Title

Adversarial Privacy Auditing of Synthetically Generated Data produced by Large Language Models using the TAPAS Toolbox

Author

Dave, Krishna

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Adversarial Privacy Auditing of Synthetically Generated Data produced by Large Language Models using the TAPAS Toolbox

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics and Data Science

by

Krishna Dave

2024

ABSTRACT OF THE THESIS

Adversarial Privacy Auditing of
Synthetically Generated Data produced by Large Language Models
using the TAPAS Toolbox

by

Krishna Dave
Master of Applied Statistics and Data Science
University of California, Los Angeles, 2024
Professor Guang Cheng, Chair

In today's world with ever increasing need for data collection, there is a rise in demand for privacy-preserving synthetic data generation and privacy auditing techniques to safeguard sensitive user information and data from privacy attacks. This paper explores the adversarial privacy auditing of synthetically generated data produced by Large Language Models (LLMs) using the TAPAS "Toolbox for Adversarial Privacy Auditing of Synthetic Data" framework. This paper uses a healthcare dataset with sensitive user information of Breast Cancer to evaluate the privacy of the data using adversarial techniques. The paper compares and contrasts the data quality, data distributions and privacy-preserving metrics of the real dataset with synthetically generated datasets from several sources including LLMs such as the GReaT framework and OpenAI's GPT4, Generative Adversarial Networks (GANs), and an AI-generated dataset produced using a proprietary technique from an industry startup, mostly.ai.

The thesis of Krishna Dave is approved.

Arash A. Amini

Nicolas Christou

Chad J. Hazlett

Guang Cheng, Committee Chair

University of California, Los Angeles

2024

To my sister, family and friends . . .

who have always been by my side

and supported me in my academic journey

from elementary school to higher education.

TABLE OF CONTENTS

1 Introduction **1**

2 Differential Privacy **4**

2.1 Background on Privacy Attacks 4

2.2 Formalizing Differential Privacy 6

2.3 Privacy Budget and Epsilon 7

2.4 Utility-Precision Trade-off in Data Release Mechanisms 7

2.5 Composition Theorem 8

2.6 Applications of Differential Privacy 9

3 Synthetic Data Generation **10**

3.1 Synthetic Data Generation Methodology 10

3.2 Tabular Synthetic Dataset 11

3.3 Overview of Statistical Techniques in Synthetic Data Generation 12

3.4 Synthetic Data Generation Models 13

3.4.1 Generative Adversarial Networks (GANs) 13

3.4.2 Market Research on Techniques for AI-generated Synthetic Data in Industry 15

3.4.3 Using LLMs to Generate Tabular Synthetic Data 16

3.5 Limitations and Challenges with Synthetic Data 17

4 Large Language Models **18**

4.1 Introduction to Large Language Models 18

4.2 GReaT Framework 20
4.3 OpenAI's GPT4 to Generate Synthetic Data 21

5 Privacy Auditing with the TAPAS Toolbox 24
5.1 Threat Modeling Framework 24
5.2 Data and Generator Knowledge of the Attacker 24
5.3 Goals and Intentions of the Attacker 26
5.4 Library of Attacks in the TAPAS Toolbox 27
5.5 Evaluation Metrics for the Attacks 28

6 Experiments 32
6.1 Breast Cancer Data 32
6.2 About the Data 33
6.3 Exploratory Data Analysis 34
6.4 Experiment Design 36
6.4.1 Generators 36
6.4.2 Experimental Attack 36
6.5 Results 38

7 Summary of the Results 44

8 Conclusion 47
8.1 Conclusion and Future Work 47

9 Appendices 49

References 68

LIST OF FIGURES

6.1 Comparing different generators on random targets from real Cancer dataset. . . 38
6.2 Random vs. outlier targets from real Cancer dataset. 39
6.3 Comparing different generators on random targets from BeGReaT Cancer dataset. 40
6.4 Random vs. outlier targets from BeGReaT Cancer dataset. 41
6.5 Comparing different generators on random targets from GPT4 Cancer dataset. . 41
6.6 Random vs. outlier targets from GPT4 Cancer dataset. 42
6.7 Comparing different generators on random targets from mostly.ai Cancer dataset. 42
6.8 Random vs. outlier targets from mostly.ai Cancer dataset. 43

