as well as information exchange. According to Gubbi, et al. [6], the IoT is ”an connectivity of sensing and actuation devices, allowing their potential to communicate data between systems through a uniform framework and establishing a common operational picture for enabling creative applications.”

The IoT ambition is to change the Internet by building networks of billions of wirelessly recognized things devices that can interact with everything and everyone, not just each other, at any time and from anywhere. Increased RFID processing capability, more wireless sensor networks (WSNs), and storage space at reduced prices are one way to do this, resulting in the establishment of a highly fragmented public resource pool connected by a complex model of networks [7].

In fact, IoT communications may occur among people and their surroundings as well as between equipment. People, automobiles, computers, books, Televisions, cell phones, clothes, food, medication, passports, baggage, and other ordinary goods all need a unique identification to connect with one another in IoT networks [8].

Citizens, corporations, and government will all benefit greatly from the IoT. From assisting governments in lowering healthcare costs and increasing living quality while lowering CO2 emissions, increasing access to training in distant underdeveloped regions, and improving transportation projects are range widely.

2.2. Missing Data

Missing data concerns were divided into three types by Rubin [9]. Every data point, according to his idea, has a chance of being absent. The missing data mechanism, also known as the response mechanism, determines these probabilities. The process model is known as the missing data model or response model.

2.2.1. Missing Completely at Random Data (MCAR)

Researchers refer to MCAR process as entirely haphazard missingness. According to the official definition of MCAR, the likelihood of missing data on a variable Y is independent to all other variables measured and unrelated to the variable y directly. MCAR is a more stringent condition than MAR since it considers that missingness is completely unrelated to the data. [10].

2.2.2. Missing at Random Data (MAR)

The probability of missing data is the same only among groups identified by the observational data, the data is missing at random (MAR). MAR is a substantially larger classification than MCAR. A weighing scale, for example, may produce more missing values when placed on a soft surface than when placed on a hard surface. As a result, such data is not MCAR. If we know the type of surface and can assume MCAR within that type of surface, however, the data is MAR. Another use of MAR is when we take a sample from a population and the chance of being included is based on a known attribute. MAR is a more generic and practical alternative to MCAR. The MAR assumption is often used in modern missing data approaches [11].

2.2.3. Missing Not at Random Data (MNAR)

The acronym MNAR (not missing at random) is also used in the literature to describe the same notion. MNAR denotes that the likelihood of going missing changes for causes we don't know about. The weighing scale mechanism, for example, may wear down with time, resulting in more missing data as time passes, although we may be unaware of this.If the heavier items are measured later in time, we will get a skewed distribution of readings. MNAR considers the potential that the scale produces more missing values for larger items, a scenario that can be difficult to spot and manage. When those with weaker opinions react less often in public opinion polls, this is an example of MNAR. The most challenging case is MNAR. Finding more information about the reasons of missing data or performing what-if analysis to evaluate how sensitive the results are under other circumstances are two strategies for dealing with MNAR.

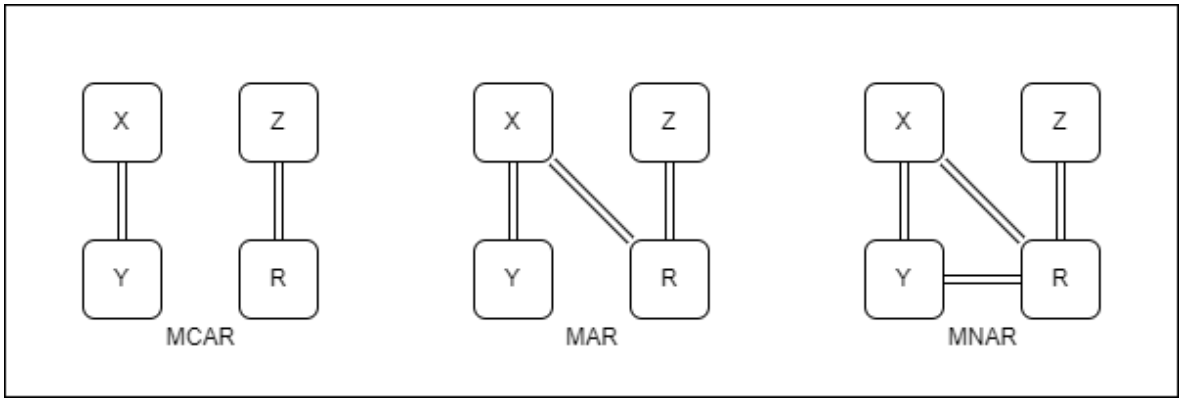

Figure 2.1 MCAR-MAR-MNAR

2.3. Traditional Imputation Methods

Imputation is appealing since it produces a full data collection. As a result, the ease of any single imputation approach is a big benefit. At first look, imputation appears to be favorable since it makes use of data that would otherwise be discarded by deletion methods. Despite their apparent benefits, single imputation approaches may have major disadvantages. Even in the ideal circumstance where the data is MCAR, the majority of techniques give biased

parameter estimations. The single exception is stochastic regression imputation, which is the only method that yields unbiased parameter estimates using MAR data. Furthermore, single imputation methods reduce standard errors. Missing values, on the surface, should raise standard errors since they introduce another layer of noise into parameter estimations. When examining a single imputed data set, however, the filled-in values are essentially treated as genuine data, so even the finest single imputation approach will underestimate sampling error [10].

2.3.1. Arithmetic Mean Imputation

Arithmetic mean imputation (also known as mean substitution and unconditional mean imputation) is a method for filling in missing data using the arithmetic mean of the available examples. The notion of using the mean to replace missing numbers is an ancient one that methodologists commonly credit to Wilks. Mean imputation, like other imputation approaches, is useful since it generates a full data set. However, even when the data is MCAR, convenience is not a convincing benefit because this strategy drastically affects the resultant parameter estimations.

2.3.2. Mod Imputation

Mode imputation is used by researchers to impute the variable's most common value. This type of imputation may properly forecast missing data, however it would alter the data set's features and create biased estimates. In any single imputation technique, error terms are undervalued, error bars are too narrow, and p-values are also too small, reflecting higher precision and proof than can be determined from the actual data. [12].

2.3.3. Hot Deck Imputation

Mode imputation is similar to Hot Deck imputation in that it employs an observed response from a comparable unit instead of the mode of a specific variable. Hot Deck imputation,

in other words, entails replacing missing data with seen values from a respondent who is comparable to the nonrespondent in terms of the attributes observed in both situations. Despite the fact that Hot Deck imputation imputes accurate values and is widely utilized in practice, it has flaws. It necessitates very excellent respondent matches that represent available covariate information, which can never be guaranteed, and the approach struggles to identify matches when the number of variables is huge [13].

2.3.4. Maximum Likelihood

Because of advances in processing capacity, more complex imputation approaches for handling missing data have been created, which, luckily, produce substantially better outcomes. In the methodological literature, the imputation techniques Maximum Likelihood and Multiple imputation, for example, are generally suggested. Since they yield unbiased estimates, these methods are seen to be preferable than the aforementioned missing data methods.

2.4. Machine Learning

Machine learning is the process of converting data into knowledge. A spam email, for example, cannot be recognized by looking at the presence of a single word; rather, looking at particular terms occurring in combination, the length of the email, and other similar criteria might assist you in recognizing it. Machine learning employs statistics as well, and it may be used to any issue that requires the interpretation and action of data, with the facts learnt subsequently being applied to a new batch of data [14].

Static programs are typically employed to tackle deterministic issues with certain solutions, but for situations that are not deterministic and lack sufficient data, we apply machine learning. Because there were insufficient datasets to train the algorithms, it was difficult to make realistic choices using machine learning in the beginning. However, with the rise of sensors and their ability to connect to the Internet, the true issue now is sorting through the avalanche of free data and using it to train machine learning algorithms [15].

The increased use of smartphones, which have numerous sensors such as accelerometers, GPS, and temperature sensors, has fueled the growth in data collecting. The present mobile computing and Internet of Things growth trends will result in the creation of increasingly relevant data in the future [16].

2.4.1. Supervised Learning

Supervised learning is when a model learns given input data (also known as training data) with given goal responses or labeling which might be a numeric value or a word.

A model is generated throughout the training or learning process that predicts the right behavior to a new example. Classification and regression are two forms of supervised techniques.

In classification, the algorithm guesses which class the test data belong to, while regression predicts a numerical value for a specific variable. Consider investing as a classification or regression problem, with the objective of teaching the computer how to make wealth-maximizing investment decisions [17].

2.4.2. Unsupervised Learning

When a model learns from data input without labeling and without a defined output, it is said to be unsupervised learning. To extract general principles from the input data, a model is built by learning the characteristics contained in the data. To remove duplication or arrange data by similarity, it is done by a mathematical technique.

Clustering, in which we group like things together, and density estimation, in which we identify statistical values that represent the date, are the two most common applications of unsupervised learning. Customer-targeted web adverts, for example, are built on this learning model, which makes recommendations based on your previous purchases.

The suggestions are based on determining which customer group you most closely match and then implying your anticipated preferences from that group [17].

2.5. Deep Learning

The terms ”artificial intelligence,” ”machine learning,” ”artificial neural networks,” and ”deep learning” are frequently interchanged in software development to denote the same or extremely comparable concepts and ideas. So it’s no surprise that, while being from separate eras and development periods, these concepts have something in common: a computer is given instructions to learn and discover the optimal solution to a problem, rather than instructions to solve a problem. This is in compared to conventional programming, which divides a larger problem into smaller jobs and instructions [18].

Deep learning is a sort of machine learning allows computers to learn from their errors and make sense of the world as a hierarchy of concepts. Because the computer learns via experience, no need for a human computer programmer to specifically specify all of the data that the computer wants. A network of these hierarchy would have numerous levels, enabling the computer to comprehend complex concepts by building them from smaller ones [19].

Since the 1990s, deep learning has been effectively used in commercial applications, but it was formerly seen as an art than a science, and only an expert could use it. True, some knowledge is required to get good outcomes from the a deep learning system. Fortunately, the level of knowledge required lowers as the quantity of training data accumulates. The algorithms that today attain human performance on difficult tasks are remarkably similar to those that struggled to solve toy problems in the 1980s, with the exception that the models that these methods train have been modified to make learning of very deep structures simpler [20].

2.6. Hyper-parameter Optimization

Hyper-parameter tuning is a critical activity that affects machine learning systems' ultimate performance. The difficulty of picking a set of acceptable hyper-parameters for a machine learning system is known as hyper-parameter optimization or tuning. A hyper-parameter is a learning process control parameter. On the other hand, some parameters, such as cluster weights, must be learnt [21].

The same machine learning model might demand specific restrictions, weight, or training rates to generalization various data patterns. These hyper-parameters must be fine-tuned so the model can tackle the machine learning task to its full potential. Hyper-parameter optimization determines a collection of hyper-parameters which leads to an ideal model that reduces a predetermined loss function on objective data. The cost linked with such a pair of hyper-parameters is returned by the objective function [21].

