Synthetic Electronic Health Records for OpenMRS using Normalising Flows and GANs

Vineeth Kumar Vellala
201719460

Supervised by Owen Johnson and Prof . Luisa Cutillo

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2024

The candidate confirms that the work submitted is his/her own and that appropriate credit has been given where reference has been made to the work of others.

UNIVERSITY OF LEEDS

School of Mathematics

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

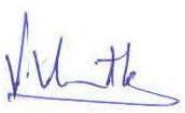
Academic integrity statement

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes. I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Name: Vineeth Kumar Vellala



Student ID: 201719460

Abstract

The growing use of Electronic Health Records (EHRs) has led to significant improvements in patient care, research, and data management. However, the sensitive nature of EHR data raises concerns about privacy and security, often limiting access for researchers. This dissertation explores how advanced data generation techniques, specifically Normalising Flows and Generative Adversarial Networks (GANs), can be used to create synthetic EHR data that closely resembles real data while safeguarding patient privacy.

Both Normalising Flows and GANs were chosen for their strengths in capturing complex data patterns. The results focused on generating realistic cardiovascular data, especially systolic and diastolic blood pressure, due to its importance in medical analysis. By successfully integrating this synthetic data into the OpenMRS platform, the research demonstrates the potential to provide researchers with valuable data while ensuring privacy protection.

The quality of the synthetic data was thoroughly tested, focussing on key metrics, including demographic details and clinical variables from key tables. The results showed that the synthetic data closely mirrors real data, effectively reducing the risk of re-identification. This makes synthetic data a promising tool for research, especially when privacy concerns limit access to real patient information.

This study contributes to the field of healthcare research by offering a practical method to generate safe and accessible EHR data. Future work will focus on improving the accuracy of time-based data and expanding the use of synthetic data across a wider range of healthcare systems.

Contents

List of Figures

List of Tables

Acknowledgements

Chapter 1

Introduction

1.1 Background

Whenever a patient visits a general practitioner (GP), they typically ask a series of questions before the diagnosis begins. The GP adds the patient's responses to their Electronic Health Record (EHR). Unlike traditional paper records, this digital record remains with the patient for life, ensuring that healthcare providers can easily access the complete medical history whenever needed.

Electronic Health Records (EHRs) have become a cornerstone of modern healthcare systems, providing a digital version of patients' paper charts and offering a comprehensive view of patient history, treatments, and outcomes (Adler-Milstein & Jha 2017). These records are invaluable for clinical research, healthcare analytics, and the development of new healthcare algorithms (Buntin et al. 2011).

However, the sensitive nature of EHRs raises significant privacy concerns, limiting their accessibility for research and educational purposes (Kruse et al. 2017). As privacy regulations such as the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and other international standards become more stringent, there is an increasing need for alternative solutions that allow researchers to access health data without compromising patient confidentiality (Rocher et al. 2019). These regulations aim to protect patient privacy by enforcing strict controls over how personal health information is accessed, shared, and stored. Moreover, just as individuals value their privacy, patients expect their personal health information to be protected and not used indiscriminately, presenting a challenge for researchers who require detailed datasets for analysis and innovation.

One promising solution is the generation of synthetic medical data, which can mimic the statistical properties (distributions, non-linear relationships, and noise)(Papamakarios et al. 2021*a*) of real EHRs while protecting patient privacy (Mohammed et al. 2011). Synthetic data can be particularly beneficial in resource-constrained settings, enabling institutions to conduct research and develop healthcare technologies without risking privacy breaches. This dissertation explores the use of Normalising Flow

models for generating continuous data and Generative Adversarial Networks (GANs) for categorical data to create synthetic EHR datasets that maintain the utility and integrity of real data (Papamakarios et al. 2021*a*).

Normalising flows are a class of generative models that provide a powerful framework for modelling complex data distributions. Generative Adversarial Networks (GANs) are another class of generative models, particularly effective in generating realistic synthetic data, including categorical data (Xu et al. 2019). By leveraging these models, we aim to learn the underlying probability density functions of Medical Information Mart for Intensive Care (MIMIC-IV), a large, publicly available database of de-identified health records, to generate realistic synthetic data that is both comprehensive and accurately reflects the diversity of real-world EHR data . This approach can facilitate the integration of synthetic data with platforms like the open-source medical records system (OpenMRS), an EHR system widely used in low-resource settings, enabling broader access to high-quality medical data for research and education (Mamlin et al. 2006).

1.2 Problem Statement

The digitalisation of healthcare data presents a dual challenge: enhancing data accessibility for meaningful research while rigorously protecting patient privacy (Rocher et al. 2019). Despite efforts to anonymise EHR data, traditional methods often fall short in preventing re-identification, especially when data is aggregated from multiple sources. This inadequacy significantly limits the availability of real EHR data for research, development, and educational purposes, hampering the development of advanced healthcare solutions.

Recent advances in generative models, particularly normalising flows, present a promising solution to this challenge. Normalising flows offer the potential to generate synthetic EHR data that closely mirrors the statistical properties of real-world data while ensuring robust privacy preservation. These models are particularly appealing because they allow for flexible and invertible transformations of data distributions, which can be crucial in maintaining the integrity of complex datasets like EHRs. However, the potential of normalising flows in this context has not been fully explored, and there is a need to rigorously evaluate their effectiveness compared to other methods, such as Generative Adversarial Networks (GANs).

This project focuses on developing and evaluating methods for generating synthetic EHR data using normalising flows and GANs, aiming to produce synthetic datasets that are both realistic and versatile for various applications (Mohammed et al. 2011).

This research will address the following key problems:

1. How can synthetic data generation techniques, particularly normalising flows, preserve the statistical properties of real EHR data while ensuring privacy?

2. What are the best practices for data preprocessing and integrating synthetic data with existing EHR platforms like OpenMRS?

3. How can the quality and utility of synthetic data be evaluated effectively?

1.3 Aims and Objectives

The primary aim of this dissertation is to develop a robust method for generating synthetic EHR data that maintains the utility of real-world data while safeguarding patient privacy. This involves exploring and implementing advanced machine learning models, particularly normalising flows, to create high-quality synthetic datasets. The specific objectives of the project, aligned with the Cross-Industry Standard Process for Data Mining (CRISP-DM)(Shearer 2000*a*), are:

1. **Business Understanding**: Define the requirements for synthetic data generation in healthcare, considering privacy regulations and ethical considerations in data management.

2. **Data Understanding**: Thoroughly explore the structure and characteristics of existing EHR datasets, such as MIMIC-IV, to facilitate the synthetic data generation process

3. **Data Preparation**: Preprocess the MIMIC-IV dataset to ensure that it is clean, consistent, and ready for modelling, addressing any issues related to data quality and formatting.

4. **Modelling**: Employ statistical techniques and deep learning models, such as normalising flows and Generative Adversarial Networks (GANs), to generate realistic synthetic EHR data (Papamakarios et al. 2021*a*).

5. **Evaluation**: Evaluate the quality, compatibility, and performance of the generated synthetic data using statistical tests along with correlation analysis to ensure accurate simulation of real-world healthcare scenarios.

1.4 Project Scope

The scope of this dissertation involves the exploration and application of normalising flow and GAN models for synthetic data generation, specifically focusing on Electronic Health Record (EHR) data. Additionally, the research includes the integration of synthetic datasets with OpenMRS and evaluates both the utility and privacy aspects of the generated data (Mohammed et al. 2011). This project not

only addresses the technical aspects of data generation but also considers the ethical implications and compliance with privacy regulations.

The project is limited to using the publicly available MIMIC-IV dataset and involves a comparative analysis with statistical techniques to identify the most effective methods for the generation of synthetic data. The study is designed to provide information on the practical challenges and opportunities of using synthetic data in real-world healthcare settings.

1.5 Dissertation Structure

This dissertation is structured as follows:

- **Chapter 2: Literature Review**: This chapter provides an overview of existing research on Electronic Health Records (EHRs), synthetic data generation, and the technologies involved, including normalising flows and EHR systems. It also discusses the ethical and privacy considerations related to healthcare data.
- **Chapter 3: Methodology and Implementation**: This chapter details the research design, data collection, and preprocessing methods used in the study. It outlines the implementation of normalising flows and Generative Adversarial Networks (GANs) for generating synthetic data and the evaluation methods employed. The integration of synthetic data with healthcare platforms like OpenMRS is also discussed.
- **Chapter 4: Results and Discussion**: This chapter presents the results of the synthetic data generation and evaluates the quality, privacy, and performance of the data. It includes a comparison with real data and discusses the limitations of the approach.
- **Chapter 5: Conclusion and Future Work**: This chapter summarises the key findings of the study and outlines the contributions to the field. It also suggests future research directions and potential areas for improvement.
- **References**: This section lists all the academic sources and references used throughout the dissertation.
- **Appendices**: These include additional materials, such as code snippets, data tables, and ethical approval documentation, which support the research but are not essential to the main narrative.

Each chapter will build upon the previous ones, culminating in a comprehensive exploration of synthetic EHR data generation and its integration with OpenMRS.

Chapter 2

Literature Review

2.1 Electronic Health Records (EHRs)

Electronic Health Records (EHRs) have fundamentally transformed how healthcare data is managed, significantly improving patient care and health system efficiency. The digitisation of health records has facilitated better disease tracking, patient monitoring, and data sharing across healthcare providers (Smith & Johnson 2015). Figure 2.1 outlines the various components that comprise an Electronic Health Record, showcasing the diverse types of information these systems manage, such as medical, demographic, administrative, and billing data.

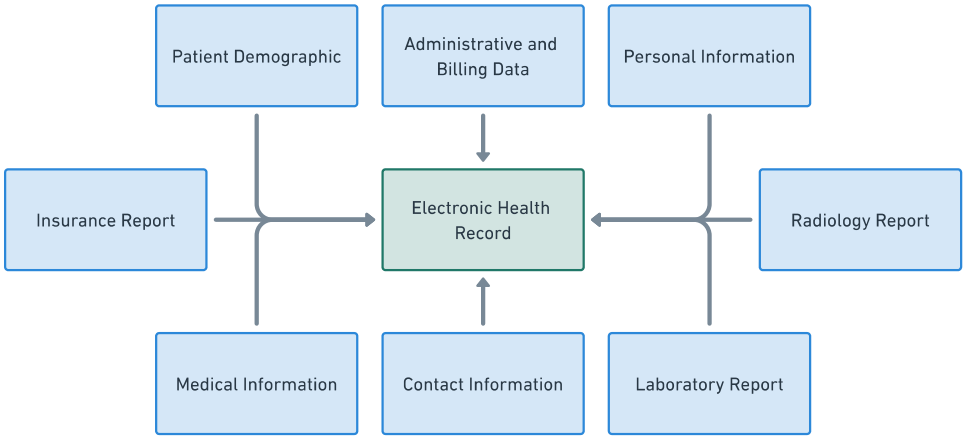


Figure 2.1: Diagram illustrating the components of an Electronic Health Record (EHR).

EHR systems are typically structured in relational databases, which allow for efficient storage, retrieval, and management of the complex and interconnected data types shown in Figure 2.1. This relational structure supports the integration of various data components, enabling healthcare providers to access comprehensive patient information in real time. The diagram illustrates how different types of

data, ranging from patient demographics to billing information, are organized within an EHR system, highlighting the complexity and depth of information managed within these systems.

Despite these advancements, EHR systems face significant challenges. Issues such as data interoperability, system usability, and concerns about data security have been persistent (Beaulieu-Jones et al. 2018). These challenges underscore the need for improved designs and standards in EHR systems.

2.2 Synthetic Data Generation

2.2.1 Synthetic Data Generation

Synthetic data refers to artificially generated data that mimics the statistical properties and relationships of real-world data. While the concept of synthetic data is not new, recent advancements in machine learning, particularly in deep learning techniques, have significantly increased interest in its potential applications within healthcare (Park & Mack 2013, Rubin 1993). The use of synthetic data is particularly relevant to address privacy concerns and alleviate the legal and ethical constraints associated with the sharing and use of sensitive personal health information (Rieke et al. 2020, Goncalves et al. 2020).

Various techniques are employed to generate synthetic data, ranging from traditional statistical methods to advanced generative models. Statistical resampling techniques like bootstrapping offer simpler methods to create synthetic datasets by sampling with replacement from the original data (Efron & Tibshirani 1994). However, more sophisticated approaches, such as Generative Adversarial Networks (GANs) and normalising flows, have gained prominence for their ability to generate high-quality synthetic data that closely resembles real data (Goodfellow et al. 2014*a*, Papamakarios et al. 2021*b*).

GANs are particularly effective in generating complex data types, such as medical images and patient records, while maintaining the intricate patterns found in the original datasets (Xu et al. 2019). Normalising flows, on the other hand, offer a flexible framework for modelling complex data distributions, making them suitable for applications that require high fidelity to the original data (Papamakarios et al. 2021*b*).

According to (Papadaki et al. 2024), there are three different ways in which synthetic data can be generated: Manual Obfuscation and Manipulation, Statistical-based Data Generation, and Machine Learning-driven Approaches (see Figure 2.1).

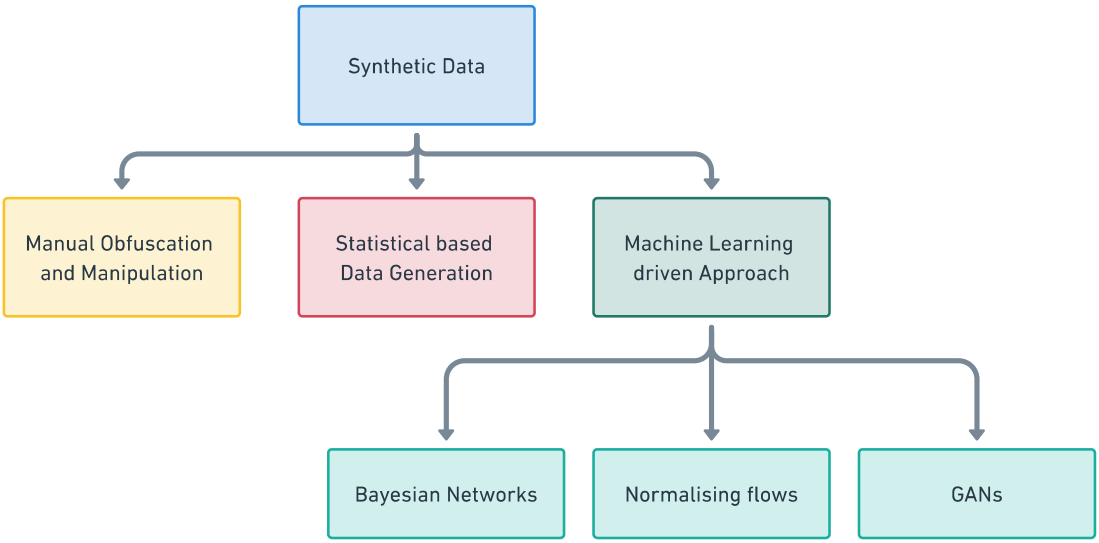


Figure 2.2: Diagram illustrating the different methods of Synthetic Data Generation.

Manual Obfuscation involves altering or masking data through predefined rules to protect sensitive information while still allowing the data to be used for analysis. However, as stated by (Witkowska 2006), manual obfuscation is clearly ineffective in maintaining the complex relationships within the data. Therefore, statistical and machine-learning approaches will be explained below, with particular emphasis on machine-learning-driven approaches.

2.2.2 Statistical Resampling Methods

Statistical resampling is a non-parametric method used in statistical inference to approximate the sampling distribution of a statistic. This technique involves repeatedly drawing samples from the original data and is particularly useful when the underlying population distribution is unknown. Bootstrapping is one of the most widely used resampling methods.

In bootstrapping, synthetic datasets are generated by sampling with replacement from the original dataset. Let $X = \{x_1, x_2, \ldots, x_n\}$ be an original dataset, where x_i represents an observation such as the age or weight of a patient. A bootstrap sample $X^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ is created by drawing observations from X with replacement. Each x_i^* is independently drawn from the empirical distribution of X.

The empirical distribution function (EDF), denoted by $F_n(x)$, is defined as:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I(x_i \leq x)$$

Where $I(\cdot)$ is the indicator function, which equals 1 if the condition inside is true and 0 other-

wise. Bootstrapping involves generating multiple bootstrap samples X^*, each of which is drawn with a replacement from the original dataset X. The statistic of interest, $T(X)$, is then computed for each bootstrap sample, denoted as $T(X^*)$. By repeating this process many times, we can approximate the sampling distribution of $T(X)$, providing insight into the variability and confidence intervals of the statistic.

