Synthetic Data Generation Using Wasserstein Conditional Gans With Gradient Penalty (WCGANS-GP)

Manhar Singh Walia
Technological University Dublin

Follow this and additional works at: https://arrow.tudublin.ie/scschcomdis

Part of the Computer Engineering Commons

Recommended Citation

This Dissertation is brought to you for free and open access by the School of Computing at ARROW@TU Dublin. It has been accepted for inclusion in Dissertations by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@tudublin.ie, arrow.admin@tudublin.ie, brian.widdis@tudublin.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License
SYNTHETIC DATA GENERATION USING WASSERSTEIN CONDITIONAL GANS WITH GRADIENT PENALTY (WCGANS-GP)

Manhar Singh Walia
D18128811

A dissertation submitted in partial fulfilment of the requirements of Technological University Dublin for the degree of M.Sc. in Computer Science (Data Analytics)

September 2020
DECLARATION

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute’s guidelines for ethics in research.

Signed: Manhar Singh Walia

Date: 01/September/2020
ABSTRACT

With data protection requirements becoming stricter, the data privacy has become increasingly important and more crucial than ever. This has led to restrictions on the availability and dissemination of real-world datasets. Synthetic data offers a viable solution to overcome barriers of data access and sharing. Existing data generation methods require a great deal of user-defined rules, manual interactions and domain-specific knowledge. Moreover, they are not able to balance the trade-off between data-usability and privacy. Deep learning based methods like GANs have seen remarkable success in synthesizing images by automatically learning the complicated distributions and patterns of real data. But they often suffer from instability during the training process. WGAN-GP has been tested for image synthesis and shown to provide stable training and strong modelling performance. WCGAN-GP is an improved variation of WGAN-GP that allows targeted generation of samples specific to the class labels. However, they have not been used in the area of Tabular data generation. This study investigates whether WCGAN-GP can learn to generate tabular synthetic data that is indistinguishable from the real data and yet does not incur information leakage. The behaviour of WCGAN-GP is compared against baseline SMOTE using three real-world datasets from different domains. The synthetic data is evaluated for data utility and privacy risks. Visualisations and Machine learning effectiveness are used to show the utility and Euclidean distance for data privacy. The results showed that WCGAN-GP provides promising results for generating tabular data as the synthetic data showed preservation of patterns, distributions and relationships of the real data. It is observed that the synthetic data from WCGAN-GP outperforms the baseline SMOTE in terms of machine learning effectiveness and ensuring better data protection. But the results also implied there should be a better approach to further improve the quality of categorical synthetic values. This work is a contribution towards tabular data generation and it can be concluded that the proposed WCGAN-GP can serve as a data synthesizer that provides nearly similar data usability as of real data and better privacy preservation.

Key words: Synthetic Data Generation, Deep Learning, Generative Adversarial Networks, Wasserstein GANs, Wasserstein Conditional Generative Adversarial Networks with Gradient Penalty, Euclidean Distance, Tabular Data Generation
# LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAN</td>
<td>Generative Adversarial Networks</td>
</tr>
<tr>
<td>CGAN</td>
<td>Conditional Generative Adversarial Networks</td>
</tr>
<tr>
<td>WGAN</td>
<td>Wasserstein Generative Adversarial Networks</td>
</tr>
<tr>
<td>WCGAN</td>
<td>Wasserstein Conditional Generative Adversarial Networks</td>
</tr>
<tr>
<td>WCGAN-GP</td>
<td>Wasserstein Conditional Generative Adversarial Networks with Gradient Penalty</td>
</tr>
<tr>
<td>SMOTE</td>
<td>Synthetic Minority Over Sampling</td>
</tr>
<tr>
<td>VAE</td>
<td>Variational Autoencoder</td>
</tr>
<tr>
<td>CART</td>
<td>Classification and Regression Tree</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>AUC ROC</td>
<td>Area under the ROC Curve</td>
</tr>
<tr>
<td>TRTR</td>
<td>Train on Real data and Test on Real data</td>
</tr>
<tr>
<td>TSTR</td>
<td>Train on Synthetic data and Test on Real data</td>
</tr>
<tr>
<td>TSTR</td>
<td>Train on Synthetic data and Test on Real data</td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
</tr>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

First of all I would like to express my sincere thanks to my supervisor Brendan Tierney for his assistance, guidance and patience throughout this MSc dissertation.

I would also like to thank Dr. Susan McKeever for her consistent support and guidance, and Dr. Luca Longo (M.Sc. thesis coordinator) for providing me with the opportunity to work on this thesis.

Last but not least, I would like to thank my friends and family for their encouragement, support and motivation during my studies. Special thanks to my parents Tarpinder and Namita for their support and encouragement throughout my studies.
# TABLE OF CONTENTS

ABSTRACT ......................................................................................................................... II

LIST OF ACRONYMS ........................................................................................................... III

TABLE OF CONTENTS ....................................................................................................... V

TABLE OF FIGURES .......................................................................................................... VII

TABLE OF TABLES ............................................................................................................. IX

1. INTRODUCTION ............................................................................................................. 1
   1.1 BACKGROUND ............................................................................................................ 2
   1.2 RESEARCH PROBLEM ............................................................................................... 4
   1.3 RESEARCH OBJECTIVES ......................................................................................... 6
   1.4 RESEARCH METHODOLOGIES ............................................................................... 7
   1.5 SCOPE AND LIMITATIONS ..................................................................................... 8
   1.6 DOCUMENT OUTLINE ............................................................................................. 8

2. LITERATURE REVIEW .................................................................................................... 11
   2.1 HISTORY OF SYNTHETIC DATA ............................................................................ 12
   2.2 MACHINE LEARNING APPROACHES ....................................................................... 15
   2.3 DEEP LEARNING APPROACHES ............................................................................. 19
      2.3.1 Vanilla GAN ....................................................................................................... 20
      2.3.2 CGAN ................................................................................................................ 26
      2.3.3 WGAN/ WCGAN ............................................................................................... 27
      2.3.4 WGAN-GP/ WCGAN-GP .................................................................................. 28
      2.3.5 Other Variations of GANs ................................................................................... 31
   2.4 EVALUATION METHODS .......................................................................................... 32
   2.5 SUMMARY ................................................................................................................ 33
      2.5.1 Summary of Literature ....................................................................................... 33
      2.5.2 Limitations and Gaps of Literature .................................................................... 39
      2.5.3 The Research Question ...................................................................................... 41

3. DESIGN AND METHODOLOGY ..................................................................................... 42
   3.1 RESEARCH DESIGN OVERVIEW ............................................................................ 42
   3.2 DATASETS ................................................................................................................ 45
      3.2.1 Default of Credit Card Clients (Banking) .......................................................... 46
      3.2.2 Cardiovascular Disease (Healthcare) ................................................................. 47
      3.2.3 Adult Census (Census) ....................................................................................... 48
   3.3 DATA PREPARATION ............................................................................................... 48
      3.3.1 Data Cleaning .................................................................................................... 49
      3.3.2 Data Transformations ....................................................................................... 50
   3.4 EXPERIMENT DESIGN ............................................................................................. 51
      3.4.1 WCGAN-GP ....................................................................................................... 52
      3.4.2 SMOTE (Baseline) ............................................................................................ 59
   3.5 EVALUATION METHODS ......................................................................................... 61
      3.5.1 Data Utility Metrics ........................................................................................... 61
## Table of Figures

**Figure 2.1.** Hierarchy of Literature Review Section ........................................... 12  
**Figure 2.2.** Different Approaches for Synthetic Data Generation ................. 15  
**Figure 2.3.** The Structure of a Generative Adversarial Network (GAN),  
Based on Hitawala (2018) .................................................................................. 23  
**Figure 2.4.** A High-Level Illustration of How the Discriminator is Trained .. 24  
**Figure 2.5.** A High-Level Illustration of How the Generator is Trained...... 24  
**Figure 2.6.** Process Flow to Illustrate the Adversarial Training in GANs ... 25  
**Figure 2.7.** Illustration of How the Distributions of Real Data (Black),  
Discriminative Distribution (Blue) and Generator Distribution (Green)  
Evolves During the Training of GANs (Goodfellow et al., 2014) .......... 26  
**Figure 2.9.** The Structure of Wasserstein Conditional Generative  
Adversarial Networks with Gradient Penalty .................................................... 29  
**Figure 3.1.** Overview of Research Design .............................................................. 44  
**Figure 3.2.** Architecture of Critic in WGAN-GP .................................................. 54  
**Figure 3.3.** Architecture of Generator in WGAN-GP .......................... 54  
**Figure 3.4.** Synthetic Data Approach Using SMOTE (Baseline) .......... 60  
**Figure 3.5.** Evaluation Framework for Machine Learning Performance of  
Real and Synthetic Data ....................................................................................... 64  
**Figure 4.1.** Box-Plots for the Numerical Variables in Credit Card Dataset.  
Blue Indicates the Real Data Points, Orange Synthetic from SMOTE and  
Green Synthetic from WCGAN-GP ................................................................. 72  
**Figure 4.2.** Histogram for Categorical Variables in Credit Card Dataset.  
Blue Indicates the Real Data Points, Orange Synthetic from SMOTE and  
Green Synthetic from WCGAN-GP ................................................................. 74  
**Figure 4.3.** Histogram for Categorical Variables in Cardiovascular Dataset.  
Blue Indicates the Real Data Points, Orange Synthetic from SMOTE and  
Green Synthetic from WCGAN-GP ................................................................. 74  
**Figure 4.4.** Histogram for Categorical Variables in Adult Census Dataset.  
Blue Indicates the Real Data Points, Orange Synthetic from SMOTE and  
Green Synthetic from WCGAN-GP ................................................................. 75  
**Figure 4.5.** Scatterplot for the Variables in Credit Card Dataset. Blue  
Indicates the Fraud Data Points, Red Indicates Normal Records .......... 76
Figure 4.6. Scatterplot for the variables in Cardiovascular Dataset. Blue indicates the ‘Have CV disease’ patients, red indicates ‘Do not have CV disease’ patients.

Figure 4.7. Histogram for the categorical variables in Adult Census Dataset. Blue indicates records with income >50K, red indicates records with income <=50K.

Figure 4.8. Correlation Matrix Evaluated using the Real data and synthetically generated datasets on the Credit Card dataset

Figure 4.9. Correlation Matrix Evaluated using the Real data and synthetically generated datasets on the Cardiovascular Dataset

Figure 4.10. Correlation Matrix Evaluated using the Real data and synthetically generated datasets on the Adult Census Dataset
TABLE OF TABLES

Table 2.1. Summary of reviewed literature on state-of-the-art approaches for synthetic data generation ................................................................. 34
Table 3.1. List of all Python software packages used in experiments ............... 45
Table 3.2. Properties of datasets used in experiments ........................................ 46
Table 3.3. List of pre-processing activities required for data generation approaches ........................................................................................................ 49
Table 3.4. Hyperparameter settings for generator model in WCGAN-GP .......... 58
Table 3.5. Hyperparameter settings for critic model in WCGAN-GP ............... 58
Table 4.1. Summary of the implemented critic and generator model configurations for WCGAN-GP ................................................................. 70
Table 4.2. Pearson correlation coefficient and Spearman correlation coefficient between variables of real and synthetic datasets on the credit card dataset ......................................................................................................................... 79
Table 4.3. Performance comparison of different predictive models for real data, synthetic data from SMOTE, and from WCGAN-GP model on credit card dataset ........................................................................................................ 80
Table 4.4. Performance comparison of different predictive models for real data, synthetic data from SMOTE, and from WCGAN-GP model on cardiovascular dataset ................................................................................................. 81
Table 4.5. Performance comparison of different predictive models for real data, synthetic data from SMOTE, and from WCGAN-GP model on Adult Census dataset .................................................................................................... 81
Table 4.6. Euclidean distance to the nearest record, the value indicates the distances (mean, standard deviation) for all samples in synthetic dataset and their most similar sample in the real data ........................................ 82
1. INTRODUCTION

“Data is the new oil of the digital economy” (Toonders & Yonego, 2014). Real-world data is commonly used in the illustration and evaluation of novel technologies in areas such as software development, data analytics, machine learning or deep learning. The amount of data produced every day is growing exponentially, but the process of collecting and sharing sensitive data has become challenging with increased privacy and security requirements. Machine learning algorithms, for instance, require immense amount of data to learn from. But in recent times, issues like data accessibility, insufficient data and privacy constraints have been the key reasons why certain models cannot be developed. One of the innovative ways to overcome hurdles for data dissemination has been the generation and use of synthetic data. A synthetic data can open up a range of opportunities for the otherwise private data and can result in faster innovation in a shorter time frame with lower risks and costs.

Synthetic data generation has been deemed useful in numerous cases and applications. Real data could contain sensitive information and researchers or software developers can use the synthetic data, instead of the original, to reduce the privacy or confidentiality concerns. The synthetic data can also be beneficial in data augmentation to generate larger datasets and address situations of imbalanced dataset problems. Industries like healthcare or financial services can highly inherit from these benefits of synthetic data. Further, synthetic data finds its application in situations where the data for test environment does not exist or needs to be simulated in order to meet the specific needs or conditions not available in real data. The business areas that benefit from these applications are self-driving car, or clinical trials, where the actual data might not exist.

Over the past few years, methods of producing a quality synthetic data from real data have gained focus and popularity, due to their capability in preserving privacy and important statistical properties from the original data. The state-of-the-art or early approaches that involve generating the synthetic data use machine learning or statistical models. But they require an immense amount of human intervention, manual curation and domain-specific expertise. Generative Adversarial Networks (GANs) has
been an innovative area within deep learning that is known to generate synthetic but realistic data. They are built using an architecture of two Neural networks that compete against each other in an adversarial manner with an attempt to generate new samples. Nevertheless, GANs are known to be difficult to train and researchers have proposed extensions of GANs to overcome its typical instability and training problems. Since their inception in 2014, GANs have seen tremendous success in synthesizing realistic images but they have seen little application to tabular data generation. The resultant paucity in the current literature is something this research seeks to address.

The research intent is to answer the question of whether an improved GAN framework can generate a high-quality synthetic tabular data that holds similar statistical patterns and provides similar usability as the real data, along with minimizing the risks of re-identification.

1.1 Background

With stricter regulations around data management due to privacy and security requirements, the process of sharing data has become difficult. Traditionally, the real-world data was anonymized or de-identified to minimize the disclosure or privacy risks. But these privacy perturbation approaches have been linked to provide poor privacy protection and also have a lacklustre usability (Bellovin, Dutta, & Reitinger, 2019). To overcome this, data can be artificially generated using statistical components of real data but it should ensure no compromise in privacy.

According to the McGraw-Hill Dictionary of Scientific and Technical Terms, Synthetic data is defined as "any production data applicable to a given situation that are not obtained by direct measurement" (Parker, 1994). The generation of synthetic data has been traditionally linked to the process of data anonymization and the synthetic data is often considered as a subset of the anonymized data. Real data often contains personal and sensitive information that cannot be disclosed to the public, therefore, synthetic data is used to ensure that the confidentiality is maintained and no personal information can be traced back to any individual. Recently, synthetic data generation has received considerable focus, owing to its benefits of addressing the data
dissemination or access restrictions by preserving the multivariate relationships and statistical integrity of the real data.

The synthetic data is broadly classified into two categories – fully synthetic and partially synthetic. With fully synthetic data, the data is completely synthetic, that means it is generated from scratch and does not contain any original data. While with partially synthetic data, only the insensitive data is replaced with the synthetic information and the rest original data is still present, implying some re-identification is still possible. While generating any type of synthetic data, it is required to find a balance and address the trade-off between data-usability and disclosure risk. With partial synthetic data, the data can have high data-usability but with some disclosure risks. On the other hand, fully synthetic data minimizes the disclosure risks and has a potential to show high usability by capturing the important relationships and distributions from real data. Because of these reasons, the generation of fully synthetic data has gained immense popularity and it is considered as a next-step solution to the real-world data sharing problems.

Interestingly, there are innovations in the field of Machine learning and deep learning that generate fully synthetic data. The approaches involving machine learning methods have limitations in the functionalities that they provide and do not guarantee a good quality synthetic data. Within deep learning, the idea of Generative Adversarial Networks (GANs) was introduced in 2014 by Goodfellow et al. and since then, GANs have provided a promising direction to overcome the data-scarcity issue and also provide privacy properties.

GANs are composed of two components - a generator G, and a discriminator D. This architecture results in a generative network after an adversarial game in which the two Neural Networks, D and G, are simultaneously trained. The generative model G creates new data instances that mimic the distribution in original data, and the discriminative model D evaluates the authenticity or quality of the new data created by Generator and estimates the probability whether the sample is real or synthetic. GANs have achieved impressive results in applications related to images like image synthesis, semantic image editing, image super-resolution, classification and so on (Alqahtani, Kavakli-Thorne, & Kumar, 2019). But GANs have not been substantially used for
structured data like mixed-data with continuous and categorical variables. Additionally, traditional GANs have limitations with discrete data as they do not consider the class labels when generating synthetic data. CGANs were proposed to generate synthetic samples by conditioning the generator and discriminator on an extra information, that is the class labels and allow targeted generation of samples of a given type (Mirza & Ossid, 2014).

Traditional GANs also have a number of common training problems like vanishing gradients, mode-collapse and non-convergence. All of these problems have negative implications that results in poor training performance and limited diversity in new samples. WGAN-GP (Wasserstein GANs with Gradient Penalty) is a proposed variation of GANs that alleviates the instability training problems of traditional GANs (Gulrajani, Ahmed, Arjovsky, Dumoulin, & Courville, 2017). It has been shown to provide strong modeling performance and stable training on large-scale image and language datasets, but has not been tested on tabular datasets. Thus, a modified variant of WGAN-GP, that is WCGAN-GP (Wasserstein Conditional GANs with Gradient Penalty) is used in the research for synthetic tabular data generation and enhance the latest successes in the field of synthetic data generation.

This research mainly focusses on generating a synthetic data using an improved variant of GAN, that is WCGAN-GP to generate high-quality samples with high data-usability and disclosure protection.

1.2 Research Problem

The machine learning approaches that are considered as the state-of-the-art for synthetic data generation have not been able to reach the required performance, as they do not guarantee a high-quality synthetic data that has high data-usability and low disclosure risks. Some of these approaches also require user-defined specifications, manual interactions and domain-specific knowledge to generate the artificial data.

The traditional GANs have emerged as a powerful framework in generating synthetic data that mimics the real data. However, they are known to be harder to train because of training problems like vanishing gradients, mode collapse and non-convergence. All
these problems have negative implications on the performance of GANs and results in producing synthetic samples of poor quality. These problems have led to development of multiple extensions of GANs such as Conditional GANs (CGANs), Wasserstein GANs (WGANs) and Wasserstein Conditional GANs (WCGANs). These variants did make progress to provide a stable training process with GANs, but still suffered to generate poor samples or failed to converge. Gulrajani et al. (2017) proposed WGAN-GP to overcome the training instability of the existing generative models but this GAN has been tested only on image and textual datasets.

The challenges of data dissemination due to data disclosure and data access restrictions can be resolved only if the synthetic data learns the desired distributions and feature correlations and provides a high data-utility. The machine learning performance can be used to evaluate the utility of synthetic data by comparing the performance of predictive models built on synthetic against the real data. But a synthetic data release to the public or researchers can only occur as long as it minimizes data privacy risks and hence the synthetic data needs to be evaluated on privacy disclosure metrics.

Under this context, the main focus of this research is defined by the research question: “To what extent can a generated synthetic data approximate the quality of a real data using Wasserstein Conditional GANs with Gradient Penalty (WCGAN-GP) by using a combination of practical utility and privacy metrics?”

This main research question can be split into multiple sub-parts that will help in answering the bigger question and will be investigated in the research:

- **Research Sub-Question:** Can a WCGAN-GP model be trained to learn the distributions and relationships of the original data with a high degree of accuracy and generate synthetic data that is indistinguishable from real data?
  - **Its importance:** Measuring and analysing the capability of synthetic data to maintain most of the valuable information and statistical properties of real data will indicate high utility performance of the data generated using WCGAN-GP.

- **Research Sub-Question:** Is there a difference in the quality of synthetic data generated using WCGAN-GP based on the types of variables (numerical or categorical) in datasets?
Its importance: Measuring and analysing the quality of synthetic data for datasets with different number of categorical and numerical columns will verify the effectiveness of GANs on the most common data types.

- **Research Sub-Question:** To what extent does the synthetic data approach using WCGAN-GP performs better or worse as compared to a baseline approach using SMOTE?
  - **Its importance:** Comparing the performance of synthetic data generated using WCGAN-GP against baseline SMOTE in addressing the trade-off between data-utility and privacy will provide evidence on the capability of WCGAN-GP to be a better synthetic data generator than existing alternatives.

### 1.3 Research Objectives

The primary aim of this research is to address the issue of data accessibility by generating a tabular synthetic data using a modified GAN framework of WCGAN-GP. The purpose is to evaluate whether an innovative approach like GANs that requires minimal user-interaction can serve as a better data generation method in the future. Thus, real-world datasets from different domains are used to train the GAN model and generate synthetic data to provide a comprehensive proof-of-concept. The research involves comparing the performance of proposed WCGAN-GP with a simple baseline method to identify its advantages and disadvantages in generation of synthetic data.