9.1 Univariate Analysis for Real Dataset 58
9.2 Univariate Analysis for Synthetic Dataset from GPT4 58
9.3 Univariate Analysis for Synthetic Dataset from GReaT framework 59
9.4 Univariate Analysis for Synthetic Dataset from mostly.ai 60
9.5 Texture1 Summary Statistics for Real Dataset 60
9.6 Texture1 Summary Statistics for Synthetic Dataset from GPT4 61
9.7 Texture1 Summary Statistics for Synthetic Dataset from GReaT framework . . 61
9.8 Texture1 Summary Statistics for Synthetic Dataset from mostly.ai 62
9.9 Explained variance ratio vs. principal components for Real Dataset 62
9.10 Explained variance ratio vs. principal components for Synthetic Dataset from GPT4 63
9.11 Explained variance ratio vs. principal components for Synthetic Dataset from GReaT framework 64

9.12 Explained variance ratio vs. principal components for Synthetic Dataset from mostly.ai 65
9.13 Correlation Matrices between original and synthetic dataset for Synthetic Dataset from mostly.ai 65
9.14 Pearson Correlation Plot of the Real Dataset 66
9.15 Pearson Correlation Plot of the GReaT Dataset 66
9.16 Pearson Correlation Plot of the mostly.ai Dataset 67
9.17 Pearson Correlation Plot of the GPT4 Dataset 67

LIST OF TABLES

4.1 Training loss during GReaT's LLM 'distilgpt2' model training 22

6.1 Breast Cancer Data. 32

7.1 Summary Metrics for Different Generators 46

9.1 Experiment metrics for the real Breast Cancer dataset with Groundhog attack . 50

9.2 Experiment metrics for the real Breast Cancer dataset with Groundhog attack (continued) 51

9.3 Experiment metrics for the synthetic Breast Cancer dataset from mostly.ai with Groundhog attack 52

9.4 Experiment metrics for the synthetic Breast Cancer dataset from mostly.ai with Groundhog attack (continued) 53

9.5 Experiment metrics for the synthetic Breast Cancer dataset from OpenAI's GPT4 with Groundhog attack 54

9.6 Experiment metrics for the synthetic Breast Cancer dataset from OpenAI's GPT4 with Groundhog attack (continued) 55

9.7 Experiment metrics for the synthetic Breast Cancer dataset from GReaT framework with Groundhog attack 56

9.8 Experiment metrics for the synthetic Breast Cancer dataset from GReaT framework with Groundhog attack (continued) 57

ACKNOWLEDGMENTS

I would like to express my gratitude to everyone who has helped me in completing my master's and thesis. Firstly, I would like to thank my advisor, Laurie Leyden, for her persistent support, advising, time and guidance, Dr. Chi-Hua Wang for his time, expertise and help with shaping the direction of my thesis, Professor Guang Cheng for his feedback and being my thesis advisor, and Director Rick Paik-Schoenberg of the MASDS program for providing the fellowship opportunity to fund my master's.

I would also like to thank other people in my life who supported me throughout my academic journey and thesis including my sister - Shyama, Isabelle, Carrie, Ola, Akshatha, Maya, Miloni, Misbah, Nina, Tanvi, Christina, Datta, Jay, Indu, Prachi, Reshma, Tiffany, Vickie, Rozeen, Lekha, Ekta, Mahima, Bindiya, Krupa, Spencer, Aaron, Ariana, Zhuri, Kaili, Hannah, Parima, Surbhi, Chotu, Shivani, Bianca, Nathan, Eddie, Perry, David, Kim, Doug, Kari, Mouli, Ding, Ryan, Mihir, Clement, Kennedy, Cyrus, Sunny, Madhav, Kajal, Dwija, Barfi, Kayla, Jon, Mary, my parents, my grandparents, Tenzin D. L., Mingyur R., Yeshua, Krushna, Radhe, and Rama - without whom my master's would not have come to fruition. I am deeply grateful for everyone's unwavering support and encouragement while I pursued this degree in school part-time while working and going through personal challenges. I would also like to thank my colleagues at work including Lynne, Matt and Josh for accommodating my schedule while I juggled many things at once. Truly grateful, feeling blessed and appreciative while recalling everyone mentioned here.