Patient ID	**Age (years)**	**Weight (kg)**
1	25	70
2	30	80
3	35	65
4	40	75
5	45	85

Table 2.1: Original dataset.

Sample Number	**Age (years)**	**Weight (kg)**
1	30	80
2	25	70
3	35	65
4	45	85
5	25	70

Table 2.2: Bootstrapped sample.

In the bootstrapped sample shown in Table 2.2, the age of 25 and weight of 70 are repeated, as highlighted in yellow in both Sample 2 and Sample 5. This repetition is a direct result of the sampling with replacement characteristics of bootstrapping. While this method is computationally efficient and captures basic data relationships, it is limited because it relies on the existing data distribution and does not generate new patterns or model complex interdependencies.

2.2.3 Bayesian Methods

Bayesian methods provide a probabilistic framework for modelling data and estimating unknown parameters. Unlike frequentist approaches, Bayesian methods incorporate prior beliefs about the parameters and update these beliefs based on observed data. This is particularly useful in synthetic data generation, where prior domain knowledge can significantly enhance the accuracy and realism of the generated data.

The core principle of Bayesian inference is Bayes' Theorem, which relates the conditional and marginal probabilities of random events. It is expressed mathematically as:

$$p(\theta|D) = \frac{p(D|\theta) \cdot p(\theta)}{p(D)}$$

Where:

- $p(\theta|D)$ is the posterior probability of the parameters θ given the data D,
- $p(D|\theta)$ is the likelihood of the data given the parameters,

- $p(\theta)$ is the prior probability of the parameters,

- $p(D)$ is the marginal likelihood of the data.

Application in Synthetic Data Generation

In the context of synthetic data generation, Bayesian methods use prior distributions to model the uncertainty in the data and parameters. The process involves:

1. **Defining Priors**: Prior distributions $p(\theta)$ represent prior beliefs about the parameters. These can be informed by historical data or expert knowledge.

2. **Likelihood Function**: The likelihood $p(D|\theta)$ models how data is generated given the parameters. For instance, if generating synthetic patient data, the likelihood could model the probability of observing patient ages and weights given certain health conditions.

3. **Posterior Sampling**: The posterior distribution $p(\theta|D)$ is computed, reflecting updated beliefs after observing the data. Sampling from this distribution allows for generating synthetic datasets that are consistent with both the observed data and the prior information.

Generating Synthetic Patient Data

Consider generating synthetic data for patient weights based on age, where the relationship is assumed to be linear. The model can be defined as:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

where y_i is the weight, x_i is the age, α and β are parameters to be estimated, and ϵ_i is the error term. The priors for α and β could be chosen as normal distributions:

$$\alpha \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \quad \beta \sim \mathcal{N}(\mu_\beta, \sigma_\beta^2)$$

Using Bayesian inference, we update these priors with observed data to obtain the posterior distributions of α and β. Synthetic data can then be generated by sampling from these posterior distributions, ensuring that the generated data reflects both the underlying statistical patterns and the prior knowledge.

Advantages and Limitations

Bayesian methods are powerful in their ability to incorporate prior knowledge and quantify uncertainty. This makes them particularly well-suited for applications where domain knowledge is critical. However, Bayesian methods can be computationally intensive, especially when dealing with large datasets or complex models, as they require techniques like Markov Chain Monte Carlo (MCMC) for posterior sampling.

2.2.4 Generative Adversarial Networks (GANs) and Conditional Tabular GANs (CTGAN)

Generative Adversarial Networks (GANs) are powerful tools in synthetic data generation. They are known for creating highly realistic samples through a competitive process between two neural network models: a generator and a discriminator. The generator aims to generate synthetic data that is indistinguishable from real data. At the same time, the discriminator evaluates the authenticity of the generated data, effectively training the generator to improve its outputs.

Architecture of GANs: The GAN architecture involves the generator starting from a random noise input and transforming it into data that mimics the true distribution. The discriminator, trained to distinguish between real and synthetic data, guides the generator through feedback, optimizing both networks until the discriminator can no longer differentiate synthetic from real data. Figure 2.3 shows an illustrative diagram of this process.

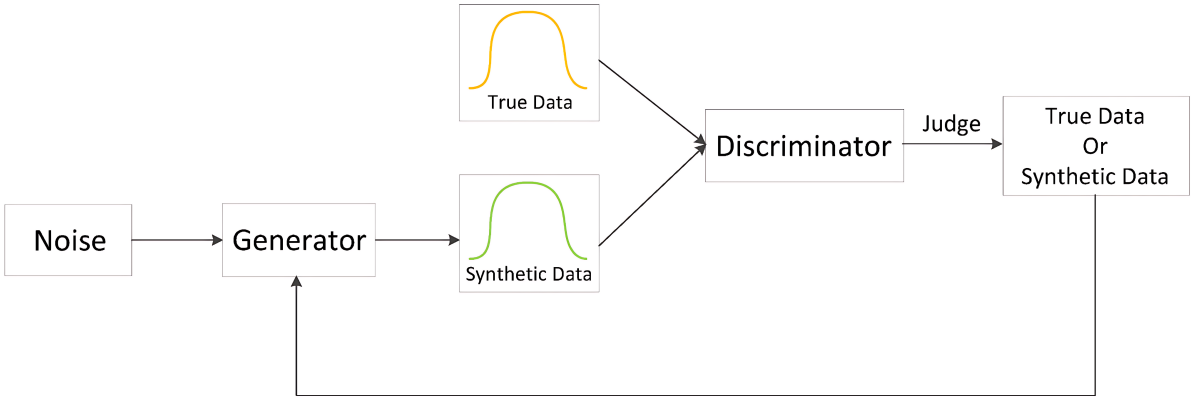


Figure 2.3: Illustration of the GAN architecture showing the flow from noise to generator and discriminator.

Extension to Conditional Tabular GANs (CTGAN): To address the unique challenges posed by tabular data, such as handling mixed data types and preserving complex data distributions, CTGAN modifies the traditional GAN architecture. CTGAN specializes in generating synthetic tabular data, employing conditional generation and mode-specific normalization techniques. CTGAN uses conditional vectors to manage categorical variables and adjusts its training strategy to account for imbalances in the class distributions, ensuring that the synthetic data maintains the statistical properties of the original dataset. This adaptation is crucial in healthcare, where maintaining the accuracy of relationships within patient data is essential.

By employing GANs and specifically CTGAN, researchers can leverage synthetic data to enhance data availability, improve model training, and ensure privacy, significantly advancing healthcare analyt-

ics and research.

2.2.5 Normalising Flows

Normalising flows are advanced generative models that model complex probability distributions through a series of invertible transformations. Their ability to preserve information during the data generation process makes them ideal for tasks that require high fidelity and detailed statistical properties (Cohen & Welling 2017). The unique properties of normalising flows, such as their invertibility and the ability to directly model complex distributions, make them particularly suitable for synthesising electronic health records (EHR). A more detailed explanation of normalising flows can be found in Chapter 3, Implementation.

2.3 Privacy and Ethical Considerations

The advent of data-driven methodologies in healthcare has intensified the focus on privacy and ethical considerations. As healthcare systems increasingly rely on data analytics for diagnostics, treatment personalization, and policy-making, safeguarding patient data privacy has never been more critical. Regulatory frameworks such as the General Data Protection Regulation (GDPR) in the EU and the Health Insurance Portability and Accountability Act (HIPAA) in the US set stringent guidelines on how patient data must be managed, emphasizing the protection of patient privacy (Xiang & Cai 2021).

2.3.1 Regulatory Frameworks

GDPR and HIPAA require the secure handling of patient data and impose conditions on consent, data anonymization, and individual rights. For instance, GDPR gives individuals the right to access, correct, and request the deletion of their personal data, which imposes specific obligations on data handlers in healthcare settings.

Implications for Synthetic Data: While synthetic data is considered a promising approach to preserving privacy, it must be generated in ways that comply with these regulatory requirements. The UK Statistics Authority's publication on the ethical considerations of using synthetic data emphasizes the importance of ensuring that synthetic data does not allow for re-identification of individuals, thereby aligning with GDPR's anonymization directives (UK Statistics Authority 2021).

2.3.2 Ethical Use of Synthetic Data

The ethical use of synthetic data extends beyond compliance with legal standards. It involves ensuring the accuracy, fairness, and non-discrimination of algorithms used in synthesizing data. Ethical

guidelines are evolving to address these issues, advocating for transparency in how data is used and how algorithms operate.

Accuracy and Fairness: Synthetic data must accurately reflect the demographic and clinical characteristics of the populations they represent to avoid bias in downstream applications. This is particularly important in training machine learning models for diagnostic tools, where biased data could lead to incorrect diagnoses or treatment plans.

Non-Discrimination: Precautionary measures must be taken to ensure that synthetic data creation does not unintentionally reinforce preexisting biases. Guidelines suggest incorporating checks and balances in the data generation process to identify and mitigate potential biases in synthetic datasets.

2.3.3 Guidelines and Best Practices

Emerging guidelines for the ethical use of synthetic data recommend:

- Rigorous validation of synthetic data to ensure it meets quality standards equivalent to the original data.
- Regular auditing of data generation algorithms to ensure compliance with ethical and regulatory standards.
- Engagement with stakeholders, including the public and patient advocacy groups, to foster trust and transparency in the use of synthetic data.

Responsibility in Implementation: As technology evolves, so does the responsibility of healthcare providers and researchers to ensure that their use of synthetic data upholds the highest ethical standards. This includes continuous education on the implications of data privacy laws and ethical guidelines, ensuring that all involved parties remain informed and vigilant in their data handling practices.

As long as the healthcare sector adheres to these principles, it can leverage synthetic data to advance medical research and patient care while maintaining ethical and privacy principles.

2.3.4 OpenMRS and Synthetic Data Integration

OpenMRS (Open Medical Record System) is an open-source platform that has significantly contributed to improving healthcare delivery in resource-limited settings. Originally designed to tackle the massive healthcare management problems in Africa, OpenMRS has evolved into a global standard for health information technologies in developing countries.

Overview of OpenMRS

OpenMRS is more than just a medical record system; it is a versatile, configurable platform designed to be adaptable to various healthcare environments. It is built on a model that allows for extensive customization, making it suitable for use in diverse medical contexts and healthcare infrastructures (Mamlin & Biondich 2006, Seebregts et al. 2009).

Key Features:

- **Modularity:** The system's architecture allows for the addition of modules to extend functionalities according to specific healthcare needs.
- **Community-Driven:** OpenMRS is supported by a vast global community that contributes to its continuous development and support, ensuring that the system is up-to-date with the latest medical and technological advancements.
- **Interoperability:** It is designed to interoperate with other health information systems, which is critical for comprehensive healthcare data management and analysis.

Integration of Synthetic Data

The integration of synthetic data into healthcare platforms like OpenMRS is pivotal, particularly in enhancing data privacy and expanding training datasets without compromising patient confidentiality. Synthetic data serves as a robust tool for testing, training, and research, enabling healthcare providers to simulate various scenarios without the risk of exposing sensitive information.

Challenges in Integration:

- **Data Compatibility:** Ensuring that synthetic data aligns well with the real-world data schema used by OpenMRS.
- **User Acceptance:** Gaining the trust of healthcare providers in the efficacy and safety of using synthetic data for clinical and operational purposes.

Case Studies and Research Implications: Several case studies have demonstrated the significant potential of synthetic data to enhance healthcare systems, especially in resource-limited environments. For example, (Health Data Research 2022) highlights the broader impact of synthetic data on healthcare research and system improvements. Building on these insights, systems like OpenMRS can be augmented using synthetic data to support medical research and healthcare services better. By addressing these

challenges, we aim to leverage OpenMRS more effectively, transforming it into a powerful tool that not only manages patient records but also plays a pivotal role in healthcare research and policy-making.

By addressing these challenges, we aim to leverage OpenMRS more effectively, transforming it into a more powerful tool that not only manages patient records but also plays a pivotal role in healthcare research and policy-making. As illustrated in Figure 2.4, the user interface of OpenMRS provides a comprehensive overview of patient details, making it an essential platform for both clinical management and data-driven healthcare initiatives.

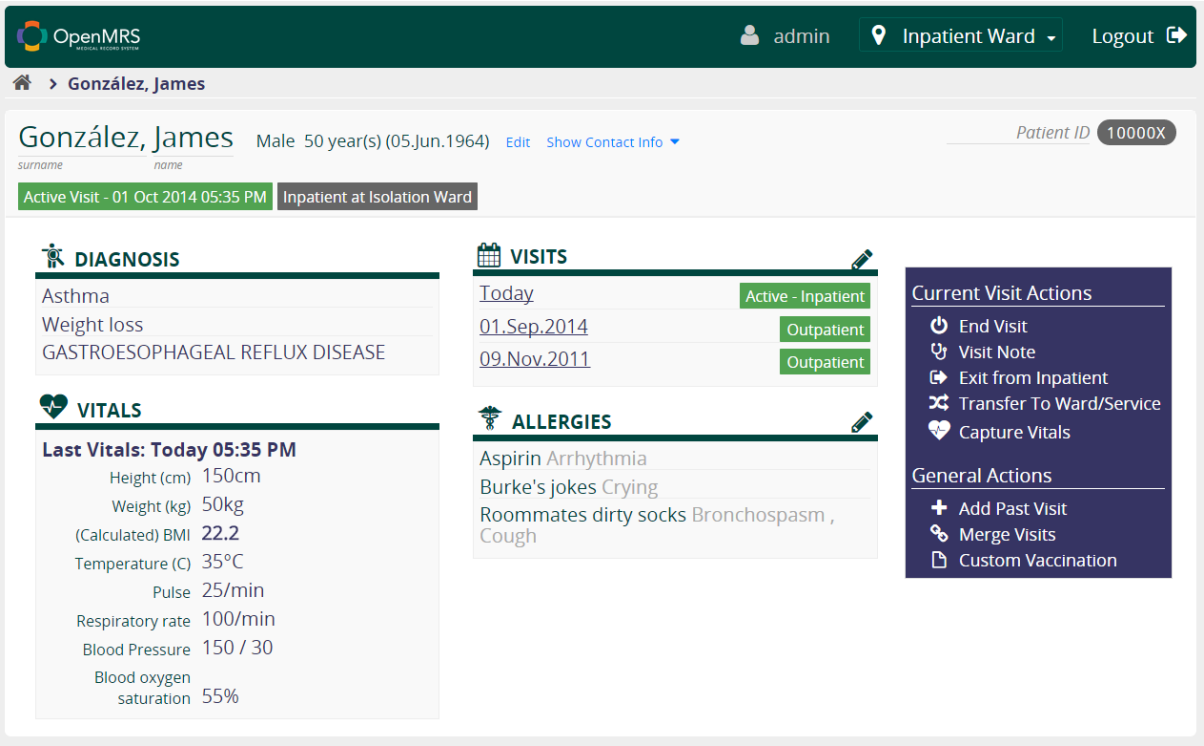


Figure 2.4: User Interface of OpenMRS showing details of a patient.

Chapter 3

Methodology and Implementation

This chapter outlines the comprehensive approach to generating synthetic medical data using normalising flows and Generative Adversarial Networks (GANs), detailing each stage from data collection and preprocessing to model implementation and evaluation. The integration of synthetic data with healthcare platforms like OpenMRS is also discussed, highlighting the practical implications of this research (Mamlin & Biondich 2006, Seebregts et al. 2009, Goodfellow et al. 2014*b*).

3.1 Research Design and Framework

The research adopts the Cross-Industry Standard Process for Data Mining (CRISP-DM) model, which provides a structured framework for tackling data mining tasks. As taught in the Data Science module (COMP5122M), using the CRISP-DM methodology is well-suited for the generation of synthetic data, guiding the process through six key phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment (Shearer 2000*b*).

3.1.1 CRISP-DM Framework

The process used in this project is based on the Cross-Industry Standard Process for Data Mining (CRISP-DM), which consists of six phases, illustrated in Figure 3.1. This framework, detailed in the Data Science module (COMP5122M), is essential for effectively organising and executing data mining projects.

- **Business Understanding**: Identify the requirements for synthetic data in healthcare.
- **Data Understanding**: Analyse existing EHR datasets, particularly MIMIC-IV.
- **Data Preparation**: Preprocess the dataset to ensure it is clean and consistent.
- **Modelling**: Develop models using normalising flows and GANs.
- **Evaluation**: Assess the quality and applicability of the synthetic data.
- **Deployment**: Integrate synthetic data with OpenMRS.