In this regard, and in light of the existing literature explored in Chapter 2, the alternate hypothesis for this project is defined as:

**Alternate Hypothesis:** If the proposed framework of WCGAN-GP is applied to create synthetic samples from real samples, then it will show better results than the baseline SMOTE and a synthetic data of higher quality with better privacy protection will be produced.

To achieve the results, the below outlined research objectives are carried out:

1. Exploring the existing state-of-the-art approaches and previous work on GANs in generating synthetic data, providing a comprehensive analysis to pave way for further research and a better GAN framework for synthetic data generation.
2. Data processing will be done and includes feature scaling, imputing missing values, and categorical data encoding.

3. Designing of a set of experiments to test the hypotheses.

4. Build and train the generative model for WCGAN-GP on real-world datasets to generate synthetic samples with exact same size of real data.

5. Run the baseline approach of SMOTE to generate synthetic samples.

6. Compare and evaluate the quality of generated synthetic data in terms of statistical properties using visualisation likes box-plots, histograms, scatterplots, correlation.

7. Train and test the Machine learning performance (using metrics such as Accuracy, F1 score and AUC-ROC) of different machine learning classifiers on the real and synthetic datasets individually.

8. Compare the machine learning performance of real and synthetic datasets as obtained from step 7.

9. Assess the privacy risks of synthetic data by calculating the Euclidean distance of synthetic data records against the real dataset.

10. Perform hyperparameter tuning of WCGAN-GP model based on results in step 6, 8 and 9 to select the final model.

11. Analyse, evaluate, compare, and record the results of different evaluation metrics for all the synthetic datasets.

12. Evaluation of success or failure of experiments against the research aims and goals.

1.4 Research Methodologies

The research conducted in this study is secondary as it relies on datasets that have already been collected and are available at Kaggle\(^1\). As a part of this existing research, a literature review is carried out for the state-of-the-art GANs to get a comprehensive idea of the project.

The research work follows the quantitative (Epidemiological) methodology and it is empirical in nature. The research is quantitative as it is related with mathematical and

\(^1\) [https://www.kaggle.com/](https://www.kaggle.com/)
statistical analysis of data using objective measurements. The experiments are carried out to evaluate the performance of the proposed GAN against the baseline generator, which is used to verify the given hypothesis. Experiment results are then evaluated to check the quality of the synthetic data using different data utility and privacy metrics. The research uses an inductive reasoning as the research focusses on the use of observations to generalise a concept and accept or reject the hypotheses of whether the synthetic data generated using WCGAN-GP is of a good quality or not.

1.5 Scope and Limitations

The research focusses on generating a synthetic data from real data using WCGAN-GP and evaluating the synthetic data quality in terms of its usability and risk of re-identification. The scope of the experiment is to generate synthetic data that can replace the need of the real data and allow the public use of synthetic data without any privacy or confidentiality concerns. The research does not focus on the applications of synthetic data in data augmentation to address data imbalance or simulating a dataset where the real data does not exist at all.

Due to time-constraints, research will have limitations as the synthetic data will not be evaluated on all the ethical and legal requirements of confidentiality. The other limitation is that only the datasets with the most common data types, that is continuous and categorical, are considered in this project. Data generation related to complex data types such as dates, geospatial, and textual data are not explored in this research. Further, there are many different variants of GANs that have been developed and can be used to find the best possible method for synthetic data generation. Due to time-constraints, the research is only limited to the implementation of WCGAN-GP against a simple baseline generator to find the best method for synthetic data generation.

1.6 Document Outline

There are five chapters remaining in this report and the rest of this dissertation is outlined in the following manner:
Chapter 2 – Literature Review: This chapter provides a comprehensive coverage of the existing work in the field of synthetic data generation and its evaluation using multiple metrics. The state-of-the-art methods that encompasses machine learning and deep learning approaches are presented and critically analysed. The chapter is mainly focussed on discussing the application and related academic studies using GANs in the area of synthetic data generation. This is followed by providing a theoretical perspective and background on GANs to allow for a better understanding of this approach. As GANs are relatively novel and an active area of research, most of the work done till date is provided and critically analysed. The chapter concludes with a summary of all the state-of-the art methods and highlighting the gaps that are being addressed through this study. This review of the existing literature is carried out to get a detailed understanding of the problem being addressed through this research.

Chapter 3 – Design and Methodology: This chapter focusses on the different phases involved in the data generation process. It summarizes the project approach in terms of design plan and experiment that is conducted, in order to test the hypothesis and eliminate the gaps identified in Chapter 2. The datasets are firstly defined to get an understanding of the key characteristics and interesting nuances within the data. This is followed by Data Preparation that deals with all the pre-processing steps including data cleaning, standardization and other required transformations. Subsequent sections provide an in-depth explanation of the specific components of the experiments to be performed. Any relevant details related to GANs implementation including model architecture, training, hyperparameter settings are highlighted in this chapter. Overall, this chapter focuses on design aspects of the major components of the project.

Chapter 4 – Results, Evaluation and Discussion: The final implementation details and results of the experiment are presented and documented here. The quality of the synthetic data generated using the proposed WCGAN-GP is evaluated on a variety of metrics and compared to a simple baseline generator, that is SMOTE. A detailed analysis of the experimental results and the evaluation of synthetic data on different metrics will be conducted in this chapter. Based on the results and analysis of observations from the findings, a decision regarding the acceptance and rejection of the proposed hypothesis will be made. It will help to understand whether the research has produced sound results and if the experimentation has worked as intended. This
chapters concludes with a discussion of the strengths of the findings and their limitations.

**Chapter 5 – Conclusion:** This chapter covers the overall accomplishments or findings of the research and the weaknesses that could be expanded upon in the future work. It provides a conclusion and an assessment of the contribution of this research to the literature. Suggestions are also put forward for direction of future work.
2. LITERATURE REVIEW

The chapter provides a detailed review of the prevailing research in the domain of synthetic data generation and particularly in the application of GANs to generate synthetic data. In this section, the previous works on data generation methods using machine learning and deep learning techniques are reviewed in detail along with discussing their shortcomings. Apart from the state of the art methods of generating artificial data, there is a review of the evaluation or performance metrics to assess the quality of synthetic data.

Figure 2.1 provides an overview of the hierarchy of sections in this chapter. The chapter commences with a section on the history of synthetic data that provides a background of the domain under consideration in Section 2.1. The motivation and necessity of synthetic data in modern times is discussed to show the relevance of efforts put by researchers in the field of synthetic data generation.

Following this, different approaches for synthetic data generation are classified into two broader categories – Machine Learning and Deep Learning methods. In Section 2.2 and Section 2.3, these data generation approaches are presented and reviewed respectively.

Section 2.2 is focused on briefly reviewing the methods involving machine learning methods. This category of data generation uses statistical modelling or machine learning classifiers to generate data either by using the handcrafted distributions specified by the user or by actually learning the distributions from the real data.

The main focus of the research is the use of deep generative models, that is GANs to synthesize data. Section 2.3 reviews and discusses the prior studies on GANs in the chosen or related domain. It includes an assessment of various GANs that have been developed for synthetic data generation. A theoretical explanation and analysis about the working of GANs is also provided to build a better understanding of the approach. These preliminary details are necessary in building the core concepts within GANs and
aid in shaping up the experiment design and implementation of GANs in Section 3 and 4 respectively.

Apart from the previous state of the art methods, there is also a brief review of the evaluation methods that are used to assess the quality of synthetic data. The chapter concludes with a summary of the reviewed studies in the field of synthetic data generation and summarizing the limitations and gaps in the literature, leading to the research question.

Figure 2.1. Hierarchy of Literature Review Section

2.1 History of Synthetic Data

The demand for sharing data to support research and innovations is growing but most of the valuable data contains sensitive or personal information, therefore releasing the original data to public or researchers poses a risk to individual privacy or possibility of information leakage. The sharing of data with any third-party also leads to potential risks of misuse by adversaries. For this reason, the data is strictly protected and kept out of reach from the researchers. In the current era, companies, governments and global businesses primarily focus on data-driven decision-making to make informed choices and smarter decisions. Thus, the availability and access to the data is necessary to explore different applications.

Historically, the release of sensitive data has been achieved by anonymization of data or with the use of other de-identification methods. Such privacy preserving techniques are focussed on altering the values in the original data to reduce privacy leaks. The most common techniques for private data release are k-anonymity, l-diversity, t-
closeness, perturbation and differential privacy (Fung, Wang, Chen, & Yu, 2010). These methods deal with removal of sensitive features, perturbing the data by adding noise, or reducing the granularity of data by grouping variables into broader categories. While the perturbed datasets do limit the ability of attackers to identify individual-specific information, it has been observed that the attackers can still identify individual information if the global distributions of sensitive attributes are known. If the attackers have access to some background knowledge about the data or related information sources, it is possible to retrieve the personal information using these attributes that are not modified (Machanavajjhala, Kifer, Gehrke, & Venkitasubramaniam, 2007).

Moreover, this trade off to boost the privacy in data also results in the loss of machine learning performance of the modified data. Researchers have shown that the balance between privacy and data utility using these de-identification methods is not fulfilling and requires more research (Park et al., 2018). These privacy preserving methods produce data that still has a corresponding row in the original data, that gives rise to re-identification risks. And the basic trade-off between data utility and privacy — an inverse relationship — still remains.

Due to limitations of these disclosure control methods, synthetically generated data has offered a promising solution and is considered as a viable alternative to the standard anonymisation techniques. Synthetic data is almost a replica of the real data and allows to keep the data private but also maintain the usefulness of the data. Researchers have explored machine learning or deep learning approaches to generate synthetic data and minimize the problems of data sharing.

Synthetic data generation has been an active area of research and used across a number of different domains (Drechsler, 2011; Howe, Stoyanovich, Ping, Herman, & Gee, 2017). Apart from privacy protection, there are also other use cases of synthetic data. There are issues of quality, quantity, and imbalance that are also associated with real data. In certain situations, the data can be insufficient or imbalanced and this motivates the need to supplement the real data using the synthetic data. The quality of data can be improved by learning the distributions within the data and repairing the real data to minimize the impact of data with a poor quality. These problems have also motivated
the need for synthetic data generation and synthetic data can be highly useful when the original data is expensive, scarce or unavailable.

The necessity of Synthetic data can be broadly summarized into the following applications (Xu, 2020):

- **Product or System Testing**: In some situations, the data does not exist and a synthetic data can be simulated to test the systems. For instance, self-driving cars make use of a simulated synthetic data to test its performance.

- **Data Augmentation**: A high-quality synthetic data can be used to augment and overcome the imbalance in data by generating more training data. This can help to enhance the performance of predictive algorithms.

- **Data Disclosure**: As data privacy is a critical issue, the generation of synthetic data that preserves all the relationships of real data and avoids any disclosure of sensitive information can be used to overcome the barriers of data sharing.

The objective of this research is focussed on the ability of synthetic data to overcome data disclosure risks and replace the real data. As a result, the synthetic data will be able to act as a reasonable proxy of the real data and promote the area of useful data dissemination forward.

**Approaches for Synthetic Data Generation**

In the next section of the chapter, all the prior works in the area of synthetic data generation to protect the confidentiality of data are reviewed in detail. Synthetic data can be generated in two ways. Firstly, by using machine learning or statistical modelling to learn from distributions (either user-specified statistical distributions or directly from the real data). Secondly, by using deep learning to learn the complicated distributions from the data with minimal user inputs. Figure 2.2 outlines the relevant approaches in the area that are reviewed in this section. There are numerous approaches that have been proposed for synthetic data generation but only the ones that are commonly referenced are discussed.
Figure 2.2. Different Approaches for Synthetic Data Generation

Over the next two sections of the chapter, the existing works within the broad categories of data generation approaches are now presented and reviewed in detail.

2.2 Machine Learning Approaches

There are multiple approaches to generate a fully-synthetic data using machine learning methods that has encouraged innovations in the field of synthetic data generation. The approaches using machine learning are promising as they have a capability to produce synthetic data with the same distributions as the real data. With these approaches, the artificial data is randomly generated with constraints to retain relationships between attributes in the real data and hide sensitive information by replacing the original data.
The Machine learning methods can be roughly classified into two categories, process-driven and data-driven methods (Goncalves et al., 2020). The process-driven methods typically consist of creating synthetic data from some handcrafted statistical distributions. These methods do not involve the use of real data directly, so the issues of re-identification are eliminated. But these methods heavily rely on human interaction and domain-specific knowledge. On the other end, data-driven methods generate synthetic data by learning the distributions from the real data. As these methods do not rely on manual curation or domain expertise, these are more transferrable. The process driven methods are widely used but the data driven methods are an area of more recent study.

The process-driven techniques are motivated by the idea to generate synthetic data by controlling statistical data characteristics, such as mean, range, covariance and so on. Lin et al. (2006) proposed a semantic graph based method to generate synthetic data by a set of rules defined manually by the user. There are rules (Independent, Intra-record and Inter-record rules) based on which a graph structure is created and a synthetic data is generated once it satisfies all the pre-defined rules. The rules are defined in terms of distributions and range of values for each attributes. Another framework is proposed by Arasu, Kaushik, and Li (2011), in which the user specifies the database schema and cardinality constraints that the synthetic data should possess. Few researchers (Albuquerque, Lowe, & Magnor, 2011; Rivera, Dominguez, Murphy, & Thorpe, 2016) introduced similar approaches where the required dimension size and structure of data is specified by user using the probability density functions. A multivariate and fully synthetic data is then generated by sampling from these distributions.

Lu, Miklau and Gupta (2014) proposed a two phased data generation mechanism where a model is trained based on statistical queries and then the data is perturbed using a differential privacy framework to create a synthetic data that maintains privacy of individual. Like previous works in the field, this approach also suffers from user bias, and high computational costs for large datasets. Again, if there are modifications in the statistical queries required, the frameworks needs to run from the scratch to generate the synthetic data. Nettleton and Salas (2016) proposed a similar framework, except it suggested k-anonymity and t-closeness in the second phase, instead of differential privacy.
All these approaches of generating data in the context of machine learning with handcrafted distributions or user defined specifications are widely used. The proposed methods are suitable to model small datasets with fully known statistical characteristics. However, these approaches do suffer from drawbacks. Firstly, it requires large number of rules for datasets with small attributes. They struggle with datasets having large dimensionality as the processing time increases significantly with large datasets. These techniques require heavy human intervention as there is a need to know about the attributes in details before the rules are defined. The user also needs to be a domain expert to accurately specify the rules, constraints, ranges and values of the attributes. These methods do not automatically use real data and are also prone to human bias introduced by the users. Consequently, the quality of synthetic data heavily relies on the user-defined rules and there is a high possibility that the synthetic data might not completely represent the real data (Surendra & Mohan, 2017).

More recently, there are data-driven approaches used to generate synthetic data by learning the intrinsic patterns from the real data. Eno and Thompson (2008) recommended a method to generate synthetic data using a decision tree model. In this technique, a decision tree is built using the real data for the attributes considering the class label. Drechsler & Reiter (2011) evaluated four different non-parametric synthesizers based on machine learning algorithms - Classification and Regression Tree (CART), Random Forest, Bagging and Support Vector Machine (SVM). The classifier model is built and trained on the real data and then used for synthetic data generation. All of these approaches of creating a classifier model have limitations and demonstrate privacy issues if the classification accuracy is high. The risk of disclosure is higher with CART and SVM as compared to other approaches. Also, the process cannot be personalized per user requirements and is dependent on the class variable selected. If the user wants to choose another attribute as a class variable, the model requires to be re-trained from the scratch to generate synthetic data.

Another approach of data generation involves using Gaussian copula (Li, Xiong, Zhang, & Jiang, 2014). In this approach, the distributions of each attribute are computed by generating histograms of the variables. Following this, the distributions are transformed into a gaussian distribution and an encoding is created for original
values in the new format. The covariance matrix is calculated using gaussian copula function and a generative model is created by combining the covariance matrix with the histogram distributions. The synthetic data is generated by random sampling from this generative model. As the modelling is done for each column individually, this approach fails to capture relationships between the columns (Brenninkmeijer, Vries, Marchiori, & Hille, 2019). Th training using the approach becomes computationally expensive with datasets having high dimensionality. Also, the process has to undergo an expensive computation of covariance matrix for any updates in the original data, making it less attractive as a data generation approach (Surendra & Mohan, 2017).

Zhang, Cormode, Procopiuc, Srivastava, & Xiao (2017) proposed a differentially private method using Bayesian network to model relationships between the variables of real data and generate the synthetic data by sampling. Once the distributions of original data are learnt, noise is added using differential privacy mechanism to create a synthetic data for public data release. This approach provides a decent quality of synthetic data, but Bayesian networks struggle with datasets having large dimensionality as it then produces a large graph for a large data. With large datasets, this approach takes extremely long time to train, despite using a subset of data (Brenninkmeijer et al., 2019).

Synthetic Minority Over Sampling (SMOTE) is another data synthesis approach that creates new instances between the real samples (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). The synthetic samples are generated using the k-nearest neighbours approach. Firstly, SMOTE selects an instance at random and finds its k nearest neighbours. Then the difference between the sample selected and its nearest neighbour is computed. It is then multiplied with a random number between 0 and 1 and added to the selected instance to generate a new artificial data point.

As the SMOTE attempts to interpolate values between real data instances to generate synthetic samples, the synthetic data can be considered to be an approximation or representative of the real data. However, SMOTE suffers from few drawbacks as it does not consider that the neighbouring instances could belong to other target class and cause overlapping of classes. This can lead to introducing an unnecessary noise in the synthetic data. There are many variants of SMOTE that have been developed by
altering the method in which the samples and the neighbours are chosen (Fernandez, Garcia, Herrera, & Chawla, 2018).

SMOTE and its related variants are originally developed to augment the data and address the imbalance problems in datasets. But they have also found application in generating synthetic data with the exact size of real data (Kaloskampis, Pugh, Joshi, & Nolan, 2019). This is achieved by replicating the entire dataset multiple times to make it imbalanced, and then adding a target class. Then SMOTE is used to oversample the minority class data and generate a full synthetic data. SMOTE is used in experiments as it is faster to run and can generate good quality synthetic data.

2.3 Deep Learning Approaches

Deep learning is an another major category within data generation that has gained immense focus in recent times. They are based on data-driven methods for generating synthetic data. These methods provide innovative solutions to the challenges of machine learning methods by showing the ability to successfully learn the complicated distributions and patterns from the real data. This field of data generation using deep learning has given rise to an area of deep generative models, with the introduction of Variational Autoencoders (Kingma & Welling, 2013) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). The success of deep generative models in the field of natural language processing and computer vision has motivated the use of deep neural networks for synthetic data generation. Recently, GANs have become quite popular because of their capacity to generate synthetic data that approximates the complex and high-dimensional statistical distributions of real data. To add, deep generative methods are data-driven and do not require human intervention or inputs and offer promising solutions to the problem of synthetic data generation.

In the coming subsections, the studies involving GAN and its variants in the domain of synthetic data are reviewed. Also, the background knowledge about these variants is provided for better understanding of the approach.
2.3.1 Vanilla GAN

GANs, in particular, have shown a remarkable performance in generating new data from images, text, and time-series data (Goodfellow et al., 2014; Choi et al., 2017; Esteban, Hyland, & Rätsch, 2017; Hitawala, 2018; Fekri, Ghosh, & Grolinger, 2019). Studies have shown successful results with GANs in producing images, such as images of faces or rooms. However, GANs haven’t been tested enough on common types of data and have been an area of research for recent studies (Torres, 2018).

Yilmaz and Masum (2019) used GAN framework to generate synthetic samples for data augmentation and oversample minority classes. The work is not directly related to the scope of this research but it demonstrates the viability of GANs in generating quality synthetic data.

Tanaka and Aranha (2019) used GANs to create synthetic database from three numerical datasets. The results showed that it is viable to create synthetic data using GANs as the synthetic data was comparable to the real data. However, GANs were only tested on a numerical dataset in the research. Generally, the datasets in real-world are mixed datasets, containing both categorical and numerical columns. Also, two of the datasets chosen in the research had a small sample size (less than 800 instances). The research also did not focus on evaluating the synthetic data in terms whether the relationships (or correlations) between variables are preserved. Additionally, the synthetic data was not tested for any privacy metrics to ensure that the produced synthetic data did not disclose any information from real data.

Lu, Wang, & Yu (2019) also used GANs for synthetic data generation on multiple datasets and evaluated them on a variety of data utility and privacy metrics. However, three out of the four datasets used were not real-world datasets and were synthetically created for the research.

Even though GANs have shown great success in the field of image generation, their training is not easy and have been noted by researchers to be slow and unstable (Radford, Metz, & Chintala, 2015; Yoon, Drumright, & Van Der Schaar, 2020). Salimans et al. (2016) has also considered the challenges with GAN’s gradient-
descent-based training process. GANs involve the training of two models simultaneously in a two-player non-cooperative game and each model updates its gradients or costs independently without considering the other player. Also, sometimes it becomes hard to find a NASH equilibrium, where the training of both models does not guarantee a convergence and the loss begins to rise to result in an unstable training.