CHAPTER 1

Introduction

As the demand for big data increases for data analysis and machine learning, new tools and techniques are emerging for generating synthetic data that mimic real datasets since they are cheaper and quicker to produce and scale. As such, there is an emergence in need for protecting sensitive user information and evaluating whether the synthetically generated datasets are truly privacy preserving while also being useful for model training and data analysis. The traditional synthetic tabular data generation methods are mainly based on probability models, and using LLMs to generate synthetic data is a very new attempt. As such, limited research has gone into the evaluation and auditing of the privacy-preserving aspects of tabular synthetic datasets generated using LLMs. This paper explores the adversarial privacy auditing of synthetically generated data produced by LLMs using the TAPAS framework presented in the paper "TAPAS: a Toolbox for Adversarial Privacy Auditing of Synthetic Data" [11].

This paper uses a healthcare dataset with sensitive user information of Breast Cancer from The Breast Cancer Wisconsin Diagnostic Database to evaluate the privacy of the dataset using adversarial techniques. Privacy attacks and evaluations are conducted on the real dataset and compared with synthetically generated datasets from several LLMs such as the GReaT framework and OpenAI's GPT4. Additionally, privacy auditing and evaluation is conducted on an AI-generated dataset from a synthetic data startup, mostly.ai, to compare the LLMs' results with privacy preserving aspects of an example start-of-the-art proprietary industry solution. The privacy metrics and results of the synthetically generated

data from LLMs are compared and contrasted with the privacy metrics and reports generated from a popular status-quo method of generator using Generative Adversarial Networks (GANs). The purpose of these experiments is to understand the data quality, data distributions and privacy preserving metrics of the synthetically generated datasets compared to the real dataset.

In Chapter 2, the mathematical definition of Differential Privacy (DP) is explained with the formal privacy guarantees it presents for privacy auditing. Subsequently, the concept of privacy budget is explained which is quantified with parameter "epsilon" (ε) which represents the amount of privacy loss an algorithm allows during its execution. The chapter also reviews the trade-off between utility and precision in differentially private algorithms which includes the trade-off between maintaining the data utility and quality at the expense of compromising privacy by adjusting the privacy budget. Finally, we examine the definition of Composition Theorem with relation to DP and explore the applications of Differential Privacy in real world applications and industry [8].

In Chapter 3, the methodologies and statistical techniques used in synthetic data generation are reviewed. First, we review the basics of tabular datasets which are commonly used during synthetic data generation. We review the popular generation models based on Generative Adversarial Networks (GANs) used to produce synthetic data with generators such as CTGAN, DPCTGAN and PATEGAN [33]. We summarize how to use LLMs to generate tabular synthetic data. Finally, we provide a summary on market research of techniques used in various industry startups for AI-generated synthetic data.

In Chapter 4, we introduce the state-of-the-art Large Language Models as a class of deep learning algorithms designed to understand and generate data in a way that is contextually relevant and mimics human interpretation and language understanding. Since most traditional methods of synthetic data generation utilize probability models, LLMs provide a new way of synthesizing tabular synthetic data while maintaining privacy and statistical properties of the data distributions. We review the architecture of LLMs based on Trans-

former Models as a departure from previous sequence-based models like RNNs (Recurrent Neural Networks) and LSTMs (Long Short-Term Memory Networks) [27]. We review the applications of LLMs particularly in relation to generating synthetic data. Next, we review the GReaT (Generation of Realistic Tabular data) framework's method which uses an auto-regressive LLM to sample synthetic data to produce highly realistic distributions [4].