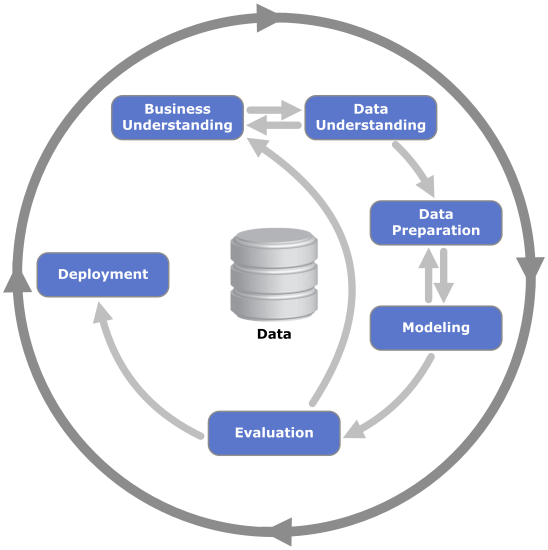


Figure 3.1: CRISP-DM Process Diagram, annotated from the CRISP Data Mining framework guide (Chapman et al., 2000).

Having reviewed the CRISP-DM framework, the Introduction chapter addresses the ”Business Understanding” phase, outlining the need for synthetic data in healthcare. The next step is ”Data Collection and Understanding,” which involves gathering and analysing the required healthcare data.

3.2 Data Collection and Understanding

3.2.1 Selection of EHR Datasets

To ensure the robustness of our synthetic data generation, it is crucial to select a dataset that is comprehensive and rich in relevant features. We considered several EHR datasets, including Health Facts, eICU Collaborative Research Database, and the MIMIC series. Here's a comparison:

Dataset	**Data Types**	**Time Period**	**Patient Count**	**Data Size (after extraction)**
MIMIC-IV	demographics, vitals, labs, notes, imaging reports	2008-2019	65,000 ICU and 200,000 ED patients	Approximately 70 GB
eICU	demographics, vitals, labs, notes, treatments	2014-2015	200,859 ICU stays	Approximately 200 GB
AmsterdamUMCdb	demographics, vitals, labs, medications	2003-2016	23,106 ICU stays	Approximately 30 GB

Table 3.1: Comparison of various EHR datasets used in healthcare research, detailing the data types, time period, patient count, and data size.

The MIMIC-IV dataset was selected due to its rich collection of ICU and ED data, which includes

a variety of numeric data types such as vitals, lab results, and imaging reports. These data types exhibit complex relationships between variables, making them particularly suitable for generating high-fidelity synthetic data. Additionally, MIMIC-IV's extensive usage in research, alongside its comprehensive and diverse patient population, ensures that the synthetic data generated from it will be broadly applicable. Furthermore, MIMIC-IV complies with stringent privacy regulations, making it a secure and ethical choice for this study.

3.2.2 Understanding the MIMIC-IV Dataset and Schema

MIMIC-IV (Medical Information Mart for Intensive Care IV) is a large-scale relational database comprising de-identified health-related data associated with patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2008 and 2019. This dataset represents a significant resource in healthcare research, providing real-world clinical data for various studies, including the development of machine learning models and clinical decision support systems.

MIMIC-IV contains comprehensive information on 383,278 distinct patients, representing 523,740 hospital admissions. The data spans over a decade, offering a longitudinal view of critical care practices and patient outcomes. This extensive timeframe allows researchers to analyse trends and changes in healthcare delivery over time.

Dataset Structure and Key Components

MIMIC-IV is structured into three primary modules, each focusing on different aspects of patient care:

Module	Component	Count	Description
Core	Patients	382,278	Unique patient records
	Admissions	523,740	Hospital admissions
	Transfers	2,075,470	Patient transfers between units
Hosp	Diagnoses	4,880,487	Diagnostic codes (ICD-9 and ICD-10)
	Procedures	2,511,668	Procedure codes
	Lab Tests	124,342,816	Laboratory test results
ICU	ICU Stays	76,540	Individual ICU admissions
	Observations	331,432,561	Charted observations (including vitals)
	Medications	13,320,280	Medication administrations

Table 3.2: MIMIC-IV Modules and Key Statistics

Each of these modules is interconnected, allowing for a comprehensive analysis of a patient's journey

through the healthcare system.

The MIMIC-IV dataset includes a diverse range of clinical data files organized into different categories, encapsulating various aspects of patient care. The dataset was provided in a compressed format (ZIP file) containing several subdirectories and files. Upon extraction, the primary contents were identified as follows :

Directory	CSV File	Description
Hosp	**admissions.csv**	**Data on patient admissions to the hospital.**
	d_hcpcs.csv	HCPCS code descriptions.
	d_icd_diagnoses.csv	ICD code descriptions for diagnoses.
	d_icd_procedures.csv	ICD code descriptions for procedures.
	d_labitems.csv	Descriptions of lab items used in the dataset.
	diagnoses_icd.csv	Data on diagnoses coded with ICD codes.
	drgcodes.csv	Diagnosis Related Group (DRG) codes and descriptions.
	emar.csv	Electronic Medication Administration Records.
	emar_detail.csv	Detailed information about medication administration.
	hcpcsevents.csv	Healthcare Common Procedure Coding System (HCPCS) events.
	labevents.csv	Laboratory measurements and results.
	microbiologyevents.csv	Microbiology test results and findings.
	omr.csv	**Outpatient Medication Records**.
	patients.csv	**Patient demographic information.**
	pharmacy.csv	Pharmacy dispensing records.
	poe.csv	Provider Order Entry (POE) records.
	poe_detail.csv	Detailed records of provider orders.
	prescriptions.csv	Prescription details.
	procedures_icd.csv	Data on procedures coded with ICD codes.
	provider.csv	Provider information.
	services.csv	Patient services data.
	transfers.csv	Data on patient transfers within the hospital.
ICU	caregiver.csv	Information on caregivers involved in ICU patient care.
	chartevents.csv	Detailed charted events, including vital signs and observations.
	d_items.csv	Descriptions of items used in the ICU.
	datetimeevents.csv	Time-stamped events in the ICU.
	icustays.csv	**Information on ICU stays.**
	ingredientevents.csv	Events related to medication ingredients administered.
	inputevents.csv	Records of inputs, such as fluids, administered to patients.
	outputevents.csv	Records of outputs, such as urine, from patients.
	procedureevents.csv	Procedures performed in the ICU.

Table 3.3: Summary of the MIMIC-IV Dataset Contents in the ‘hosp‘ and ‘icu‘ Directories.

Hosp Directory: Contains 21 tables related to general hospital records, including patient demographics, admissions, diagnoses, lab tests, and prescriptions.

ICU Directory: Contains 9 tables specific to ICU stays, capturing detailed clinical observations and treatments.

Data Types and Formats

- **Numerical data:** Includes vital signs, laboratory results, and medication dosages.
- **Categorical data:** Encompasses diagnoses, procedures, and demographic information.
- **Time-series data:** Represents continuous monitoring of patient conditions in the ICU.
- **Text data:** Includes clinical notes and discharge summaries.

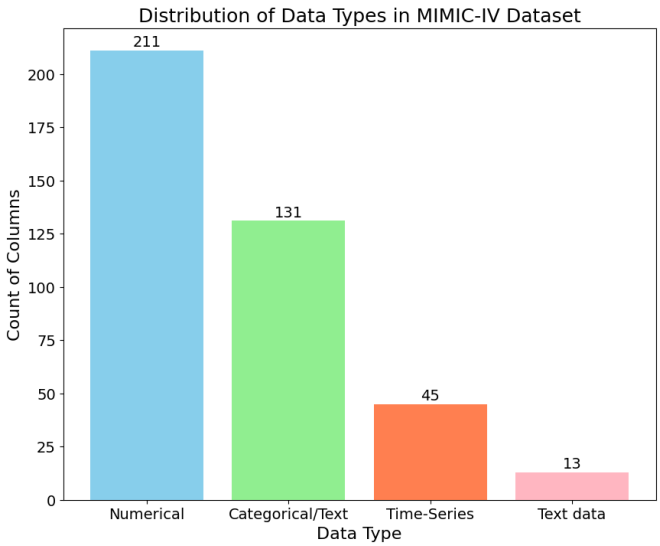


Figure 3.2: Data Types in MIMIC-IV

As shown in Figure 3.2, the dataset primarily comprises numerical and categorical/text data, with a smaller proportion of time-series data. This study will focus on analyzing the numerical and categorical data, which constitute most of the dataset.

The dataset utilizes standardized coding systems ICD-9 (International Classification of Disease) and ICD-10 for diagnoses, enhancing interoperability and facilitating comparative studies. Timestamps are recorded with minute-level precision, allowing for detailed temporal analysis of patient care. On average, each patient record contains approximately **1,200** distinct data points, highlighting the richness and density of the available information.

Data Linkage and Relationships

MIMIC-IV employs a robust system of unique identifiers to link data across different tables and modules:

- **SUBJECT_ID:** Uniquely identifies each patient
- **HADM_ID:** Represents individual hospital admissions
- **STAY_ID:** Denotes specific ICU stays

These identifiers enable us to trace a patient's entire clinical journey, from admission through various hospital departments and ICU stays, to discharge. The dataset maintains high data completeness, with core demographic information available for over 99% of patients. It is important to note that these

identifiers and timestamps are not synthetically generated; they are preserved in their original form within the database schema to uphold the ACID properties (Haerder & Reuter 1983), ensuring that the data remains consistent, reliable, and true to the original clinical records.

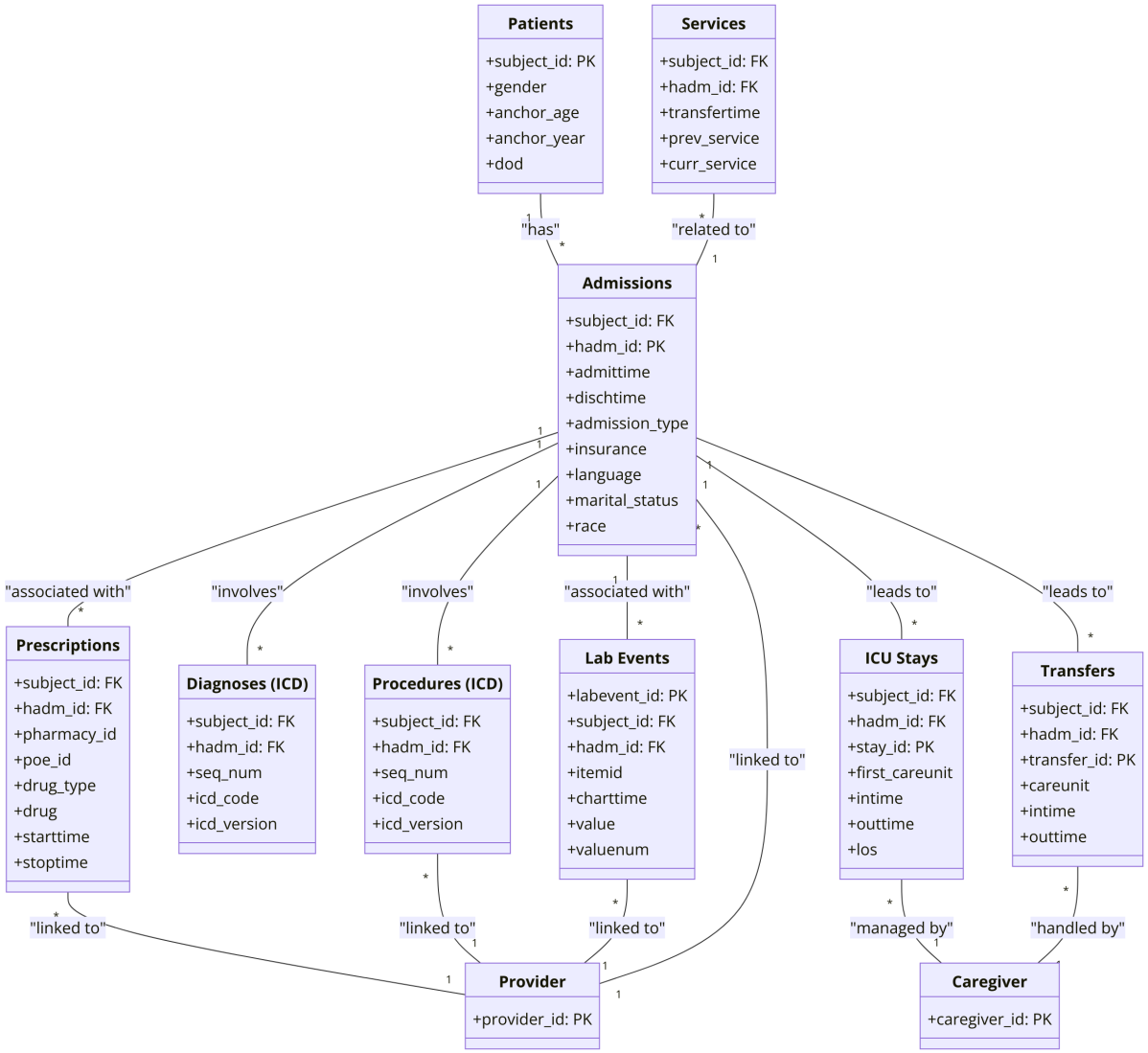


Figure 3.3: Simplified concept level Entity Relationship (ER) diagram of MIMIC-IV Dataset

The 3.3 illustrates the important tables present in the MIMIC-IV dataset, the high-level EHR structure defined in 2.1, highlighting each table's primary (PK) & foreign keys (FK) and the various one-to-many relationships between these tables. This diagram provides a clear overview of the interconnected tables, reflecting the complex data structure used to model patient care and hospital operations in the dataset.

Understanding these relationships and the structure of MIMIC-IV is crucial for effectively leveraging this rich dataset in healthcare research and for developing robust synthetic data generation models.

3.2.3 Understanding OpenMRS

OpenMRS (Open Medical Record System) is an open-source platform widely used to support healthcare delivery in resource-limited environments. It is built on a flexible data model with a concept dictionary that allows healthcare providers to meet their varied needs by customising forms, data structures, and reports without requiring programming knowledge.

Architecture and Features

The architecture of OpenMRS is highly modular, allowing for extensive customization to meet the specific needs of various healthcare environments. The figure below depicts this modular design, which illustrates how different modules interact within the OpenMRS ecosystem to provide comprehensive healthcare services.

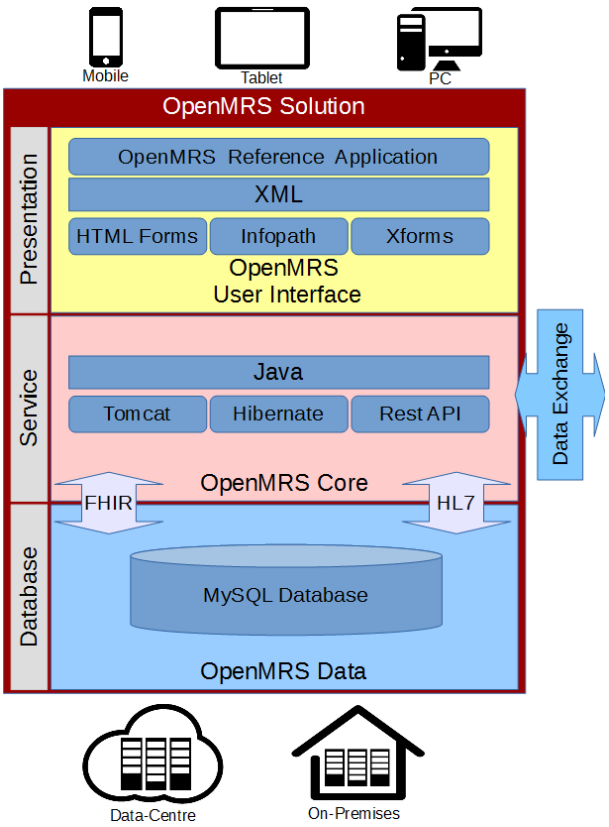


Figure 3.4: The modular architecture of OpenMRS

As shown in Figure 3.4, OpenMRS uses a core application that supports a variety of modules. Each module can be independently developed and integrated into the core, enhancing functionality without altering the core code. This architecture enables robust API integrations for seamless data exchange and CRUD (Create, Read, Update, Delete) operations on synthetic data. We effectively managed synthetic records by leveraging the OpenMRS database layer, ensuring their seamless incorporation into existing healthcare workflows.