Further, Arjovsky and Bottou (2017) discussed the problems of vanishing gradients with GANs. With this training problem, the generator fails to create good fake samples. This arises from situations when the discriminator is not able to provide useful feedback for the generator to improve. This occurs in two situations. First, when the discriminator performs poorly and the generator does not receive a valid feedback. Alternatively, if the discriminator is perfect, the gradient of the loss functions falls to zero, resulting in a situation of no gradients to update the loss and slowing down the learning process.

Mode collapse (Arjovsky, Chintala, & Bottou, 2017) is another form of GAN failure, in which the generator collapses to always produce similar and less diverse samples. This happens when the generator produces plausible outputs to trick the discriminator and gets stuck to produce only those small sets of outputs over and over again. As a result, each iteration of generator over-optimizes for the discriminator and the discriminator gets stuck in a local minima. The discriminator is never able to come out the local minima or force generator to widen the variety of the synthetic samples generated.

Thus, GANs are known to be highly sensitive to hyperparameters and take a long time to converge. There have been suggestions to stabilize and overcome the training problems of GANs by incorporating a different loss function, that is Wasserstein loss instead of the traditional mini-max loss function (Arjovsky, Chintala, & Bottou, 2017).

**Preliminaries**

The idea of GANs was introduced by Goodfellow et al. in 2014. GANs (Generator Adversarial Networks) is a class of deep generative models used to generate data that has realistic characteristics, thus the term “Generative”. The framework of GANs is
based on two Neural Networks: a generator G, and a discriminator, D. The idea is that the two neural networks (Generator & Discriminator) are pitting against each other (thus, the term “adversarial”) with an attempt to learn the probability distribution. With the help of an adversarial game against each other, both the networks improve their performance and the generator learns to successfully generate synthetic samples which are not distinguishable. As a result, the discriminator is fooled and the synthetic samples are no longer distinguishable by the discriminator. The term “networks” in GANs comes from the fact that both Generator and Discriminator are made of vanilla neural networks or any type of related deep neural networks.

GANs are a clever way of training a generative model by considering the problem as a supervised learning with two sub-models, where the generator generate new samples and discriminator classifies the samples as real or fake. Figure 2.3 presents a brief overview of how GANs work. The role of discriminator and generator is listed below:

- A discriminator, D tries to evaluate the authenticity of generated data. It attempts to distinguish the samples whether they belong to the true distribution (that is taken from original dataset) or the model distribution (that is created by generator). The discrimination takes original data and the synthetic data from generator as inputs and predicts the probability of samples to be real or fake.
- A generator, G attempts to create realistic but fake data that is as close to original data. In most situations, the input data to a model is available for the task. However, the generator model works differently as it outputs new instances. The generator model receives an input of a fixed-length noise vector z that has a random distribution. The role of the random noise vector as an input is to offer diversity in the synthetic samples. Generator tries to analyse and learn the variations within the real data and generate new samples with a probability distribution that mimics the distribution in real data.

The concept of GANs is also visualized as a two-way game between counterfeiter and policeman. The generator is considered as a counterfeiter that generates fake money and discriminator as a policemen that attempts to detect the fake money. The idea is that with this constant competition, both the counterfeiter and policeman will improve but ultimately the counterfeiter learns to produce a fake money that is indistinguishable from the original ones. The concept can also be understood from the Figure 2.3 below.
Training of GANs: The training of GANs is carried out by alternating between the training of discriminator and generator. The generator captures the data distribution and discriminator estimates the probability of sample as real or fake. The training of neural networks is implemented by updating the weights, in order to reduce the loss function or errors. As the generator is connected to the discriminator, the purpose of training (both generator or discriminator) is to adjust the weights based on discriminator’s output. The whole training process is accomplished using the concept of backpropagation and updating of the weights to improve the performance.

Training the Discriminator: It is important to note that during the training of discriminator, the generator is not trained. The weights and biases of the generator model are then kept constant while it continues to create samples for discriminator to train on. The discriminator is related to two loss functions, the discriminator loss and generator loss. During training of discriminator, the discriminator only uses the discriminator loss and ignores the generator loss. The generator loss is only accounted during the generator model training. When the discriminator is trained, it attempts to classify original data as 1 and 0 otherwise and predicts a probability value. Based on the discriminator loss, the discriminator is penalized for misclassification of the fake instances as original or original instances as fake. In this manner of backpropagation,
discriminator’s weights are updated using the discriminator loss. Figure 2.4 shows how the training of discriminator is carried out.

**Figure 2.4. A High-Level Illustration of how the Discriminator is trained**

**Training the Generator:** During generator training, the generator learns to create fake samples by implementing the feedback from discriminator. Based on the discriminator output, the generator loss is calculated to obtain gradients and penalize the generator for failing to confuse the discriminator (although the discriminator does not update its weights during generator training). Using the gradients, the weights of generator are updated. Figure 2.5 shows how the training of generator is carried out.

The objective of generator is to generate fake samples that the discriminator classifies as original and predicts a probability of 0.5 as it cannot differentiate between fake or original samples. During generator training, the discriminator training is stopped and hence, its weights remain fixed. The reason for this is to avoid the discriminator model from becoming extremely strong to be beaten by the generator. If the discriminator training is kept on during generator training, the generator will never converge or be able to learn the distributions because then the discriminator will continue to improve and become an expert in identifying the fake samples.

**Figure 2.5. A High-Level Illustration of how the Generator is trained**
The adversarial training of GANs is achieved by firstly training the discriminator for at least one epoch, followed by training of generator for at least one epoch. The process of training discriminator and then the generator is repeated until the pre-defined epochs are completed. The discriminator is trained first because one would initially need a classifier that can distinguish between original and synthetic data, even with an untrained random generator output. If the discriminator cannot distinguish the simpler classification task, the GAN training cannot start and succeed as planned. The flow can be observed in the Figure 2.6 below.

A generator is termed to be trained perfectly once the discriminator predicts a 50% accuracy, that is probability of flipping a coin to make prediction. However, this process is way more complicated than it sounds and often results in convergence problems. If the GANs are over-trained, the discriminator has a tendency to provide random feedback to generator and generator gets trained on an inaccurate feedback and produces data with low quality.

The graphs in Figure 2.7 summarize how the distributions of real data (black), discriminative distribution (blue) and generator distribution (green) evolve during the training of GANs. The process is also explained below for a better understanding:

a. Before GAN Training: The distributions before training of GANs represent an untrained model. The discriminator distribution is in a chaotic form and there is a big difference in the generated synthetic data and real data distribution.
b. After Discriminator Training: As the discriminator training progresses, it learns to distinguish between original and synthetic data distribution

c. After Generator Training: When the generator is trained while keeping discriminator constant, the generator distributions reaches closer to the original data distribution

d. Equilibrium after Adversarial Training: As the process of b. and c. is repeated multiple times, the synthetic data distribution approximates the original data distribution and the discriminator distribution denotes uniformity.

![Diagram showing distribution evolution](image)

*Figure 2.7. Illustration of how the distributions of real data (black), discriminative distribution (blue) and generator distribution (green) evolve during the training of GANs (Goodfellow et al., 2014)*

### 2.3.2 CGAN

Researchers have highlighted the drawbacks of GANs when generating data in a classification problem (Sagong, Shin, Yeo, Park, & Ko, 2019). This has led to the development of Conditional GANs (CGANs), in which an extra condition of class labels is taken into account by both discriminator and generator (Mirza & Osindero, 2014). Rezaei, Yang, and Meinel (2018) implemented an architecture using conditional GAN to handle the issues of imbalanced data in the domain of magnetic resonance images (MRI) and achieved promising results.

Vega-Márquez et al., (2019) used a conditional adversarial neural network (CGAN) on a numerical dataset to generate a new synthetic data that can replace the real data. It was observed that the synthetic data performed similar on a classification task, when
compared with the real data. The trained model had similar Accuracy, F1 scores and AUC score (Area under ROC curve) for the model trained on real and on synthetic data. Additionally, the synthetic data did not have any correlation with the variables of real data, indicating that the privacy was preserved in the synthetic data. However, the scatterplots of the variables of synthetic data were completely different from the real data and showed signs of mode collapse with CGAN. The authors of the research tested CGANs only on a numerical dataset but not on mixed type datasets, hence, the approach cannot be generalized for its application on a mixed type dataset.

Preliminaries

The proposed framework of CGANs is almost similar to vanilla GANs. The only change is where the discriminator and generator are both conditioned on an extra information such as class labels. This is carried out by passing an additional input layer that contains the condition vector (class labels) to both generator and discriminator.

2.3.3 WGAN/ WCGAN

The limitations of vanilla GANs to suffer from training problems led to the development of Wasserstein GANs (WGAN) and Conditional Wasserstein GANs (WCGAN) (Cao, Liu, Long, & Wang, 2018). With WGANs, a meaningful distance measure is used as a loss function, namely the Wasserstein loss to alleviate the training failures of GANs (Arjovsky et al., 2017). Recently, Ba (2019) in his work used four different GAN frameworks (GAN, CGAN, WGAN, WCGAN) for data augmentation in tabular datasets. Again, the research is not directly related to the objective of this research but demonstrates the ability of GANs in generating quality synthetic data.

Preliminaries

In WGANs, the discriminator is actually called as a critic instead of a discriminator. Rather than classifying samples as real or fake, the critic predicts values that are large for real and small for fake samples. The critic tries to maximize the output on real and minimize the output on synthetic data.
The Wasserstein loss measures the distance between the probability distributions of synthetic and real samples. WGAN offers improvement over vanilla GANs as the Wasserstein distance attempts to minimize the difference between the two probability distributions. Wasserstein distance is also known as Earth Mover’s distance (EM) as it is easily explained using an example of transferring piles of dirt. The EM distance is the cost of optimal transport of a pile from one place to another and is represented by the amount required to be moved multiplied by the distance to be travelled.

In addition, WGAN involves clipping of the weights of critic network using the 1-Lipschitz function (that is functions where the gradient norm has a constant upper bound of 1). The weights are clipped between [-1 to 1] or [-0.01 to 0.01]. This is mainly implemented to overcome the training instability of GANs that occurs when the critic (or discriminator) outputs explosive gradients. This means that when the discriminator tries to provide feedback to generator, it randomly wants generator to change its weights by large values. Ideally it is not desirable as discriminator asks generator to change its weights by a large factor. Following which, the discriminator also changes its own weights by a large factor and this goes back and forth until eventually they both collapse completely. As a result, none of the synthetic data looks like real samples and are of a poor quality. Therefore, the weights are clipped using the 1-Lipschitz function and the rate of change is bounded by this function. This metric results in faster convergence as the training provides reasonable gradients when the difference in distributions are high. The mathematical details and working of Lipschitz function is out of scope for this research and will not be discussed in details. For more details on the function, refer to Heinonen (2001).

2.3.4 WGAN-GP/ WCGAN-GP

WGAN-GP stands for Wasserstein GANs with Gradient Penalty and was proposed by Gulrajani et al (2017). The authors of WGANs had noted that weight clipping to enforce Lipschitz constraint was not an effective and the best solution (Arjovsky et al., 2017). While the Wasserstein loss function alleviates some training problems, but WGAN still suffered from unstable training, slower convergence after weight clipping (when the clipping range is large) and vanishing gradients (when the clipping range is small). Therefore, an alternative approach of gradient penalty is proposed to comply
with the 1-Lipschitz constraint instead of clipping the weights. WGAN-GP enforces a regularization term, that is the gradient penalty, to force the norm of gradients to be 1. As a result, the critic discriminator becomes more stable and less explosive. The gradient penalty is increased or decreased depending on how far the gradient is from being 1-Lipschitz.

WGAN-GP uses Wasserstein distance and Gradient Penalty that reduces the chances of mode collapse. The WGAN-GP model is expected to converge faster and smoother along with an improved performance over traditional GANs. WGAN-GP was evaluated on image and text generation tasks and the results showed superior performance of WGAN-GP over WGAN and few other GAN architectures. The proposed WGAN-GP showed no signs of mode collapse and demonstrated an improved performance. It has been shown to overcome the training problems with GANs and show strong modelling performance.

WGAN-GP can be easily extended to create a WCGAN-GP framework by passing an additional input layer that contains the condition vector (class labels) to both generator and discriminator. The difference between WCGAN-GP and WGAN-GP is the same as between CGAN and GAN. The structure of WCGAN-GP is shown in Figure 2.9.

Figure 2.8. The Structure of Wasserstein Conditional Generative Adversarial Networks with Gradient Penalty
Preliminaries

Once the generator and discriminator models are defined, the training of a WCGAN-GP model is carried out as proposed by Arjovsky et al. (2017). The objective of the training is to make the output of generator’s distribution as similar as possible to the distribution of original data. The theoretical background behind the training and working of WCGAN-GP remains similar to the basic fundamentals explained and discussed earlier for GANs or CGANs. But there are few noticeable changes that are discussed below.

1. **Loss function:** The GAN variants like WCGANs or WCGAN-GP use Wasserstein loss and vanilla GANs use Binary Cross Entropy Loss. For the generator in WCGAN-GP, it is only the Wasserstein loss used to update its weights. On the other hand, critic uses Wasserstein loss for original and generated samples, and the gradient penalty loss. With Wasserstein loss function, the decrease of critic loss implies a better generator.

However, this is not the case in vanilla GANs as the discriminator loss (that is Binary Cross Entropy) can continue to rise despite improvement in quality of synthetic samples. Further, the loss function in WCGAN-GP framework does not suffer from the typical training problems of traditional GANs. In vanilla GANs, the imbalanced generator and discriminator could fail to converge and cause training problems.

2. **Gradient Penalty:** Vanilla GANs do not use any gradient clipping, while WGANs based variants use gradient clipping for the critic model. On the other hand, WCGAN-GP replaces weight clipping with gradient penalty to penalizes the critic network when its gradient norm moves away from 1. This is included to comply with 1-Lipschitz functions (i.e. functions where the gradient norm has a constant upper bound of 1). The original WGAN paper enforced this by clipping weights to very small values [-0.01, 0.01]. However, this was shown to drastically reduce the network capacity. Hence, penalizing the gradient norm is more natural and suggested.

In Improved WGANs, the 1-Lipschitz constraint is enforced by adding a regularization term to the loss function that penalizes the network if the gradient norm moves away from 1. The gradient penalty increases or decreases depending on how far the gradient
is from being 1-Lipschitz. The clipping of weights in no longer required with WGAN-GP as the use of gradient penalty has been found to be more effective across a variety of architectures and datasets.

3. Activation Function for Critic’s output layer: The vanilla GANs have the discriminator as a sigmoid output to predict the probability that samples are real or fake. In the GAN variants like WCGANs or WCGAN-GP that use Wasserstein loss, the output is linear with no activation function. Instead of being restricted between 0 and 1, the discriminator attempts to make the distance between its output for real and fake samples as large as possible.

4. Updating Critic more than Generator for each Iteration: With WGANs or WGAN-GP related frameworks, the critic is trained more times than the generator. A new hyperparameter is defined to control this setting and is proposed as 5 by the authors of WGAN and WGAN-GP (Arjovsky et al., 2017; Gulrajani et al., 2017). This in different from vanilla GANs training, where both the discriminator and generator are trained in equal amounts. The reason for more training of critic is to produce more reliable gradients of the Wasserstein metric and train the critic model till optimality.

5. Class Labels for Real and Fake Samples: Before the training begins for critic, the class labels are set for real and fake samples. -1 is set for real data and 1 for fake data (instead of 1 and 0 in vanilla GANs).

2.3.5 Other Variations of GANs

There have been many more variations of GANs that have been developed like MedGAN, VEEGAN, TableGAN and so on. MedGAN uses autoencoders in the architecture of GANs (Choi et al., 2017). However, the implementation of the paper has limitations as it was only tested for binary and numeric variables. Further, it is not designed to support the two different data types in the same model and requires building two separate models for each of the data types (Brenninkmeijer et al., 2019). TableGAN is proposed using convolutional neural networks to capture the correlations from the real data (Park et al., 2018). It is shown to work well for numerical datasets but suffers from mode collapse with categorical variables (Brenninkmeijer et al., 2019;
Xu, 2020). VEEGAN is another GAN variant that has been used for tabular data generation. This GAN variant is not originally proposed for generating tabular data but is designed to handle issues of mode collapse. However, VEEGAN also suffers from mode collapse when used for tabular mixed-type datasets (Xu, 2020). As these GAN variants are not the focus of this research, any further details or theoretical explanations on their working will not be discussed.

2.4 Evaluation Methods

With the use of synthetic data approaches, a synthetic data with the exact size of real data can be released to the public or researchers. Hence, it is critical to evaluate the quality of generated data for its practical utility and usability. The goal of a synthetic data generation is to generate data, which performs similar to what an original data does in various analytics tasks and maintain the privacy quotient at the same time. Researchers have employed methods to evaluate the data quality of the synthetic data. A common measure involves a qualitative comparison of scatter plots of variables within the real and synthetic dataset. This is done to understand the difference in distributions between the two datasets. Additionally, a correlation matrix is computed to check whether the attribute correlations in the real data remains the same in the generated dataset (Lu et al., 2019). Another important measure involves performing Pearson or Spearman correlation between real and generated data (Beaulieu-Jones et al., 2019). However, these metrics are not sufficient to evaluate the data utility of the synthetic data.

Machine Learning performance metric is the most crucial metric to evaluate the quality of synthetic data and involves comparing the performance metrics of a machine learning or predictive model built on synthetic and on real data. This property is also known as model compatibility. As the synthetic data can approximate the statistical properties and distributions of real data, it has a tendency to exhibit similar performance or results as compared to real data. This well-known approach of quality assessment involves comparing the performance of a machine learning model trained and tested on real (TRTR) and synthetic data (TSTST) (Heyburn et al., 2018; Jordon, Yoon, & van der Schaar, 2018; Lu et al., 2019). Esteban et al. (2017) proposed a “Train on Synthetic, Test on Real” (TSTR) approach for evaluating the synthetic data.
The idea is to use the synthetic data to train the predictive model and then test the model on the real data. The main aim of synthetic data is to generate data for use in Machine learning tasks, however, the approach of TSTR has limitations. If the quality of synthetic data is poor, the generated data will not capture the diversity of real data and hence, the results from the TSTR approach could be misleading.

In addition, measures are proposed to evaluate privacy using Euclidean distance (Giannella, Liu, & Kargupta, 2013). A distance based metric like Euclidean distance is used to determine the likelihood to identify personal information from synthetic data and a record with zero distance implies individual identification. Further, the number of identical records between real and synthetic data can be computed to understand if there are direct matches between real and synthetic data. Any matches between the two datasets poses risks for re-identification (Lu et al., 2019).

Sometimes, human evaluation is also sought after to evaluate the quality of generated data (Choi et al., 2017). The real and synthetic records are randomly shuffled and combined. The human experts are asked to evaluate and differentiate between the real and synthetic records. However, this metric creates a dependency and reliance on judgements from human experts. Also, the measure of synthetic data’s quality can be subjective and imprecise using this metric and thus quantitative metrics are preferred as they are more reliable and can quantify the quality of data.