We used grid search to assign the most appropriate values to our hyper parameters. In the implemented WGAN-GP method, how many epochs the model will be trained for each data set, the optimal value of the batch size, and the appropriate constant values for the Gradient penalty were determined using the grid search technique.

2.6.1. Grid Search

The hyper-parameter is a property of a model whose value cannot be estimated from data and is independent of the model. Prior to starting the learning process, the quantity of the hyper - parameter should be determined. For example, how many layers will be used in the neural network or what the k number will be in the Knn algorithm are hyper-parameters. [22].

Grid search is a strategy for determining the best hyperparameters for a model. Finding hyperparameters in training data, unlike parameters, is impossible. As a result, we develop a model for each combination of hyperparameters in order to determine the best hyperparameters. Because we are essentially "brute-forcing" all conceivable

combinations, grid search is considered a fairly classic hyperparameter optimization strategy. Cross-validation is then used to assess the models. Naturally, the model with the highest accuracy is regarded as the best.

2.7. Generative Adversarial Networks (GAN)

In the field of deep learning, Generative Adversarial Networks is a game-changing generative approach. Goodfellow, Pouget-Abadie, Mirza, et al. [23] announced GANs in 2014, and they consist of two competing neural networks.

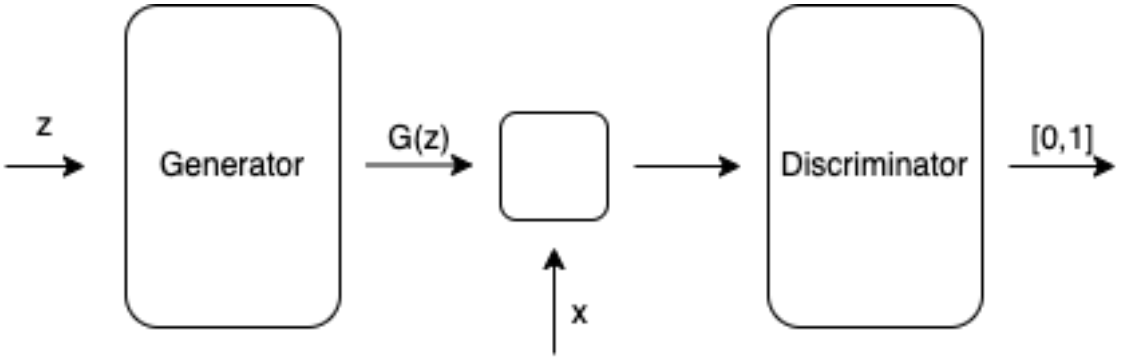

Figure 2.2 GAN Diagram

In Figure 2.2, the noise vector z is sent into the generator, which maps it to $G(z)$. The discriminator then gets either x (actual data) or $G(z)$ as input (generated data). If the supplied data is either false (0) or real (1), the discriminator generates a prediction. The network is trained using these outputs.

Two multilayer perceptrons, a discriminator D, and a generator G make up the model. The generator's goal is to learn a data distribution across the data that it generates from noise. The synthesized data is supplied to the discriminator to offer feedback on how effectively the generator worked. The discriminator calculates the likelihood that the data supplied was created by the generator. The discriminator is given data from both the distribution and the data, and its aim is to maximize the estimated probability. The generator's purpose is to deceive the discriminator into believing that the synthesized data is real. In each training

iteration, the discriminator and generator networks are trained and compete with one another to reduce and maximize the objective in equation 1.

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[1 - \log D(G(z))] \tag{1}$$

2.8. Wasserstein GAN

In their 2017 book Wasserstein GAN, [24] presented the Wasserstein GAN, or WGAN for short. It's a GAN extension that seeks for a new way to train the generator model so that it can better imitate the data distribution found in a given training dataset.

Rather than using a discriminator to categorize or forecast the chance of produced pictures being genuine or false, the WGAN replaces it with a critic who reviews the realness or fakeness of a particular picture.

The theoretical logic for this update is that the gap between the distribution of data seen in the training dataset and the distribution observed in created instances should be reduced when training the generator. The WGAN has the benefit of being more stable during training and less subject to architectures and hyperparameter configurations. Most importantly, the loss of the discriminator seems to be tied to the visual quality of the generator.

WGAN architecture is shown in the Figure 2.3. G represents Generator and C represents Critic

To overcome the sensitive and unstable difficulties of GANs, Martin Arjovsky et al. [24] created the Wasserstein GAN. They focus on the various methods for determining how close the produced distribution and the true distribution are. The Earth-Mover (EM) distance, which is a measure of the distance between two probability distributions over an area , is a novel technique to estimate the distance between two distributions.

Many concerns, such as generator instability and gradient disappearance in GAN training, can be avoided to some extent by using the new distance measurement. However, severe

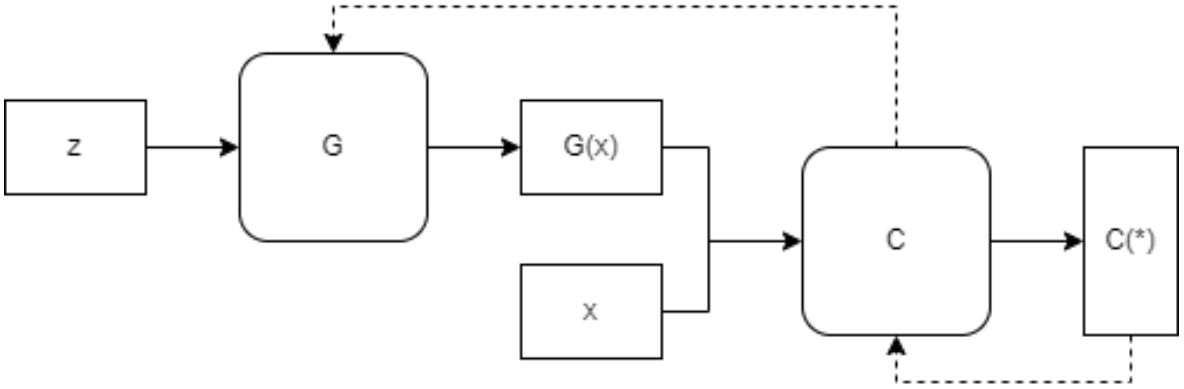

Figure 2.3 Wasserstein GAN Architecture

constraints on the network during the calculation of the EM distance may result in capacity underuse as well as an exploding and disappearing gradient. Ishaan Gulrajani et al. [25] present a novel clipping weight approach based on penalizing the gradient's norm to the important section concerning its input .

2.9. Monitoring of the Heart Rate

Heart rate monitoring may be done in a variety of ways. Electrocardiography and photoplethysmography are two essential approaches for this inquiry. Electrocardiography (ECG) is the practice of employing electrodes implanted on the skin to keep track the heart's electrical activity [24]. The minute electrical changes on the skin caused by the heart muscle's electrophysiologic pattern of depolarization during each heartbeat are detected by these electrodes. This approach is used in medical settings, such as hospitals, with ten electrodes implanted on the patient's limbs and chest surface. A combo of photo diodes and LEDs is used in photoplethysmography, commonly known as optical heart rate detection to measure heart-rate. Blood absorbs green light, which is why it turns red. When a portion of the body is put on top of a light source, the light is partially absorbed and partially reflected by the blood. A photo diode collects the reflected light. Although PPG is a low-cost approach for measuring heart rate, it has certain drawbacks. During workouts and free living situations, motion artifacts have been demonstrated to be a correct limiting factor outcomes.

Measurement errors can also be caused by individual differences. For example, varying blood perfusion causes varied light absorption, which might lead to discrepancies in readings.

3. RELATED WORK

The Internet of Things is a worldwide network of physical objects embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems through the Internet [26]. IoT provides seamless integration of sensors, actuators and communication devices, paving the way for new application development in many areas such as health, industry, automotive, transportation, and environment [6]. The rapid increase in the amount of developed applications causes an exponential increase in the number of devices connected to the Internet [27].

On the other hand, all connected devices generate huge amounts of data from connected sensors. In order for the applications to be run effectively and efficiently in accordance with their development purposes, generated data must be collected, analyzed, interpreted and delivered to the end user quickly [1]. At this point, the quantity and quality of the collected data is important in data analysis processes [2]. This becomes critical in applications that affect human life, require fast response and demand high service quality. However, due to the nature of IoT, the collected data may be inaccurate, missing and noisy due to various reasons, such as collision, unstable network communication, malfunctioning devices, and manual system closure [3]. It is hard to avoid the missing data problem, and dealing with it is extremely tough. Therefore, in order to conduct an effective and meaningful data analysis, missing data needs to be handled appropriately [28]. The simplest way to deal with missing data is to eliminate missing records. However, this elimination process causes serious information loss in areas where small and limited amount of data can be collected [29]. Moreover, the data integrity, accuracy, and on-time delivery requirements of IoT healthcare applications are stricter than other applications [30]. Particularly, current trends in healthcare domain, are shifting from treatment-oriented health to prevention-oriented health services. Also, health trends are shifting towards approaches that focus on personalized treatments rather than general treatment approaches [31]. At this point, in order to provide optimal and patient-centered health services, a small amount of data that can be collected should be analyzed without deleting it.

For real-time applications, the Internet of Things (IoT) allows for the seamless integration of sensors, actuators, and communication devices. IoT based smart systems, that are established on the basis of such devices are beginning to be employed in automobiles, houses and other infrastructure systems [32].

With the rise of big data in the biomedical and healthcare communities, precise medical data analysis helps early illness diagnosis, patient treatment, and community services. When the quality of medical data is poor, the analytical accuracy suffers [33]. Therefore, missing data problem is a very hot topic in health domain. In the literature, there are many studies based on statistical methods, fuzzy logic and machine learning to solve the missing data problem. However, due to the page limitation, we summarize the most appropriate ones for our research problem in this subsection.

Jerez and colleagues[34], compared machine learning methods in a large breast cancer dataset, such as multi-layer perceptron (MLP), self organizing maps (SOM), and k-nearest neighbor (KNN) to traditional statistical imputation methods and found that machine learning imputation methods performed better.

Turabieh et al. [35] proposed a dynamic layered recurrent neural network (D-LRNN) for IoMT applications to impute missing data. The authors solved two medical instances that simulated real IoT applications, and after recovering the missing data, the performance of the IoMT program increased.

In the study [33], in order to complete the missing data, belong to health data they build on an autoencoder to create a deep learning architecture capable of learning the hidden representations of data even when the data is skewed by missing values. The results of the study were compared with well-known methods like KNN.

In another study [36], the advantages of using the random forest method in completing missing data were mentioned and missing data predictions were made with random forest methods.