In Chapter 5, we introduce Adversarial Privacy Auditing techniques from a software toolbox called TAPAS from the paper "TAPAS: A Toolbox for Adversarial Privacy Auditing of Synthetic Data" and the role that it plays in assessing the privacy of synthetic datasets. The different privacy attacks in the toolbox are explained such as shadow-modeling attack, local neighborhood attack and inference-on-synthetic attack. We describe what it means for the attacker to have the knowledge of the generator, the common attacker goals, and methods of evaluating the attacks and generated reports with the privacy-preserving metrics. We also define a threat model which forms the foundation of the adversarial evaluation of synthetic data using the TAPAS toolbox [11].

In Chapter 6, the Breast Cancer dataset from the Breast Cancer Wisconsin Diagnostic Database is presented and summarized [31]. The unique variables of the dataset are listed out and explained from the data source. Moreover, results from the Exploratory Data Analysis (EDA) for the real dataset and the synthetic datasets are summarized to understand the similarities and differences in the data distributions produced. The experimental design is explained with the threat model implementation for the experimental attacks. The resultant reports generated by the TAPAS toolbox are discussed including comparison of different generators on random targets, and random vs. outlier targets metrics from the different datasets. Finally, in Chapter 7, the Summary of Results are presented for the different generator types. The metrics such as accuracy, true positive rate, false positive rate, mia advantage, privacy gain, auc value and effective epsilon values from the different generator methods are compared and contrasted. In Chapter 8, conclusion from the analysis is described as well as future work and next steps are discussed.

CHAPTER 2

Differential Privacy

As the world becomes increasingly data-oriented and more data than ever is collected and stored, there is a need to evaluate whether the datasets we use and the algorithms that produce them are truly privacy preserving or not for an individual whose data is collected. The main goal of data privacy is to enable data analysis and the usage of data for model training without revealing specific information about the individual instances present in the data. As such, to evaluate data privacy, there is a necessity for a mathematical definition that formalizes the notion of privacy, which can be used for privacy auditing. Differential Privacy (DP) is the formal mathematical notion of privacy. Simply put, differential privacy is a property for auditing privacy of an algorithm that produces data — so, if the algorithm satisfies the mathematical guarantees posed by DP, the dataset produced by it also satisfies DP. In this section, we review the historical context of the development of DP by reviewing the limitations of the traditional methods of privacy preservation such as de-identification attacks, linkage attacks, k-Anonymity, the properties of differential privacy, privacy budget and epsilon, utility precision trade-off in data release mechanisms, Composition theorem and the applications of differential privacy [17].

2.1 Background on Privacy Attacks

The challenge of privacy-preserving data analysis spans and affects many disciplines and industry fields. A simple and traditional way of privacy preservation is called de-identification, sometimes called anonymization. There is no formal definition for identifying information,

but generally, it is the information of an individual that can be used to uniquely identify a person from the data collected such as their name, zip code, social security number, cell number etc. So, the notion of personally identifiable information (PII) comes from information that can be identifying of the person. De-identification is the term used to describe removing PII from the dataset during data pre-processing. However, this method is not quite effective because the removed data can be used as auxiliary data by attackers to perform a re-identification attack which can be as simple as using a *join* on two tables [17].

An example of re-identification attack is a linkage attack. Linkage attacks are performed by overlapping data columns between the de-identified dataset we are trying to attack and the auxiliary data. This method is quite straightforward to perform if the attacker gets their hands on the auxiliary data. According to a data privacy research done at Carnegie Mellon University called "Simple Demographics Often Identify People Uniquely" by Latanya S. [24], staggering 87% of people in the US can be uniquely re-identified by auxiliary data combinations including their birth day, gender and zip.