2.5 Summary

2.5.1 Summary of Literature

The reviewed literature of the state-of-the-art approaches in the area of synthetic data generation has been summarized in Table 2.1 for easy comprehension of the related work. It is important to note that the approaches reviewed in the section are not exhaustive but only the latest and relevant developments in synthetic data generation are being presented.
<table>
<thead>
<tr>
<th>Author, (Year)</th>
<th>Synthetic Data Approach</th>
<th>Algorithm Used</th>
<th>Type of data used?</th>
<th>Evaluation Criteria</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin et al. (2006)</td>
<td>Machine Learning (Process-Driven)</td>
<td>Semantic Graph based</td>
<td>Tabular - both numerical and categorical (mixed-data type)</td>
<td>Does not state any evaluation criteria used but successful generation of a large dataset (million records) in hours is stated.</td>
<td>Prone to human bias as data built using user-defined rules. Requires domain-specific knowledge. Does not use real data to learn the distributions.</td>
</tr>
<tr>
<td>Arasu et al. (2011)</td>
<td>Declarative mechanism (using cardinality constraints)</td>
<td>Sampling from probability density function specified by user</td>
<td>Tabular Data. The variables type is not clearly specified but one of the datasets used 'Abalone' is Numerical.</td>
<td>Visual Evaluation and Cost estimated using running time</td>
<td>Does not use real data Requires manual curation and domain-specific knowledge.</td>
</tr>
<tr>
<td>Albuquerque et al. (2011)</td>
<td>Statistical queries specified by user and data perturbed using differential privacy framework</td>
<td></td>
<td></td>
<td>Visual Evaluation (Scatterplot, Orthographic projection)</td>
<td>Does not use real data. Prone to human bias introduced by the user. Processing time increases exponentially with data dimensionality.</td>
</tr>
</tbody>
</table>

Table 2.1: Summary of reviewed literature on state-of-the-art approaches for Synthetic data generation
<table>
<thead>
<tr>
<th>Author, (Year)</th>
<th>Synthetic Data Approach</th>
<th>Algorithm Used</th>
<th>Type of data used?</th>
<th>Evaluation Criteria</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rivera et al. (2016)</td>
<td>ML (Process-Driven)</td>
<td>COCOA (Rules and constraints of probability density functions defined by user)</td>
<td>Tabular - both numerical and categorical</td>
<td>PCA to assess dis(similarity), Generalized information loss (GenILoss), Execution time, average CPU, memory utilizations</td>
<td>Does not use real data. Prone to human bias. Computationally expensive for large dimensionality datasets.</td>
</tr>
<tr>
<td>Nettleton &amp; Salas (2016)</td>
<td>Statistical queries specified by user and data perturbed using k-anonymity and t-closeness</td>
<td>N.A. (Not specified)</td>
<td>N.A.</td>
<td>Privacy issues if classification accuracy of model is high. Disclosure risks or privacy preservation not considered or evaluated.</td>
<td></td>
</tr>
<tr>
<td>Eno &amp; Thompson (2008)</td>
<td>Decision Tree</td>
<td>Tabular - both numerical and categorical</td>
<td>Misclassification rate using Decision tree model, Comparing probability of samples at each node of decision tree.</td>
<td>SVM and CART provide higher utility, but risk of disclosure is also high. SVM is highly sensitive to tuning. Only tested on numerical dataset and not on categorical.</td>
<td></td>
</tr>
<tr>
<td>Drechsler &amp; Reiter (2011)</td>
<td>CART, Random Forest, Bagging, Support Vector Machines</td>
<td>Tabular - both numerical and categorical</td>
<td>Visual Evaluation (Scatterplots), Disclosure risk by calculating identical between real &amp; synthetic</td>
<td>Computationally expensive when scaled up to large datasets. Addition of noise using differential privacy reduces synthetic data utility.</td>
<td></td>
</tr>
<tr>
<td>Zhang et al. (2014)</td>
<td>PrivBayes (Differential Privacy with Bayesian Networks)</td>
<td>Tabular - both numerical and categorical</td>
<td>Machine Learning Performance (TRTR vs. TSTS)</td>
<td>Computationally expensive when scaled up to large datasets. Addition of new rows to original data requires re-computation of covariance matrix that is time-consuming.</td>
<td></td>
</tr>
<tr>
<td>Li et al. (2014)</td>
<td>DPSynthesizer (Differential Privacy with Gaussian copula function)</td>
<td>Tabular - both numerical and categorical</td>
<td>Visual Evaluation (Histograms)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author, (Year)</td>
<td>Synthetic Data Approach</td>
<td>Algorithm Used</td>
<td>Type of data used?</td>
<td>Evaluation Criteria</td>
<td>Limitations</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Heyburn et al. (2018)</td>
<td>ML (Data-Driven)</td>
<td>Synthpop package (R language)</td>
<td>Tabular - both numerical and categorical</td>
<td>Visual evaluation (distributions), Machine learning performance</td>
<td>Not Tested on Numerical dataset</td>
</tr>
<tr>
<td>Goodfellow et al. (2014)</td>
<td>Deep Learning (DL) (Data-Driven)</td>
<td>GAN (Invented in this paper)</td>
<td>Images</td>
<td>Visual Evaluation but not relevant to this research</td>
<td>Not related to the objective of structured data generation as tested on Images</td>
</tr>
<tr>
<td>Fekri et al. (2020)</td>
<td></td>
<td>GAN (Recurrent-GANs)</td>
<td>Time Series data (Energy Consumption)</td>
<td>Visual evaluation, Machine Learning Performance (TSTR, TRTR, TRTS, TSTS, MAPE, MAE), statistical measures - Kruskal-Wallis H &amp; Mann-Whitney U test</td>
<td>Not related to the objective of structured (tabular) data generation as tested on Time series data.</td>
</tr>
<tr>
<td>Yilmaz &amp; Masum (2019)</td>
<td></td>
<td>GAN</td>
<td>Tabular - both numerical and categorical</td>
<td>Visual evaluation (distributions), Machine learning performance but scope not directly related</td>
<td>Tested for data augmentation and not for generating new synthetic database</td>
</tr>
<tr>
<td>Tanaka and Aranha (2019)</td>
<td></td>
<td>GAN</td>
<td>Tabular - Numerical</td>
<td>Visual evaluation (distributions), Machine learning performance but scope not directly related</td>
<td>Tested for data augmentation and not for generating new synthetic database</td>
</tr>
<tr>
<td>Author, (Year)</td>
<td>Synthetic Data Approach</td>
<td>Algorithm Used</td>
<td>Type of data used?</td>
<td>Evaluation Criteria</td>
<td>Limitations</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Lu et al. (2019)</td>
<td>DL (Data-Driven)</td>
<td>GAN</td>
<td>Tabular - both numerical and categorical</td>
<td>Visual Evaluation (Correlation Matrix), Classification Accuracy, and some utility metrics used in PWSCUP Competition Hitting Rate, Record Linkage, Euclidean distance, and some privacy metrics used in PWSCUP competition.</td>
<td>Does not use real-world datasets. For generalizability and validity, the input datasets should be real-world data. But 3 out of 4 input datasets are synthetic, so the study cannot be generalizable.</td>
</tr>
<tr>
<td>Mirza &amp; Osindero (2014)</td>
<td></td>
<td>CGAN (Proposed by paper)</td>
<td>Images</td>
<td>Visual Evaluation but not relevant to this research</td>
<td>Not tested on a mixed type tabular dataset</td>
</tr>
<tr>
<td>Rezaci et al. (2018)</td>
<td></td>
<td>CGAN</td>
<td>Images</td>
<td>Metrics not relevant to this research</td>
<td>Not tested on a mixed type tabular dataset</td>
</tr>
<tr>
<td>Sagong et al. (2019)</td>
<td></td>
<td>CGAN</td>
<td>Images</td>
<td>Visual Evaluation but not relevant to this research</td>
<td>Not tested on a mixed type tabular dataset</td>
</tr>
<tr>
<td>Vega-Márquez et al. (2019)</td>
<td></td>
<td>CGAN</td>
<td>Tabular - Numerical</td>
<td>Correlation between real and synthetic data columns, Machine learning performance.</td>
<td>Only tested on numerical dataset and not on a mixed type dataset. Visual evaluation and Privacy metrics not evaluated</td>
</tr>
<tr>
<td>Arjovsky et al. (2017)</td>
<td></td>
<td>WGAN (Proposed by paper)</td>
<td>Images</td>
<td>Visual Evaluation but not relevant to this research</td>
<td>Not tested on a mixed type tabular dataset</td>
</tr>
<tr>
<td>Author, (Year)</td>
<td>Synthetic Data Approach</td>
<td>Algorithm Used</td>
<td>Type of data used?</td>
<td>Evaluation Criteria</td>
<td>Limitations</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Torres (2018)</td>
<td>Tabular - Mixed-type and textual</td>
<td>WGAN and WCGAN (with RNN), Inverse Transform Sampling (ITS)</td>
<td>Euclidean distance, Wasserstein distance, Perplexity (for textual data)</td>
<td>Machine learning performance is a crucial parameter for synthetic data evaluation and is not tested in the paper.</td>
<td></td>
</tr>
<tr>
<td>Ba (2019)</td>
<td>Tabular - Numerical</td>
<td>GAN, CGAN, WGAN, and WCGAN</td>
<td>Machine learning performance but paper's scope is not directly related to this research</td>
<td>Tested for data augmentation to enhance the fraudulent detection in credit card transactions &amp; not for generating a new synthetic database</td>
<td></td>
</tr>
<tr>
<td>Gulrajani et al. 2017</td>
<td>Images, language</td>
<td>WGAN-GP (Proposed)</td>
<td>Visual Evaluation but not relevant to this research</td>
<td>Not tested on a mixed type tabular dataset</td>
<td></td>
</tr>
<tr>
<td>Choi et al. (2017)</td>
<td>Tabular - Binary and Numerical</td>
<td>Med-GAN (using autoencoder and GANs), Random Noise, Independent Sampling, Stacked RBM (DBM), Variational Autoencoder</td>
<td>Visual Evaluation (scatterplots, distribution), Machine Learning performance (Logistic regression). Human Evaluation, Euclidean distance</td>
<td>Limited to test only for binary and integer values. Variables and variable like patient demographics or related ordinal data types not tested in the research. Not designed to support the two different data types in the same model and requires building two separate models for each of the data types.</td>
<td></td>
</tr>
<tr>
<td>Park et al. (2018)</td>
<td>Tabular - both numerical and categorical</td>
<td>ARX, sdcMicro, table-GAN</td>
<td>Statistical comparison (distributions), Machine learning performance for Data-utility, Euclidean distance for Privacy risks</td>
<td>More statistical evaluation (using box-plots, scatterplots, correlation matrices) for Data-utility and checking of duplicates between synthetic and real data for privacy can be assessed.</td>
<td></td>
</tr>
<tr>
<td>Xu (2020)</td>
<td>Tabular - both numerical and categorical</td>
<td>TVAE, CLBN, PrivBN (differential privacy and Bayesian Network), MedGAN, VEEGAN, TableGAN</td>
<td>Machine Learning efficacy (Classification and regression problem datasets), Nearest neighbour</td>
<td>TVAE and CTGAN both give better results than alternative on machine learning performance and privacy risks. MedGAN, VeeGAN, and TableGAN suffer from mode collapse on mixed-type datasets, suggesting they are sensitive to hyperparameter tuning and suffer from training problems if not fine-tuned rigorously. More statistical evaluation (using box-plots, scatterplots, correlation matrices) for Data-utility and checking of duplicates between synthetic and real data for privacy can be assessed.</td>
<td></td>
</tr>
</tbody>
</table>
2.5.2 Limitations and Gaps of Literature

Following such a review, current gaps in the literature are identified.

The use of privacy preserving methods to de-identify or anonymize has been studied for many years and these methods do help in reducing the likelihood of any privacy leaks. However, every row in the perturbed data still has a corresponding instance in the original data and the re-identification is still possible. Additionally, these approaches also adversely affect the usability of the data.

As it can be inerferenced from the reviewed literature, there are numerous approaches proposed in the area of synthetic data generation. The approaches to generate synthetic data using machine learning or statistical methods are widely used. Machine learning or statistical models are considered as the state-of-the-art approaches to generate synthetic data using process-driven (by hand-crafted distributions) or data-driven (by learning distributions from real data). The studies that use process-driven methods involve manual curation and efforts as the distributions within the synthetic data are manually specified by the user. Consequently, these methods require the user to have domain-specific knowledge and the synthetic data is prone to suffer from human biases. On the contrary, the data-driven methods produce synthetic data by learning from the real data and do not require any user-interaction or expert knowledge. Thus, they can be more readily deployed to new scenarios. However these approaches that use machine learning or statistical models also have limitations. The training of machine learning or statistical models becomes highly expensive with large datasets. Further, some of these methods also indicate high disclosure risks with synthetic data and it makes them less favourable for data generation approaches.

The use of deep learning to synthesize data using learned distributions is a novel and an active area of research. GANs, in particular, have found success as a data synthesizer in the field of synthetic but realistic images. The motivation to use GANs comes from its capacity to learn complicated high-dimensional distributions without any manual intervention. The ability of GANs to draw samples from images has been widely explored but hasn’t been tested enough on tabular datasets.
In the existing research, there are few studies that implement GANs for synthetic data generation. But the researchers have instead used GANs for data augmentation rather than creating a full synthetic datasets to replace the real data. Some of the studies that do focus on creating a full synthetic data are tested only on numerical data and not on categorical data. Most of the real-world datasets contain mixed data types (continuous, categorical and so on.), thus it is an active area of future research. Also, the synthetic data can only overcome the data-disclosure risks of real data if it is assessed on privacy metrics or risk of de-identification. Most of the researchers have not evaluated the synthetic data on privacy metrics.

Further GANs are likely to suffer from training problems such as mode collapse, non-convergence and so on. GANs need an extensive hyperparameter tuning to create a stable generative network. Hence, there are variants such as WGAN, WGAN-GP that provide a more stable training framework. WGAN-GP is an alternative that provides a more stable algorithm for training GANs and a strong modelling performance on image and text datasets. WGAN-GP can be easily extended to WCGAN-GP by inputting the condition vector, that is target labels, to the generator and discriminator network. This enables the GAN framework to learn the distributions specific to each class label and produce higher quality samples for both the class labels.

To address the limitations and gaps presented in this chapter, the thesis is focussed on implementing WCGAN-GP on mixed-type datasets and evaluated the quality of synthetic data using a mix of data utility and privacy metrics. Further, SMOTE is used as a baseline approach because of its simplicity and common usage in synthetic data generation.

This chapter presented an overview of the literature on the different types and necessity of synthetic data. Further, various approaches for synthetic data generation were reviewed with a detailed discussion of the GAN related approaches. The fundamentals of GANs, incorporating a delineation of the framework and algorithms used in the research experimentation ensued. Then a glimpse of the evaluations methods that have been incorporated to evaluate the quality of synthetic data were presented. The final section offered a summary of the reviewed literature and analysis.
of limitations or gaps in literature that are addressed by this study. Before deep-diving into the design and methodology section, the subject of Chapter 3, this chapter concludes with a reaffirmation of the research questions below.

### 2.5.3 The Research Question

The evidences lead to a theory that the use of WCGAN-GP to generate tabular synthetic datasets can overcome the limitations of previous state-of-the-art approaches. The proposed framework can produce a higher quality synthetic data with stronger privacy guarantees and better data utility than the baselines. It will also be able to overcome all the training problems with GANs and show strong modelling performance. Thus, the limitation and gaps in the literature can be addressed by the research question given as:

“To what extent can a generated synthetic data approximate the quality of a real data using Wasserstein Conditional GANs with Gradient Penalty (WCGAN-GP) using a combination of practical utility and privacy metrics?

The next chapters will describe the research design, implementation and evaluation of the experiments that will help in addressing the research question.
3. DESIGN AND METHODOLOGY

This chapter provides a detailed overview of the underlying project approach and experiment design to aid in understanding the plan and experiment conducted to address the research hypothesis. It helps in setting up a preliminary plan to organize the experiment and accomplish the objectives by defining a step-by-step plan for the study.

3.1 Research Design Overview

The research objective focuses on synthetic data generation using multiple approaches and thus the chapter is organized to follow the data generation process as shown in Figure 3.1. A high-level explanation of the research design is presented in this section. As outlined in Figure 3.1, this chapter firstly focuses on gaining understanding and familiarity with the multiple datasets in section 3.2. This involves defining the datasets to understand the data types, variables and the key properties of the datasets.

Further, this is followed by the Data preparation section 3.3, which outlines the required pre-processing steps like cleaning, standardization or transformations for numerical variables and label encoding for categorical variables. The need for data transformations varies based on the synthetic data generation approach, that is whether the approach is SMOTE or WCGAN-GP. It is imperative to mention that standardization or feature scaling is a crucial pre-processing step especially when using the GANs to facilitate faster convergence and better quality results. In WCGAN-GP, the data transformations need to be carried out so that the real data is easily understood by the generative adversarial algorithms. While with SMOTE, it is not a necessity to standardize or transform the numerical data and thus SMOTE does need any transformations on real data. Like with WCGAN-GP, the categorical variables in real data for SMOTE would also be processed with label encoding for the convenience of comparison. More details on pre-processing are explained in details in section 3.3.

The approach to generate synthetic data using SMOTE is presented in section 3.4. SMOTE is generally used for oversampling the minority class or overcome imbalance
in datasets. However, the approach can also be used beyond the purposes of data augmentation and create new data that has the same size as real data. After this, the main focus of the research, that is the experiment design of WCGAN-GP is presented. The design and modelling process of the WCGAN-GP framework is explained. The section provides details of methodology adopted for designing the experiment. It also explicates the model architecture, parameters and their chosen values, along with the rationale for the choice of parameter settings.

Once the WCGAN-GP model is trained, the synthetic data is generated using the trained generator to have the exact number of records as in real dataset. As the real data is transformed before the training of WCGAN-GP model, the output of generator, that is the synthetic data, needs to be reverse transformed to convert the format of synthetic data back into the format of original raw data. However, the process with SMOTE is relatively trivial and there is no need to train any model. SMOTE does not need any parameter optimizations and the data is simply generated using its standard package. Once the new samples are generated using SMOTE as outlined in Section 3.4.2, the synthetic samples are extracted and removed from the mix of original and synthetic samples to form the synthetic dataset. As SMOTE does not requires any data transformations on real data, it does not need any reverse transformations.

This then leads to the final evaluation section 3.5. The synthetic data generated is evaluated on the basis of data utility, such as whether statistical distributions or correlations are preserved in synthetic data, and whether the machine learning performance of synthetic data is comparable with the real data. Based on the evaluation, the WCGAN-GP model is fine-tuned and re-run until the best model is achieved. Based on the performance of synthetic data against real data, the best model with the optimal settings for WCGAN-GP is selected. Further, the synthetic data is evaluated for privacy metric using Euclidean distance to offer perspective on how similar the samples are between synthetic and real datasets.

The chapter concludes with an assessment of the limitation and strengths of the proposed design and experiment setup. The use of detailed visual representations is carried out in the entire chapter to aid in better understanding of the design and considerations in deciding the model architecture for the experiment setup.

43
Figure 3.1. Overview of Research Design
Software Implementation: The implementation of experiments is carried out using Python 3.7 and specifically, Keras and TensorFlow. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow. Both Keras and TensorFlow are open source deep learning libraries and most appropriate for the implementation of GANs because of the flexibility they offer. The machine learning performance of the real and synthetic data is carried out with the classifiers from scikit-learn library and Xgboost. The Python libraries for the tasks in the research have been listed in Table 3.1.

Table 3.1. List of all Python Software Packages used in experiments

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keras</td>
<td>2.0.9</td>
<td>High level neural networks - GANs</td>
</tr>
<tr>
<td>Tensorflow</td>
<td>1.3.0</td>
<td>Machine Learning framework</td>
</tr>
<tr>
<td>Scikit-learn</td>
<td>0.23.2</td>
<td>Data analysis, Machine Learning algorithms</td>
</tr>
<tr>
<td>Xgboost</td>
<td>1.3.0</td>
<td>Xgboost Implementation</td>
</tr>
<tr>
<td>Imblearn</td>
<td>0.0</td>
<td>Re-sampling technique - SMOTE</td>
</tr>
<tr>
<td>Scipy</td>
<td>1.0.0</td>
<td>Statistical functions</td>
</tr>
<tr>
<td>Pandas</td>
<td>0.22.0</td>
<td>Data manipulations</td>
</tr>
<tr>
<td>Numpy</td>
<td>1.14.2</td>
<td>Data manipulations</td>
</tr>
<tr>
<td>Matplotlib</td>
<td>3.3.1</td>
<td>Data visualizations</td>
</tr>
<tr>
<td>Seaborn</td>
<td>0.10.1</td>
<td>Data visualizations</td>
</tr>
<tr>
<td>Dython</td>
<td>0.4.2</td>
<td>Data analysis</td>
</tr>
</tbody>
</table>

3.2 Datasets

Before implementing the modelling techniques and experiments on the datasets, it is fundamental to first gain knowledge and understand about the datasets. This section focusses on gaining familiarity with datasets and the nature of each dataset.

To remind the reader, the main objectives of this research is to understand the benefits of WCGANs with Gradient Penalty (WCGAN-GP) in the generation of synthetic data. Therefore, multiple publicly available and sensitive datasets pertaining to different domains are explored to ensure reliability and validity of the designed experiments. These datasets are selected for multiple reasons. First, these datasets belong to the case of classification problems and some of the datasets are even imbalanced, implying
uneven distribution of classes. Such datasets are pretty common occurrence in real-world domains. Secondly, the datasets consist of a mix of categorical and numerical variables and can be potentially categorized as medium-sized datasets as they comprise of 30,000 – 70,000 samples. For the reasons discussed above, three datasets belonging to different verticals focussed on Banking, Healthcare and Census respectively are used. The properties of each dataset is summarized in Table 3.2.

Table 3.2. Properties of Datasets used in experiments

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th># Rows</th>
<th># Features</th>
<th># Categorical</th>
<th># Numerical</th>
<th>Class Imbalance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default of Credit Card</td>
<td>Banking</td>
<td>30,000</td>
<td>23</td>
<td>9</td>
<td>14</td>
<td>✔</td>
</tr>
<tr>
<td>Cardiovascular Disease</td>
<td>Healthcare</td>
<td>70,000</td>
<td>11</td>
<td>6</td>
<td>5</td>
<td>✗</td>
</tr>
<tr>
<td>Adult Census</td>
<td>Census</td>
<td>31,562</td>
<td>14</td>
<td>9</td>
<td>5</td>
<td>✔</td>
</tr>
</tbody>
</table>

Note: The Table depicts the number of features in the data and not the total columns. The target label is binary class for each dataset, so the number of columns will be an addition of 1 to the number of features. Also, ID variables have been excluded from the list of features or attributes.