In [37], Liu et al. employed decision trees, Naive Bayesian classifiers, and feature selection

approaches to a geriatric hospital dataset in order to predict inpatient duration of stay, particularly for extended stay patients.

Enders et al. [38] presented an overview of two contemporary analytic alternatives, direct maximum likelihood (DML) estimate and multiple imputations, and outline recent methodological developments linked to missing data (MI). They provided a brief overview of classic missing data strategies, as well as DML and MI. They provided a brief overview of classic missing data strategies, as well as DML and MI. Then, the authors presented an exemplary analysis based on the collection of life quality data.

Beaulieu et al. [39] evaluated the performance of common multiple assignment methodologies with a highly trained autoencoder (PRO-ACT) on the Pooled Resource Open-Access ALS Clinical Trials Database. The authors evaluated the performance of the methods used by looking at their estimation accuracy on values that is either completely missing or not missing at all. They also investigated how different imputation methods predicted ALS disease progression. In the study, Autoencoders were found to have the best performance in terms of disease progression prediction accuracy. Unintentional bias can occur due to a variety of reasons for missing data in EHR data.Using the Pooled Resource Open-Access ALS Clinical Trials Database, Beaulieu et al. [39] evaluate the performance of common multiple imputation methodologies with a highly trained autoencoder (PRO-ACT).

To assess performance, they looked at imputation accuracy for known values that were either fully missing at random or missing not at all. They also investigated how different imputation methods predicted ALS disease progression. Autoencoders performed well in terms of imputation accuracy; and they helped to create the best disease progression prediction. Finally, they showed that, despite clinical variability, ALS disease progression appears to be homogeneous, with the most relevant predictor being time from onset.

Norris et al. [40] examined several ad hoc approaches to handling missing data using a clinical database of 6,065 cardiac patients. Instead of deleting the missing values, the authors used it as if it indicates the deficiency of a risk factor, enriching it with an administrative database. They looked at utilizing full cases alone (rather than deleting partial instances),

considering missing values as though they indicated the lack of a risk factor, and enhancing data by combining clinical data with an administrative database.

Nazabal et al. [41] present a broad framework for designing VAEs that are suited for fitting incomplete heterogeneous data in this research. The proposed HI-VAE comprises likelihood models for real-valued, positive real-valued, interval, categorical, ordinal, and count data, as well as correct missing data estimate (and maybe imputation). Furthermore, in supervised tasks, HI-VAE outperforms supervised models when trained on partial data, outperforming supervised models.

Hegde et al. [42] evaluated 116 dental variables with incomplete values produced at random to compare Probabilistic Principal Component Analysis (PPCA) with Multiple Imputation using Chained Equations (MICE). To produce a smaller dimensional space for the dataset, PCA was employed for dimensionality reduction. The missing values were retrieved from the compressed information distribution calculated by the PCA approach, therefore this attribute was used to impute the incomplete values. The EM technique was then used to iteratively estimate the MLE of an incomplete dataset. Instead, MICE used regression models to impute the missing data numerous times, taking into account the statistical uncertainty in the imputations.

Duan et al. [43] presented a DL model called stacked denoising autoencoders (SDAE) for traffic data imputation. The proposed model is built by taking into account both spatial and temporal aspects. The experimental findings suggest that the DL model is effective at imputation of traffic data and has potential in smart transportation.

Missing data exist in nearly every research, even in well-designed and controlled studies. Missing data can decrease a study's statistical power and create skewed estimates, resulting in incorrect findings [44].

Matte et al. [45] presented MIWAE, a technique based on the importance-weighted autoencoder (IWAE) that maximizes a potentially tight lower bound on the observed data's log-likelihood. Due to the missing data, their technique has no additional computational cost

when compared to the original IWAE. They also use a DLVM trained on an incomplete data set to build Monte Carlo algorithms for single and multiple imputation. They demonstrate their method by using imperfect static binarisations of MNIST to train a convolutional DLVM. Furthermore, they show that MIWAE produces extraordinarily accurate single imputations and is highly competitive with state-of-the-art approaches on a variety of continuous data sets.

Luo et al. [46] present a method for data imputation using generative adversarial networks. A modified GRU cell (dubbed GRUI) is presented for processing incomplete time series in order to learn the unfixed time delays between two observed values. The ”noise” input vector of the generator is learned and generates suitable values for imputation once the GAN model with GRUI cell has been trained. The adversarial architecture may learn the dataset's temporal correlations, inner-class similarities, and distribution in this way. Experiments demonstrate that their technique outperforms the baselines in terms of missing value imputation accuracy, and that it has applications downstream.

Li et al. [47] provide a GAN-based framework for learning from high-dimensional, partial data in this study. The proposed system simulates the missing data distribution by learning a complete data generator as well as a mask generator. They also show how to impute missing data by using an adversarially trained imputer in our architecture. They test the suggested framework in a series of experiments with various sorts of missing data procedures under the premise that data is missing totally at random.

Yoon et al. [48] present Generative Adversarial Imputation Nets (GAIN), this imputation approach that generalizes the well-known GAN and may operate well even when complete data is missing. The discriminator's purpose in GAIN is to discriminate between observed and imputed components, whereas the generator's goal is to properly impute missing data. The generator is trained to maximize the discriminator's misclassification rate while the discriminator is taught to minimize classification loss when distinguishing which components were seen and which were imputed. As a result, an adversarial approach is used to train these two networks. GAIN builds on and extends the conventional GAN architecture

to achieve this purpose. The GAIN architecture offers the discriminator with additional information in the form of ”hints” to guarantee that the outcome of this adversarial process is the intended aim. This hints that the generator should create samples based on the genuine underlying data distribution.

4. THE PROPOSED METHOD

In this study, unlike conventional methods, leverage the power of GAN models is proposed in order to impute missing data. To properly clarify this suggested method, it is necessary to first explain the fundamentals of these generative models. Vanilla GAN, the first example of Generative Adversarial Network models, should be considered as a system wherein two distinct networks compete in a zero-sum min-max game. While these two networks compete against one another, they also train each other though. The first of these networks is known as a generator, and the second is named a discriminator. The generator strives to produce data that is similar to the original data, starting with a random set of values as its name suggets. The Discriminator, which is trained with both the original data and the fake data produced by the generator attempts to determine whether the provided data is real data or not. With the feedback it provides, the Discriminator drives the Generator to create data that has closer to the distribution of the real dataset.As a generator that can produce better data is obtained, the discriminator also begins to have difficulty distinguishing between real and fake data and is compelled to improve. Thus, the result of the network system reaching equilibrium has a generator that strives to produce data as close as possible to the real data, and a discriminator that specializes in distinguishing between real data and fake data. Equation 2 shows Min-Max Game Objective of Vanilla GAN.

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[1 - \log D(G(z))] \tag{2}$$

There are studies in the literature where Vanilla GAN, which was originally designed to work on 2D images, is utilized for a variety of purposes and applications.There are many studies that use only Generator, only Discriminator or a combination of both in this network architecture that promises strong results. By modifying the overall structure of these networks and/or their inputs-outputs, some researchers have constructed new GAN variants that can suit a variety of applications. Some of these models extends Discriminator with classification ability, while others supply specific condition inputs to the generator in order

to create data under certain situations. These are only a few instances of this structure, which has a very broad range of applications. Regardless of the objectives, this network structure is relatively harder to train compared to other neural network models. There are various alternative GAN models that are recommended for more stable training. One of the most well-known and powerful among these is the Wasserstein Generative Adversarial Network (WGAN). The proposed WGAN differs from the vanilla GAN in terms of both structure and training procedure. The Discriminator is renamed as Critic, and this structure now creates the realness/fakeness score rather than stating whether the data is real or fake. For a more stable training process, this method suggests training the critic more than the generator. Furthermore, it aims to eliminate the problems in education by weight clipping. Weight clipping , on the other hand, is not always the best option. If the clipping window is not appropriately selected, it might result in slow or even unstable training or encountering with vanishing gradient problems. Thus, the researchers proposed WGAN with Gradient Penalty by replacing the proposed WGAN's weight clipping technique with a gradient penalty term and promised to define a more stable training process. Loss functions for the generator and critic are defined by Equation 3, respectively. Equation 4 denotes the WGAN model's objective, where f is stated as a 1-Lipschitz function.

$$\begin{aligned} Loss_G &= \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log (D(G(z^{(i)})] \\ Loss_C &= \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log (D(G(z^{(i)}) \end{aligned} \tag{3}$$

$$\min_G \max_{||f||_L \leq 1} E[f(x)] - E[f(\hat{x})] \tag{4}$$

The original WGAN-GP loss function, which is obtained by modifying the WGAN loss function by adding the Gradient penalty term, is given in Equation 5. Where P_x represents the distribution of real samples, $P_{\hat{x}}$ is distribution of generated samples which are generated by generator. The coefficient λ is used to weight the penalty term in the loss.

$$L = \underset{\hat{x}\sim P_g}{E}[f(\hat{x})] - \underbrace{\underset{x\sim P_r}{E}[f(x)]}_{\text{critic loss}} + \lambda \underbrace{\underset{\hat{x}\sim P_{\hat{x}}}{E}[(||\nabla_{\hat{x}} f(\hat{x})||_2 - 1)^2]}_{\text{gradient penalty}} \tag{5}$$

In this study, WGAN-GP was chosen as the base GAN model since it promises more stable training. Layers of the generator and the critic networks have been reconfigured, input/outputs have been updated in accordance with the defined problem, and losses have been reconsidered. At the end of the training, it is aimed to complete the missing data with the generator in the model. In this context, the input of the generator has been transformed from a latent vector to data with missing parts and auxiliary data that will express which areas are absent in the data. The generator combines these two inputs and generates a new data with the leveraging of convolutional layers and fills the missing parts in the given data. As a result, it varies from the original GAN models in that the data is used as input to the generator. The Critic network, like the original WGAN-GP, calculates the realness/fakeness score for both real and fake samples it receives. Figure 4.1 shows an overview of the proposed MIWGAN-GP model. While the data containing the missing data is given to the model as a 4x4 matrix, the mask matrix that marks the regions with missing data is also given to the model in the same dimensions. The value m_{ij} of mask matrix is defined at space N, where x_{ij} is the cell in given data matrix which addressed with i^{th} row and j^{th} column,

$$m_{ij} = \begin{cases} 1 & \text{if } x_{ij} \text{ — exists} \\ 0 & \text{if } x_{ij} \text{ — absent} \end{cases}$$

Unlike the original GAN studies, we have revised the loss functions for both the generator and the critic to fit the nature of the problem defined and we have also modified the objective

function with gradient penalty computation respect to original WGAN-GP study. Equation 6 defines critic loss for proposed model,

$$\begin{aligned} Loss_C &= Loss_{RF} + \lambda \times Loss_{GP} \\ Loss_{RF} &= C(X) - C(G(X)) \end{aligned} \tag{6}$$

where

$$\begin{aligned} x &= \text{Data matrix} \\ M &= \text{Mask matrix} \\ X &= \text{Combined input from given} [x, M] \\ G(X) &= \text{Generated samples from given} X \\ C(X) &= \text{Realness/fakeness score for given} X \end{aligned}$$