3.2.4 Data Mapping Between MIMIC-IV and OpenMRS

Data mapping is a crucial step in integrating MIMIC-IV with OpenMRS. The table below details how specific fields from the MIMIC-IV dataset map to the corresponding entities within OpenMRS, facilitating the use of MIMIC-IV data for various healthcare applications in OpenMRS.

Table Name	MIMIC-IV Field	OpenMRS Entity	Description
PATIENTS	subject_id	person_id	Unique identifier for each patient
	dob	birthdate	Date of birth of the patient
	gender	gender	Gender of the patient
ADMISSIONS	hadm_id	encounter_id	Identifier for hospital admissions
	admittime	start_date (Encounter)	Admission time for encounters
	dischtime	end_date (Encounter)	Discharge time for encounters
ICUSTAYS	stay_id	visit_id	Identifier for ICU stays
	chartevents	obs	General observations recorded during ICU stays
DIAGNOSES ICD	diagnosis	diagnosis_concept	ICD-coded diagnoses
	icd_code	coded_diagnosis	Coded diagnoses using ICD standards
LABEVENTS	charttime	observation_datetime	Timestamp for clinical observations
	labname	lab_test_name	Name of the laboratory test
	labresult	test_result	Result of the laboratory test
PRESCRIPTIONS	prescription	medication_order	Details of medication prescribed
	dose	dose_value	Dosage of medication administered
	route	administration_route	Route of medication administration
MICROBIOLOGY EVENTS	microbiology	microbiology_result	Microbiology results and findings
PROCEDURES ICD	procedure	procedure_concept	Medical procedures performed

Table 3.4: Comprehensive mapping of MIMIC-IV fields to OpenMRS entities, demonstrating the integration process.

A Python script was written to map all the fields from the 8 tables mentioned in the 3.4 MIMIC-IV to OpenMRS to facilitate the integration process. This script automates the conversion of data formats, ensures consistency in data mapping, and addresses any discrepancies between the datasets, thus streamlining the data integration and validation process. The script is available in the GitHub repository linked in the Appendix section, providing access to the code and further documentation.

3.3 Data Preprocessing

The primary dataset utilized in this dissertation is the MIMIC-IV dataset, a comprehensive collection of de-identified health records. This section details the preprocessing methods to prepare the data for modelling (Johnson et al. 2016).

3.3.1 Analysis of Key Tables

Initial Data Inspection

We began by inspecting key tables in the MIMIC-IV dataset to understand their structure and content. The table below summarizes the number of rows, columns, and key columns of interest for each of the main tables.

Table Name	Rows	Columns	Key Columns
`Admissions`	431,231	16	`admittime`, `dischtime`, `deathtime`, `admission_type`
`Patients`	299,712	6	`subject_id`, `gender`, `anchor_age`, `anchor_year`, `dod`
`ICU stays`	73,181	8	`subject_id`, `hadm_id`, `stay_id`, `intime`, `outtime`
`OMR`	6,439,169	5	`subject_id`, `chartdate`, `result_name`, `result_value`

Table 3.5: Summary of Key Tables in the MIMIC-IV Dataset

The following figures provide a visual analysis of the Admission and Patients table.

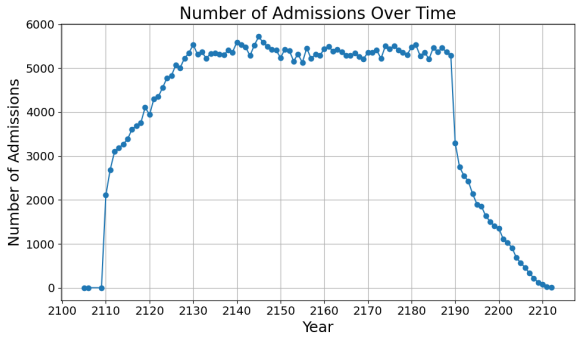


Figure 3.5: Number of Admissions Over Time

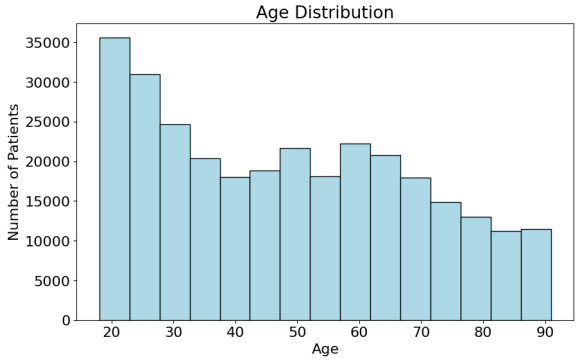


Figure 3.6: Age Distribution of Patients

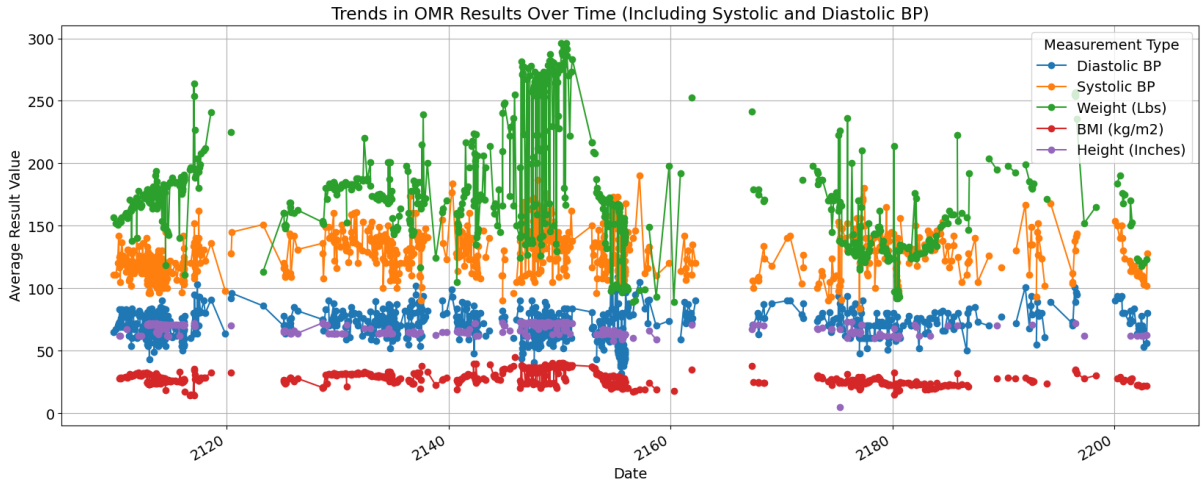


Figure 3.7: Trends in OMR (Outpatient Medication Records) Results Over Time

Figure 3.5 illustrates the distribution of admissions over time, showing trends in the number of

hospital admissions. Figure 3.6 provides an overview of the age distribution among patients, highlighting a concentration of middle-aged to elderly individuals.

Figure 3.7 tracks the trends in OMR results over time, showcasing consistent monitoring of key metrics like blood pressure, which are crucial for patient care.

Note: The OMR table will be reviewed in detail in further chapters.

3.3.2 Date and Time Conversion

Date and time columns across all tables were standardized to `datetime` formats to ensure consistency in analysis. This was particularly important for tables like `Admissions` and `ICU Stays`, where accurate event timing (e.g., `admittime`, `dischtime`, `intime`, `outtime`) is crucial.

The original dataset featured dates ranging from 2110 to 2201 due to a de-identification process that shifted real dates forward to protect patient privacy. These dates were adjusted back using a proportional scaling method to align with the actual data collection period (2008 to 2019).

Proportional Scaling Method

The dates were adjusted by compressing the original date range (2110 to 2201) into the real-world range (2008 to 2019). This process maintained the relative timing between events while ensuring the dates fit within the actual data collection period. Given the original and target date ranges, each date t_{original} in the original range was scaled to a new date t_{scaled} in the target range using the following transformation:

$$t_{\text{scaled}} = t_{\text{target\_min}} + \frac{t_{\text{target\_max}} - t_{\text{target\_min}}}{t_{\text{max}} - t_{\text{min}}} \times (t_{\text{original}} - t_{\text{min}})$$

Here:

- t_{scaled} represents the date in the scaled range.
- $t_{\text{target\_min}}$ and $t_{\text{target\_max}}$ are the minimum and maximum dates in the target (real-world) range, which span from 2008 to 2019.
- t_{min} and t_{max} refer to the minimum and maximum dates in the original de-identified range (2110 to 2201).

Example

For instance, a date in the original range (e.g., July 15, 2150) was mapped to a corresponding date in the target range (e.g., late October 2012) using this method, preserving its relative position within the date range. Figure 3.8 illustrates the transformation, comparing the original de-identified dates with the adjusted dates aligned to the actual data collection period.

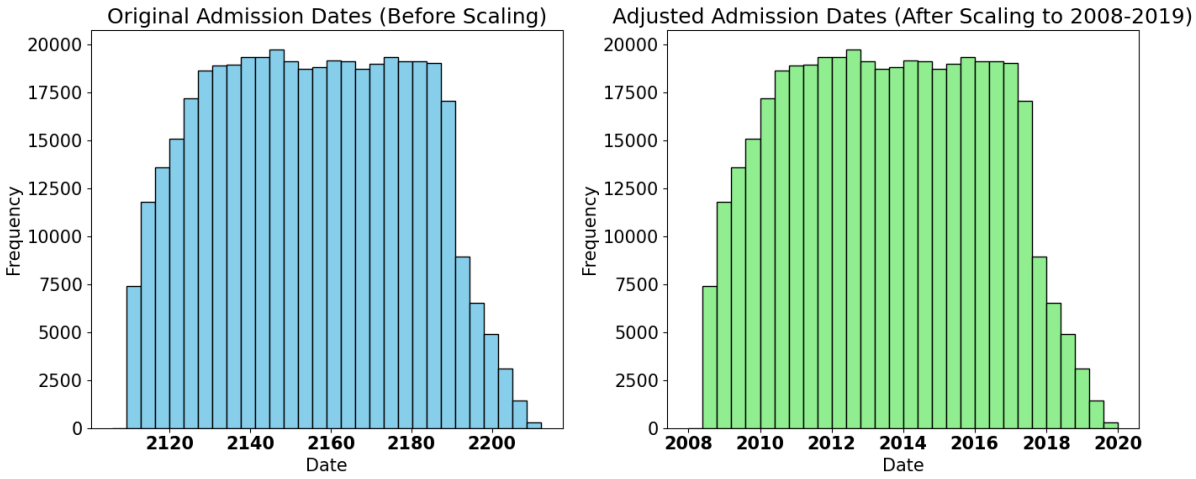


Figure 3.8: Comparison of Admission Dates Before and After Scaling

3.3.3 Handling Missing Values

Missing values were handled thoughtfully across the datasets. For example, in the `Patients` dataset, 270,636 entries lacked a date of death (`dod`), signifying that these patients were alive at the study's conclusion. These missing values were intentionally retained as NaNs to preserve the integrity of the data for further analysis.

Placeholder values, such as "?" in the `language` column of the `Admissions` dataset, were replaced with 'Unknown' to maintain consistency. Categorical data, such as `gender` in the `Patients` dataset, was checked for consistency with no errors, ensuring valid and expected values.

Moreover, since the goal is to generate synthetic data that closely mirrors real-world data, it is also important to reflect the presence of missing values in the synthetic dataset. In real-world scenarios, missing values are common, and imputing them could distort the authenticity of the synthetic data. Therefore, if a column in the original dataset has missing values, the synthetic data should also exhibit similar patterns of missingness to maintain realism.

3.3.4 Feature Engineering

Feature engineering is the process of transforming raw data into meaningful features that enhance model performance:

- **Creating New Features:** New features were created by splitting existing variables to extract additional insights, such as separating systolic and diastolic values from blood pressure readings in the OMR dataset. The below tables 3.6 and 3.7 illustrate this transformation, with the original blood pressure readings highlighted in yellow in the first table and the corresponding separated systolic and diastolic values highlighted in the second table.

subject_id	seq_num	result_name	result_value
10000032	1	Blood Pressure	110/65
10000032	1	Weight (Lbs)	94
10000032	1	BMI (kg/m2)	18.0
10000032	1	Height (Inches)	60
10000032	1	Weight (Lbs)	92.15

Table 3.6: Overview of the First Few Entries in the Outpatient Medical Records

subject_id	BMI (kg/m2)	Height (Inches)	Weight (Lbs)	Systolic_BP	Diastolic_BP
10000032	18.0	60	94	110	65
10000117	19.6	64.5	121	124	66
10000635	31.7	70	221	140	86
10000719	37.0	67	236	144	88
10000826	20.5	68	98.9	88	58

Table 3.7: Overview of the First Few Entries in the **transformed** Outpatient Medical Records

- **Normalizing Continuous Variables:** Continuous variables, such as lab results and vital signs, were normalized to ensure that each feature contributes equally to the distance calculations during model training. This step is crucial for neural networks, which are sensitive to the scale of input data.

  Normalisation was performed using the `MinMaxScaler` from `scikit-learn`. This method scales each continuous variable to a specified range, typically between 0 and 1. The scaling process is defined by the following transformation.

$$x_{\text{scaled}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

  Where x is the original value, x_{min} is the minimum value of the feature in the dataset, and x_{max} is the maximum value of the feature in the dataset. This transformation ensures that all features are on a comparable scale, mitigating the risk of any single feature dominating the model due to its magnitude.

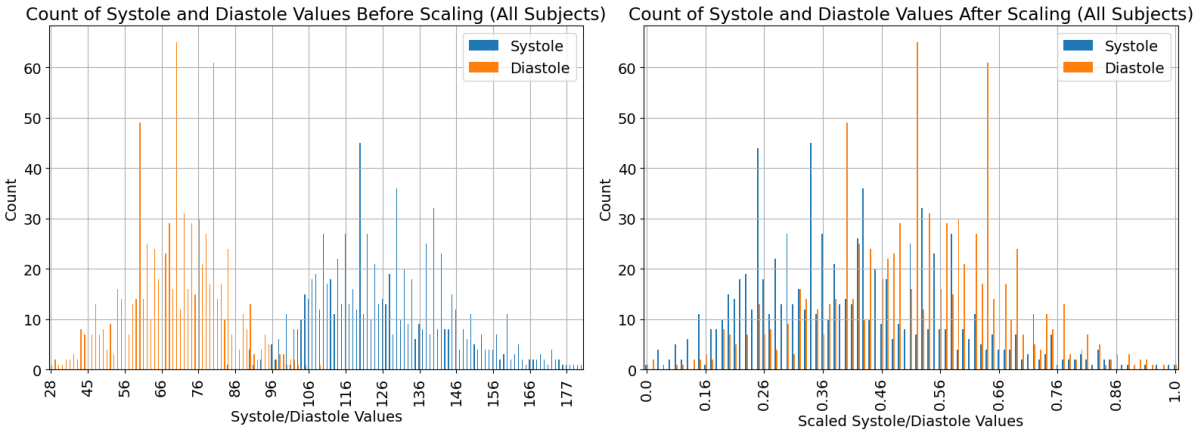


Figure 3.9: Comparison of Systole and Diastole Values Before and After Scaling

To illustrate the effect of this normalisation, Figure 3.9 shows the distribution of systole and diastole values before and after scaling. The figure demonstrates how these variables, originally on

different scales, were transformed to fit within the same range [0, 1], ensuring that both contribute equally during model training.

3.3.5 Data Splitting

To comprehensively evaluate the model, the dataset was divided into three subsets:

- **Training Set:** Comprising 70% of the data, approximately 301,861 records from the 431,231 in the Admissions table were used to fit the model, enabling it to learn patterns and relationships.
- **Validation Set:** 15% of the dataset, about 64,685 records, was allocated for hyperparameter tuning and performance monitoring, helping to prevent overfitting.
- **Test Set:** The final 15%, also 64,685 records, was reserved for evaluating the model's ability to generalise to new, unseen data, providing a robust measure of real-world performance.

These preprocessing steps are critical to enhancing data quality and model robustness, ensuring that the data set is well prepared for subsequent modelling tasks. The preprocessing pipeline was automated using Python scripts, which are available in the GitHub repository linked in the Appendix section, facilitating reproducibility and transparency in the research process.