3.2.1 Default of Credit Card Clients (Banking)

The “Default of Credit Card Clients” dataset is available at UCI Machine Learning Repository² (Lichman, 2013). It contains transactions made by credit card clients in Taiwan from April 2005 to September 2005. The dataset is related to the Banking industry and enables banks to identify defaulters of credit card payments.

The dataset has 23 features and contain details about demographics (like age, gender, education), historical information of bill statements and payments by customers. The values for variables such as Limit_bal, bill statements and payments are in NT dollars. It contains one column as ID variable, which contains a set of natural numbers in ascending order, such as 1,2,3 and so on. As the ID variable will not be used in

² https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset
models, it is not considered as a feature. Besides the ID variable, there are 23 features (categorical: 9, numerical: 14) and the number of rows in data are 30,000.

The target class is “Default payment” and signifies if the client will do the payment in the next month. The target value is 1 in case of a default and 0 otherwise. Among the 30,000 customers, 6636 (22.1%) instances are positive (that is default) and the rest are zero (that is not a defaulter). The dataset is considered to be heavily imbalanced or lopsided as the percentage of default transactions (22.1%) are much lesser than the non-fraudulent transactions (77.88%). The data descriptions of Credit Card Default data can be found in appendix in Table A.1.

3.2.2 Cardiovascular Disease (Healthcare)

The Kaggle Cardiovascular Disease dataset\(^3\) contains results of medical examinations of patients. The dataset is related to healthcare industry and supports a classification problem to predict whether a patient has a cardiovascular disease or not. The dataset is useful to identify patients who are likelihood to develop a heart disease.

The dataset has 11 features, that contains details about demographics (such as age, gender, height), results of medical examination (such as cholesterol, glucose, blood pressure), and other details pertaining to the patient’s activities. The total number of features are 11 (categorical: 6, numerical: 5) and the number of patients in this dataset are 70,000. Among the 70,000 patients, there is approximately an equal proportion of patients having cardiovascular disease (49.97%) vs. patients who are healthy (50.03%). It also contains a column as ID variable, which contains a set of natural numbers in ascending order and has been excluded from the list of features or an attributes.

The target class “cardio” equals 1, when the patient has a cardiovascular disease and it’s 0 when patient is healthy and has no cardiovascular disease. The task is to predict the presence or absence of cardiovascular disease using the examination results. Unlike the Credit card data, this dataset does not suffer from data imbalance. Data descriptions of Cardiovascular Disease data can be found in appendix in Table A.2.

\(^3\) https://www.kaggle.com/sulianova/cardiovascular-disease-dataset
3.2.3 Adult Census (Census)

The US Adult Census data extracted from the 1994 Census bureau database is taken from the Kaggle website\(^4\) (Kohavi & Becker, 1996). It contains a mixture of categorical and numerical columns like age, education, marital status, relationship and so on. This dataset has a higher number of categorical variables than numerical variables. The dataset is used for classification problems to predict whether the yearly income for a person is over $50,000 or not.

The total number of features are 14 (categorical: 9, continuous: 5). The target class “income” is 1 in case of an income over $50K a year and 0 otherwise. There are a total of 32,561 records, out of which 7841 (24%) people have income over $50k and the rest below $50K. The dataset is considered to be imbalanced as the percentage of people with income over $50K are much less than the other label class. Data descriptions of Adult Census data can be found in appendix in Table A.3.

3.3 Data Preparation

In this phase, all the pre-processing activities are performed, and the resultant dataset is then fed into the data generation algorithms. Data manipulations like imputing missing values, transformations like feature scaling and label encoding are presented in this section. The pre-processing is limited when working with GANs or SMOTE as we aim to generate synthetic data as similar to the real data. Thus, outlier treatment or feature engineering are not carried out as part of the pre-processing activities.

Transformations are mandatory for GANs to ensure the data is easily ingested into the algorithms and produce good quality synthetic samples. There are multiple ways to achieve these based on the data types involved. The continuous variables are often normalized using standardization or normalization. However, for categorical variables, encoding is performed.

\(^4\) https://www.kaggle.com/uciml/adult-census-income
With the baseline synthetic data approach, that is SMOTE, only the pre-processing activities like removal of missing values and use of encoding for categorical variables are performed. On the contrary, GAN related approach is relatively non-trivial and requires building and training of models to generate synthetic data. Thus, it not only needs removing of missing values and encoding, but also requires feature scaling or normalization. A summary of the pre-processing activities can be found in Table 3.3.

**Table 3.3. List of Pre-processing activities required for data generation approaches**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Pre-Processing</th>
<th>SMOTE</th>
<th>GANs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Cleaning</td>
<td>Missing Value Treatment</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Data Transformation</td>
<td>Feature Scaling/Normalization</td>
<td>Not required</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Label Encoding</td>
<td>Not required but implemented for comparison purposes</td>
<td>✔</td>
</tr>
</tbody>
</table>

Further, a decision is also made on what columns will be generated using the data generation approaches. Datasets (like Default of Credit Card, Adult Census) have a column “ID” that represent a unique ID, consisting of a range of numbers in ascending order. This column does not have any correlation with the other features and is not a useful column as it does not carry any important information. Thus, this column is not considered and removed from any analysis. Such columns can be manually added after the synthetic data is generated. Since these “ID” column have a range of numbers in ascending order starting from 1, the process of adding back these unique IDs can be straightforward. For instance, the numbers in the range of 1 to n are considered, where n is the length of synthetic data and then the numbers can be randomly sampled without replacement to place them in the unique ID column.

### 3.3.1 Data Cleaning

It is important to handle missing values as most of the deep learning or machine learning algorithms do not work well with missing data. Missing values generate problems like reducing the sample representativeness or its statistical power and ultimately causing bias in estimation of parameters of GANs.
The Credit Card Default and Cardiovascular Disease datasets do not have any missing values, so there is no need of imputing missing values in those datasets. However, Adult Census data does comprise of missing values in some columns. One straightforward approach is to drop the records with missing values, but the data is imbalanced and hence, this approach might further reduce the number of minority class samples. Therefore, it makes more sense to retain the records with missing values and perform data imputations. Tabachnik and Fidell (2013) noted that if missing data represents less than 5% of the total data and is missing in a random pattern from a large dataset, then any procedure of handling missing values provide similar results. The dataset has less than 5% of missing records, thus the missing value treatment using mode substitution will be performed. The imputation method will be similar for both the SMOTE and GAN approaches for the convenience of comparison.

### 3.3.2 Data Transformations

In order to generate a good quality synthetic data, the input data to the GANs needs to be in an appropriate representation. The success of GANs is attributed to cleaning and pre-processing activities. The type of data transformations varies for the data types involved. It is important to note that these transformations are only needed on real data for GANs and are not required with the SMOTE approach. But the label encoding of categorical variables is done for SMOTE only for comparison purposes.

**Label Encoding**

The datasets in the experiments are mixed data types and contain both continuous and categorical variables. GANs are known to have challenges with producing a quality categorical synthetic data. The reason is because of the decision to be made at generator output for categorical values. This implies that if the data contains categorical data, it is desired to encode them into numbers before running the generative models. The categorical (both nominal and ordinal) variables can be converted into 0 and 1 using one hot encoding, but it will create sparse data problems and high memory consumption. Henceforth, label encoding is used in the experiments to assign a unique integer to each label based on an alphabetical ordering.

**Standardization/ Normalization**
The continuous features (that includes the numerical as well as label encoded) are normalized to bring them into the same scale or standardize the range of features (Brenninkmeijer et al., 2019). This step is outlined as one of the crucial steps for successful implementation of GANs as these manipulations are responsible for faster convergence, better processing and ease of reproducibility of results. There are many numerical variables present in datasets and one of the issues is that the range of features could differ significantly from each other. If the original scale is used during modelling, there is a tendency to put more weight on the features with a larger range. Thus, techniques such as Standardization or Normalization are applied to rescale all the features to almost the same scale. This transformation ensures that each feature gets an equal importance and avoid any biases due to the scale of any specific attribute. It also makes the data easier to process by GANs as it reduces the range of values that the generator has to generate.

Standardization (or Z-score Standardization) is a common approach in which the features are rescaled to ensure that the mean and standard deviation is 0 and 1, respectively. The features are scaled using the sklearn StandardScaler() function. Another common approach is the Max-Min Normalization (Min-Max scaling). This technique changes all the values of features between the interval of 0 and 1. For every feature, the minimum value of that feature gets transformed into 0, the maximum value gets transformed into 1 and every other value gets transformed into a decimal between 0 and 1. The features are normalized using the sklearn MinMaxScaler() function. The choice of using standardization or normalization is problem and algorithm specific and there is no one-size-fits-all approach. Thus, the impact of both normalization and standardization will be experimented and compared to identify the better approach.

### 3.4 Experiment Design

This section focusses on the implementation and technical details of the vital parts of the proposed GAN framework, that is Wasserstein Conditional GANs with Gradient penalty. The key focus of this research is based on GANs, hence the implementations mainly focusses on this improved GAN variant. It also focusses on the implementation
of the baseline synthetic data generation approach – SMOTE, which is used for comparison purposes.

3.4.1 WCGAN-GP

In order to design the WCGAN-GP model, there are various hyperparameters that need to be determined. The inventor of GANs (Goodfellow et al., 2014), author of original paper of WGAN (Arjovsky et al., 2017) and WGAN-GP (Gulrajani et al., 2017) have provided recommendations and guidelines on the choice of hyperparameters that have been proved to be successful for many tasks. Therefore, instead of randomly choosing the hyperparameters, the hyperparameters or settings defined in this section are based of the suggestions from the experts in the area.

It is important to note that discriminator is renamed as “critic” in WCGAN-GP, but the critic plays a similar role as the discriminator, that is to evaluate whether the data is fake or real.

3.4.1.1 Hyperparameters of WCGAN-GP model

The hyperparameters defined for the proposed WCGAN-GP model are presented in this section.

**Neural Network Architecture:** Most of the traditional GAN implementations are focussed in the field of creating synthetic images, but this research deals with structured or tabular data generation. Therefore, the WCGAN-GP model will be designed in a different manner as compared to the way it is commonly defined. The standard designs of GANs for image generation have deep convolutional neural networks as its generator and critic. However, this research will have dense or fully connected layers for both generator and critic. Both the networks will be simple feed-forward neural networks, implying regular neural network layers. This architecture of GANs for structured data generation has resulted in good quality synthetic data and will be used in the experiments.
**Number and Size of Hidden Layers:** The number of layers for both the generator and critic networks will be experimented with 1 up to 3 hidden and fully connected layers. For a particular iteration, the number of layers in both the networks will be the same. The number of layers is one of the parameter settings and optimum number will be decided based on outcomes of different iterations.

Prior works for tabular data generation using GANs denote promising results when the nodes for generator are ordered in ascending and for critic (or discriminator) in descending order (Tanaka & Aranha, 2019). The authors have shown success in synthetic data generation by defining the size of hidden layers as powers of 2.

The size of hidden layers is important as a size too small or large can adversely impact the quality of synthetic data. The use of few neurons in hidden layers can result in underfitting, that is hidden layers fail to adequate learn the probability distributions from the original dataset. In case of too many neurons, the critic improves during the training but fails during validation as it has started to memorize the training data. This can also lead to a form of mode collapse where the critic has overfitted on the data and always distinguishes between real or fake. A large size of neurons in hidden layers also increases the training time of the network and results in high computational costs. Consequently, both the number of hidden layers and their size (that is the number of neurons) will be altered in the different iterations to find a balance between too many and few hidden layers or nodes.

**Dimension of the Models:** As the WCGAN-GP framework includes a conditional element to allow targeted generation of samples for each label class, the generator will receive random noise and a conditional vector as the inputs. Similarly, the critic receives the output of generator or original data and a conditional vector as an input. Another reason for conditioning both generator and critic with an extra information of class labels is to minimize biasness due to dominance of majority class samples.

The output of critic network is a fully connected layer with a single node. But the output of generator is a fully connected layer with N units, which is the dimension size or number of columns in real data. The generator and critic have an architecture of a feed-forward neural networks and is illustrated in Figure 3.2 and Figure 3.3.
**Figure 3.2. Architecture of Critic in WGAN-GP**

- **Input layer:** N units, where N is size of data dimension or number of columns in original data.
- **Output layer:** 1 units
- **Hidden layers:** 1 to 3 Layers
- **Number of Neurons:** Descending orders of Power of two ($2^x$, where x is 7, 8, 9, or 10).
  For instance, 1st hidden layer has $2^x$ neurons, 2nd hidden layer has $2^{x-1}$ neurons and so on.
- **Output:** $-\infty / +\infty$

**Figure 3.3. Architecture of Generator in WGAN-GP**

- **Input layer:** 32 or 100 units (Size of noise vector)
- **Output layer:** N units, where N is size of data dimension or number of columns in original data.
- **Hidden layers:** 1 to 3 Layers
- **Number of Neurons:** Ascending orders of Power of two ($2^x$, where x is 7, 8, 9, or 10).
  For instance, 1st hidden layer has $2^x$ neurons, 2nd hidden layer has $2^{x-1}$ neurons and so on.

**Dropout:** Since the network architecture comprises of deep neural networks, the use of dropout is incorporated to eliminate some neurons during the training and minimize the loss function stochastically to prevent overfitting (Srivastava et al., 2014). This is in alignment with the theoretical framework proposed by Goodfellow et al. (2014), permitting the use of dropout at intermediate layers of generator and discriminator. The performance with and without a dropout layer in the generator and critic is investigated. The combination of dropout values (0.1 up to 0.5) is used. Values higher than 0.5 for dropout will be avoided, as it will reduce training of majority of neurons and result in significant quality drop.
**Activation Function:** Activation function is one of the building blocks of neural networks and without an activation function, the neural networks performs linear transformations on the inputs using the weights and biases. A network with linear transformations is simpler, but the resulting network is less powerful to learn the complex patterns within the data. Thus, the experiments with the vanilla GANs will use non-linear transformations, introduced by the use of activation functions.

Researchers in the field like Radford et al. (2015) have suggested architecture guidelines on the choice of activation functions for GANs. The use of ReLU activation (Nair & Hinton, 2010) is recommended in generator with the exception of output layer. Further, leaky rectified activation (Maas, Hannun, & Ng, 2013; Xu, Wang, Chen, & Li, 2015) also works well with the generator and critic for all except the last layer. As some of the traditional applications of GANs are related to image generation, this research will have different iterations involving the use of LeakyReLU or ReLU activation in the input layers of generator and critic to find the optimal setting.

With vanilla GANs, the sigmoid activation function is used in the output layer of discriminator. But in WCGAN-GP, the output is not sigmoid and does not represent a probability. Thus, the critic model requires a linear activation to predict the score of realness for a given sample. The output of critic model should be as large and negative as possible for generated inputs and as large and positive as possible for real inputs. This will be achieved by setting activation to linear in the output layer of critic model. The linear activation is the default activation for a layer, so there is no need to specify any activation function for the output layer of critic.

For the generator, both tanh and linear activation function are tested to find the optimal setting. Hyperbolic tangent is experimented as it has been observed that the use of a bounded activation such as tanh allows the model to learn more quickly to saturate and learn the training distributions effectively (Liew, Khalil-Hani, & Bakhteri, 2016).

**Optimizer:** Adam optimizer (Kingma & Ba, 2014) with tuned hyperparameters is chosen as it is the suggested optimizer by authors of WGAN-GP and GAN (Gulrajani et al., 2017; Goodfellow et al., 2014; Radford et al., 2015). Additionally, RMSProp is
recommended by authors of WGAN (Arjovsky et al., 2017). Thus, both the generator and critic are trained with Adam and RMSProp to find the optimal setting.

**Learning Rate and Momentum:** In the GAN architecture, the learning rates are suggested to be small to produce stabilized GAN models (Tanaka & Aranha, 2019). Reason for this is that smaller values result in slower movement across the gradient slope, making sure that the local minima is not missed. But higher learning rates might cause the gradient descent to overshoot the minima and as a result, the resulting model fails to converge to minima and leads to training failure of GANs. Therefore, a learning rate (0.001, 0.005, 0.0001, 0.0002) will be tested. Further, to demonstrate training problems at higher learning rate, learning rate between 0.01 and 0.05 will also be investigated.

The momentum term $\beta_1$ and $\beta_2$ are suggested as 0.5 and 0.9 respectively by the authors of WGAN-GP (Gulrajani et al., 2017). For this reason, the experiments will have these settings for the hyperparameters of the optimizer.

**Batch Size:** Batch size determines the number of instances passed through the model before the backpropagation for each epoch. The experiments will use various mini-batch sizes (more than 1 and less than total samples in training data). Smaller batch size results in updating the error gradients based on smaller batch of samples and offers a regularization effect to reduce the generalization error. However, there is evidence which suggests that increasing the minibatch size improves the performance of GANs (Brock, Donahue, & Simonyan, 2018). But a bigger batch size is expected to adversely impact the performance, as the initial training of critic using a lot of samples will result in overpowering the generator. Thus, the experiment will involve iterations by increasing the batch sizes by a factor of 2 (like 8, 16, 32, 64, 128, 256).

**Loss function:** In the WCGAN-GP model, the two networks are trained using different loss functions. One of the key alterations introduced with WCGAN-GP requires the use of Wasserstein loss function instead of the Binary Cross-entropy loss. The generator is optimized using the Wasserstein loss, but for the critic, gradient penalty is also added to the Wasserstein loss. The new losses are implemented as a custom function in Keras/ TensorFlow. Arjovsky et al., (2017) have provided evidence
that the cross-entropy loss function causes training problems in GANs and make the training unstable. For this underlying reason, Wasserstein distance was proposed to measure the difference in the distributions between real and fake data, instead of classifying the samples. The loss predicts a score of how real or fake the given sample is and this score can be extremely large for poor quality synthetic samples.

3.4.1.2 Model Implementation

The code implemented for WCGAN with Gradient Penalty is based on the GitHub repository available online\(^5\). This repository contains implementation of some GAN frameworks in Python using Keras and Tensorflow libraries.

**Building the Generator model:** The generator model is built using the Keras Sequential model along with Dense and Batch Normalization layers. The input is the noise vector and conditional vector (that is the target class labels) and the output is desirable synthetic data. The activation function used is either Leaky ReLu or ReLU. The generator model can be divided into several blocks. One block consisting of Dense Layer, Activation, and Dropout. Depending on the number of hidden layers chosen, such blocks are added. The final block has linear activation or Tanh as the activation function and does not have any Dropout layers. Note that the size of hidden layer increases in an ascending order for every subsequent hidden layer. Finally, the generator model is compiled using the Wasserstein loss. The hyperparameter settings for generator model are summarized and shown in Table 3.4.

**Building the Critic model:** The critic takes in samples from either the original data or the generated synthetic samples along with a conditional vector (that is the target labels) as an input. There is no activation function in the output layer, implying a linear activation function by default. The critic model uses Wasserstein Loss and Gradient Penalty loss, which are defined as custom functions in Keras/ TensorFlow. A new hyperparameters called n_critic is also defined so that the critic is trained for more iterations than the generator in a 5:1 ratio (Arjovsky et al., 2017). The hyperparameter settings for critic model are summarized and shown in Table 3.5.

---

\(^5\) [https://github.com/codyznash/GANs_for_Credit_Card_Data](https://github.com/codyznash/GANs_for_Credit_Card_Data)
Table 3.4. Hyperparameter Settings for Generator Model in WCGAN-GP

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network architecture</td>
<td>Feed-forward networks</td>
</tr>
<tr>
<td>Hidden layers</td>
<td>{1,2,3} fully connected</td>
</tr>
<tr>
<td>Number of neurons</td>
<td>{2^x, where x is 7,8,9,10 in ascending order}</td>
</tr>
<tr>
<td>Dropout</td>
<td>{No, Yes (alpha: 0.1, 0.2, 0.3, 0.4, 0.5)}</td>
</tr>
<tr>
<td>Activation function - Input</td>
<td>{ReLu, Leaky ReLu}</td>
</tr>
<tr>
<td>Activation function - Output</td>
<td>{Linear, Tanh}</td>
</tr>
<tr>
<td>Loss function</td>
<td>Wasserstein Loss</td>
</tr>
<tr>
<td>Optimizer</td>
<td>{Adam, RMSProp}</td>
</tr>
<tr>
<td>Learning rate</td>
<td>{0.01 - 0.05, 0.001 - 0.0005}</td>
</tr>
<tr>
<td>Optimizer’s hyperparameters</td>
<td>(\beta_1 = 0, \beta_2 = 0.9)</td>
</tr>
</tbody>
</table>

Table 3.5. Hyperparameter Settings for Critic Model in WCGAN-GP

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network architecture</td>
<td>Feed-forward networks</td>
</tr>
<tr>
<td>Hidden layers</td>
<td>{1,2,3} fully connected</td>
</tr>
<tr>
<td>Number of neurons</td>
<td>{2^x, where x is 7,8,9,10 in ascending order}</td>
</tr>
<tr>
<td>Dropout</td>
<td>{No, Yes (alpha: 0.1, 0.2, 0.3, 0.4, 0.5)}</td>
</tr>
<tr>
<td>Activation function - Input</td>
<td>{ReLu, Leaky ReLu}</td>
</tr>
<tr>
<td>Activation function - Output</td>
<td>{Linear}</td>
</tr>
<tr>
<td>Loss function</td>
<td>Wasserstein Distance, Gradient Loss Penalty</td>
</tr>
<tr>
<td>Optimizer</td>
<td>{Adam, RMSProp}</td>
</tr>
<tr>
<td>Learning rate</td>
<td>{0.01 - 0.05, 0.001 - 0.0005}</td>
</tr>
<tr>
<td>Optimizer’s hyperparameters</td>
<td>(\beta_1 = 0, \beta_2 = 0.9)</td>
</tr>
</tbody>
</table>

**Data Generation:** Once the model has been trained, the trained generator is used to produce synthetic data samples. In this phase, the size of random noise and required sample size of the synthetic data is given as an input to trained generator. After the training of GANs is completed, the generation of synthetic samples from the trained generator is quite fast.