The loss function of the generator is defined as specified in Equation 7. Here, $Loss_{G_{full}}$ is derived by averaging the positive distance from the original data set of the data generated by the generator over the full parts in the data set. Whereas, $Loss_{G_{miss}}$, on the other hand, takes into account the average of the positive distance calculated over the missing parts for the given samples. As a result, the generator imputation loss $Loss_{G_{imp}}$ is evaluated with the weighted sum of those two components where coefficients are indicated with α for full part and β for missing part, respectively. Finally, the generator loss $Loss_G$ is weighted combination of imputation loss of generator $Loss_{G_{imp}}$ and critic loss $Loss_C$.

$$\begin{aligned} Loss_{G_{full}} &= mean(|-(M \cdot x_{org}) + (M \cdot x_{gen})|) \\ Loss_{G_{miss}} &= mean(|-((1-M) \cdot x_{org}) + ((1-M) \cdot x_{gen})|) \\ Loss_{G_{imp}} &= \frac{\alpha \times Loss_{G_{full}} + \beta \times Loss_{G_{miss}}}{\alpha + \beta} \\ Loss_G &= -\frac{\rho \times Loss_C + \theta \times Loss_{G_{imp}}}{\rho + \theta} \end{aligned} \tag{7}$$

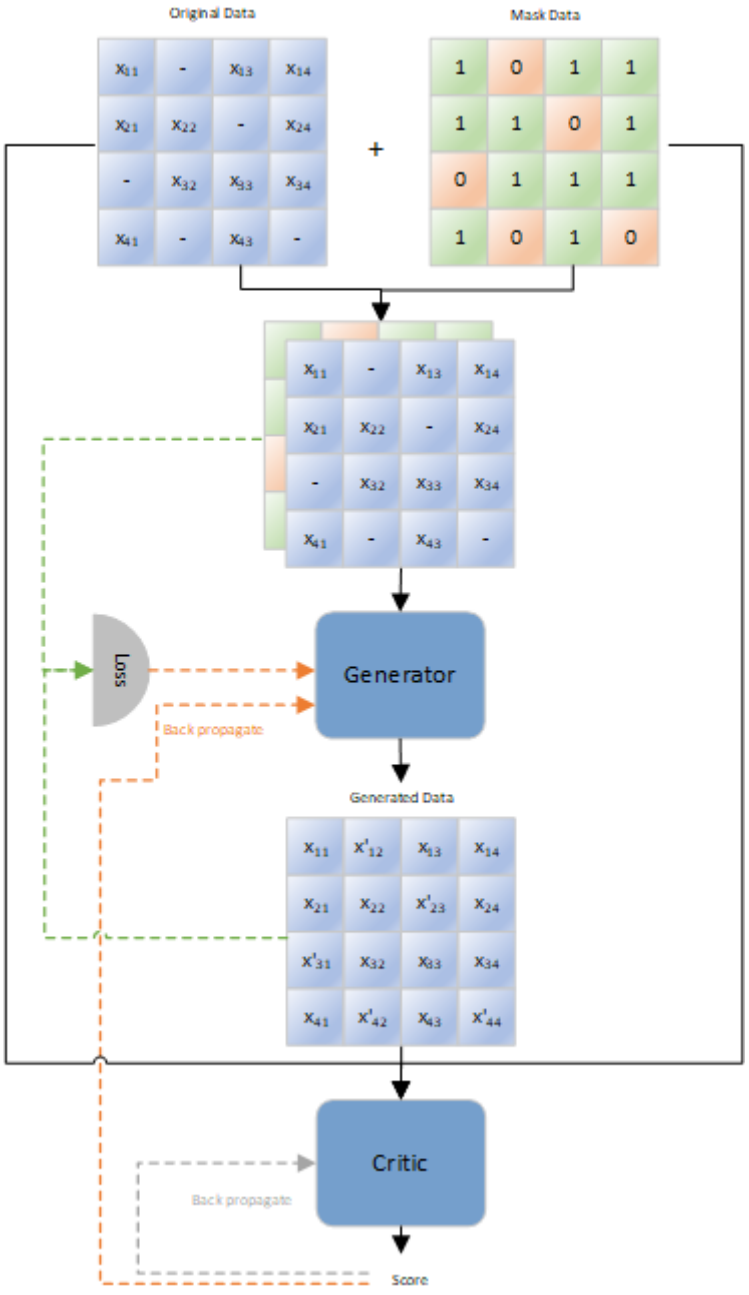

Figure 4.1 Proposed MIWGAN Architecture

Figure 4.2 shows the inner architecture of generator model. Instead of getting latter vector as an input to network, the generator gets the data that will be imputed into neural network layers with the mask matrix. In the input stage of the network, these two inputs concatenate in the manner of overlay in order to build multiple channel input matrix. As a result, the input for the first trainable Convolutional 2D layer is in size $nxnx2$ where n is defined as

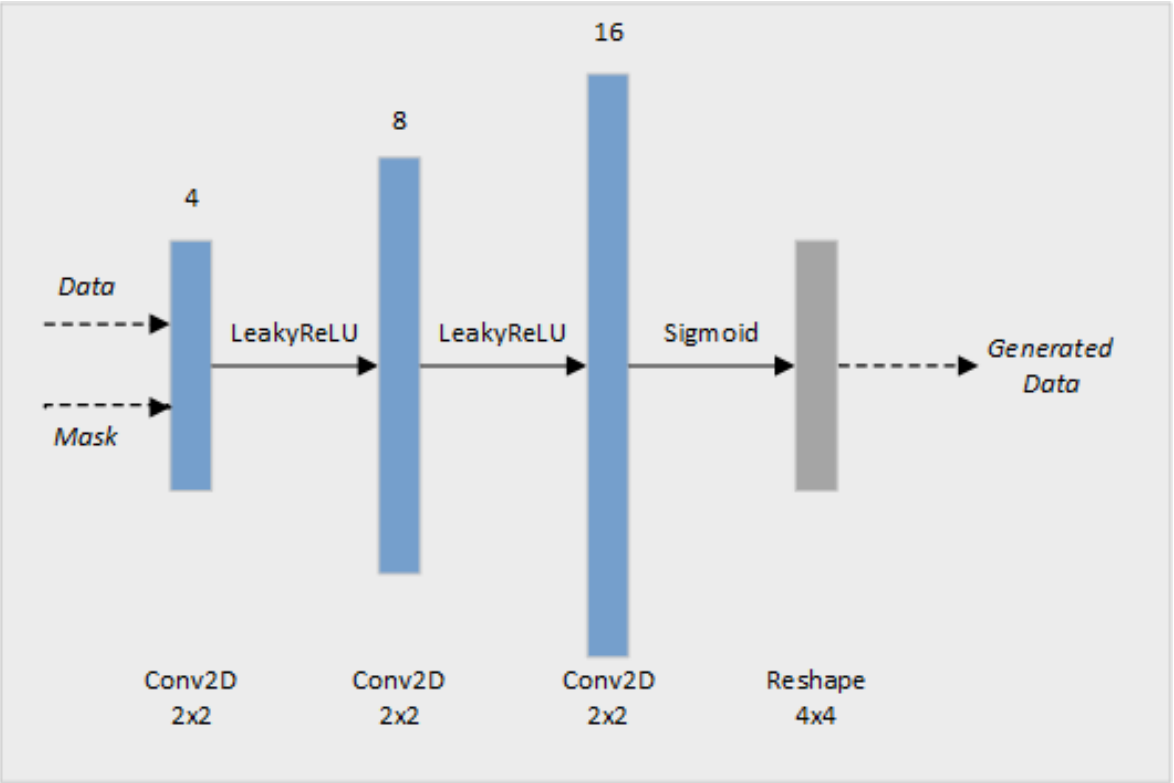

Figure 4.2 Generator Layers Architecture

value 2 in the figure. After each convulutional layer, except the last layer, the LeakyReLU activation function is preferred. The output layer is reshape layer that follows Sigmoid activation function which is responsible from squeeze the output data to interval $[0, 1]$. In the end of layers, the output of the network is a imputed data which in the same size with the data matrix nxn. We also want to emphasize that the data to be imputed is normalized between 0 and 1 before being fed to the network. The method we propose can be reshaped and scaled to a given input size n.

Similar to the generator model, the critic model also accepts the mask matrix as an input in addition to the data matrix. In the neural network, which is requested to produce the realness/fakeness score, the Convolutional 2D layers are advanced and the LeakyReLU activation function is employed until the last layer. In the last layer, it is desired to obtain a value that can express the data at hand by using global max pooling 2D and the network is finished with a dense layer and a realness/fakeness score is produced.

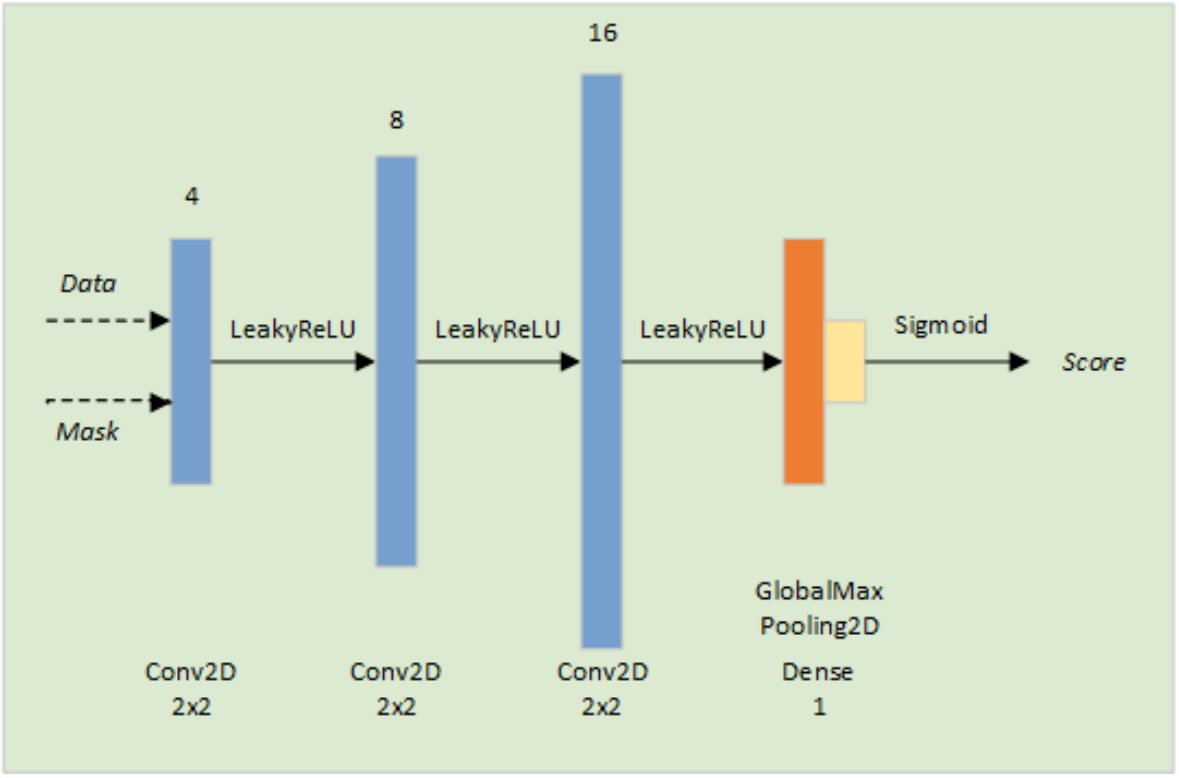

Figure 4.3 Critic Layers Architecture

5. EXPERIMENTAL RESULTS

In this section, the data sets used in this thesis, experiments, evolution metrics and evaluation results are presented.