3.4 Model Implementation: Normalising Flows and GANs

The core of this research lies in implementing advanced generative models—specifically, normalising flows and Generative Adversarial Networks (GANs) to generate synthetic Electronic Health Record (EHR) data that accurately represents the underlying real-world distributions. These models are selected for their ability to model complex, high-dimensional data distributions, which is particularly challenging in the context of EHR data. This section provides a detailed description of the mathematical foundations, architecture, and training processes of these models. The structure of the normalising flows model, which plays a pivotal role in this research, is depicted in Figure 3.10 (Dinh et al. 2017*a*, Kingma & Dhariwal 2018).

3.4.1 Normalising Flows

Normalizing flows is a powerful method for constructing complex probability distributions by transforming a simple base distribution through a series of invertible mappings. The key idea is to start with an easy-to-sample distribution (like a standard normal) and then apply a sequence of bijective functions to "flow" this distribution into a more complicated target distribution.

Formally, let $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ be a random variable drawn from a simple prior distribution $p_{\mathbf{z}}$, typically a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Normalizing flows apply a series of N invertible transformations $f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_N}$ to $\mathbf{z}$, resulting in a transformed variable $\mathbf{x}$ that follows the target distribution $p_{\mathbf{x}}(\mathbf{x})$:

$$\mathbf{x} = f_{\theta_N} \circ f_{\theta_{N-1}} \circ \cdots \circ f_{\theta_1}(\mathbf{z})$$

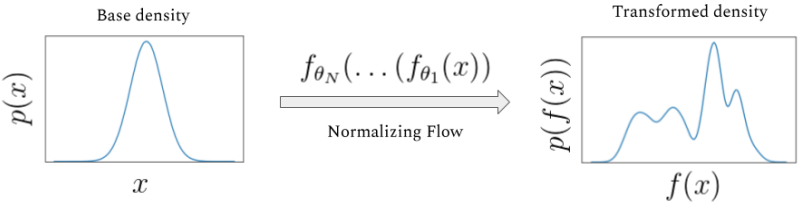


Figure 3.10: The process of normalizing flows

The key property of normalizing flows is that they allow us to compute the exact likelihood of the data under the model using the change of variables formula:

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{z}}(\mathbf{z}) \left| \det \left(\frac{\partial f^{-1}}{\partial \mathbf{x}} \right) \right|$$

Where f^{-1} is the inverse of the transformation f, and $\det \left(\frac{\partial f^{-1}}{\partial \mathbf{x}} \right)$ is the determinant of the Jacobian matrix of the inverse transformation.

The Jacobian matrix, denoted by $\mathbf{J}_{f^{-1}}(\mathbf{x}) = \frac{\partial f^{-1}}{\partial \mathbf{x}}$, captures how the transformation f^{-1} changes the space around each point $\mathbf{x}$. In other words, it tells us how much the transformation is stretching or compressing the space at each location. This is crucial because, in probability, we need to adjust the density accordingly: if the space is stretched, the probability density should decrease, and if it's compressed, the density should increase.

The determinant of the Jacobian matrix, $\det(\mathbf{J}_{f^{-1}}(\mathbf{x}))$, measures this volume distortion introduced by the transformation f^{-1}. In the context of normalising flows, this determinant ensures that as we transform the base distribution $p_{\mathbf{z}}$ into the target distribution $p_{\mathbf{x}}$, the probabilities are adjusted correctly, maintaining the integrity of the distribution throughout the transformation process.

Figure 3.10 visually illustrates how a base density, such as a standard Gaussian distribution, is transformed into a more complex density through a series of invertible transformations. The change in the shape of the distribution reflects the flexibility of normalising flows to capture intricate patterns in the data. The computation of the Jacobian determinant at each step ensures that the probability mass is appropriately adjusted.

In high-dimensional spaces, computing the determinant of the Jacobian can be computationally intensive. However, specific architectures used in normalising flows, such as RealNVP (Dinh et al. 2017*b*)

and Glow (Keller et al. 2021), are designed to ensure that this determinant can be computed efficiently, often by exploiting the structure of the transformations (e.g. using affine transformations or invertible convolutions).

Model Architecture

In this research, we employ normalising flows to model the complex distributions inherent in Electronic Health Record (EHR) data, specifically focusing on generating numerical data/columns. EHR datasets often contain a mix of continuous and categorical variables, which are typically correlated and may exhibit complex dependencies. Traditional generative models may struggle to capture these intricate structures. However, normalising flows are particularly well-suited for generating numerical data because they can model complex distributions through invertible transformations, maintaining the relationships between numerical variables. This makes them ideal for generating realistic synthetic numerical EHR data, preserving both the statistical properties and the privacy of the original dataset.

Specifically, we utilise advanced architectures inspired by RealNVP and Glow, which are well-suited for high-dimensional numerical data due to their invertible transformations. The normalising flows model consists of a series of transformations that progressively map the base distribution, typically a multivariate Gaussian, to the target complex distribution. Figure 3.11 provides a conceptual overview of the normalising flow architecture, illustrating the flow from the base distribution through a chain of invertible transformations to the final output distribution. This figure highlights the different types of flows, including coupling layers and autoregressive flows, which are key components in the architecture used in this research.

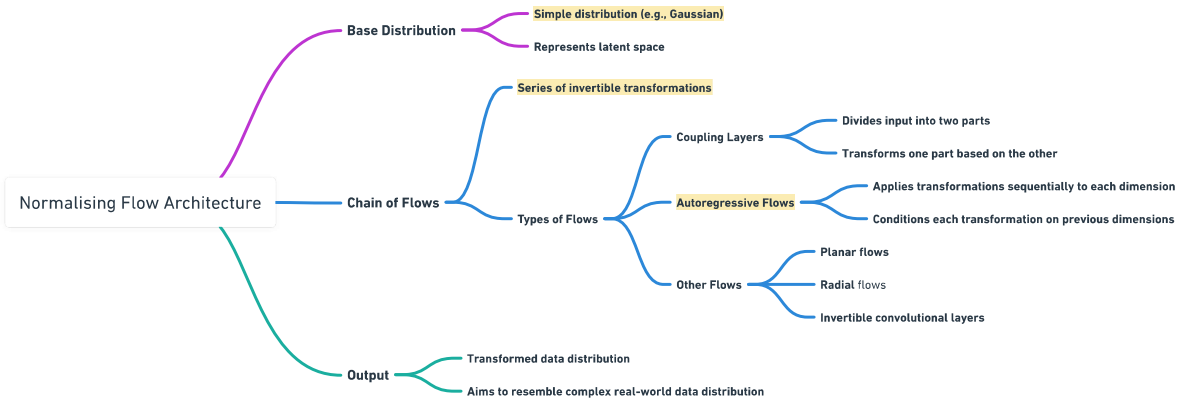


Figure 3.11: Conceptual overview of a typical normalising flow architecture, showing the progression from the base distribution to the target distribution through a series of invertible transformations.

In our implementation, each transformation in the flow is composed of two main components: a Random Permutation and a Masked Affine Autoregressive Transform. The Random Permutation layer

shuffles the input features, adding a layer of complexity that enhances the model's flexibility. Following this, the Masked Affine Autoregressive Transform applies an affine transformation to the data, conditioned on the previous outputs, using neural networks to parameterise the scale and translation functions. Mathematically, this can be represented as:

$$\mathbf{y}_1 = \mathbf{x}_1, \quad \mathbf{y}_2 = \mathbf{x}_2 \odot \exp(s(\mathbf{x}_1)) + t(\mathbf{x}_1)$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are partitions of the input data, and $s(\mathbf{x}_1)$ and $t(\mathbf{x}_1)$ represent the scale and translation functions. In our model, we use 5 layers of transformations, with each Masked Affine Autoregressive Transform consisting of a hidden layer of 32 features. The sequential application of these layers, as shown in Figure 3.12, allows the model to capture the intricate dependencies in the numerical EHR data.

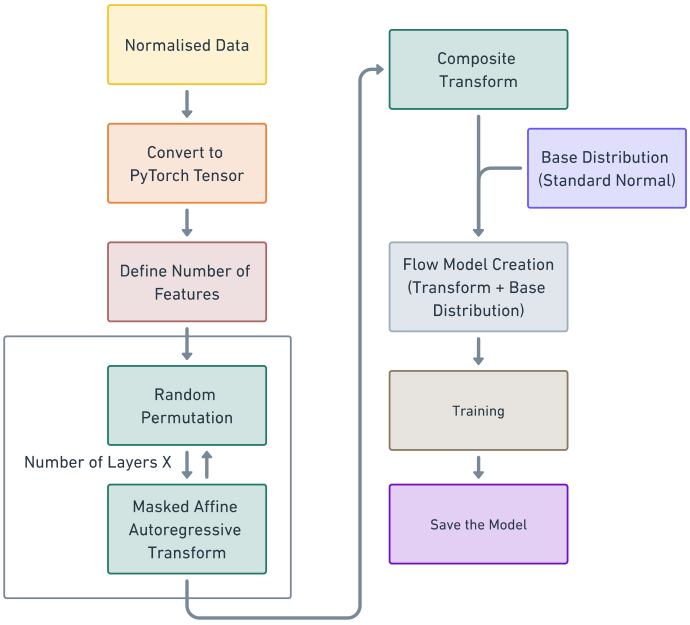


Figure 3.12: Architecture of the normalising flows model, illustrating the sequence of Random Permutation and Masked Affine Autoregressive Transform layers.

Invertibility and Latent Space: A critical property of normalising flows is the invertibility of the transformations, ensuring a bijective mapping between the latent space and the data space. This feature allows for the exact computation of likelihoods and enables back-and-forth mapping between observed data and latent representations. The latent space, typically modelled as a simple Gaussian distribution, provides a lower-dimensional representation of the data. This invertibility is not only essential for generating synthetic numerical data that accurately mirrors real EHR data but also for exploring the underlying structure of the data. The analysis of the latent space can reveal patterns and relationships that are not immediately obvious in the raw data.

As depicted in Figures 3.11 and 3.12, the model architecture consists of a sequence of transformations, starting with Random Permutation layers that rearrange the input features to enhance the flexibility of the model. This is followed by Masked Affine Autoregressive Transforms, which apply affine transformations conditioned on previous outputs, enabling the model to learn complex dependencies within the numerical data. Each layer is crucial in allowing the model to map the base distribution to the target distribution, capturing the intricate patterns in the EHR data.

Training Procedure: The training process for normalising flows focuses on maximising the log-likelihood of the observed data under the model, which effectively corresponds to minimising the negative log-likelihood. In our implementation, the Adam optimiser with a learning rate of 1×10^{-3} is used to adjust the model parameters iteratively. Throughout the training, gradient clipping with a maximum norm of 1.0 is applied to prevent the explosion of gradients, ensuring stable and efficient learning. The training process is conducted over multiple epochs (200 epochs), with checkpoints saved every 50 epochs to facilitate recovery and analysis at different stages. We also monitor key evaluation metrics, such as log-likelihood on a validation set, to assess model performance and guide training adjustments.

Evaluation and Regularisation: During training, we employ the regularisation technique of weight decay to prevent overfitting and ensure the generalisability of the model. The model is evaluated by analysing the learned latent space representations and comparing them with known patterns in the EHR data. This evaluation helps fine-tune the model and understand the underlying structure of the data.

Following the use of normalising flows for generating numerical data, we utilise the CTGAN (Conditional Tabular GAN) model to generate synthetic categorical data.

3.4.2 Conditional Tabular Generative Adversarial Networks (CTGAN)

Conditional Tabular GANs (CTGANs) are an advancement over traditional GANs, designed to address the unique challenges of generating synthetic tabular data. Unlike traditional GANs, which are mainly used for image and text generation, CTGANs are specifically tailored for structured datasets common in domains such as electronic health records (EHR), finance, and more. This architecture is particularly relevant for datasets with mixed data types and complex inter-column dependencies, making it an essential tool in these fields.

Data Preprocessing

The first step in implementing CTGAN involved preprocessing the admissions dataset, which included categorical features such as `admission_type`, `admission_location`, `insurance`, `marital_status`, `race`. These features were selected due to their importance in the dataset and their categorical nature, which required encoding.

Label Encoding: Each categorical column was transformed using label encoding, converting categorical variables into numerical labels that CTGAN could process. Label encoders were used to maintain a mapping of the original categories to their corresponding numerical labels, allowing for potential reverse transformation if needed.

Handling Missing Values: The dataset was examined for missing values before training. Any missing entries were imputed with a placeholder value (`-1`) to ensure that the dataset was complete and ready for training, as missing values can disrupt the model's learning process.

CTGAN Model Initialization

The CTGAN model was initialized with specific hyperparameters tailored to the dataset. A learning rate of 2×10^{-4} was set for both the generator and discriminator, with a batch size of 500. These settings were chosen to optimize the model's performance, balancing the learning speed with the quality of the generated synthetic data.

Training Process and Loss Tracking

The model was trained over 100 epochs. While CTGAN typically does not return a loss value, a proxy loss was simulated for demonstration purposes to monitor the model's progress throughout training.

Epoch Loop: The model was trained incrementally, one epoch at a time. After each epoch, the model's progress was tracked, and every 50 epochs, the model was saved as a checkpoint. This approach allowed for detailed monitoring and adjustment during training, and the checkpointing strategy ensured that the model could be preserved and resumed if needed.

Model Saving and Synthetic Data Generation

To preserve the trained model, it was saved to a file every 50 epochs using the Python `pickle` module. This allows the model to be reloaded and used in future analyses without retraining from scratch.

Generating Synthetic Data: After the training was completed, the CTGAN model was used to generate synthetic data samples. This synthetic data can then be analyzed to assess the quality of the model and ensure that it preserves the statistical properties of the original dataset.

CTGAN Architecture

The CTGAN architecture addresses several challenges inherent in tabular data generation:

- **Mixed Data Types:** CTGAN effectively handles continuous and categorical variables within the same model, a crucial feature for datasets like EHRs containing diverse data types.
- **Conditional Generation:** The generator receives noise input concatenated with conditional vectors representing categorical variables, allowing it to generate samples conditioned on specific categories. This ensures that the synthetic data mirrors the distribution of the original dataset.
- **Privacy Preservation:** CTGAN generates synthetic data without direct access to individual records, which is crucial for sensitive data like patient information. The model learns underlying patterns, enabling it to produce realistic synthetic datasets that protect privacy while retaining essential characteristics.

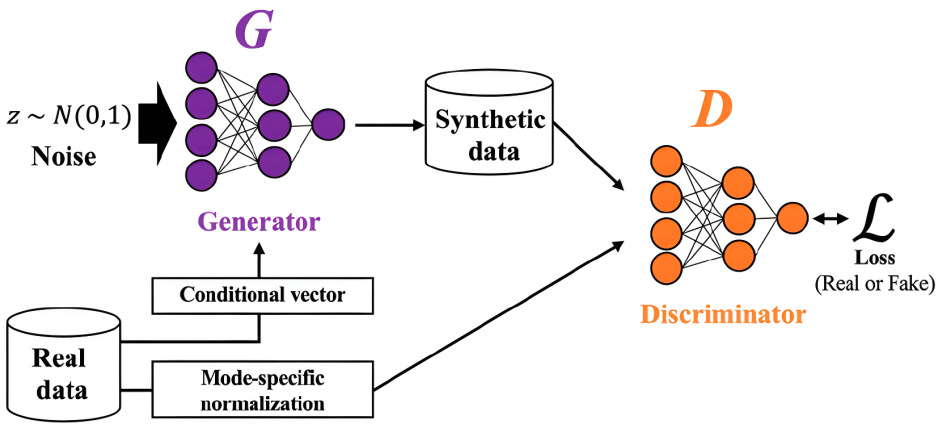


Figure 3.13: CTGAN architecture showing the generator and discriminator components (Watanuki et al. 2024).

Handling Categorical Variables

CTGAN employs several sophisticated techniques to effectively manage categorical and numerical data.

- **Mode-specific Normalization:** This technique captures the conditional distributions of continuous variables given categorical variables, preserving complex relationships within the data. For instance, it helps maintain the accurate relationship between a patient's age (a continuous variable) and their diagnosed conditions (categorical variables).
- **Conditional Vectors:** Each categorical variable is represented by a conditional vector, a mechanism used to control the generation process in GANs. A conditional vector is essentially a binary or one-hot encoded vector that specifies the category (or combination of categories) for which the generator should produce data. For example, when generating data for a variable like "Gender,"

the conditional vector would indicate whether the generated data should correspond to ”male” or ”female.” This ensures that the synthetic data respects the real-world distribution, such as an equal or realistic proportion of male and female records.

- **Training-by-Sampling:** CTGAN uses conditional sampling during training to ensure that rare categories are adequately represented in the synthetic data, addressing potential class imbalances in the MIMIC-IV dataset and generating balanced datasets.