**Reverse Transformations:** The synthetic data generated using WCGAN-GP is in a standardized or rescaled range because of the initial transformations that are applied on the real input data. Once the synthetic data has been generated, the transformations are reversed (with respect to the initial transformations) to undo the scaling according to the original feature range. This is done using sklearn’s inverse_transform function. As a result of this reverse feature engineering, the synthetic de-normalized data reflects
the original scale or range of real input data. Using the inverse transformations changes the interval [0,1] back to +/-infinity. The categorical variables not only undergo the inverse transformations but are also further rescaled based on the minimum and maximum values of each label encoded variables in real data.

3.4.2 SMOTE (Baseline)

An alternative synthetic data approach is SMOTE (Chawla et al., 2002) that does not involve GANs has been implemented for comparison purposes. SMOTE is chosen as a baseline approach due to its popularity and common usage. Compared to GANs, the process with SMOTE is trivial for generating the synthetic data. SMOTE synthesises new data instances between original instances and is traditionally used to address problems of data imbalance. It creates synthetic samples from the minority class rather than creating copies of data from minority class. It selects the similar records from minority class and alters the records by changing one column at a time by a random amount to balance the data. This approach does not require to train any model or any parameter optimization. Thus, it offers some advantages over the GAN-based approach as the synthetic data can be generated instantly.

SMOTE is used to generate synthetic data using the following approach and also can be visualized as shown in Figure 3.4 (Kaloskampis et al., 2019):

- At first, the original dataset goes through the pre-processing activities, that is imputation of missing values and one-hot encoding of categorical variables.
- The original dataset (with n instances) is replicated to create copies of the dataset and made imbalanced in a ratio of 2 to 1. A new target label is assigned with label as 1 for majority class and 0 for minority. This results in two groups, with the majority class having 2*n instances and minority with n instances. The resulting dataset has the same number of variables as original but also has an additional target label created earlier.
- SMOTE is run to generate synthetic data samples using the imbalanced-learn library. This generates new synthetic samples with n new instances.
- The balanced data now contains synthetic samples in the ‘0’ target label and the new samples are appended at the end of original samples by default. As a final
step, the synthetic samples are then removed from the original data and the target label created in step 2 is dropped. This results in generating synthetic samples with the exact size of instances and variables as in real data.

- Lastly, identical records are found between the synthetic and real records and are removed. Because of the way how the SMOTE algorithm works to generate the synthetic samples, there are few matches expected in the synthetic data against the real records. This removal of identical matched records is only done with the SMOTE approach and not with the data from WCGAN-model.

![Diagram](image)

*Figure 3.4. Synthetic data approach using SMOTE (Baseline)*
3.5 Evaluation Methods

Multiple metrics for data-utility and privacy are used to evaluate the quality of synthetic data generated by WCGAN-GP and SMOTE method. The data-utility of synthetic samples is assessed in multiple stages. Firstly, the visualisations are assessed to verify whether the patterns or relationships are intact. This is more like a qualitative assessment done using boxplots, histogram and scatterplots between variables of real between variables of synthetic data. Then the similarity of data is assessed using two correlation indicators to check whether the variables in the synthetic and original data are correlated or not. Finally, data utility of synthetic data is evaluated using its machine learning performance. Basically, the behaviour of both real and synthetic data is checked when faced with a classification task. The machine learning algorithm is trained and tested using Real data (TRTR) and similar process is carried out with Synthetic data (TSTS). The results are evaluated using performance metrics like Accuracy, F1 score, and AUC-ROC.

The evaluation of data to check whether privacy is maintained with the synthetic data is an another critical criteria in the research. If the synthetic data leaks information about the real data, it cannot be shared with the public and can cause issues. With respect to evaluating whether the privacy is maintained in the synthetic data, measure like Euclidean distance is recorded. This is done to verify whether the risk of re-identification is minimized with the generation of synthetic data and the synthetic records cannot be used to link back or identify any sensitive information about individuals. Although not a criteria of evaluation, the identical records between synthetic data from WCGAN-GP and real data are computed and noted to show the capability of WCGAN-GP to show privacy preserving properties.

3.5.1 Data Utility Metrics

3.5.1.1 Visual Evaluation

The visual inspection of synthetic data quality can provide great insights that the quantitative evaluation cannot. The research focusses to generate synthetic data that preserves the distribution and patterns from the real data. Thus, a visual evaluation is performed in multiple stages.
Univariate analysis is performed in the first stage to observe the Box and Whisker plots for the numerical variables by differentiating the target class by colours. Box Plots will verify whether the synthetic samples have values within similar ranges and if the distributions or spread remains intact or not. In addition, the histogram distribution of categorical variables will also be observed. The synthetic data quality will be qualitatively assessed using the visualisations.

The second stage looks at bi-variate analysis of variables (with respect to each target class) to compare the scatterplots between variables in the synthetic data against the variables in the real data. This will help to affirm whether the relationship and patterns within the real data are preserved in the synthetic data or not. This analysis will also indicate the existence of mode collapse, and if the generator has been successful in fooling the critic. Another evaluation considers to check the correlations between the columns of each dataset. Creating a correlation matrix using heatmaps is important to understand the relationships between the columns and whether they are preserved in the synthetic data. Heatmaps are useful as the trends or patterns can be observed by looking at how the cell colours change across each axis.

3.5.1.2 Similarity of Data

The goal of synthetic data generation is to understand how close the synthetic data is to a real data but the variables between the two datasets should not correlated. Thus, the focus is on estimating the similarity by doing correlation analysis. The purpose of this statistical metric is to measure the relationship between variables in real and synthetic data. The correlation between real and synthetic data should be minimal when comparing variables in datasets, indicate there is no correlation between variables of real and synthetic dataset (Vega et al., 2020). This aspect is important if synthetic data needs to be shared with the public, as there is a need of a synthetic data that does not preserve correlations with real data but behaves in the same manner in machine learning tasks.

In this section, two correlation indicators, that is Pearson’s correlation coefficient, and Spearman’s correlation coefficient will be calculated to check whether the variables in the real and synthetic data are correlated or not.
Pearson’s correlation coefficient (Sedgwick, 2012) suits well as it is a metric that determines the linear correlation between two variables and in this case, the optimal relation is linear. The results of these metrics generally range between $[-1, 1]$ and indicates the correlation coefficient where $-1$ means a negative correlation (that is, when $X$ increases, $Y$ decreases or vice versa), $0$ means no correlation and $1$ means positive correlation (when $X$ increases, $Y$ too increases or vice versa).

Some of the columns can contain extremely large numbers or outliers and can impact the results of Pearson’s correlation coefficient. To counter this, Spearman’s correlation coefficient (Schober et al., 2018) will also be used. It is a rank correlation metric and calculates the ranks of the values of each of the two variables instead of their actual values. The result is a metric similar to Pearson’s, but less perceptive to outliers in the tails. Spearman correlation coefficient assesses whether there is a monotonic relationship instead of a linear one (as in Pearson’s). It is used to summarise the strength between the two variables when the two variables are not related by linear relationship. Like in Pearson’s correlation coefficient, the range varies between $-1$ and $1$ where $-1$ indicates a negative correlation and $1$ means positive correlation. As part of evaluating the similarity between the two sets of data, both these correlation indicators will be used as a part of the evaluation metric.

### 3.5.1.3 Machine Learning Performance

The machine learning performance is the most important metric of this study. The synthetic data is valuable if it can replace the real data and perform in the same way as the real data does in machine learning tasks (Heyburn et al., 2018). The datasets in the research have a target class and require a classification task. The goal of classification would be to accurately predict the target class for each sample in the data.

The classifiers are chosen because of their common usage and not for any specific performance on these datasets. As the target columns are binary, classifiers such as Decision Tree, Random Forest, Support Vector Machine, and Adaboost are chosen. XGBoost is also chosen as it is a popular implementation of gradient boosted decision trees designed for speed and performance. It has gained popularity and attention in recent times and has been rated as the algorithm of choice in many machine learning
competitions (Sandulescu & Chiru, 2016). The process of comparing the machine learning performance is summarized and explained in Figure 3.5 below.

**Figure 3.5.** Evaluation framework for Machine Learning performance of real and synthetic data

Two sets of machine learning models are created (ER and ES) for Real and Synthetic data respectively. The steps of the evaluation are described below:

1. Real data R and Synthetic data S are split in a 5-fold cross validation split for train/test and resulting in 4 different datasets of TrainR, TestR, TrainS, TestS.
2. Machine learning models for Real data (ER) are defined. They are trained on TrainR and tested on TestR. Similarly, models on Synthetic data (ES) are separately trained on TrainS and tested on TestS.
3. The performance of both models (ER and ES) is evaluated on the respective test datasets using the performance metrics.

Note that the purpose of this machine learning metric is not aimed at finding out the best classifier and performance in the prediction tasks. But the objective is to get comparable results between the synthetic and real data. Thus, the imbalanced datasets are not balanced or oversampled using resampling techniques for the tasks.

For the evaluation of the effectiveness of classifier, the performance metrics - Accuracy, F1 score (harmonic mean between the precision and recall), and AUC-ROC are recorded. Two of our datasets, Credit card and Adult Census are imbalanced datasets, hence, it makes more sense to refer to metrics other than Accuracy (Jeni et al., 2013) for all these datasets.
A confusion matrix is created for each model and then Accuracy, F1 score and AUCROC are calculated. Below is a quick summary of the metrics:

- **Accuracy**: It is a widely used metric for measuring the performance of a classifier. It is calculated as the percentage of the correctly classified positive and negative class samples.

- **F1 score**: F-measure also called as F1 score is the harmonic mean of precision and recall. 
  \[ \text{F1 score} = \frac{2 \times \text{Precision} \times \text{recall}}{\text{Precision} + \text{Recall}} \]

- **ROC-AUC score**: ROC curve is a plot between True Positive Rate (TPR) and False Positive Rate (FPR). ROC-AUC is a measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. Higher the score, better the model is at distinguishing samples between classes.

### 3.5.2 Privacy Metrics

Euclidean distance is the main measure that is proposed to evaluate the disclosure risk or risk of re-identification with synthetic data (Giannella et al., 2019; Lu et al., 2019).

#### 3.5.2.1 Euclidean distance

A distance based metric like Euclidean distance is commonly used to determine the likelihood to identify a personal information from the synthetic data (Giannella et al., 2019; Park et al., 2018). It offers perspective into how similar the individual records are between any two datasets. Euclidean distance to nearest record \((d)\) is the mean distance between synthetic sample and its closest record in original data. A record with zero distance would imply leakage of information and low privacy. The desired outcome is a high mean and low standard deviation. If the standard deviation is high (or close to the mean), it implies that many synthetic records have high similarity with the real records.

#### 3.5.2.2 Identical Records between Synthetic and Real Data

This metric is not included in the main evaluation metrics but is a relevant metric to record. It is a simple analysis to check if any samples in the synthetic data are identical to corresponding samples in the real data.
With SMOTE, it is expected to get duplicates between synthetic data and original data, due to the nature of the approach. Hence, the removing of matched records was proactively carried out. But with GANs, this metrics gives us an idea whether mode collapse occurred or not and how well distributed the data is. Even though, this is not a strong metric to evaluate the privacy but is recorded as it is a basic standard that needs to be fulfilled.

3.6 Summary

For conducting the experiment, first the dataset is gathered from the Kaggle website and pre-processed to convert it into the format that is easily digestible for GAN model. After this, the WCGAN-GP model is designed with the hyperparameters suggested by the inventor of GANs (Goodfellow et al., 2014) and experts in the field of GANs (Arjovsky et al., 2017; Gulrajani et al., 2017). The baseline SMOTE is run using its standard package and the designed WCGAN-GP model is built and trained. Hyperparameter tuning is carried out with WCGAN-GP model to find the optimal settings. Finally, the evaluation of the quality of synthetic dataset from multiple approaches is carried out using the evaluation metrics.

3.6.1 Strengths

In this section, the strengths of the Design and Methodology are discussed in brief.

**Designed to handle Mixed-type Datasets:** Mixed-type datasets from different domains are chosen for this research to show the validity and generalizability of WCGAN-GP for synthetic data generation. The proposed WCGAN-GP is designed to handle the mixed-type datasets (such as categorical and numerical) at the same time and in the same model. There is no need to build and train separate models for each data type.

**Pre-processing of Real Data:** The pre-processing activities like standardizations of numerical variables and label encoding of categorical variables is used to improve the
performance of the WCGAN-GP model. These transformations will help to improve the quality of synthetic samples generated.

**Provide Stable Training and No Mode Collapse:** The proposed WCGAN-GP model uses Wasserstein loss and Gradient penalty to penalize the explosive gradients from critic and improve the stability of training process. This will help to reduce the chances of mode collapse and other training problems associated with GANs. Thus, the proposed framework will demonstrate stable training with tabular and mixed type datasets.

**Stronger Modelling Performance and Better Modelling of Data Distributions:** As the WCGAN-GP model is designed to provide more stability in the training phase, the proposed WCGAN-GP is likely to show better modelling performance in learning the complicated and messy distributions of the real data.

**No User-Defined Specifications or Constraints:** Most of the existing machine learning approaches suffered from drawbacks as they required user-specification of rules and constraints. With the proposed WCGAN-GP framework, there is no need to input any set of rules or relationships, the model is trained to automatically learn the characteristics of real data and generate the synthetic samples that look similar to the real samples.

### 3.6.2 Limitations

In this section, the limitations of the Design and Methodology will be discussed.

**Computationally Expensive:** The major limitation with the GANs is their tendency to take a lot of time to train the model. The training time increases with the size and dimension of the real data.

**No Reliable Metric of Evaluation during Training:** The generator and critic loss during training is not a useful metric to evaluate the quality of synthetic data generated by WCGAN-GP model. The initial trend of the losses and the generated synthetic data does not guarantee any signs of progress. After the training is completed,
Visualisations and Machine learning performance is used to assess the performance improvement for any hyperparameter tuning that is done. Therefore, a lack of metric to evaluate the synthetic data during training is a limitation of the research as it becomes difficult to do a model comparison and find the best model in a single run. As a result, the process of hyperparameter tuning also becomes complicated.

**No Comparison with other baselines or state-of-the-art approaches:** In the literature, there are many data generation approaches that have been proposed. There are many variants of GANs that have been developed. The experiment design does not compare the WCGAN-GP model with the other GAN variants or existing state-of-the-art approaches due to time-constraints. With numerous baselines available, the comparison is challenging and difficult, but is an interesting avenue for research.
4. RESULTS, EVALUATION AND DISCUSSION

The chapter focuses on the results and outputs of the different data generation approaches. The generated synthetic data from different approaches is assessed using evaluation methods described in the Section 3.5. Following the assessment and evaluation of results against the research questions, the hypothesis is tested. The chapter concludes with a discussion of the strengths and limitations of research findings along with future enhancements that can be incorporated into the design.

4.1 Implementation Details

The data transformations steps are kept constant to get comparable results. Each categorical variables are label-encoded to convert them into numerical format. Following this, the continuous variables (including the label-encoded categorical variables) are standardized to bring all the variables into the same range. For more details, please see Section 3.3.

4.1.1 WCGAN-GP

The architecture for WCGANs-GP was tested using different parameters and tuning of hyperparameters to optimize the quality of synthetic data generated. Overall, WCGAN-GPs provided a much stable training without any hyperparameter tuning and provided good quality results even with the basic settings. The transformation of data using standardization showed better results than the normalization of features. The learning rate and the number of layers, in particular, seemed to impact the performance significantly higher as compared to the other settings. Based on the difference in results from visual evaluation, and machine learning performance, the parameters were chosen for the final experiment. The chosen parameters provided comparable results when evaluated on different machine learning models where Accuracy, F1 score and ROC values were recorded.

The depth of the generator and critic is set to 3. For generator, the size of nodes in hidden layer are ordered in an ascending size, that is d, d*2, d*4 (where d is 128). The
critic has the same hidden nodes but ordered in a descending size, that is \(d^*4, d^*2, d\) (where again \(d\) is 128).

Leaky ReLU is used as the activation function for each layer except the output layer which uses linear activation function and the batch size is 64 for both generator and critic. The use of dropout in generator is used as it minimizes overfitting and improves the quality of synthetic data. The learning rate is 0.0001 and Adam Optimizer is used to minimize the loss function. The model is trained for 5000 epochs as overtraining had started to deteriorate the quality of synthetic data.

The random noise vector has a length of 32. Once the training of the model was completed, the synthetic data was generated with the exact size as of real data. The code used in this project is available online and can be found here\(^6\). The details of the implemented settings for the final WCGAN-GP model has been outlined in Table 4.1.

Table 4.1. Summary of the Implemented Critic and Generator Model Configurations for WCGAN-GP

<table>
<thead>
<tr>
<th>critic C</th>
<th>generator G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input - Dimension of real data</td>
<td>Input - Random Noise: 32</td>
</tr>
<tr>
<td>512 Leaky RELU (alpha: 0.2)</td>
<td>128 Leaky RELU (alpha: 0.2), Dropout (0.3)</td>
</tr>
<tr>
<td>256 Leaky RELU (alpha: 0.2)</td>
<td>256 Leaky RELU (alpha: 0.2), Dropout (0.3)</td>
</tr>
<tr>
<td>128 Leaky RELU (alpha: 0.2)</td>
<td>512 Leaky RELU (alpha: 0.2), Dropout (0.3)</td>
</tr>
<tr>
<td>Output - 1, Linear activation</td>
<td>Output - Dimension of real data, Linear activation</td>
</tr>
</tbody>
</table>

**Other Parameters:**
Learning rate: 0.0001; Adam Optimizer; Batch size: 64; Epochs: 5000

4.1.2 SMOTE (Baseline)

The implementation of SMOTE is pretty straightforward and is achieved using the Imblearn package. SMOTE does not require any parameter optimizations, so there is no need to build or train any model. Using the approach described in Section 3.4.2, the synthetic dataset is generated.

\(^6\)https://drive.google.com/drive/folders/1LhFulVYPNiciW9yMNqR_NZXj_wXy_Z3M?usp=sharing
There are few identical records found between real and synthetic data and are removed in the final synthetic data for further evaluation and analysis.

4.2 Evaluation

The three resulting synthetic datasets are assessed using our evaluations methods discussed in section 3.5. With these experiments, it is shown that the proposed WCGAN-GP model performs better than the alternative SMOTE in terms of machine learning tasks and protecting privacy. In this section, the performance of the synthetic data from WCGAN-GP model is discussed when compared to those from alternative approaches for each of the evaluation metrics. Only the relevant and crucial results or visuals are presented in the section.

4.2.1 Data Utility Metrics

4.2.1.1 Visual Evaluation

The visual analysis provides insights about the underlying properties of the generated synthetic data and these help in confirming the hypothesis whether the properties within the real data are preserved with the synthetic data using different data synthetiser approaches.

Univariate Analysis

Box-Plots: Starting with the first evaluation, the box plots are observed for the numerical variables of the real and synthetic datasets. For the Credit card dataset, it is clear from the box plots in Figure 4.1 that the numerical samples synthesized from SMOTE and WCGAN-GP have a similar distribution as compared with the real data. Both the approaches have been able to capture the basic properties and have similar range, median, IQR and so on. The synthetic data from WCGAN-GP verifies that the model has successfully generated data that captures the properties for the numerical columns.
However, the original data seems to have few outliers located outside the whiskers of box plot and this characteristic is also reflected in the generated synthetic data. It is also observed that WCGAN-GP model has generated slightly more synthetic samples outside the IQR, that are outliers. To add, there are outliers towards both the extremes and variables such as ‘Age’ contain negative values with WCGAN-GP method. The hypothesis is that the outliers in original data led to updates in weights of neural networks based on the extreme samples and generated more synthetic instances near the neighborhood of outliers. The outlier treatment was not applied in original data as the purpose of the research was to create synthetic data with the same properties, including any presence of outliers. Nonetheless, the generated synthetic data requires outlier treatment to cap the extreme instances and have synthetic data make more logical sense. The similar behavior is continued and was repeated with the other two datasets as well.