5.1. Dataset

Within the scope of this thesis, two different datasets, human activity and fitbit, were used. These 2 data sets are explained under this title.

5.1.1. Human Activity Dataset

The study of human movement and activities has spawned a slew of previous studies, the majority of which involved the placement of sensors on the subject's body. Smart devices have been increasingly popular in recent years since they are already ubiquitous and contain precise miniature sensors. Each device, whether it's a smartphone, a smartwatch, or a pair of smart glasses, can be used to describe additional data such as emotions, specific movements, or environmental conditions.

In July 2017, a data set[49] has been gathered. The smartphone was kept in the pocket for a significant amount of time during the data collection. Every day, the smart glasses were worn for a few hours. SWIPE is a platform that uses smartwatches and cellphones to sense, record, and interpret human dynamics.

Watch Metric	Source	Recording Rate	Description
Heart rate	Optical heart rate sensor	Event-based	The optical heart rate sensor provides the heart rate in beats per minute. Each number has an accuracy value that represents the monitor's status during the reading.
Step Detector	Accelerometer	Event-based	Whether or whether the user is taking a step is indicated.
Step Counter	Accelerometer	Event-based	The number of steps taken by the user, as measured by the accelerometer, as identified by the Android system.
Battery	Android	5,000ms	Battery level

Table 5.1 Human Activity Dataset Descriptions

	count	mean	std	min	25%	50%	75%	max
Heart Rate	91250	68.98	11.99	0.00	61.00	67.00	74.00	167.00
Pressure	14900	982.53	7.15	962.31	979.45	984.36	987.91	1005.81

Table 5.2 Human Activity Dataset Summary

Table 5.2 shows summarize of Human Activity dataset.

Before using the Human activity dataset, it was preprocessed. In the data set, the values 2 before and 2 after each instance's own location were placed side by side and used as features. By selecting heart rate and pressure values from this dataset, missing data was estimated for these values. Neighboring values from before 2 and after 2 were used for each instance.In addition, the inputs were given to network as 4 each. Thus, if we compare the data type that is the input to the network to a picture, it looks like a 4*4 picture is given as an input.

5.1.2. Fitbit Dataset

Between 03.12.2016 and 05.12.2016, responders to an Amazon Mechanical Turk distributed survey created these datasets [50]. Thirty Fitbit members who completed the requirements consented to have their personal tracking data, which includes minute-level output for physical exercise, heart rate, and sleep monitoring, submitted. The export session ID or timestamp can be used to parse individual reports. The variation in output indicates the usage of various Fitbit trackers as well as individual tracking practices and preferences.

	count	mean	std	min	25%	50%	75%	max
Step Total	22099	320.16	690.38	0	0	40	357	10554
Calories	22099	97.38	60.70	42	63	83	108	948
Total Intensity	22099	12.03	21.13	0	0	3	16	180
Average Intensity	22099	0.20	0.35	0	0	0.05	0.26	3

Table 5.3 Fitbit Dataset Summary

Table 5.3 shows summarize of FitBit dataset.

5.2. Evaluation Metrics

5.2.1. Mean Squared Error (MSE)

The mean error is the average error between the predicted values predicted by a machine learning model and the actual values. Error in this context is the uncertainty in a measurement, or the difference between the estimated value and the true value.

$$MSE = \sum_{i=1}^{n} (d_i - f_i)^2 \tag{8}$$

5.2.2. Root Mean Squared Error (RMSE)

It's a quadratic metric that calculates the size of a machine learning model's mistake and is frequently used to calculate the distance between the predictor's predicted and true values. The standard deviation of the estimating errors is the RMSE. The RMSE is a measure of how prevalent these residues are; residuals are a measure of how distant the regression line is from the data points. To put it another way, it shows how dense the data is in the vicinity of the best-fitting line. The RMSE value might be anything from 0 to 1. Scores that are negatively orientated, or have lower values, perform better. A RMSE of 0 indicates that the

model made no mistakes. Because RMSE penalizes big errors more severely, it may be better suited to specific scenarios. In many mathematical calculations, RMSE precludes the use of undesirable absolute values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - f_i)^2} \tag{9}$$

5.2.3. Mean Absolute Error (MAE)

The difference between two continuous variables is measured by the mean absolute error. The average vertical distance between each true value and the best-fitting line is known as MAE. The average horizontal distance between each data point and the best-fit line is also known as MAE. The MAE value is commonly utilized in regression and time series issues because it is simple to comprehend. The MAE is a linear score that weighs all individual mistakes equally on the mean and assesses the mean magnitude of errors in a series of predictions without considering their direction. The MAE value might be anywhere between zero and infinity. Scores that are negatively orientated, or have lower values, perform better.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|d_i - f_i| \tag{10}$$

5.2.4. Mean Absolute Percentage Error (MAPE)

The accuracy of a company's forecasting process is measured by the mean absolute percentage error (MAPE). It shows, on average, how accurate the anticipated quantities were in relation to the actual amounts by averaging the absolute percentage errors of each entry in a dataset. MAPE is useful for studying huge datasets, however it is impossible to compute the MAPE of datasets with zero values. This is due to the fact that the computation would need zero division, which is impossible.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|d_i - f_i|}{d_i} \tag{11}$$

In Equations 8, 9, 10 and 11 d_i is the generated data and f_i is the original data.

5.3. Experiments

The proposed approach was tested on the three data sets indicated above, and performance measures were collected. The results of the described metrics were compared to the results of the traditional imputation techniques such as mean, median, and mode. In addition, graphs of loss and metric measures of the trained model are provided in this section. These experiments were carried out by separating each data set into train and test datasets using k-fold split where k was chosen as 5 for this study. In addition, 10%, 20%, 25%, 40% and 50% of the total number of instances were randomly deleted from each complete original train and test datasets in order to obtain missing datasets. Unless otherwise stated, the presented figures and the given results were created by carrying out the experiments on datasets whose 20% of the samples are missing. All experiments were conducted with Python using Tensorflow and Keras v2.5 libraries on Nvidia GeForce GTX 860M graphics card and took from 2 hours up to 20 hours depending on the chosen epoch number and batch size. The numeric results, figures and tables of all these experiments are also given in following subsections in details. In the proposed model, grid-search is used for hyperparameter optimization in order to improve network performance. In order to measure imputation success, the dataset obtained by completing the missing data with the proposed method was classified using various classification methods such as Naive Bayes, KNN, Decision Tree, Random Forest and gboost, and the results were compared with mean, median and mode imputed versions of the same dataset.

Figure 5.1 Human Activity - Heart Rate Critic and Generator Loss Graphs

5.3.1. Human Activity Dataset-Heart Rate Results

Figure 5.1 shows the critic and generator loss during the training process of the proposed method on Human Activity Heart Rate dataset. In this training, where the generator was stabilized after 150 epochs, loss graphs similar to the original GAN study were obtained. Furthermore, the loss graphics prove that healthy training process has taken place.

Figure 5.2 indicates the evaluated metrics during training process over test dataset. MSE, RMSE, MAE and MAPE metrics reveal that the training of the proposed model has reached a stable state and the generator and critic models have hit the equilibrium level, in line with the loss charts given in Figure 5.1. Table 5.4 demonstrates the comparison of imputation results between our proposed method and classical imputation methods such as mean,

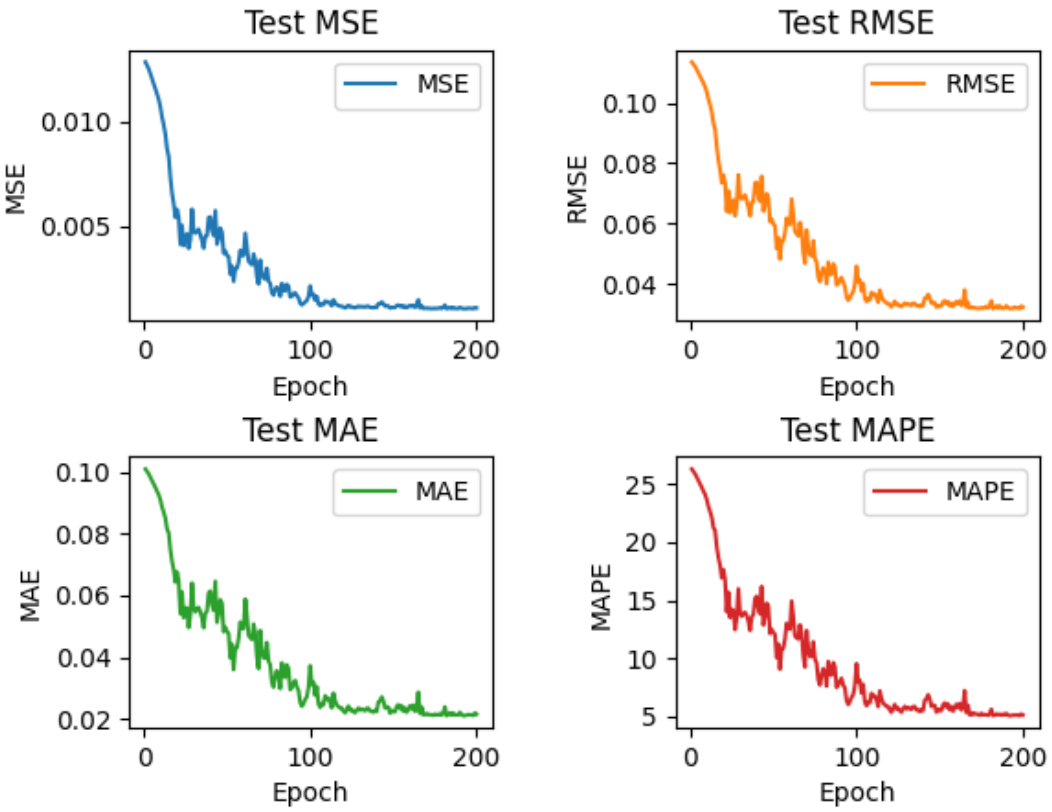

Figure 5.2 Human Activity - Heart Rate Critic and Generator Results Metrics

median and mode on evaluation metrics.