Training Dynamics

The adversarial training process in CTGAN is carefully designed to manage the complexities of tabular data. The generator aims to produce synthetic samples that are indistinguishable from real data by learning to create realistic combinations of features. Meanwhile, the discriminator is trained to identify synthetic entries, continuously pushing the generator to improve its output's realism.

Balancing the learning rates of the generator and discriminator is critical. If the discriminator learns too quickly, it may reject all generated samples, stalling the generator's learning process. Conversely, if the generator learns too quickly, it may produce poor-quality samples that the discriminator cannot effectively challenge, leading to suboptimal results. Maintaining this balance is essential for preserving the integrity of the generated data.

Loss Functions

CTGAN utilises specific loss functions tailored to the challenges of tabular data generation:

- **Wasserstein Loss:** This loss function is particularly effective in stabilising GAN training. Unlike traditional GAN loss functions, Wasserstein Loss provides a meaningful gradient even when the discriminator is near optimal, helping generate diverse samples and avoiding the problem of mode collapse, where only a limited variety of samples are produced. The application of Wasserstein Loss in GANs has been widely studied and validated, showing its effectiveness in improving the quality of generated data (Arjovsky et al. 2017).

- **Information Loss:** This loss function ensures that the generated categorical variables match the distribution in the real data, maintaining the correct prevalence of various categories and reflecting the diversity and distribution of the original dataset.

By leveraging these advanced techniques, CTGAN generates high-quality synthetic data that preserves the statistical properties and complex relationships found in real tabular datasets. This capability is particularly valuable in applications involving sensitive or complex data, such as electronic health

records, where it facilitates research and system testing while protecting individual privacy. The following section will explore how synthetic data generated from Normalising flows and CTGAN can be integrated into real-world systems for further analysis and testing.

3.5 Synthetic Data Generation and Integration

The process of generating synthetic data involves sampling from the trained models and integrating the data into the healthcare platform OpenMRS.

The figure on the right illustrates the ETL (Extract, Transform, Load) process flow used for integrating synthetic data into OpenMRS, as implemented by the (Epistasis Lab 2023) in their MIMIC-LOADING module . This process involves extracting data generated by the Normalising Flows and CTGAN models, transforming it to match the OpenMRS schema, and loading it into an MSSQL relational OpenMRS database. The `MIMIC-LOADING` module provides a robust framework for ensuring that the synthetic data is accurately and efficiently mapped, enabling the testing and validation of OpenMRS workflows with synthetic patient records.

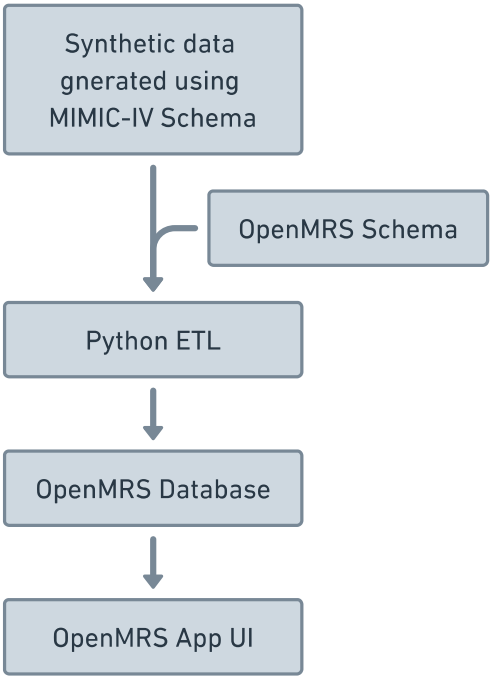


Figure 3.14: ETL Process Flow for Integration with OpenMRS (Epistasis Lab 2023).

3.6 Evaluation Methods

Evaluating the synthetic data generated in this study involves multiple mathematical and statistical approaches to ensure it meets the required realism, utility, and privacy standards. The following methods comprehensively assess the quality and security of the synthetic Electronic Health Records (EHR) data.

3.6.1 Descriptive Statistical Comparison

Descriptive statistics is the primary method used to evaluate synthetic data. Key metrics such as mean (μ), median, standard deviation (σ), and range are computed for both the synthetic and real

datasets. These metrics are compared to assess whether the synthetic data faithfully replicates the statistical properties of the real data. The comparison ensures that the synthetic data maintains the same overall distribution as the original dataset, which is crucial for its usability in research.

- **Mean (μ)**: The average value of a dataset is calculated by summing all the values and dividing by the number of values.
- **Standard Deviation (σ)**: A measure of the dispersion of data points around the mean. It indicates the degree of variation within the dataset.
- **Median**: The middle value (50^{th} percentile) of a dataset when it is ordered from least to greatest. If the dataset has an even number of observations, the median is the average of the two middle numbers.
- **Range**: The difference between the maximum and minimum values in a dataset.

3.6.2 Kolmogorov-Smirnov Test for Continuous Data

The Kolmogorov-Smirnov (K-S) test is used to compare the distributions of a continuous variable in the synthetic and real datasets. This non-parametric test measures the maximum difference between the empirical cumulative distribution functions (ECDFs) of the two datasets.

- **Empirical Cumulative Distribution Function (ECDF)**: A step function that estimates the cumulative distribution function (CDF) of a sample. It represents the proportion of data points less than or equal to a certain value.
- **Supremum (**$\sup$**)**: The least upper bound or the maximum value in the context of the differences between the ECDFs of two samples.

Let $F_{\text{real}}(x)$ and $F_{\text{synthetic}}(x)$ represent the ECDFs of the real and synthetic datasets, respectively. The K-S statistic is defined as:

$$D_{n,m} = \sup_x |F_{\text{real}}(x) - F_{\text{synthetic}}(x)|$$

where $\sup_x$ denotes the supremum (maximum) over all x values.

In K-S test the null hypothesis H_0 is that the two samples are drawn from the same distribution. A significant $D_{n,m}$ value indicates a difference between the distributions.

3.6.3 Correlation Analysis

Correlation analysis evaluates the relationships between different variables within the synthetic data. The Pearson correlation coefficient r is calculated for pairs of continuous variables in both datasets to assess how well the synthetic data preserves multivariate dependencies.

- **Pearson Correlation Coefficient (r_{XY})**: A measure of the linear relationship between two continuous variables X and Y, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no correlation.
- **Mean (μ_X and μ_Y)**: The average values of the variables X and Y, respectively.
- **Number of Observations (n)**: The total number of paired observations or data points used in the correlation calculation.

The Pearson correlation coefficient is defined as:

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_X)^2 \sum_{i=1}^{n}(y_i - \mu_Y)^2}}$$

where x_i and y_i are individual data points for variables X and Y, respectively, μ_X and μ_Y are the means of X and Y, and n is the total number of observations.

3.6.4 Visual Inspection

Visual inspection supplements statistical evaluations by providing a graphical comparison of data distributions. Techniques such as histograms, box plots, and scatter plots are employed to visually compare the distributions of individual variables and their interactions. This approach helps identify discrepancies or anomalies in the synthetic data that might not be evident from statistical tests alone.

- **Histogram**: A graphical representation of the distribution of numerical data, where the data is divided into intervals (bins), and the frequency of data points within each bin is shown as bars.
- **Box Plot**: A graphical representation showing the distribution of data through their quartiles, highlighting the median, lower, and upper quartiles, and any potential outliers.
- **Scatter Plot**: A graph in which the values of two variables are plotted along two axes, revealing any potential relationship between them.

3.6.5 Privacy Risk Assessment

Privacy risk assessment focuses on ensuring that the synthetic data does not compromise patient confidentiality. Re-identification risk analysis is used to determine the likelihood of matching synthetic

records with real ones. Techniques like k-anonymity and differential privacy are employed to measure and mitigate this risk.

- **Re-identification Risk Analysis**: A process to evaluate the probability that a synthetic data record could be linked to a real individual in the original dataset.
- **k-Anonymity**: A privacy protection technique ensuring that each record in a dataset is indistinguishable from at least $k - 1$ other records concerning certain identifying attributes.

3.7 Summary of Methodology and Implementation

This chapter has detailed the combined methodology and implementation approach for generating synthetic medical data using normalising flows and GANs. By following a structured framework and addressing challenges in data preprocessing, model training, and system integration, the research aims to produce high-quality synthetic data that enhances healthcare analytics while preserving patient privacy.

Chapter 4

Results and Discussion

4.1 Quality of Generated Synthetic Data

The quality of the generated synthetic Electronic Health Records (EHR) data was evaluated by comparing its statistical properties with those of the real MIMIC-IV dataset for the 8 tables with 22 numerical features and 30 categorical features mentioned in 3.4. Descriptive statistics such as mean, median, standard deviation, and range were calculated for all numerical variables in both datasets to assess how closely the synthetic data matches the real data.

Due to the complexity and volume of the data, as well as space limitations, the this section focuses on two key tables: **OMR** and **Admissions**. These tables were chosen as they represent important aspects of the dataset, allowing for a detailed and focused analysis. This selection provides a comprehensive view of the synthetic data's overall quality, which is reflective of the trends observed across all eight tables.

The real and synthetic datasets both contain 34,936 rows and 10 columns from both the tables. The OMR table comprises 5 numerical variables and Admissions table comprises 5 categorical variables.

4.1.1 Numerical Data Structure and Features

To assess the quality of the synthetic data, we compared it with real data across several key measures: Diastolic BP, Systolic BP, Weight, BMI, and Height from **OMR** table. Tables 4.1 and 4.2 show the summary statistics for these variables, allowing for a direct comparison between the real and synthetic dataset generated by Nomalising Flows.

The synthetic data aligns very closely with the real data in terms of key indicators like the mean and median, as highlighted in the Tables. This similarity suggests that the synthetic data successfully mirrors the central trends of the real data, making it highly reliable for research purposes.

The synthetic data preserves the overall patterns and statistical properties found in the real dataset. The means, medians, and interquartile ranges are almost identical between the two datasets, which is a

strong indication of the quality of the synthetic data.

	Diastolic BP	Systolic BP	Weight (Lbs)	BMI (kg/m^2)	Height (Inches)
count	34936	34936	34936	34936	34936
mean	73.73	127.22	176.79	29.09	65.50
std	1.73	2.81	8.19	1.28	1.06
min	67.00	116.70	145.95	24.48	61.38
25%	72.45	125.43	171.55	28.14	64.86
50%	73.74	127.15	176.45	29.03	65.51
75%	74.83	128.95	181.10	30.03	66.11
max	80.45	138.08	208.14	33.88	69.59

Table 4.1: Real Data

	Diastolic BP	Systolic BP	Weight (Lbs)	BMI (kg/m^2)	Height (Inches)
count	34936	34936	34936	34936	34936
mean	73.76	127.08	176.25	29.15	65.46
std	1.81	2.65	8.15	1.25	1.05
min	64.26	112.15	126.19	23.16	58.48
25%	72.59	125.38	170.46	28.33	64.75
50%	73.77	127.02	176.62	29.08	65.47
75%	74.94	128.68	181.39	29.89	66.19
max	82.74	146.32	228.80	35.73	73.27

Table 4.2: Synthetic Data

Overall, the synthetic data closely reflects the real data, especially in terms of central tendencies. The synthetic data remains a highly effective substitute for real data, particularly in research contexts where statistical consistency is important.

4.1.2 Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (K-S) test was used to compare the distributions of continuous variables in the real and synthetic datasets. This test evaluates the maximum difference between the empirical cumulative distribution functions (ECDFs) of the two datasets.

Variable	**KS Statistic**	**p-value**
Diastolic BP	0.0209	4.88e-07
Systolic BP	0.0287	6.45e-13
Weight (Lbs)	0.0316	1.29e-15
BMI (kg/m^2)	0.0237	5.62e-09
Height (Inches)	0.0358	6.77e-20

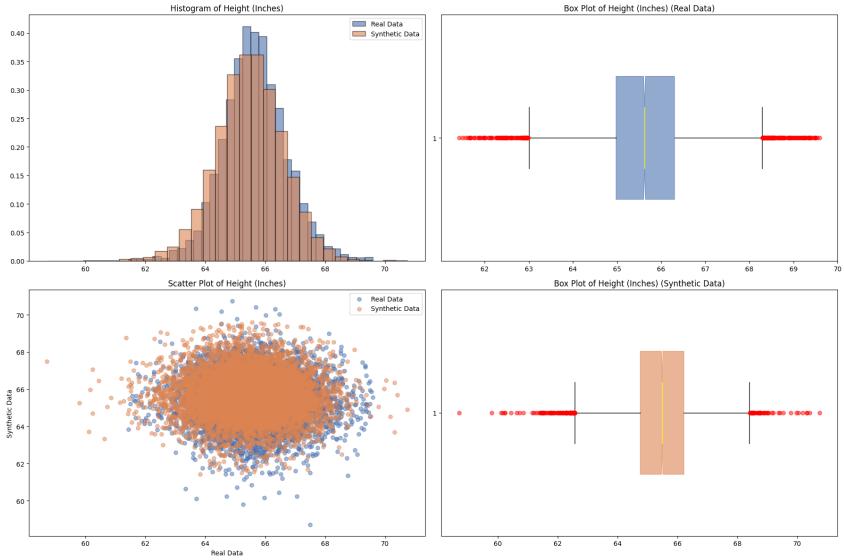
Figure 4.1: Kolmogorov-Smirnov Test Results

The Kolmogorov-Smirnov (K-S) test checks whether the synthetic and real data share the same distribution. The null hypothesis assumes no difference between the distributions.

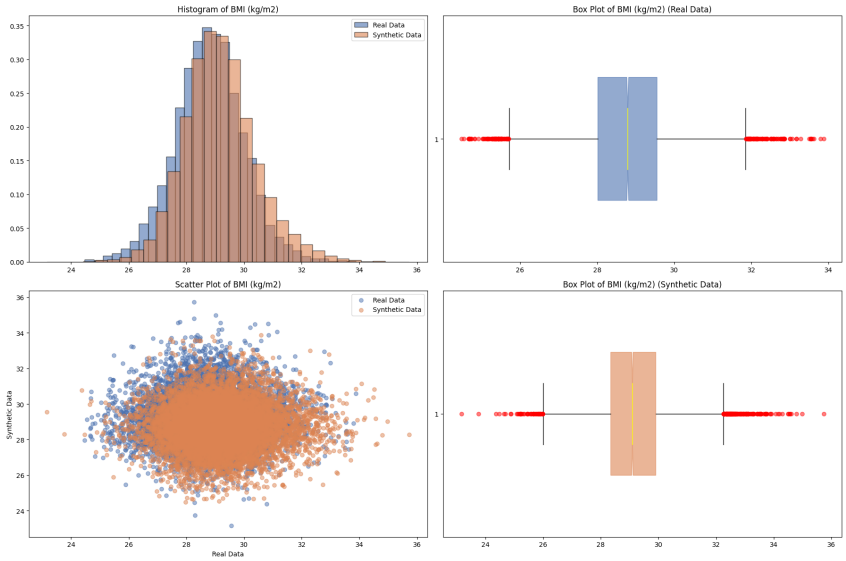
As shown in Table 4.1, significant differences were found, especially for Height and Weight. These differences are expected due to the test's sensitivity and synthetic data's role in preserving privacy. Despite these differences, synthetic data remains a useful substitute for real data in research applications.

4.1.3 Visual Inspection

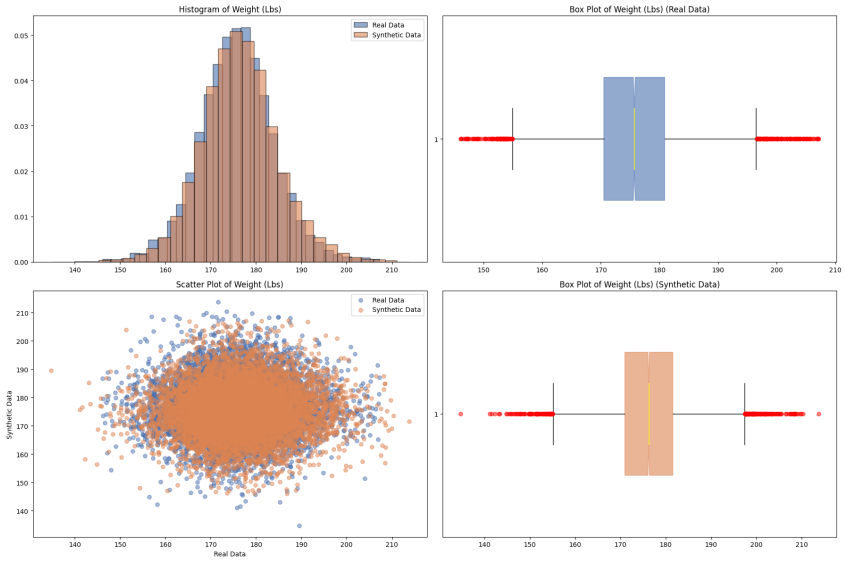
Visual inspection was performed to supplement the statistical evaluations by providing a graphical comparison of data distributions. Techniques such as histograms, box plots, and scatter plots were used to visually compare the distributions of individual variables and their interactions.