![Box-plots](image)

**Figure 4.1.** Box-plots for the numerical variables in Credit Card Dataset. Blue indicates the real data points, orange synthetic from SMOTE and green synthetic from WCGAN-GP
Histories: The frequency distribution of categorical variables using histograms is another visualisation metric to evaluate the synthetic data. Overall, both SMOTE and WCGAN-GPs are able to approximately capture the frequency distributions of categorical variables, with a few exceptions. For instance, the imbalance problem still remains in the synthetic datasets for Credit Card and Adult Census from WCGAN-GP method (See Figure 4.2 and 4.4 for results). The Cardiovascular dataset had an equal distribution of classes and its corresponding synthetic data from WCGAN-GP also reflects the same property. In addition, columns like ‘Gender’ and ‘Active’ in Cardiovascular dataset have matching distributions and convey nearly similar patterns as of real data (see Figure 4.3).

However, it is worth noting that the distributions are also not a perfect replica and do not exactly match with either of the data synthesizer approaches. There are gaps observed between the distributions for some categorical variables. For instance, Figure 4.2 shows that the frequencies for ‘Gender’ and ‘Education’ in Credit Card dataset do differ by a certain amount, when compared against the real data.

Credit card has a mix of numerical and categorical variables, with a good number of numerical variables. However, Adult Census data has more categorical variables than numerical ones and its categorical columns also contain multi-levels. From Figure 4.4, it is observed that WCGAN-GP has a hard time capturing the distributions as compared to SMOTE for categories with more than 3 levels. The presence of many categories in a variable seems to be an obstacle for WCGAN-GP. This can be observed in ‘Workclass’, ‘Marital status’, ‘Relationship’ variables in the Adult Census data.

Though not perfect, both SMOTE and WCGAN-GP performed equally well with Credit card and Cardiovascular datasets. WCGAN-GP method does better in maintaining the class imbalance. But the baseline SMOTE has an advantage over WCGAN-GP when it comes to synthesizing datasets with multiple-category categorical variables.
Figure 4.2. Histogram for categorical variables in Credit Card Dataset. Blue indicates the real data points, orange synthetic from SMOTE and green synthetic from WCGAN-GP.

Figure 4.3. Histogram for categorical variables in Cardiovascular Dataset. Blue indicates the real data points, orange synthetic from SMOTE and green synthetic from WCGAN-GP.
Figure 4.4. Histogram for categorical variables in Adult Census Dataset. Blue indicates the real data points, orange synthetic from SMOTE and green synthetic from WCGAN-GP

**Multivariate Analysis**

**Scatterplots:** Figure 4.5, 4.6 and 4.7 present the scatterplot graphs between the numerical variables for each synthesizer and original dataset. It can be seen from the graphs that the generated data using SMOTE and WCGAN-GP seems to establish and maintain the relationships between the variables. This pattern is repeated in all the three datasets.

It can also be inferred that the WCGAN-GP has not suffered the problem of mode collapse, which is a common training problem. The samples produced are diverse enough and the model is able to learn and reproduce the distributions of the real-world data.

The assessment of synthetic data in this section is qualitative. And as noticed from the box-plots, there are slightly more synthetic points towards the extremes from WCGAN-GP method. As a result, SMOTE again has a slight advantage over WCGAN-GP. But it is important to note that WCGAN-GP has still shown a great capability to reproduce almost similar patterns.
Figure 4.5. Scatterplot for the variables in Credit Card Dataset. Blue indicates the fraud data points, red indicates normal records.

Figure 4.6. Scatterplot for the variables in Cardiovascular Dataset. Blue indicates the ‘Have CV disease’ patients, red indicates ‘Do not have CV disease’ patients.
Figure 4.7. Histogram for the categorical variables in Adult Census Dataset. Blue indicates records with income >50K, red indicates records with income <=50K.

**Correlation Matrix:** The results of correlation matrix between the columns of each dataset are presented in the Figure 4.8, 4.9 and 4.10. It is observed that the column correlations in synthetic datasets are almost similar to the original data correlations. There is not a big difference in the correlation values of synthetic and real data. The same trend is reproduced and repeated in all the three datasets.

Figure 4.8. Correlation Matrix Evaluated using the Real data and synthetically generated datasets on the Credit Card dataset

Figure 4.9. Correlation Matrix Evaluated using the Real data and synthetically generated datasets on the Cardiovascular dataset
Figure 4.10. Correlation Matrix Evaluated using the Real data and synthetically generated datasets on the Adult Census dataset

### 4.2.2 Similarity of Data

Table 4.2 represents the Pearson’s correlation coefficient and Spearman’s correlation coefficient between the real and synthetic dataset for each of the columns of Credit Card data. These two indicators are calculated to find out whether the variables in the real data and synthetic data are correlated or not. The objective is to obtain minimal or zero values for the correlation.

Table 4.2 shows that the coefficients for Pearson and Spearman are close to zero. This indicates that the two datasets are not correlated and the synthetic data is not directly correlated to the original data. Although, this is a simple metric to compute and correlations would be naturally expected to come around zero, this assessment is necessary to strike out any possibility of any direct relation or link between the real and synthetic datasets. Based on the results so far, it can be inferred that the synthetic datasets do not have any correlation with the real data but they preserve the relationship and statistical properties of the real data. This same pattern was repeated for both Cardiovascular and Adult Census datasets.

### 4.2.3 Machine Learning Performance (Model-compatibility)

To validate the quality of the synthetic data by WCGAN-GP and SMOTE, the predictive performance of synthetic data is compared against the real data. There are three different training-testing settings performed. Setting A: train the predictive models on the real training data, test the performance of the trained model on the real test set. Setting B: For the synthetic data generated by SMOTE, train on the generated
synthetic train data and test on synthetic test data. Setting C: For the synthetic data generated by WCGAN-GP, train on the generated synthetic train data and test on synthetic test data.

All the three datasets are used for a classification problem. The results show that WCGAN-GP model consistently performs better than SMOTE in the machine learning tasks. Across all the datasets, the Accuracy, F1 scores and ROC AUC of the predictive model built on WCGAN-GP’s synthetic data is comparable to the predictive model built on the real data. On the contrary, the Accuracy, F1 scores and ROC AUC values of the classification model on SMOTE’s synthetic data are far-off from the predictive model on real data. This trend is repeated for almost all of the machine learning algorithms and performance metrics reviewed.

Table 4.2. Pearson Correlation Coefficient and Spearman Correlation Coefficient between variables of real and synthetic datasets on the Credit Card dataset

<table>
<thead>
<tr>
<th>Column</th>
<th>Pearson Correlation Coefficient</th>
<th>Spearman Correlation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SMOTE</td>
<td>WCGAN-GP</td>
</tr>
<tr>
<td>LIMIT_BAL</td>
<td>0.0036</td>
<td>0.0020</td>
</tr>
<tr>
<td>SEX</td>
<td>0.0079</td>
<td>0.0079</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>-0.0002</td>
<td>0.0026</td>
</tr>
<tr>
<td>MARRIAGE</td>
<td>0.0002</td>
<td>0.0005</td>
</tr>
<tr>
<td>AGE</td>
<td>-0.0089</td>
<td>-0.0090</td>
</tr>
<tr>
<td>PAY_0</td>
<td>-0.0011</td>
<td>0.0009</td>
</tr>
<tr>
<td>PAY_2</td>
<td>0.0060</td>
<td>0.0115</td>
</tr>
<tr>
<td>PAY_3</td>
<td>-0.0017</td>
<td>0.0013</td>
</tr>
<tr>
<td>PAY_4</td>
<td>-0.0035</td>
<td>0.0009</td>
</tr>
<tr>
<td>PAY_5</td>
<td>0.0060</td>
<td>0.0111</td>
</tr>
<tr>
<td>PAY_6</td>
<td>0.0041</td>
<td>0.0082</td>
</tr>
<tr>
<td>BILL_AMT1</td>
<td>0.0051</td>
<td>0.0057</td>
</tr>
<tr>
<td>BILL_AMT2</td>
<td>0.0049</td>
<td>0.0078</td>
</tr>
<tr>
<td>BILL_AMT3</td>
<td>0.0014</td>
<td>0.0009</td>
</tr>
<tr>
<td>BILL_AMT4</td>
<td>0.0021</td>
<td>0.0044</td>
</tr>
<tr>
<td>BILL_AMT5</td>
<td>0.0011</td>
<td>0.0074</td>
</tr>
<tr>
<td>BILL_AMT6</td>
<td>-0.0020</td>
<td>0.0035</td>
</tr>
<tr>
<td>PAY_AMT1</td>
<td>0.0010</td>
<td>0.0015</td>
</tr>
<tr>
<td>PAY_AMT2</td>
<td>0.0035</td>
<td>-0.0050</td>
</tr>
<tr>
<td>PAY_AMT3</td>
<td>-0.0034</td>
<td>-0.0049</td>
</tr>
<tr>
<td>PAY_AMT4</td>
<td>0.0005</td>
<td>0.0039</td>
</tr>
<tr>
<td>PAY_AMT5</td>
<td>0.0027</td>
<td>-0.0102</td>
</tr>
<tr>
<td>PAY_AMT6</td>
<td>0.0044</td>
<td>-0.0047</td>
</tr>
<tr>
<td>Class</td>
<td>0.0010</td>
<td>0.0010</td>
</tr>
</tbody>
</table>
As the synthetic data is derived from the original data, the Accuracy, F1 and ROC AUC scores for the synthetic data should ideally be equal to or less than the scores from real data. The scores on the synthetic data should not be higher as the synthetic data is expected to match or experience a slight deterioration in machine learning usability tasks. With the Credit card data results (Table 4.3), the performance scores with WCGAN-GP are approximately equal or slightly less than the scores on real data. However, synthetic data generated with SMOTE shows much higher Accuracy and ROC AUC scores as compared to the real data, indicating a relatively lower quality of synthetic data. The F1 scores with SMOTE’s synthetic data are lower than real data but they are significantly lower, suggesting SMOTE’s average performance in comparison to WCGAN-GP.

Table 4.3. Performance comparison of different predictive models for Real data, Synthetic data from SMOTE, and from WCGAN-GP model on Credit Card dataset

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>F1 Score</th>
<th>ROC AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>SMOTE</td>
<td>GAN</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>73%</td>
<td>91%</td>
<td>72%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>82%</td>
<td>95%</td>
<td>81%</td>
</tr>
<tr>
<td>XgBoost</td>
<td>82%</td>
<td>94%</td>
<td>82%</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>82%</td>
<td>94%</td>
<td>81%</td>
</tr>
<tr>
<td>Linear SVM</td>
<td>67%</td>
<td>94%</td>
<td>67%</td>
</tr>
<tr>
<td>Average</td>
<td>77%</td>
<td>93%</td>
<td>77%</td>
</tr>
</tbody>
</table>

Things get more interesting with the other two datasets. In Cardiovascular data (Table 4.4), the performance with WCGAN-GP is close but looks significantly lower than the real dataset. The hypothesis for this drop in performance with this dataset could be the potential presence of more categorical variables. While credit card had more numerical than categorical variables, the cardiovascular dataset has an equal proportion of categorical columns and it has been noticed earlier that there were gaps in distribution of categorical columns with WCGAN-GP. With SMOTE, the trend is similar as observed with credit card data. To conclude the findings on this dataset, the performance with WCGAN-GP still remains to be better than SMOTE. But the performance scores with WCGAN-GP do not seem to be significantly comparable to original data and suggest an opportunity for an improvement in the quality of synthetic data.
Further, WCGAN-GP is again better than SMOTE on the Adult Census dataset (Table 4.5). But the scores on performance measures are higher (instead of being lower) and the differences between the synthetic and real data remain significant. This opposite trend occurs because of the possibility of dominance of categorical variables, compared to numerical with Adult Census dataset (9 categorical out of 14 features).

In short, WCGAN-GP fares better than SMOTE in classification tasks. With datasets having more numerical columns than categorical, WCGAN-GP performs very good in approximating similar machine learning performance. But with categorical, the performance needs an improvement. To conclude, the results have shown that the performance of synthetic data generated by WCGAN-GP for a classification task is similar to the performance of a real data.

Table 4.4. Performance comparison of different predictive models for Real data, Synthetic data from SMOTE, and from WCGAN-GP model on Cardiovascular dataset

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>F1 Score</th>
<th>ROC AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>SMOTE</td>
<td>GAN</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>63%</td>
<td>77%</td>
<td>57%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>72%</td>
<td>82%</td>
<td>65%</td>
</tr>
<tr>
<td>XgBoost</td>
<td>74%</td>
<td>80%</td>
<td>67%</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>73%</td>
<td>79%</td>
<td>66%</td>
</tr>
<tr>
<td>Linear SVM</td>
<td>52%</td>
<td>57%</td>
<td>50%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>67%</td>
<td>75%</td>
<td>61%</td>
</tr>
</tbody>
</table>

Table 4.5. Performance comparison of different predictive models for Real data, Synthetic data from SMOTE, and from WCGAN-GP model on Adult Census dataset

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>F1 Score</th>
<th>ROC AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>SMOTE</td>
<td>GAN</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>76%</td>
<td>92%</td>
<td>77%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>80%</td>
<td>95%</td>
<td>85%</td>
</tr>
<tr>
<td>XgBoost</td>
<td>80%</td>
<td>94%</td>
<td>85%</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>81%</td>
<td>94%</td>
<td>85%</td>
</tr>
<tr>
<td>Linear SVM</td>
<td>67%</td>
<td>93%</td>
<td>76%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>77%</td>
<td>94%</td>
<td>81%</td>
</tr>
</tbody>
</table>
4.2.2 Privacy Metrics

Euclidean Distance to Nearest Record

Euclidean distance is used as the privacy metric to calculate the mean and standard deviation to the closest record from the synthetic data to real data. To comply with privacy requirements, it is required to have a large mean and small standard deviation. A synthetic record with a zero mean distance implies that there is a leakage of the individual information. Additionally, it makes sense to do the comparisons only within the dataset because of the manner in which the measure is calculated.

It is observed from the Table 4.6 that synthetic data using WCGAN-GP consistently has a higher Euclidean distance as compared to data generated using SMOTE. The findings from this metric confirms the hypothesis that synthetic data using WCGAN-GP performs better at protecting the individual’s privacy and can be used to provide better privacy protection.

Table 4.6. Euclidean Distance to The Nearest Record, the value indicates the distances (mean, standard deviation) for all samples in synthetic dataset and their most similar sample in the real data

<table>
<thead>
<tr>
<th></th>
<th>SMOTE</th>
<th>WCGAN-GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Card</td>
<td>1.18 ± 0.95</td>
<td>3.07 ± 1.24</td>
</tr>
<tr>
<td>Cardiovascular</td>
<td>0.37 ± 0.38</td>
<td>0.81 ± 1.37</td>
</tr>
<tr>
<td>Adult Census</td>
<td>1.45 ± 0.57</td>
<td>2.59 ± 0.50</td>
</tr>
</tbody>
</table>

Duplicate Records

We did not see any duplicate records between the real and synthetic datasets generated using WCGAN-GP. This indicates that the GAN approach did not suffer from mode-collapse. Additionally, it shows that there is no direct link between the real and synthetic datasets and this aspect of privacy is also satisfied.
Further, it was expected to find duplicates in synthetic data generated using SMOTE and it was proposed to remove them while designing the experiment. But it is worth noting that there were around 0.1% identical matches produced with the SMOTE approach. While the percentage of match rate is very low, but a re-identification of even a single individual can create high legal and confidentiality issues. Hence, it makes sense to remove synthetic samples generated by SMOTE that have direct matches with real data. In our evaluations, only the revised synthetic dataset after removing all the matched records is considered. This is a valuable privacy metric to strike out possibility of copied records in synthetic data.

4.3 Discussion

4.3.1 Observation From Results

A number of methods were employed to assess the quality of generated synthetic data. As a first metric, the qualitative evaluation was done using visualisations to inspect the basic statistical properties, distributions, and relationships in synthetic data. Both the synthetization approaches produce high-quality synthetic data as the visualisations indicate that the synthetic data nearly matches with the real data, with a few differences. Looking at the favourable findings, the box-plots for the numerical variables shows that the synthetic data by both WCGAN-GP and SMOTE has identical distributions with the real data, particularly for median, quartiles (interquartile range), minimum and maximum values. Unlike SMOTE, WCGAN-GP shows signs of producing some outliers (or unusual data points) on both the extremes. The reason for this might be presence of outliers in the original data that biased the weights of neural networks to produce more extreme samples. In this research, the outliers were intentionally not removed or treated as the focus was to generate a synthetic data that had similar properties even in terms of containing the outliers. Thus, there is a scope of improvement identified with the pre-processing techniques when using GANs. Nonetheless, it is recommended from the findings that the synthetic data should undergo Synthetic sample treatment to handle the Out of Range values, indicating that GANs do require some minor manual work to improve the quality of synthetic samples. But this seems to be valuable and a necessary step.
Furthermore, the histograms for categorical variables in synthetic data reveal that both SMOTE and WCGAN-GP have the ability to reproduce the categorical data pretty well. But SMOTE does better than WCGAN-GP with respect to synthesizing categorical variables and its synthetic data has more similar frequency distributions as with the real data. The class imbalance is better preserved with WCGAN-GP, which is due to the fact that conditional vector was added to the input of generator and critic during the WCGAN-GP training. The WCGAN-GP successfully handles the imbalanced data when generating synthetic data. Further, when there are more categorical and less continuous variables in a dataset (such as Adult Census that has 9 categorical out of 14 features), the performance of synthetic data with the proposed WCGAN-GP struggled especially with multiple-category variables. Generally, the synthetic data for categorical variables with more than 3 levels had a slightly different frequency distribution as compared to the real data.

One of the reasons for challenges with categorical data generation in this research could be the use of linear activation function in the generator’s output layer. With continuous variables, the linear activation function works well to model the distributions, but with categorical, a corresponding SoftMax function in the generator’s output with a dimensionality equal to number of levels within a category might yield better results. SoftMax function is a generalized logistic activation function that is popularly used for multiclass classification and could be the solution to problems of synthesizing categorical data.

Although few gaps are identified with synthetic data’s quality through the univariate analysis, the bivariate analysis demonstrates high success in preserving and reproducing the same patterns and underlying distributions of the real dataset. It is observed from the scatterplots and correlation matrix that SMOTE and WCGAN-GP generated samples that preserved the relationships between variables. Nevertheless, if a winner had to be chosen out of these two methods for the qualitative assessment so far, SMOTE would surpass WCGAN-GP by a small margin due to reasons discussed already.

Further, the assessment of synthetic data for data utility, that is its usability for machine learning tasks, was conducted using different classification models.
Identifying the best classifier for the datasets was not the aim of this research but the objective was to compare if the evaluation metrics achieved from a predictive model (trained and tested) using synthetic data are similar to the evaluation metrics achieved from a predictive model (trained and tested) using a real dataset. Since two of the datasets used in the research dealt with a class-imbalanced problem, evaluation metrics (F1 score, ROC AUC) other than accuracy were used to get more insights and fully understand the performance of synthetic data. In contrast to findings with the SMOTE method, it is observed that the model performance of synthetic data from WCGAN-GP is more comparable to the model performance on real data. As WCGAN-GP showed promising results with distributions and correlations matching closely with the real data and was better than SMOTE in synthesizing the imbalanced columns, the predictive model using synthetic data from WCGAN-GP turned out to be more reflective and similar to the model built on real data. This shows that even though the synthetic data from SMOTE had a more close match with the real data in terms of distributions, the class variable plays a crucial role in such machine learning models. The results confirm that the synthetic data using WCGAN-GP can facilitate machine learning and predictive modelling tasks without the need of a real data.

Finally, the assessment from a privacy perspective revealed that WCGAN-GP was better than the SMOTE approach in creating synthetic samples that are better at minimizing the disclosure risks. Also, there were no duplicate records found between the synthetic data from WCGAN-GP and the original data. Therefore, as the WCGAN-GP framework has been successful in providing a high quality synthetic data in terms of data utility and risk of re-identification, the hypothesis is accepted.

4.3.2 Strengths of Findings

In this section, the strengths of the findings are discussed in brief.

Proof-of-concept for Tabular (Structured) Synthetic Data Generation Based on WCGAN-GP: The primary strength of the findings is that they provide a comprehensive proof-of-concept for generation a tabular or structure synthetic data. As a result of the qualitative and quantitative assessment of the quality of generated synthetic data, it is demonstrated that an improved variant of GAN – WCGAN-GP can
yield a high quality structured data that captures the complex distributions, patterns and relationships of the original data.