Imputation Methods	**MSE**	**RMSE**	**MAE**	**MAPE**
MIWGAN-GP (ours)	0.001055	0.032490	0.021552	5.0798000
Mean	0.005411	0.073565	0.052109	12.340543
Median	0.005543	0.074456	0.050823	11.667576
Mode	0.005717	0.075614	0.050924	11.518874
KNN	0.004878	0.064987	0.045690	9.963214

Table 5.4 Human Activity-Heart Rate Dataset Imputations Results for Metrics

Figure 5.3 Human Activity - Pressure Critic and Generator Loss Graphs

5.3.2. Human Activity Dataset-Pressure Results

Figure 5.3 shows the critic and generator loss during the training process of proposed method on Human Activity Pressure dataset. In this training, where the generator was stabilized after 125 epochs, loss graphs similar to the original GAN study were obtained. Furthermore, the loss graphics prove that healthy training process has taken place.

Figure 5.6 indicates the evaluated metrics during training process over test dataset. MSE, RMSE, MAE and MAPE metrics reveal that the training of the proposed model has reached a stable state and the generator and critic models have hit the equilibrium level, in line with the loss charts given in Figure 5.3. Table 5.5 demonstrates the comparison of imputation results between our proposed method and classical imputation methods such as mean, median and mode on evaluation metrics.

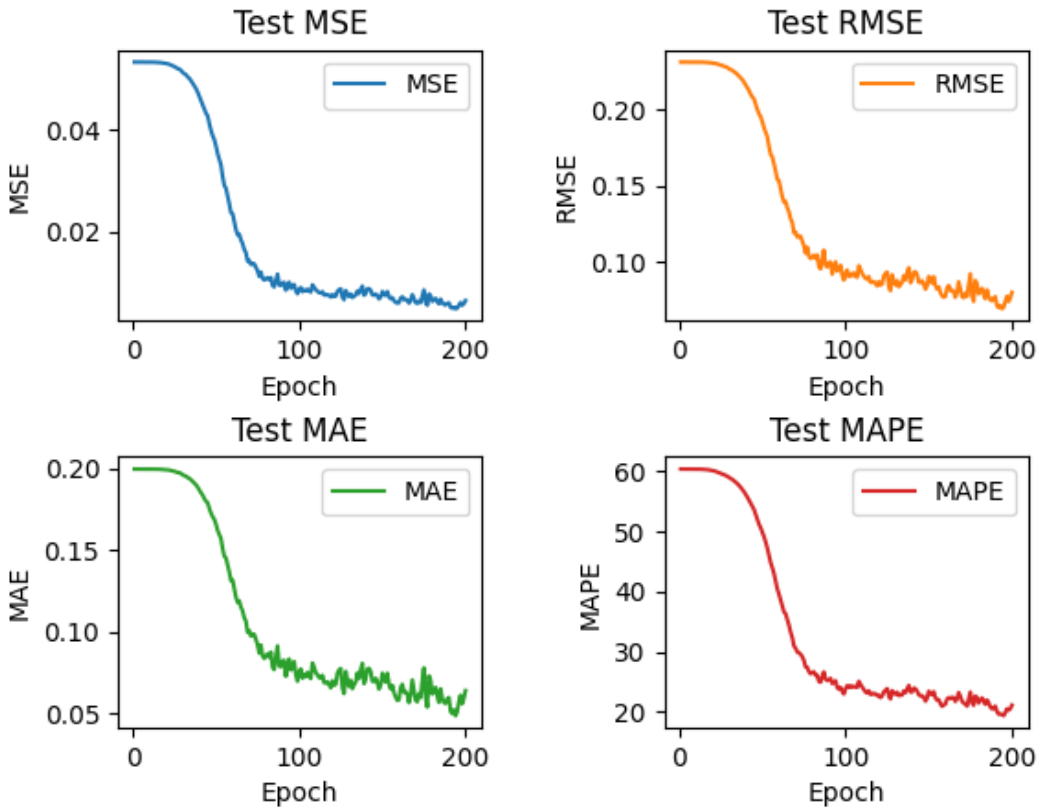

Figure 5.4 Human Activity - Pressure Data Set Results Metrics Graphs

Imputation Methods	MSE	RMSE	MAE	MAPE
MIWGAN-GP (ours)	0.013746	0.117243	0.090462	35.201672
Mean	0.045652	0.213665	0.172999	65.371374
Median	0.048155	0.219444	0.167971	70.200228
Mode	0.050985	0.225799	0.169271	72.934199
KNN	0.035894	0.198576	0.159873	55.957420

Table 5.5 Human Activity-Pressure Dataset Imputations Results for Metrics

5.3.3. FitBit - Hourly Activity Dataset Results

Figure 5.5 shows the critic and generator loss during the training process of proposed method on Hourly Activity dataset. In this training, where the generator was stabilized after 1500 epochs, loss graphs similar to the original GAN study were obtained. Furthermore, the loss graphics prove that healthy training process has taken place.

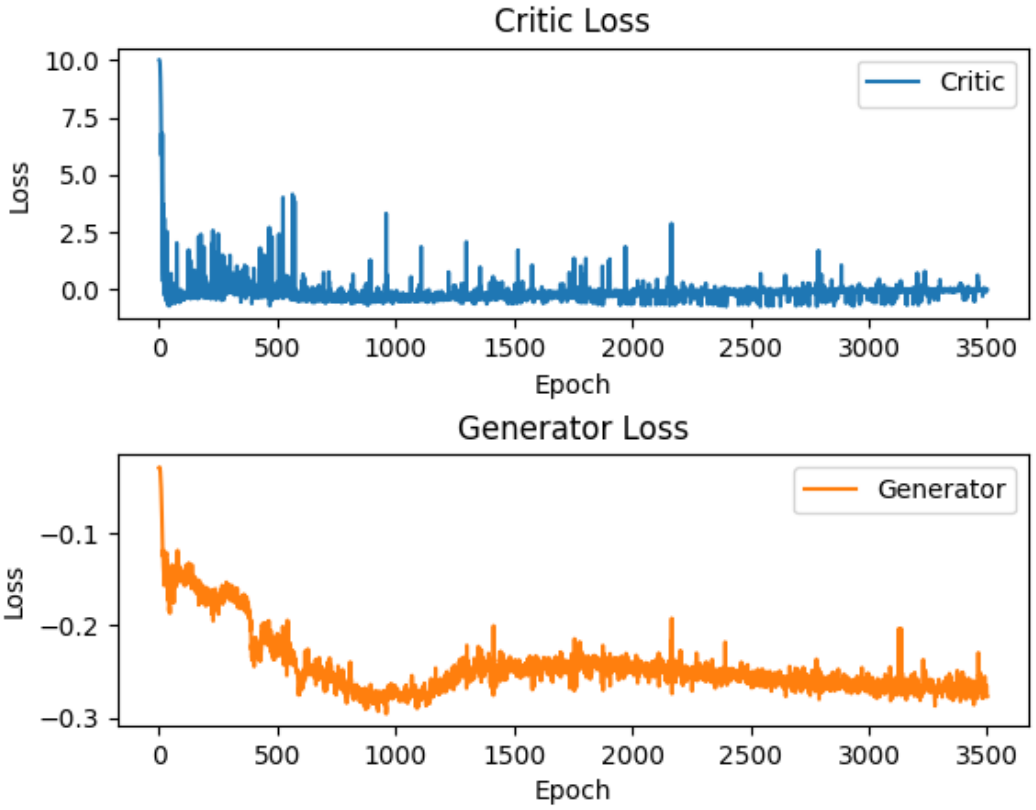

Figure 5.5 Hourly Activity Critic and Generator Loss Graphs

Imputation Methods	MSE	RMSE	MAE	MAPE
MIWGAN-GP (ours)	0.004234	0.065070	0.043052	116.308198
Mean	0.005047	0.071043	0.054218	169.648734
Median	0.004557	0.067508	0.036078	160.311322
Mode	0.005933	0.077031	0.040015	158.570859
KNN	0.005014	0.070649	0.045893	147.327898

Table 5.6 Hourly Activity Dataset Imputations Results for Metrics

5.3.4. Classification Results After Data Imputation Process

This section presents the results of the classification of the dataset with missing data after it has been imputed. The classification accuracy results of the data set imputed with the proposed MIWGAN-GP model were compared with the classification results of the same data set filled with the traditional imputation methods mean, median and mode with using the same classifiers. All these experiments were carried out by producing datasets with

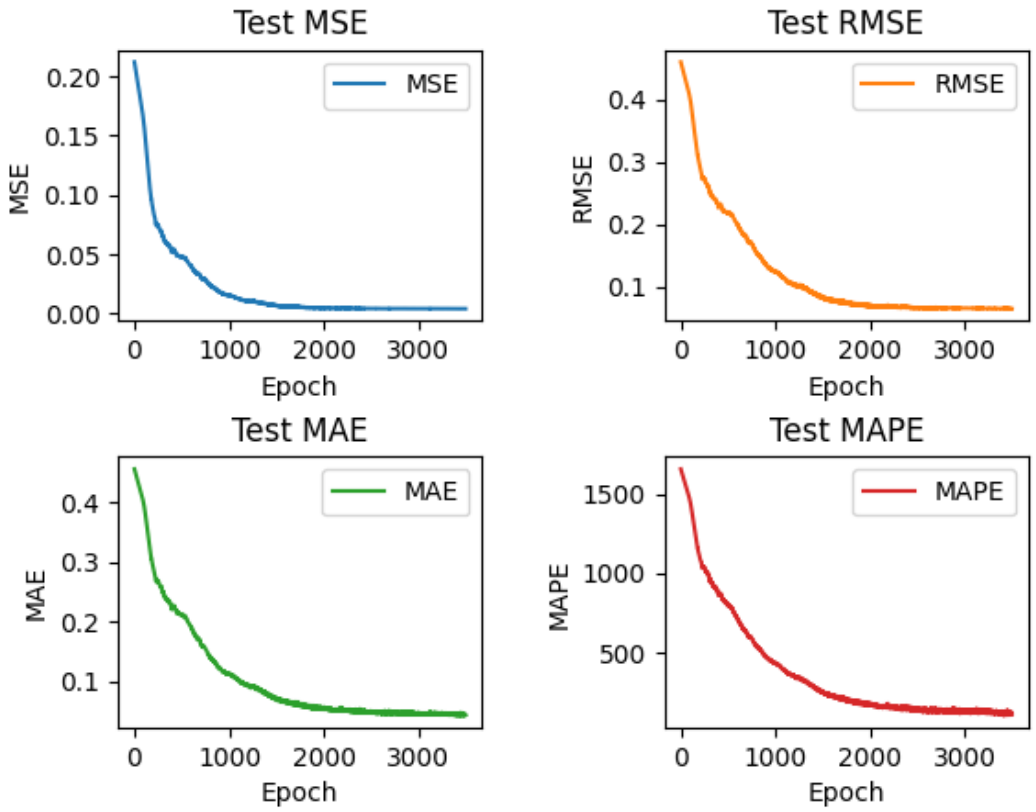

Figure 5.6 Hourly Activity Data Set Results Metrics Graphs

10%, 20%, 25%, 40% and 50% deficiencies from the selected data set independently. The classifiers were chosen as KNN, NaiveBayes, Decision Tree, Random Forest and xgboost, which are well known and frequently used in the literature, and the results for each are given in the following tables. While examining the results, we would like to draw attention to the fact that each classifier should be analyzed independently and the focus should be on the imputation methods compared, not on the classifiers chosen. Thus, we want to emphasize the imputation strength of the proposed method in various classifiers rather than the independent success of the classifiers. Furthermore, as expected, classification accuracy drops as the missing rate in the data set increases. It can be observed that a similar circumstance arises in the proposed method and shows parallelism with traditional methods. MIWGAN-GP's outperformed results demonstrate the usability of imputation of missing data with our suggested method as a preprocessing step for classification tasks. When all