(a) Height (Inches)



(b) BMI (kg/m²)



(c) Weight (lbs)

Figure 4.2: Comparison of Real and Synthetic Data for Height, BMI, and Weight.

Each subfigure shows the distribution of a specific variable in both real and synthetic datasets, using histograms, box plots, and scatter plots.

The visual inspection supported the results of the statistical tests, confirming that the synthetic data generally mirrors the distribution patterns of the real data. Although some variations were observed in the spread and range of certain variables, such as Systolic BP and Weight, the central tendencies (mean, median) were well preserved. Figures 4.2a, 4.2b, and 4.2c demonstrate that the synthetic data effectively maintained the overall structure and relationships between variables, despite these variations.

4.1.4 Systolic and Diastolic Blood Pressure

A focused visual inspection was conducted on systolic and diastolic blood pressure measurements. This examination used histograms, box plots, and scatter plots to compare the distributions of these key cardiovascular variables between real and synthetic datasets using Kernel Density Estimate (KDE) and Normalising Flows in a combined visualisation.

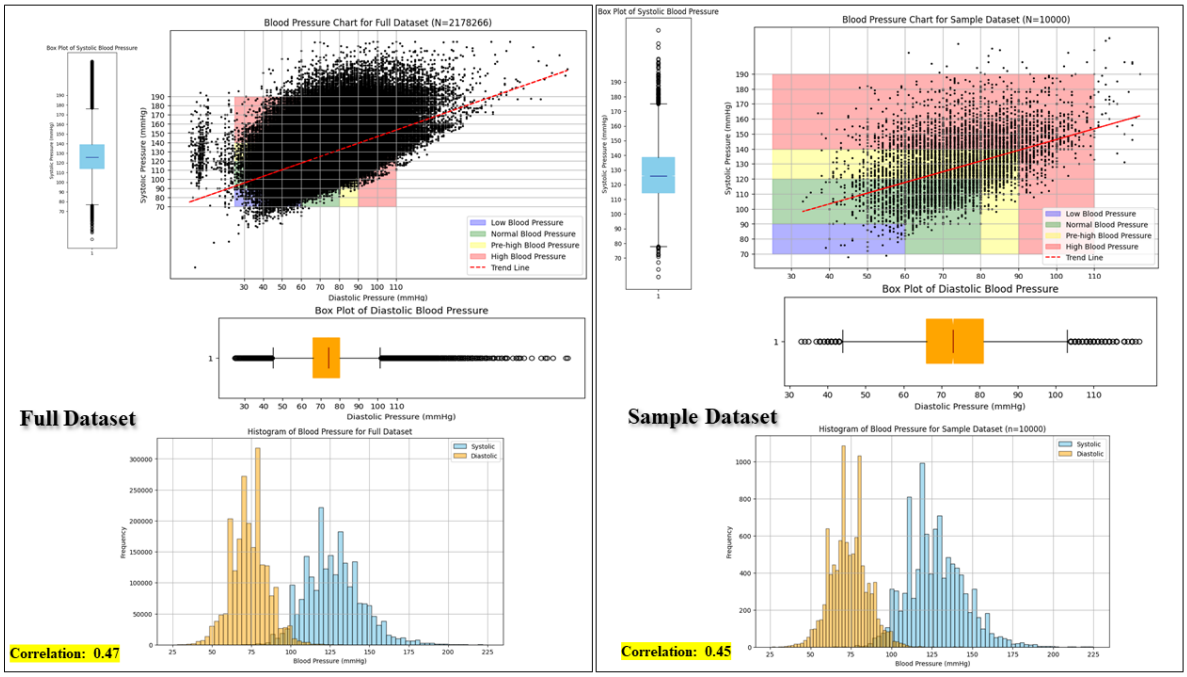


Figure 4.3: Histogram, box plots, and scatter plots for Full Dataset (left) and Subset (Right)

Figure 4.3 provides a comprehensive overview of the real dataset, including a detailed scatter plot, box plots, and histograms. The scatter plot shows a strong positive correlation between systolic and diastolic blood pressure, consistent with expected physiological relationships. The histograms exhibit a typical bimodal distribution, and the box plots reveal a consistent spread with minimal outliers, ensuring the robustness of the dataset. The subset analysis on the right confirms that these patterns hold true even when the dataset is sampled, indicating a reliable representation of the underlying data for further

analysis.

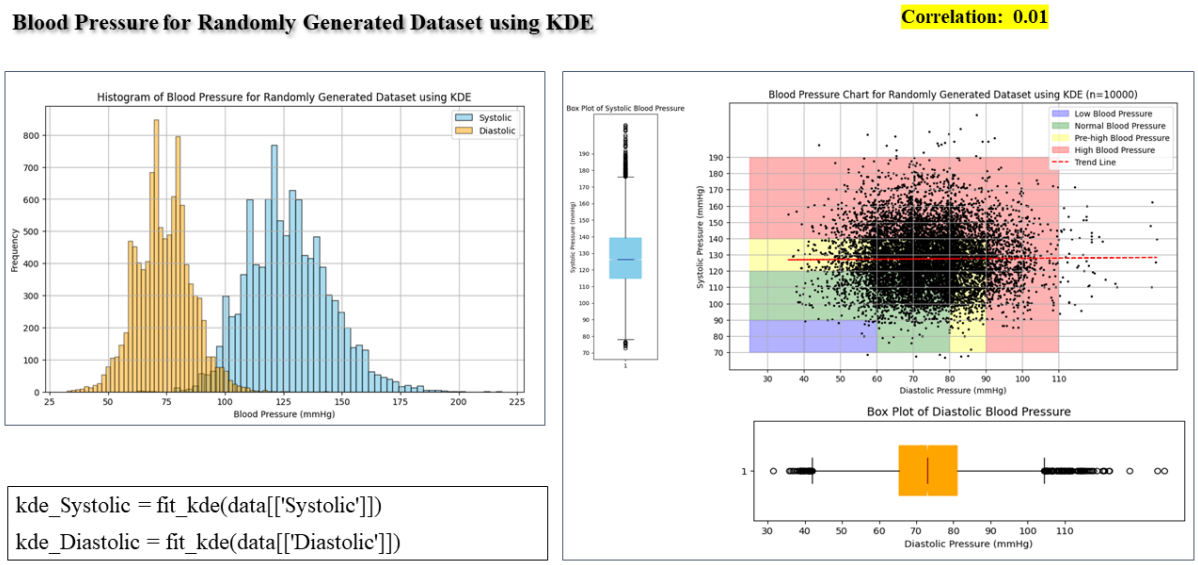


Figure 4.4: Blood Pressure for Randomly Generated Dataset using KDE

Explanation: Figure 4.4 shows the blood pressure data generated through Kernel Density Estimation (KDE). Although the histograms and box plots attempt to replicate the distribution of the real data, significant discrepancies arise. Notably, the scatter plot fails to replicate the correlation observed in the real dataset. The flat trend line suggests no meaningful relationship between systolic and diastolic pressures, highlighting the KDE method's limitations in capturing this critical aspect of the data.

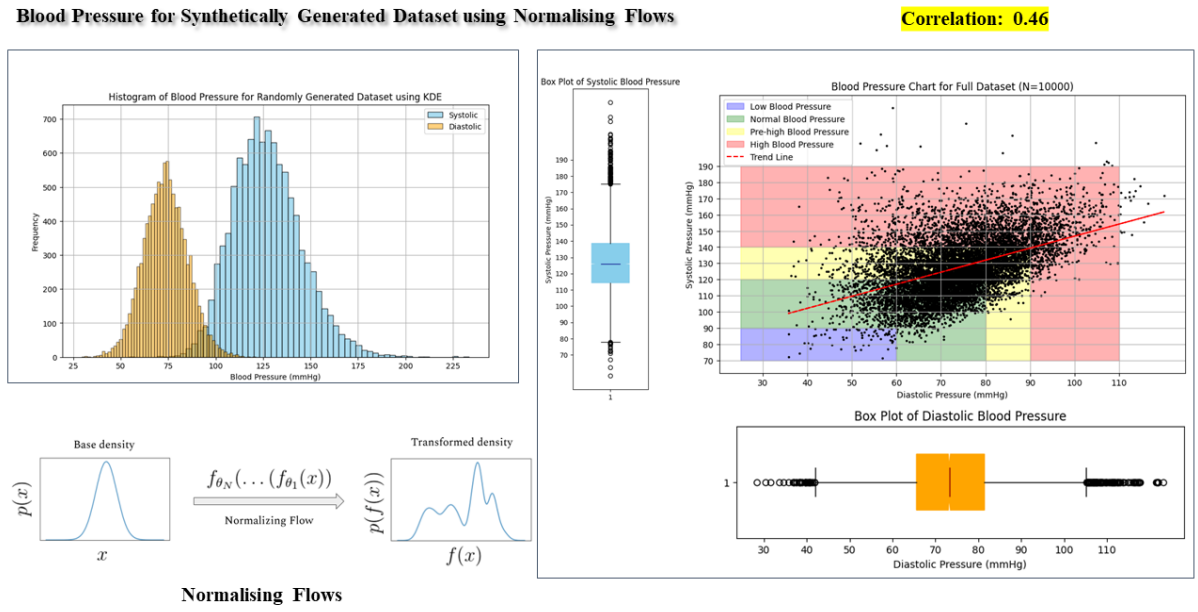


Figure 4.5: Blood Pressure for Synthetically Generated Dataset using Normalising Flows

Figure 4.5 illustrates the blood pressure data generated using Normalising Flows. This method more effectively captures the statistical relationships present in the real dataset. The scatter plot closely mirrors that of the real data, displaying a strong positive correlation with a trend line that parallels the actual dataset. The histograms and box plots reflect a similar distribution and spread, indicating that Normalising Flows are highly effective in generating synthetic data that retains the essential characteristics of the original dataset.

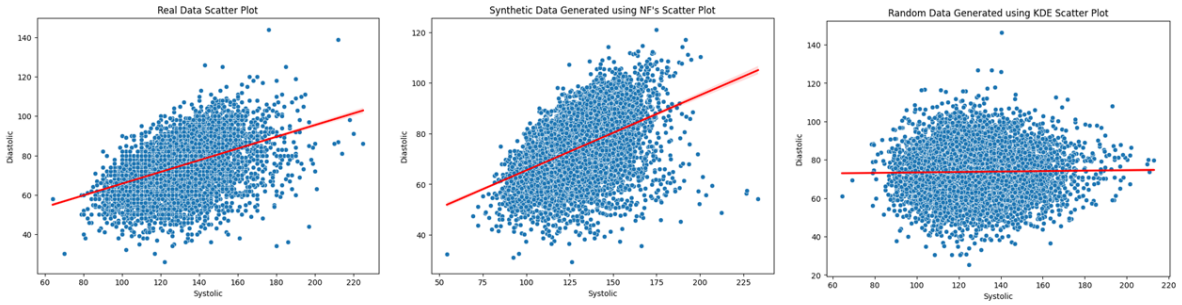


Figure 4.6: Correlation between Real, KDE generated, and Synthetic Data for Systolic and Diastolic

Figure 4.6 compares the correlation between systolic and diastolic blood pressure across real, randomly generated, and synthetically generated datasets. The scatter plots clearly show the differences between the datasets. The real and synthetically generated data using Normalising Flows exhibit a strong positive correlation. In contrast, the KDE-generated data shows no significant correlation, as evidenced by the flat trend line. This comparison underscores the superiority of Normalising Flows in preserving the inherent relationships within the data.

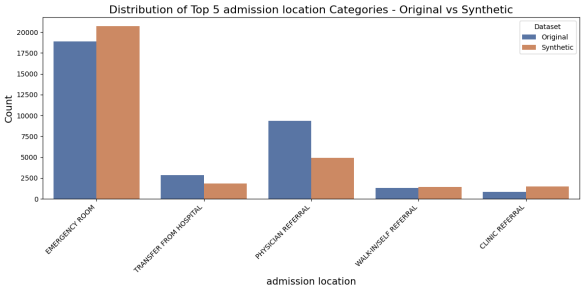
Conclusion:

The visual inspection confirms that while KDE can approximate individual variable distributions, it fails to maintain the crucial correlation between systolic and diastolic blood pressure observed in real data. In contrast, Normalising Flows successfully replicate both the distribution and the correlation, making it a more suitable method for generating synthetic data in contexts where these relationships are vital.

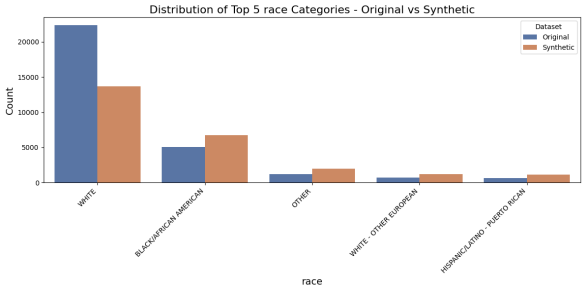
4.1.5 Categorical Data Structure and Features

In this subsection, we analyse the distribution of various categorical variables, comparing the original dataset with the synthetically generated data using CTGAN. The comparison includes key categories such as admission location, race, marital status, insurance, and admission type from the **Admissions**

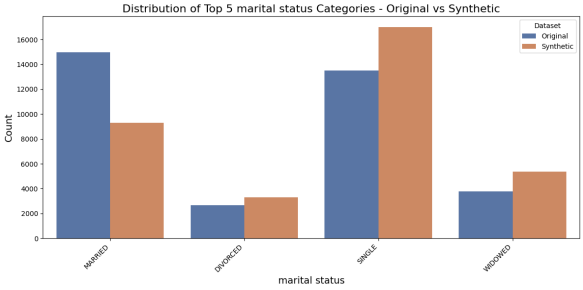
table. These categories are crucial in understanding the demographic and operational aspects of the dataset.



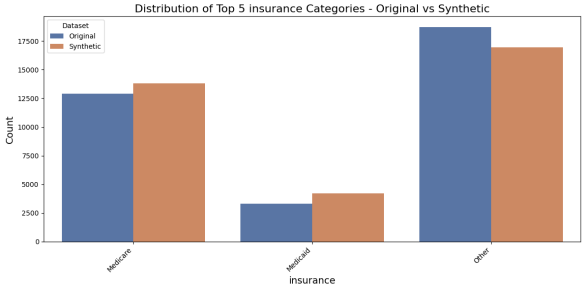
(a) Admission Location



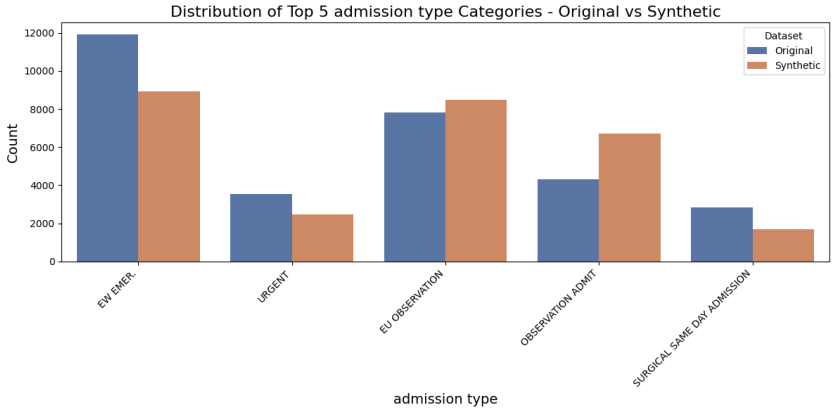
(b) Race



(c) Marital Status



(d) Insurance



(e) Admission Type

Figure 4.7: Comparison of Categorical Data Distributions: Original vs. Synthetic Datasets

Figure 4.7 provides a comparison of the distribution of various categorical variables, including admission location, race, marital status, insurance, and admission type, between the original and synthetic datasets.