**High Data-Utility and Disclosure Risk Protection:** The trade-off between data utility and privacy is often difficult to achieve and the synthetic data has to strike a balance between these two concepts. The expectation out of synthetic data is to provide high data-utility, that is to achieve a comparable machine learning performance with the original dataset. At the same time, the synthetic data should provide minimum risks of re-identification and solve the problem of private data-release. The results indicate that the generated synthetic data using WCGAN-GP framework behaves nearly similar in terms of machine learning tasks in some cases (Credit Card, Cardiovascular disease). In addition, the mean of Euclidean distance between the records of synthetic data and their nearest record in real data is more than 0, indicating that the synthetic data provides a level of privacy protection. Lastly, the number of identical records between the synthetic data from WCGAN-GP and real data is zero, implying that the records in the synthetic data do not have any match in the real data. The findings from this research do indicate that GANs would provide better disclosure protection than existing approaches.

**Reasonable Success with Mixed-types datasets:** Another key strengths in this research is attaining reasonable success with WCGAN-GP to synthesize the common data types, that is continuous and categorical variables.

**Success with imbalanced Datasets:** The results of WCGAN-GP have demonstrated success in maintaining the imbalanced distribution in the synthetic data. It has also shown the capability of WCGAN-GP in capturing the patterns and distributions for even the minority classes and without any biasedness towards the majority class samples.

**Better Performance than the baseline SMOTE:** While SMOTE is a standard package and provides good results without any requirement of hyperparameter optimisation, the WCGAN-GP was better at producing synthetic samples with a similar machine learning efficacy as the real data and at providing lower disclosure risks. As data privacy is going to be the main reason for a synthetic data release, GANs
like WCGAN-GP are naturally bound to become the recommended choice for synthetic data generation problems.

**Improved Stability and Modelling Performance:** The findings in this research have confirmed that WCGAN-GP does not experience any signs of mode collapse and provides a stable GAN training.

### 4.3.3 Weaknesses of Findings

In this section, the weakness of the findings will now be discussed.

**Necessity for Synthetic Sample Treatment:** Even though, GANs have an advantage to automatically learn and generate synthetic samples from real data with minimal human interaction. But it has been observed that the generated synthetic data can contain values that might not make sense and are out of range. Hence, there arises a need to have a domain knowledge to understand the bounds required to be set for the variables and ensure that the data values makes logical sense. The synthetic samples are suggested to undergo treatment to handle the out of range values and improve the quality of synthetic data.

**Require more Exploration of Data-Transformation Techniques:** While the synthetic data needs to undergo a sample treatment after it is generated, data transformations like log-transform on the real data can be explored as a pre-processing technique to tackle the skewed data and outliers before the training of GAN model. The use of such transformations might help in reducing the presence of out of range values in the synthetic data.

**Longer Training Time than SMOTE:** The experiments in this research has showed that SMOTE is much faster to run as compared to GANs. On an average, the GANs took around 30 minutes to one hour to train on the datasets and SMOTE just took around a minute to run. Thus, if there are time-constraints and no time available for training the GAN model, it becomes difficult to use GANs for synthetic data generation.
**Challenge with generating Categorical data having multiple levels:** The findings showed that the datasets dominated with categorical variables (Adult Census) had a slightly deteriorated performance as compared to the other datasets (Credit Card). This does indicate that the generation of categorical data is challenging using WCGAN-GP. The use of different activation layers like softmax on one-hot encoded categorical data will be an interesting thing to explore further. The use of label encoding has worked well to provide satisfactory results, but there is definitely an opportunity to improve the quality of categorical synthetic data.

**Limited hyperparameter Tuning:** Multiple iterations were performed to optimize the performance of WCGAN-GP model. But due to time constraints for the thesis and slow training process of GANs, it was difficult to train the model with all the possible hyperparameters. An inclusion of grid search to find a better combination of hyperparameter settings can be a fruitful area of future work.
5. CONCLUSION

This chapter provides an overview of the study conducted. It summarizes the findings achieved with respect to the research questions that were defined at the beginning of the research. The outcome of the study is summarised and its contribution to the field of synthetic data generation is assessed. The chapter concludes by highlighting the potential areas for future work and improvements.

5.1 Research Overview

The main objective of this research was to investigate the field of synthetic data generation and evaluate whether a WCGAN-GP framework can be used to generate high-quality synthetic data that demonstrates high usability and minimizes privacy risks. WCGAN-GP was originally proposed to provide a stronger modelling performance on unstructured datasets (like image and language). This research is a contribution of WCGAN-GP towards generating a structured dataset (that is tabular dataset with mixed data types).

Multiple datasets belonging to different domains and mixed data types were selected for this research. In order to reach the research objective, a WCGAN-GP model was designed and trained on the real dataset and the performance of the generated synthetic data was compared with the real dataset on the same evaluation metrics. The proposed GAN framework was compared and evaluated against a simple baseline generator SMOTE to identify the advantages or disadvantages of GANs for synthetic data generation.

The results showed that WCGAN-GP offers a promising framework to generate continuous and categorical data as the synthetic data showed preservation of patterns, distributions and relationships of the real dataset. For qualitative assessments like scatterplots and correlation matrix, WCGAN-GP showed comparable results with the SMOTE approach. The machine learning performance for synthetic data from SMOTE was far off from the performance of the real data, however, the synthetic data from WCGAN-GP showed comparable performance with real data in the classification
tasks. Thus, WCGAN-GP performed better in terms of machine learning performance than the baseline SMOTE. Furthermore, the synthetic data from WCGAN-GP also offered better privacy protection than SMOTE as the mean Euclidean distance was higher for the synthetic samples generated by WCGAN-GP. The synthetic samples from WCGAN-GP also did not have any identical matches with the real data.

Therefore the alternate hypothesis was accepted that if WCGAN-GP is used to create synthetic samples from real samples, then it will show better results than the baseline SMOTE and a synthetic data of higher quality with better privacy protection will be produced.

5.2 Problem Definition

Most of the real-world datasets contain sensitive or personal information and hence, the data cannot be shared with researchers or public to support innovations or research. Data dissemination has become challenging due to increased privacy and security requirements. For these reasons, there has been a significant rise in the need for synthetic data and how the synthetic data can be a game-changer to accelerate data-driven innovations. Synthetic data has the potential to keep the data private and at the same time maintain the usefulness of the real data.

Over the years, there have been many state-of-the-art approaches that were proposed to generate synthetic data. The traditional approaches included machine learning methods to either learn from handcrafted distributions or directly from the real data. Before the rise of deep learning, the approaches that attempted to generate synthetic datasets suffered from drawbacks and challenges. These early methods relied largely on human intervention and manual curation as they required manual input of user-specified rules and constraints. While these initial approaches maintained valuable information and patterns of the real data, but they had an adverse impact on the privacy of real data. This is because data utility and privacy are inversely related and it is difficult to achieve both at the same time.

Since its inception in 2014, GANs within deep learning have been very successful in generating sharp and realistic images, but have seen little application with tabular data
generation. Their capability to replicate the complex patterns of real data is much sought after. As this field of deep generative model is quite novel, it is an active area of research. However, GANs suffer from common training problems that ends up producing poor quality samples. There are variants of GANs like WGAN or WGAN-GP that are introduced to overcome these limitations. WCGAN-GP is a simple extension of WGAN-GP that takes the class labels into consideration during the training of models and overcomes any biases that could arise due to target classes. The framework has been tested on image generation tasks but hasn’t been tested on tabular datasets. In order to solve the issues in the existing literature, a WCGAN-GP is designed, trained, and tested to generate synthetic data.

The research problem was defined by the main research question: “To what extent can a generated synthetic data approximate the quality of a real data using Wasserstein Conditional GANs with Gradient Penalty (WCGAN-GP) using a combination of practical utility and privacy metric?”

The main research question was further divided into multiple sub-questions to answer the bigger question. The answers to the sub-questions and the main research question will be discussed in the next section.

5.3 Addressing the Research Questions

For WCGAN-GP model, multiple experiments were performed with different parameter settings and the model was fine-tuned to observe improvement in the results. Similarly, experiment was run with SMOTE to generate synthetic data for comparison purposes. Once the synthetic data is generated, the quality and performance of the approaches were assessed using different evaluation metrics to measure the data usability and privacy risks. The synthetic data generated from WCGAN-GP was compared against the real data and then against the synthetic data from SMOTE to answer the research questions and test against the hypothesis presented in Section 1.3.

With the findings observed through the experiments, the research sub-questions defined in Section 1.2 can now be answered clearly:
Research Sub-Question: Can a WCGAN-GP model be trained to learn the distributions and relationships of the original data with a high degree of accuracy and generate synthetic data that is indistinguishable from real data?

Answer: Based on the findings from Section 4.2.1, it can be observed that WCGAN-GP does show a strong ability to preserve the statistical properties of the real data. For the numerical variables, the distributions or spread are almost similar to the real data, with the exception of few extra outliers (or out of range values). The impact of the out of range values can be minimized by applying outlier treatment methods.

For the categorical variables, the frequency distributions are aligned with the distributions in real data but with categorical variables having more than 3 levels, there is a gap in the frequency distributions for some values.

WCGAN-GP yields great results in terms of preserving relationships and that is evident from the scatterplots and correlation matrices that almost look similar to the real data. Overall, the machine learning effectiveness provides satisfactory results but the results vary for different datasets:

- For Credit Card data, the performance metrics achieved for the predictive models (trained and tested) using synthetic data are almost similar to the metrics for the model (trained and tested) using the real data.
- For Cardiovascular data, the performance metrics using synthetic data are nearly similar to those achieved from real data but there is a difference of 5%.
- Unlike other datasets, Adult Census performs the worse for machine learning tasks. The difference in evaluation metrics is not only higher but the magnitude of the difference is also larger than other datasets.

Research Sub-Question: Is there a difference in the quality of synthetic data generated using WCGAN-GP based on the types of variables (numerical or categorical) in datasets?

Answer: The quality of synthetic data for data usability was observed to be impacted by the types of variables in datasets. Synthetic data for Credit Card dataset, which had
more numerical variables than categorical, provided comparable machine learning performance with the real data. This indicates the high usability of synthetic data.

However, the synthetic data for Adult Census dataset, which had more categorical columns than numerical, showed a significant difference in the machine learning performance when compared with the real data. This indicates a scope of improvement in the quality of synthetic data. The reason for this deteriorated performance with categorical variables might not have to do with the WCGAN-GP framework but it could be driven by certain architectural changes (like softmax for categorical values) required by neural networks to handle the categorical variables. As there is a difference in the quality of different mixed-type datasets, this question is answered but still remains an area for future research and open for further exploration.

**Research Sub-Question**: To what extent does the synthetic data approach using WCGAN-GP performs better or worse as compared to a baseline approach using SMOTE?

**Answer**: Both SMOTE and WCGAN-GP gave competitive results, but WCGAN-GP performs better than SMOTE due to its superior performance on machine learning tasks and privacy metrics. Both data utility and privacy are crucial metrics for this research to solve the data accessibility and release issues. The synthetic data generated should be able to provide a high utility as well as better privacy protection and WCGAN-GP outperforms SMOTE in both these metrics.

**Main Research Question**: To what extent can a generated synthetic data approximate the quality of a real data using Wasserstein Conditional GANs with Gradient Penalty (WCGAN-GP) using a combination of practical utility and privacy metric?

**Answer**: Based on the findings from Section 4.2, the synthetic data using WCGAN-GP provides promising results in approximating the quality of real data. While the results are not remarkable with datasets that have multiple categorical variables, the assessment that showed that the results are satisfactory and within the acceptable limits. With datasets like credit card, the optimal performance of WCGAN-GP is
achieved. Thus, the assessment shows that the synthetic data is able to maintain the statistical properties of the real data and behave similar as the real data.

Finally, the assessment from a privacy perspective revealed that WCGAN-GP was better than the SMOTE approach in creating synthetic samples that are better at minimizing the disclosure risks. Also, there were no duplicate records found between the synthetic data from WCGAN-GP and the original data. Therefore, as the WCGAN-GP framework has been successful in providing a high quality synthetic data in terms of data utility and risk of re-identification, the hypothesis is accepted.

5.4 Contributions and Impact

The investigation in this research has shown that generative models like GANs can be useful for the generation of tabular or structured data. GANs have had a huge success with image and language datasets, but the success of WCGAN-GP over SMOTE to generate a synthetic dataset of the same size as the real dataset with mixed data types contributes literature for future research. As WCGAN-GP does not require any user-defined specifications and human interaction to learn from the real data, it is better than most of the machine learning or statistical modelling approaches that rely heavily on user interaction.

Further, GANs are often termed to be extremely hard to train and there are drawbacks related to its training. The innovation of this work is that it shows that WCGAN-GP framework provides a strong modelling performance and stable training on structured or tabular datasets. This framework does not show any signs of mode collapse, indicating that this extension of GANs works well with the tabular data. Additionally, the WCGAN-GP model is robust and not heavily sensitive to hyperparameter tuning as it provided good results without any tuning. Nonetheless, fine-tuning is desirable with WCGAN-GP to improve the performance of synthetic data.

Synthetic data is a powerful tool if the data is limited or there are barriers to data sharing due to privacy or security requirements. The results in this research indicate a promising growth in the direction of tabular data generation. With the proposed
WCGAN-GP framework, the generated synthetic data can successfully replace the real data and get rid of the privacy bottleneck. When data privacy is a requirement, the usage of synthetic tabular data by GANs can be recommended and suggested for industry applications like healthcare, finance, insurance and so on.

5.5 Future Work & recommendations

GANs are an active area of research and its new variants are frequently developed to overcome the limitations of existing GANs and offer improvements. An interesting area of future research could be to experiment with different GAN variations to identify a better framework that shows improvement in the preservation of patterns in synthetic data for different data types. Additionally, GANs can be combined with privacy-preserving mechanisms like differential privacy to provide more formal and stronger guarantees of privacy with the synthetic data. At the same time, it will also be important to ensure that the data-utility continues to remain high and a balance between data-utility and privacy remains accomplished.

The generation of categorical data is tricky and challenging with GANs. The use of label encoding provides satisfactory results, but there is a scope of further improvement in the quality of categorical synthetic data. The use of softmax or gumbel softmax activation on one-hot encoded categorical variables and linear activation for numerical variables in the output layer of generator can be explored as a future area of research to improve the quality of categorical data.

The real data can have many usages and applications beyond the machine learning classification tasks. Therefore, the real-world datasets dealing with other supervised learning tasks like regression and unsupervised learning tasks like clustering and segmentation can be further explored. This could provide more evidence to showcase the generalizability of GANs for synthetic data generation and high usability of synthetic data in many machine learning tasks.

Another future recommendation is to explore other data transformation techniques like log-transform or box cox transform to handle skewed real datasets and assess if it helps
to generate fewer out of range values in the synthetic data and provide better utility results.

The future work can also involve more changes in hyperparameters by using exhaustive grid-search to improve the quality of synthetic data.
BIBLIOGRAPHY


### APPENDIX A

*Table A.1. Data description - Default of Credit Card Clients*

<table>
<thead>
<tr>
<th>Variables</th>
<th>Measurement Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Interval</td>
<td>ID of each customer</td>
</tr>
<tr>
<td>LIMIT_BAL</td>
<td>Ratio</td>
<td>Amount of credit in New Taiwan (NT) dollars (includes individual &amp; supplementary credit)</td>
</tr>
<tr>
<td>SEX</td>
<td>Nominal</td>
<td>Gender</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>Ordinal</td>
<td>Level of Education</td>
</tr>
<tr>
<td>MARRIAGE</td>
<td>Nominal</td>
<td>Marital status</td>
</tr>
<tr>
<td>AGE</td>
<td>Ratio</td>
<td>Age in years</td>
</tr>
<tr>
<td>PAY_0</td>
<td>Nominal</td>
<td>Repayment status in September, 2005 (-1=paid duly, 1= delay for one month, 2= delay for two months,..9= delay for nine months and above)</td>
</tr>
<tr>
<td>PAY_2</td>
<td>Nominal</td>
<td>Repayment status in August, 2005</td>
</tr>
<tr>
<td>PAY_3</td>
<td>Nominal</td>
<td>Repayment status in July, 2005</td>
</tr>
<tr>
<td>PAY_4</td>
<td>Nominal</td>
<td>Repayment status in June, 2005</td>
</tr>
<tr>
<td>PAY_5</td>
<td>Nominal</td>
<td>Repayment status in May, 2005</td>
</tr>
<tr>
<td>PAY_6</td>
<td>Nominal</td>
<td>Repayment status in April, 2005</td>
</tr>
<tr>
<td>BILL_AMT1</td>
<td>Ratio</td>
<td>Amount of bill statement in September, 2005</td>
</tr>
<tr>
<td>BILL_AMT2</td>
<td>Ratio</td>
<td>Amount of bill statement in August, 2005</td>
</tr>
<tr>
<td>BILL_AMT3</td>
<td>Ratio</td>
<td>Amount of bill statement in July, 2005</td>
</tr>
<tr>
<td>BILL_AMT4</td>
<td>Ratio</td>
<td>Amount of bill statement in June, 2005</td>
</tr>
<tr>
<td>BILL_AMT5</td>
<td>Ratio</td>
<td>Amount of bill statement in May, 2005</td>
</tr>
<tr>
<td>BILL_AMT6</td>
<td>Ratio</td>
<td>Amount of bill statement in April, 2005</td>
</tr>
<tr>
<td>PAY_AMT1</td>
<td>Ratio</td>
<td>Amount of previous payment in September, 2005</td>
</tr>
<tr>
<td>PAY_AMT2</td>
<td>Ratio</td>
<td>Amount of previous payment in August, 2005</td>
</tr>
<tr>
<td>PAY_AMT3</td>
<td>Ratio</td>
<td>Amount of previous payment in July, 2005</td>
</tr>
<tr>
<td>PAY_AMT4</td>
<td>Ratio</td>
<td>Amount of previous payment in June, 2005</td>
</tr>
<tr>
<td>PAY_AMT5</td>
<td>Ratio</td>
<td>Amount of previous payment in May, 2005</td>
</tr>
<tr>
<td>PAY_AMT6</td>
<td>Ratio</td>
<td>Amount of previous payment in April, 2005</td>
</tr>
<tr>
<td>default.payment.next.month</td>
<td>Binary</td>
<td>Default payment (1=yes, 0=no)</td>
</tr>
</tbody>
</table>
Table A. 2. Data description - Cardiovascular Disease

<table>
<thead>
<tr>
<th>Variables</th>
<th>Measurement Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>Ratio</td>
<td>Age</td>
</tr>
<tr>
<td>height</td>
<td>Interval</td>
<td>Height</td>
</tr>
<tr>
<td>weight</td>
<td>Ratio</td>
<td>Weight</td>
</tr>
<tr>
<td>gender</td>
<td>Nominal</td>
<td>Gender</td>
</tr>
<tr>
<td>ap_hi</td>
<td>Interval</td>
<td>Systolic blood pressure</td>
</tr>
<tr>
<td>ap_lo</td>
<td>Interval</td>
<td>Diastolic blood pressure</td>
</tr>
<tr>
<td>cholesterol</td>
<td>Nominal</td>
<td>Cholesterol</td>
</tr>
<tr>
<td>gluc</td>
<td>Nominal</td>
<td>Glucose</td>
</tr>
<tr>
<td>smoke</td>
<td>Binary</td>
<td>Smoking</td>
</tr>
<tr>
<td>alco</td>
<td>Binary</td>
<td>Alcohol intake</td>
</tr>
<tr>
<td>active</td>
<td>Binary</td>
<td>Physical activity</td>
</tr>
<tr>
<td>cardio</td>
<td>Binary</td>
<td>Has cardiovascular disease or not</td>
</tr>
</tbody>
</table>

Table A. 3. Data description - Adult Census

<table>
<thead>
<tr>
<th>Variables</th>
<th>Measurement Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>Numerical</td>
<td>Age</td>
</tr>
<tr>
<td>workclass</td>
<td>Categorical</td>
<td>Type of employment</td>
</tr>
<tr>
<td>fnlwgt</td>
<td>Numerical</td>
<td>Sampling weight</td>
</tr>
<tr>
<td>education</td>
<td>Categorical</td>
<td>Level of education</td>
</tr>
<tr>
<td>education.num</td>
<td>Numerical</td>
<td>Numeric representation of education</td>
</tr>
<tr>
<td>marital.status</td>
<td>Categorical</td>
<td>Marital status</td>
</tr>
<tr>
<td>occupation</td>
<td>Categorical</td>
<td>Occupation</td>
</tr>
<tr>
<td>relationship</td>
<td>Categorical</td>
<td>Relationship status</td>
</tr>
<tr>
<td>race</td>
<td>Categorical</td>
<td>Ethnicity</td>
</tr>
<tr>
<td>sex</td>
<td>Categorical</td>
<td>Gender</td>
</tr>
<tr>
<td>capital.gain</td>
<td>Numerical</td>
<td>Income from investment sources, apart from wages/salary</td>
</tr>
<tr>
<td>capital.loss</td>
<td>Numerical</td>
<td>Losses from investment sources, apart from wages/salary</td>
</tr>
<tr>
<td>hours.per.week</td>
<td>Numerical</td>
<td>Worked hours per week</td>
</tr>
<tr>
<td>native.country</td>
<td>Categorical</td>
<td>Native country</td>
</tr>
<tr>
<td>income</td>
<td>Binary</td>
<td>Income &lt;=50k, or &gt; 50k</td>
</tr>
</tbody>
</table>