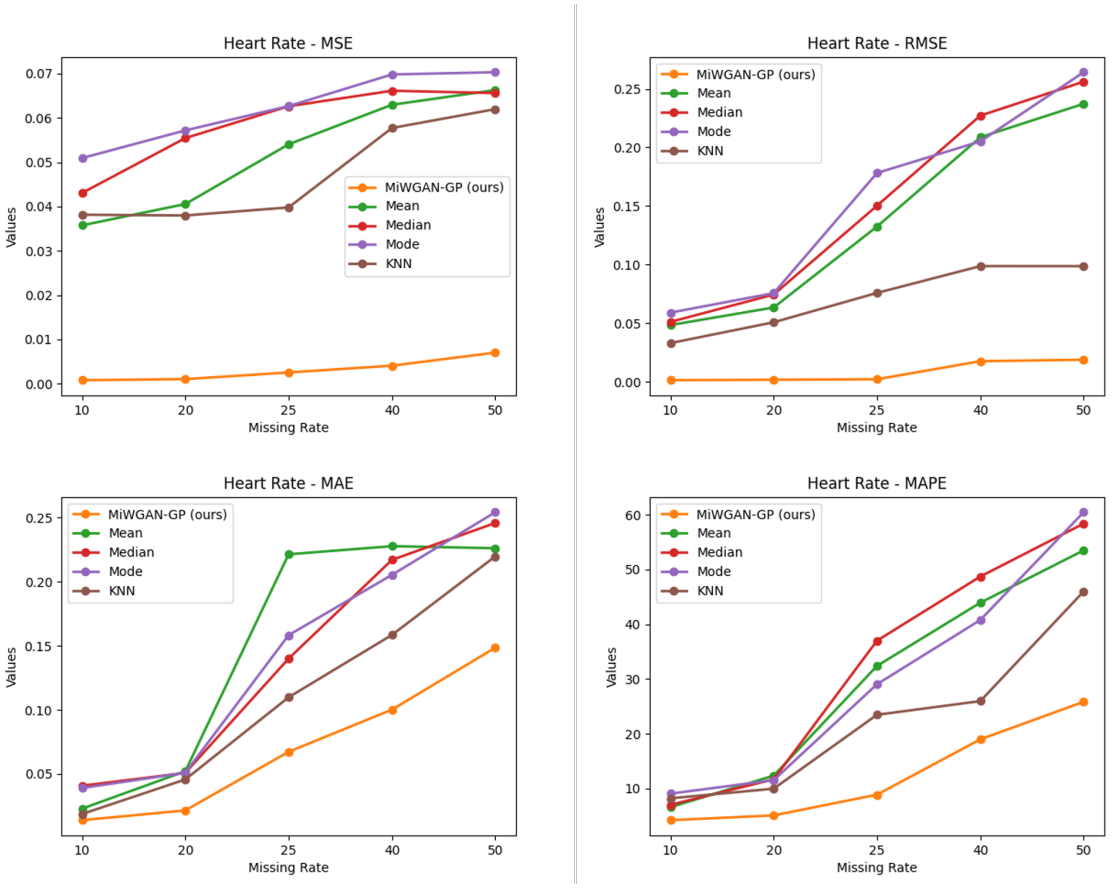

Figure 5.7 Human Activity - Heart Rate Dataset Method Comparison over Metrics

of these classification results are evaluated collectively, it is undeniable that our proposed method contributes to obtaining better results compared to traditional imputation methods, even for classifiers that can achieve average performance.

Missing Rates (%)	Zero Imp.	Mod Imp.	Average Imp.	Median Imp.	KNN Imp.	MIWGAN-GP Imp.	Different Value
10	56.194	60.964	58.926	58.924	62.752	63.389	0.637
20	56.190	60.614	57.896	57.880	60.598	61.167	0.553
25	56.534	60.272	56.872	56.872	59.127	60.986	0.714
40	56.138	58.761	56.872	56.872	57.560	59.237	0.476
50	56.528	56.524	55.500	55.500	56.854	57.681	0.827

Table 5.7 Naive Bayes Classification Accuracy on Fitbit Dataset

Table 5.7 shows the result for Naive Bayes classifier which has been carried out on the dataset. When the last column containing the classification accuracy results of the proposed

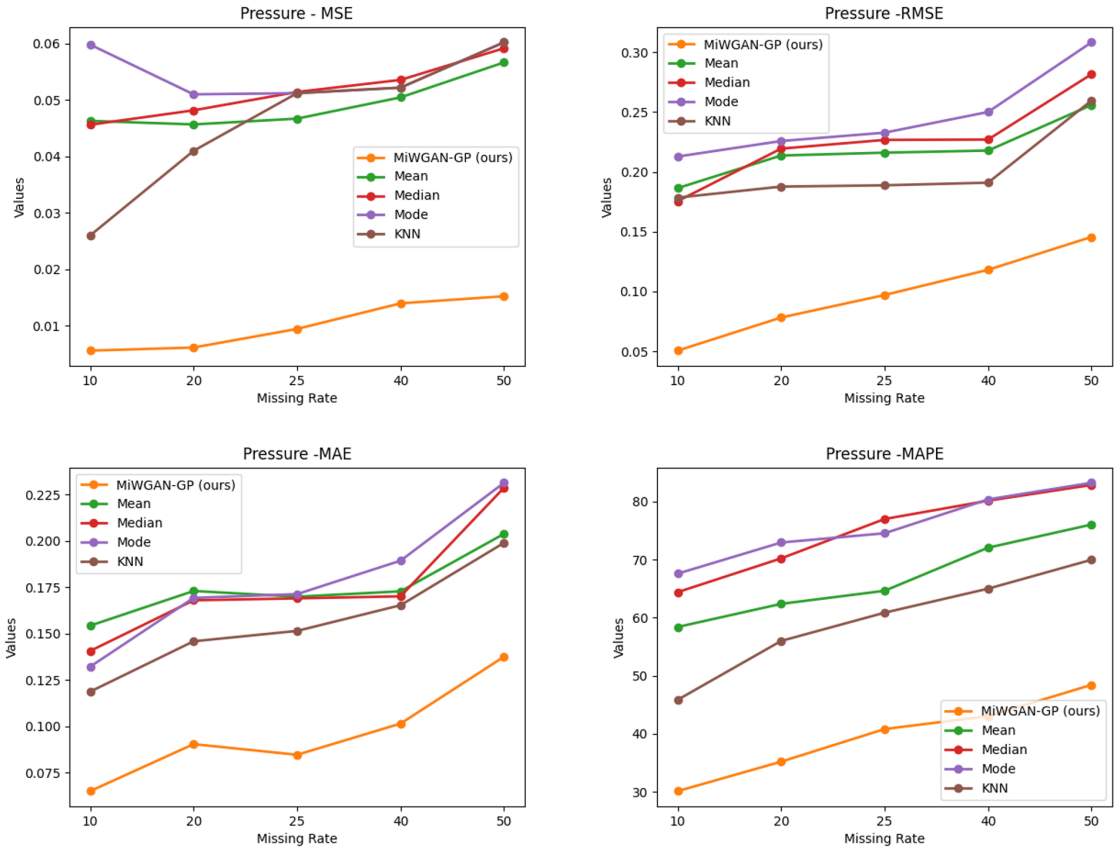

Figure 5.8 Human Activity - Pressure Dataset Method Comparison over Metrics

MIWGAN-GP method is examined, it is shown that it outperforms other imputation methods for each missing rate.

Missing Rates (%)	Zero Imp.	Mod Imp.	Average Imp.	Median Imp.	KNN Imp.	MIWGAN-GP Imp.	Different Value
10	51.730	53.098	54.120	53.436	56.183	57.496	1.313
20	50.564	49.668	48.966	48.970	51.422	52.731	1.309
25	46.012	50.706	49.672	48.990	50.527	51.159	0.453
40	44.726	44.766	45.828	47.756	49.257	50.678	1.421
50	43.146	40.426	43.494	43.150	43.985	45.916	1.931

Table 5.8 KNN Classification Accuracy on Fitbit Dataset

Similar to the Naive Bayes classifier scores, KNN classification results demonstrate that better classification accuracy is acquired when compared to other conventional imputation techniques. The KNN classification accuracy is given in Table 5.8.

The Decision Tree classification results, which show significantly better results than previous classification types in terms of overall classification accuracy, are given in Table 5.9. The employment of the approach we offer in this study helps to the improvement in accuracy for Decision Tree classification method, which may reach %70 accuracy on average, as we have seen in previous classification results.

Table 5.10 presents the results of the Random Forest classifier, which has the highest classification accuracy for this data set among the other classifiers examined. In addition, our proposed method improved classification accuracy for this test set and surpassed the traditional imputation methods in all missing ratios. Our method, which managed to increase the classification success by approximately %5 on average even in the data set with %50 missing data, has proven itself as an imputation method that can be utilized in the preprocessing stage of classifiers.

The results of the XGBoost classifier, which produces results remarkably similar to the Random Forest classifier type, are given in Table 5.11. The results of the tests in this test set and classification combination are quite encouraging, as we are able to raise the average classification accuracy at the highest rate than conventional imputation approaches.