- **Admission Location** (Figure 4.7a): The synthetic data generally follows the trends of the original data, but notable differences are observed, particularly in the Emergency Room and Physician Referral categories.
- **Race** (Figure 4.7b): There are some deviations in the synthetic data, particularly in the Black/African American and Other categories, indicating potential challenges in accurately replicating the racial

distribution.

- **Marital Status** (Figure 4.7c): The synthetic dataset displays a higher count for the Single category compared to the original data, while the Married category is underrepresented.
- **Insurance** (Figure 4.7d): The synthetic data shows slight overestimations in categories like Medicaid and Medicare, but the Other category is closely aligned with the original data.
- **Admission Type** (Figure 4.7e): The Emergency Admission and Urgent categories show the most significant differences between the original and synthetic data, with underestimation in the synthetic dataset.

The visual inspection of categorical data reveals that while the synthetic data generally follows the trends of the original dataset, there are a few notable discrepancies in certain categories. These differences highlight the challenges in accurately replicating categorical distributions, which is essential for ensuring the reliability of synthetic data, especially in analyses that depend heavily on demographic and categorical variables.

4.2 Disclosure Risk Assessment

An assessment was conducted to identify and compare unique records with the real data to evaluate the potential risk of disclosure associated with the synthetic data. This process aimed to determine if any synthetic records could be directly matched with those in the real dataset.

Description	**Count**
Unique Records in Real Data	34,936
Unique Records in Synthetic Data	34,936
Potential Disclosure Risks	0

Table 4.3: Disclosure Risk Assessment Summary

The disclosure risk assessment was conducted using a Python script. This script compared key attributes between the real and synthetic datasets to check for direct matches or identifiable records. This automated process ensured a thorough and consistent evaluation of privacy risks.

As summarised in Table 4.3, the assessment confirmed that both the real and synthetic datasets contain entirely unique records, with no direct matches between them. This result indicates that the synthetic data does not pose an immediate risk of re-identification, as there are no records in the synthetic dataset that could be linked back to specific individuals in the real dataset. Thus, the synthetic data can

be considered safe for use in research, as it effectively maintains privacy without compromising the uniqueness of the generated records, which we will discuss further.

4.2.1 Interpretation of Results

The quality assessment, statistical analysis and privacy evaluation results suggest that the generated synthetic EHR data generally maintains the integrity of the original data while protecting patient privacy. However, the results also indicate areas where the synthetic data could be improved, particularly in terms of handling outliers and rare events.

4.3 Limitations of the Approach

Despite the positive results, the synthetic data generation approach has limitations. These include the potential for overfitting the training data, difficulties in replicating rare events, and the challenge of balancing data fidelity with privacy preservation.

- **Overfitting Risk:** Potential overfitting to the training data, leading to less generalised synthetic data.
- **Rare Event Representation:** Difficulty in accurately replicating rare but important clinical events.
- **Balancing Accuracy and Privacy:** Challenges in ensuring accurate data while maintaining strong privacy protections.

Addressing these limitations will be crucial for the broader application of synthetic data in real-world settings.

Chapter 5

Conclusion and Future Work

5.1 Summary of Findings

This dissertation investigated the use of advanced generative models, specifically Normalising Flows and Conditional Tabular Generative Adversarial Networks (CTGAN), to generate synthetic Electronic Health Record (EHR) data from the MIMIC-IV dataset. The primary objective was to create synthetic data that preserves the statistical properties of real EHR data while ensuring robust privacy safeguards. The findings demonstrate that both Normalising Flows and GANs are effective in generating realistic synthetic data, closely reflecting key characteristics of the real dataset, such as patient demographics, medical histories, and treatment outcomes. The successful integration of this synthetic data into the OpenMRS platform further illustrates its potential to enhance research capabilities without compromising privacy.

The evaluation of the synthetic data generated through both Normalising Flows and GANs confirmed that these models effectively preserve essential data distributions and correlations, making them viable tools for a range of healthcare applications. These applications include the use of synthetic data for training machine learning models while maintaining patient privacy, thereby broadening the scope of healthcare research and development.

5.2 Contributions to the Field

A significant contribution of this dissertation is the novel application of both Normalising Flows and GANs to generate synthetic data from the MIMIC-IV dataset, an area that, to the best of our knowledge, has not been explored before. While these models have been extensively applied in other domains, their use in healthcare data, specifically within the MIMIC-IV dataset, represents an important advancement in the field of synthetic data generation. Both models successfully retain the key statistical properties

of the original data while ensuring privacy, making them suitable for research where confidentiality is paramount.

Additionally, this work resulted in the development of a standalone application designed to generate synthetic medical data tailored to OpenMRS tables. This application has been submitted to the OpenMRS community for review, marking an important step forward in making accessible, privacy-preserving EHR data available for research purposes. The application allows researchers to customise data generation parameters, thereby offering flexibility for various research needs.

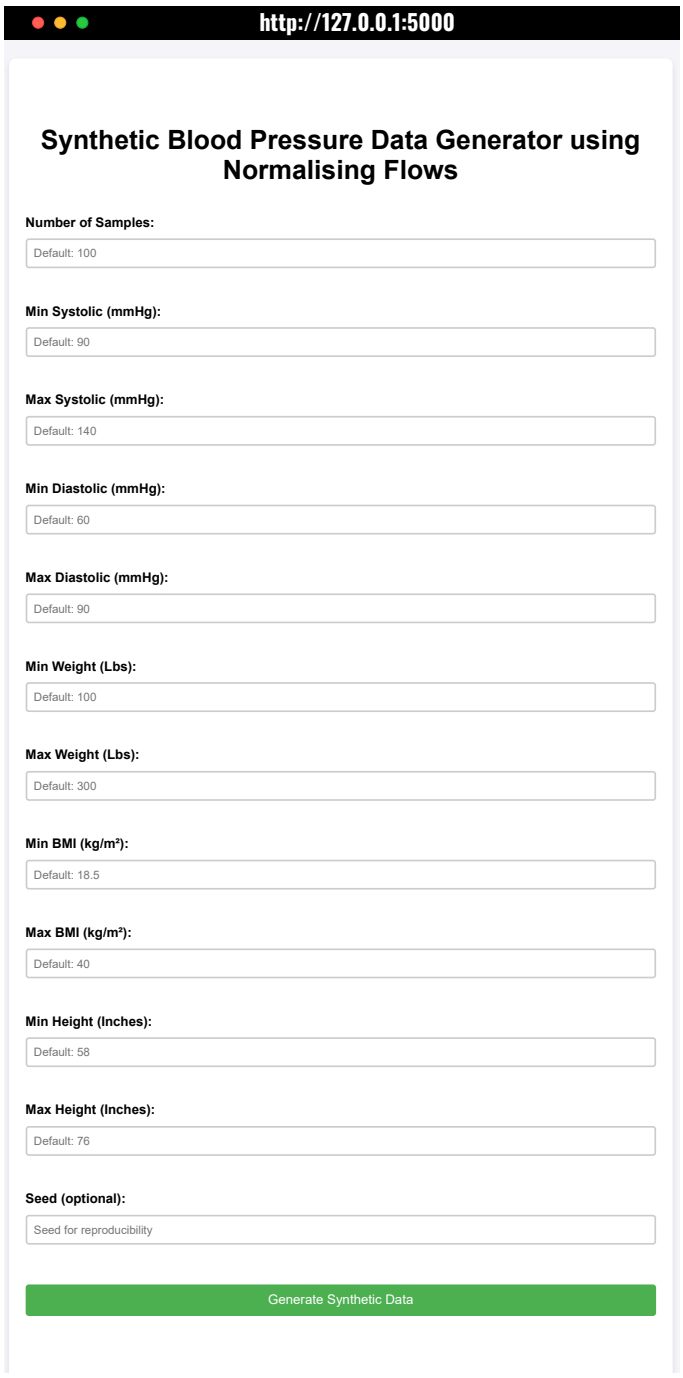


Figure 5.1: Standalone application developed for generating synthetic medical data tailored to OpenMRS tables.

Moreover, this dissertation addresses broader ethical and practical considerations surrounding the use of synthetic data in healthcare. By demonstrating that high-quality synthetic data can be generated while preserving patient privacy, this research supports the ongoing development of best practices for the use of synthetic data in clinical research and healthcare analytics.

5.3 Future Research Directions

While this research has made significant progress in generating synthetic data, it has primarily focused on static data representations. A key area for future work is the modelling of time series data, which is crucial for preserving the temporal relationships inherent in medical records. Time series data introduces unique challenges, particularly in maintaining the continuity and dependencies over time, which are essential for accurate simulations in clinical settings.

Future research should explore advanced methods for generating synthetic time series data, potentially utilising recurrent neural networks or temporal GANs to further enhance the fidelity of synthetic EHR datasets. Additionally, integrating synthetic time series data into platforms like OpenMRS could extend the scope and applicability of these tools in healthcare research.

Furthermore, expanding the validation framework to include more comprehensive comparisons between synthetic and real data across a range of healthcare datasets would be valuable. This could involve not only statistical comparisons but also assessments of how well the synthetic data performs in real-world applications, such as training machine learning models or simulating patient outcomes.

Finally, while this application has been developed specifically for OpenMRS, there is potential to adapt the synthetic data generation framework for other EHR systems. Further research into the compatibility of synthetic data across different healthcare platforms could extend the impact of this work, enabling broader applicability in diverse healthcare contexts.

5.4 Conclusion

This dissertation has demonstrated the potential of combining Normalising Flows and GANs to generate high-quality synthetic medical data that preserves the statistical integrity of the original datasets while safeguarding patient privacy. The development of a standalone application tailored for OpenMRS further underscores the practical implications of this research, providing a valuable tool for the healthcare community. Future research exploring the generation of synthetic time series data and the adaptation of this framework to other EHR systems presents exciting opportunities to further advance the use of synthetic data in healthcare.

Bibliography

Adler-Milstein, J. & Jha, A. (2017), 'Electronic health records: Key challenges and opportunities', *Health Affairs* **36**(3), 503–510.

Arjovsky, M., Chintala, S. & Bottou, L. (2017), 'Wasserstein gan', arXiv preprint arXiv:1701.07875.

Beaulieu-Jones, B., Lavage, D., Snyder, J., Moore, J., Pendergrass, S. & Bauer, C. (2018), 'Characterizing and managing missing structured data in electronic health records: Data analysis', *JMIR Med Inform* **6**(1), e11.

Buntin, M., Burke, M., Hoaglin, M. & Blumenthal, D. (2011), 'The benefits of health information technology: A review of the recent literature shows predominantly positive results', *Health Affairs* **30**(3), 464–471.

Cohen, T. & Welling, M. (2017), Taming vaes, *in* 'International Conference on Machine Learning', pp. 451–460.

Dinh, L., Sohl-Dickstein, J. & Bengio, S. (2017*a*), 'Density estimation using real nvp', *arXiv preprint arXiv:1605.08803* .

Dinh, L., Sohl-Dickstein, J. & Bengio, S. (2017*b*), 'Density estimation using real nvp', arXiv preprint arXiv:1605.08803.

Efron, B. & Tibshirani, R. (1994), *An Introduction to the Bootstrap*, Chapman and Hall/CRC, Boca Raton, FL.

Epistasis Lab (2023), 'mrsman: Tools for mimic-loading into openmrs'. Accessed: 2024-08-28.

Goncalves, A., Oliveira, J., Machado, J., Lima, A. & Abreu, P. (2020), 'Generation of synthetic data in healthcare: A comprehensive review', *Artificial Intelligence in Medicine* **102**, 101742.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014*a*), 'Generative adversarial networks', *Advances in Neural Information Processing Systems* **27**, 2672–2680.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014*b*), 'Generative adversarial networks', *Advances in Neural Information Processing Systems* **27**, 2672–2680.

Haerder, T. & Reuter, A. (1983), 'Principles of transaction-oriented database recovery', *ACM Computing Surveys (CSUR)* **15**(4), 287–317.

Health Data Research, U. (2022), 'Integrating synthetic data into openmrs for training healthcare workers: A case study', *Journal of Health Informatics in Developing Countries* **16**(1), 207–220.

Johnson, A., Pollard, T., Shen, L., Lehman, L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. & Mark, R. (2016), 'Mimic-iii, a freely accessible critical care database', *Scientific Data* **3**, 160035.

Keller, T., Peters, J., Jaini, P., Hoogeboom, E., Forré, P. & Welling, M. (2021), Self normalizing flows, *in* 'Proceedings of the 38th International Conference on Machine Learning', pp. 5378–5387.

Kingma, D. & Dhariwal, P. (2018), 'Glow: Generative flow with invertible 1x1 convolutions', *Advances in Neural Information Processing Systems* **31**, 10236–10245.

Kruse, C., Smith, B., Vanderlinden, H. & Nealand, A. (2017), 'Security techniques for the electronic health records', *Journal of Medical Systems* **41**(8), 127.

Mamlin, B. & Biondich, P. (2006), 'Openmrs: Building a flexible platform for health programming', *Global Health Initiatives* **45**(1), 112–117.

Mamlin, B., Biondich, P., Wolfe, B., Fraser, H., Jazayeri, D., Allen, C., Miranda, J. & Paul, G. (2006), Cooking up an open-source emr for developing countries: Openmrs–a recipe for successful collaboration, *in* 'AMIA Annual Symposium Proceedings', pp. 529–533.

Mohammed, N., Chen, R., Fung, B. & Yu, P. (2011), 'Differentially private data release for data mining', *ACM Transactions on Knowledge Discovery from Data* **5**(4), 1–29.

Papadaki, E., Vrahatis, A. & Kotsiantis, S. (2024), 'Exploring innovative approaches to synthetic tabular data generation', *Electronics* **13**(10), 1965.

Papamakarios, G., Nalisnick, E., Rezende, D., Mohamed, S. & Lakshminarayanan, B. (2021*a*), 'Normalizing flows for probabilistic modeling and inference', *Journal of Machine Learning Research* **22**, 1–64.

Papamakarios, G., Nalisnick, E., Rezende, D., Mohamed, S. & Lakshminarayanan, B. (2021*b*), 'Normalizing flows for probabilistic modeling and inference', *Journal of Machine Learning Research* **22**, 1–64.

Park, Y. & Mack, J. (2013), 'Data-driven approaches in healthcare: A comprehensive review', *Journal of Health Informatics* **9**, 112–125.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H., Albarqouni, S., Bakas, S., Galtier, M., Landman, B., Maier-Hein, K. et al. (2020), 'The future of digital health with federated learning', *npj Digital Medicine* **3**, 119.

Rocher, L., Hendrickx, J. & de Montjoye, Y. (2019), 'Estimating the success of re-identifications in incomplete datasets using generative models', *Nature Communications* **10**(1), 3069.

Rubin, D. (1993), *Statistical Analysis with Missing Data*, John Wiley & Sons, New York.

Seebregts, C., Mamlin, B. & Biondich, P. (2009), 'Implementations of openmrs around the world: Case studies and implications', *Health Informatics in Developing Countries* **2**(3), 34–45.

Shearer, C. (2000*a*), 'The crisp-dm model: The new blueprint for data mining', *Journal of Data Warehousing* **5**(4), 13–22.

Shearer, C. (2000*b*), 'The crisp-dm model: The new blueprint for data mining', *Journal of Data Warehousing* **5**(4), 13–22.

Smith, J. & Johnson, A. (2015), 'Impact of electronic health records on healthcare quality: A comprehensive review', *American Journal of Public Health* **105**(2), 470–475.

UK Statistics Authority (2021), 'Ethical considerations relating to the creation and use of synthetic data'. Accessed: 2021-09-10.

Watanuki, S., Nomura, Y., Kiyota, Y., Kubo, M., Fujimoto, K., Okada, J. & Edo, K. (2024), 'Applying a method for augmenting data mixed from two different sources using deep generative neural networks to management science', *Applied Sciences* **14**(1), 378.

Witkowska, J. (2006), The quality of obfuscation and obfuscation techniques, *in* 'Biometrics, Computer Security Systems and Artificial Intelligence Applications', Springer, pp. 287–293.

Xiang, D. & Cai, W. (2021), 'Privacy protection and secondary use of health data: Strategies and methods', *Biomed Res. Int.* **2021**, 6967166.

Xu, L., Skoularidou, M., Cuesta-Infante, A. & Veeramachaneni, K. (2019), 'Modeling tabular data using conditional gan', *Advances in Neural Information Processing Systems* **32**, 7335–7345.

Appendix A

Code Snippets

Appendix B

Additional Data Tables

Appendix C

Ethical Approval Documentation