To summarize all of these results, the same data sets which have been created by reducing the same data set at the specified missing rates have been imputed with traditional imputation methods as well as the method we recommend in the preprocessing step of six different classification methods, and classification accuracy values are presented in tables. As can be clearly seen from the results in the tables, the MIWGAN-GP method that we propose may

Missing Rates (%)	Zero Imp.	Mod Imp.	Average Imp.	Median Imp.	KNN Imp.	MIWGAN-GP Imputation	Different Value
10	68.072	69.464	72.906	70.154	73.655	74.257	0.602
20	68.428	66.734	70.998	65.366	70.120	71.757	0.759
25	65.230	63.210	68.870	68.790	68.650	69.752	0.882
40	63.753	62.984	66.752	65.459	66.798	67.476	0.678
50	62.958	62.976	62.624	64.344	66.027	66.744	0.717

Table 5.9 Decision Tree Classification Accuracy on Fitbit Dataset

Missing Rates (%)	Zero Imp.	Mod Imp.	Average Imp.	Median Imp.	KNN Imp.	MIWGAN-GP Imp.	Different Value
10	76.676	77.696	78.042	76.334	77.548	78.682	0.640
20	74.292	73.262	75.316	74.646	75.857	76.244	0.387
25	75.318	74.970	73.262	73.250	75.190	75.981	0.663
40	75.318	74.970	73.262	73.250	74.851	75.198	0.228
50	69.826	69.488	70.526	69.484	72.573	72.953	0.380

Table 5.10 Random Forest Classification Accuracy on Fitbit Dataset

Missing Rates (%)	Zero Imp.	Mod Imp.	Average Imp.	Median Imp.	KNN Imp.	MIWGAN-GP Imp.	Different Value
10	72.898	73.942	75.634	75.634	77.691	78.230	0.539
20	70.496	72.226	70.868	69.850	74.797	75.774	0.977
25	69.304	71.618	69.286	69.286	74.124	74.913	0.789
40	68.874	70.122	68.913	68.755	71.549	72.105	0.0559
50	67.438	68.824	68.128	68.134	70.272	71.231	0.959

Table 5.11 XGBoost Classification Results on Fitbit Dataset

be described as a powerful imputation method that can be applied in the preprocessing step regardless of the classifier employed and can be placed ahead of the traditional methods.

6. CONCLUSION

In this thesis, another GAN based missing data imputation method is proposed with leveraging the generative models. In order to serve this purpose, a variant of the Wasserstein GAN with Gradient Penalty has been developed in line with the nature of the topic. Wasserstein GAN with GP has been described in the literature and has proven itself in many fields and various applications. Although the model we present is based on Wasserstein GAN, it varies from the original work in terms of generator structure and utilization. The generator uses two matrices as input in this study, where the trained model promises to impute the missing parts in the provided sample relying on the distribution of the data seen during the training process. In addition to the data matrix containing the missing parts, the network input is built with the auxiliary mask matrix to express which parts are missing. In order to follow the similar approach to 2D images, in which color information has been provided as an input to the network as a channel, in this problem, the mask matrix is utilized as an additional channel in the input matrix. Furthermore, in order to take into account the information in neighboring cells, convolutional layers is used in the model structure, again similar to the original work. As an output, imputed version of the given sample is generated. Generator and critic loss functions have been reconsidered in order to adapt to the imputation challenge and to make the imputed data as similar to the original data as possible. In order to determine the success of this proposed method, experiments were carried out on three distinct data sets. In order to measure success, four different metrics that are well-known and widely preferred in this field were utilized to assess effectiveness of study. In addition, five different classification methods have been performed on the imputed data to demonstrate the efficacy of the proposed method and its applicability in classification tasks. During these classification experiments, comparisons with traditional imputation approaches are also included. To mimic data loss at varied densities, all these classification processes were repeated with 5 different loss rates. These experiments performed on data produced by IoT devices show that the solution we propose can be a solution to the problem of completing missing data in IoT devices due to essential benefits of method such as tolerance to data loss,

performance, and lightweight structure of network. With the modification of the Gradient Penalty version of Wasserstein GAN, it is aimed to prevent problems such as vanishing gradients and mode collapse, which are among the most known drawbacks of the GAN models.

As future work, it will be beneficial to expand the study by evaluating the model we propose on data sets produced in other domains, other than IoT devices. It might also be worthwhile to work on including approaches like auto encoder in the proposed model or applying other unsupervised learning methods on data imputation may be interesting.

REFERENCES

[1] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414–454, **2013**.

[2] Metehan Guzel, Ibrahim Kok, Diyar Akay, and Suat Ozdemir. Anfis and deep learning based missing sensor data prediction in iot. *Concurrency and Computation: Practice and Experience*, 32(2):e5400, **2020**.

[3] İbrahim Kök and Suat Özdemir. Deepmdp: A novel deep-learning-based missing data prediction protocol for iot. *IEEE Internet of Things Journal*, 8(1):232–243, **2021**. doi:10.1109/JIOT.2020.3003922.

[4] Kevin Ashton et al. That 'internet of things' thing. *RFID journal*, 22(7):97–114, **2009**.

[5] Pradyumna Gokhale, Omkar Bhat, and Sagar Bhat. Introduction to iot. *International Advanced Research Journal in Science, Engineering and Technology*, 5(1):41–44, **2018**.

[6] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, **2013**.

[7] Tuhin Borgohain, Uday Kumar, and Sugata Sanyal. Survey of security and privacy issues of internet of things. *arXiv preprint arXiv:1501.02211*, **2015**.

[8] Li Li, Timo LM Ten Hagen, Azadeh Haeri, Thomas Soullié, Csilla Scholten, Ann LB Seynhaeve, Alexander MM Eggermont, and Gerben A Koning. A novel two-step mild hyperthermia for advanced liposomal chemotherapy. *Journal of Controlled Release*, 174:202–208, **2014**.

[9] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, **1976**.

[10] Craig K Enders. *Applied missing data analysis*. Guilford press, **2010**.

[11] Alan C Acock. Working with missing values. *Journal of Marriage and family*, 67(4):1012–1028, **2005**.

[12] Jaap Brand. *Development, implementation and evaluation of multiple imputation strategies for the statistical analysis of incomplete data sets*. **1999**.

[13] Rebecca R Andridge and Roderick JA Little. A review of hot deck imputation for survey non-response. *International statistical review*, 78(1):40–64, **2010**.

[14] Issam El Naqa and Martin J Murphy. What is machine learning? In *machine learning in radiation oncology*, pages 3–11. Springer, **2015**.

[15] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, **2019**.

[16] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, **2015**.

[17] Ramadass Sathya, Annamma Abraham, et al. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38, **2013**.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, **2015**.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, **2016**.

[20] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, **2017**.

[21] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, **2011**.

[22] Steven M LaValle, Michael S Branicky, and Stephen R Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):673–692, **2004**.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, **2014**.

[24] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, **2017**.

[25] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, **2017**.

[26] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics*, 14(11):4724–4734, **2018**.

[27] Eunhye Ko, Taeeun Kim, and Hwankuk Kim. Management platform of threats information in iot environment. *Journal of Ambient Intelligence and Humanized Computing*, 9(4):1167–1176, **2018**.

[28] Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, and Bailin Yang. Lstm-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305, **2018**.

[29] Joo-Chang Kim and Kyungyong Chung. Multi-modal stacked denoising autoencoder for handling missing data in healthcare big data. *IEEE Access*, 8:104933–104943, **2020**.

[30] Dongwook Lee, Junyoung Kim, Won-Jin Moon, and Jong Chul Ye. Collagan: Collaborative gan for missing image data imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2487–2496. **2019**.

[31] Ljiljana Trtica Majnarić, František Babič, Shane O'Sullivan, and Andreas Holzinger. Ai and big data in healthcare: towards a more comprehensive research framework for multimorbidity. *Journal of Clinical Medicine*, 10(4):766, **2021**.

[32] Tao Liu and Dongxin Lu. The application and development of iot. In *2012 International Symposium on Information Technologies in Medicine and Education*, volume 2, pages 991–994. IEEE, **2012**.

[33] Son Phung, Ashnil Kumar, and Jinman Kim. A deep learning technique for imputing missing healthcare data. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6513–6516. IEEE, **2019**.

[34] José M Jerez, Ignacio Molina, Pedro J García-Laencina, Emilio Alba, Nuria Ribelles, Miguel Martín, and Leonardo Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial intelligence in medicine*, 50(2):105–115, **2010**.

[35] Hamza Turabieh, Amer Abu Salem, and Noor Abu-El-Rub. Dynamic l-rnn recovery of missing data in iomt applications. *Future Generation Computer Systems*, 89:575–583, **2018**.

[36] Fei Tang and Hemant Ishwaran. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6):363–377, **2017**.

[37] Peng Liu, Lei Lei, Junjie Yin, Wei Zhang, Wu Naijun, and Elia El-Darzi. Healthcare data mining: Prediction inpatient length of stay. In *2006 3rd International IEEE Conference Intelligent Systems*, pages 832–837. IEEE, **2006**.

[38] Craig K Enders. A primer on the use of modern missing-data methods in psychosomatic medicine research. *Psychosomatic medicine*, 68(3):427–436, **2006**.

[39] Brett K Beaulieu-Jones, Jason H Moore, and POOLED RESOURCE OPEN-ACCESS ALS CLINICAL TRIALS CONSORTIUM. Missing data imputation in the electronic health record using deeply learned autoencoders. In *Pacific symposium on biocomputing 2017*, pages 207–218. World Scientific, **2017**.

[40] Colleen M Norris, William A Ghali, Merril L Knudtson, C David Naylor, and L Duncan Saunders. Dealing with missing data in observational health care outcome analyses. *Journal of clinical epidemiology*, 53(4):377–383, **2000**.

[41] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, **2020**.

[42] Harshad Hegde, Neel Shimpi, Aloksagar Panny, Ingrid Glurich, Pamela Christie, and Amit Acharya. Mice vs ppca: Missing data imputation in healthcare. *Informatics in Medicine Unlocked*, 17:100275, **2019**.

[43] Yanjie Duan, Yisheng Lv, Yu-Liang Liu, and Fei-Yue Wang. An efficient realization of deep learning for traffic data imputation. *Transportation research part C: emerging technologies*, 72:168–181, **2016**.

[44] Hyun Kang. The prevention and handling of the missing data. *Korean journal of anesthesiology*, 64(5):402, **2013**.

[45] Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning*, pages 4413–4423. PMLR, **2019**.

[46] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, 31, **2018**.

[47] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data with generative adversarial networks. *arXiv preprint arXiv:1902.09599*, **2019**.

[48] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pages 5689–5698. PMLR, **2018**.

[49] Sébastien Faye, Nicolas Louveton, Sasan Jafarnejad, Roman Kryvchenko, and Thomas Engel. An open dataset for human activity analysis using smart devices. **2017**.

[50] Keith M Diaz, David J Krupka, Melinda J Chang, James Peacock, Yao Ma, Jeff Goldsmith, Joseph E Schwartz, and Karina W Davidson. Fitbit®: An accurate and reliable device for wireless physical activity tracking. *International journal of cardiology*, 185:138–140, **2015**.