Another common way of preserving privacy is using aggregations and summary statistics. Many times, aggregate statistics are calculated by dividing the larger dataset into smaller ones. The purpose of this method is that aggregation on smaller groups is supposed to preserve privacy of the individual but also allow for real insights from the data analysis. However, the problem this method poses is in the division into smaller groups in the cases where the group size contains only one instance or few instances of the data. In this case, the individual's info can be easily revealed through the summary statistics. A larger group sizes are not fully privacy-preserving as well because methods like calculating difference on multiple aggregate statistics over the same data can reveal individual's information. When these datasets and summary statistics are made public, it is not quite possible to determine the maliciousness or intent of the client in the cases where they are performing multiple queries over the same dataset [17].

k-Anonymity is a formal privacy definition which states that auxiliary information should

not narrow down the set of possible instances or records to a specific individual. In simpler terms, this formalizes the intuition that an individual can blend into the data population. We can say that a dataset is k-Anonymized if each instance in the dataset is part of a group size of at least k size such that each member of the group shares a selected few data columns called quasi-identifiers with other members of the group. So, the goal is to deter from privacy attacks such that it maybe possible to narrow down an individual instance to a group, but not the target group member itself. However, k-Anonymity is not immune to privacy attack by a class of attacks called homogeneity attack. This attack leverages the similarities of the quasi-identifiers within a group and exploits the similarity among the attributes making it challenging to achieve privacy for an individual instance. The attack vector is defined based on the degree of similarity in the data's quasi-identifiers and sensitive information is inferred using determination of group membership. In this attack, if the attacker has a background knowledge on the individual, k-Anonymity becomes highly susceptible to attacks [17].

A more robust anonymization technique to improve privacy preservation while also being immune to the presence of background knowledge is differential privacy.

2.2 Formalizing Differential Privacy

Differential privacy is a formalized notion of privacy. A function which satisfies differential privacy is referred by the term, a *mechanism*. A mechanism F satisfies differential privacy if for all *neighboring datasets* x and x', and all possible sets of outputs S,

$$\frac{\Pr[F(x) \in S]}{\Pr[F(x') \in S]} \leq e^{\varepsilon} \tag{2.1}$$

This can be re-written as the following equation. A randomized *mechanism* $M : X \times \Omega \rightarrow S$ over datasets provides (ε, δ)-Differential Privacy if, for all $x \approx x' \in X$ and all $S \subseteq S$,

$$\Pr[F(x) \in S] \leq e^{\varepsilon} \Pr[F(x') \in S] + \delta \tag{2.2}$$

where the equation requires distributions inducted by F to be close to each other when

datasets vary by addition or deletion of one record [11]. The δ value means that with a probability of $1-\delta$ the mechanism/algorithm will be successful. As mentioned earlier in the introduction of the chapter, differential privacy is a property of the algorithm and so, if the algorithm is differentially private, than the data distribution produced by it is also going to be differentially private. This property can be explained in post-processing such that if F is (ε, δ)-DP, then for any random operation $N : \Omega \times O \to O'$ the composition of F and N is also (ε, δ)-DP. Therefore, this provides privacy guarantees that can not be broken with any operation [11].

In simpler terms, this definition formally defines an algorithm as differentially private if the outcome of the algorithm does not significantly change, regardless of whether any individual's data is included or excluded from the dataset. In other words, mathematically, the probability of any output is almost the same, with or without the presence of any individual's data. This is accomplished by addition of controlled noise to the output, providing strong privacy guarantees.

2.3 Privacy Budget and Epsilon

The concept of privacy budget, also known as privacy parameter, is quantified with parameter "epsilon" (ε), which represents the amount of privacy loss an algorithm allows during its execution. A smaller ε indicates a higher chance of preserving privacy of individual records, but it may result in noisier query results. Generally, in practice, ε is supposed to be less than or equal to 1, and values of ε over 10 do not guarantee much privacy [17].

2.4 Utility-Precision Trade-off in Data Release Mechanisms

While discussing differential privacy, it is important to review the trade-off between utility and precision in differentially private algorithms. Increasing the privacy budget (larger ε) can

improve data utility but may compromise privacy. On the other hand, decreasing the privacy budget (smaller ε) provides stronger privacy but may lead to less accurate analysis results. There are various algorithms and privacy data release mechanisms that achieve differential privacy, such as Laplace mechanism, exponential mechanism, and smooth sensitivity-based approaches. Each mechanism is tailored to specific types of queries and data structures, resulting in a balance between privacy and utility [17].

2.5 Composition Theorem

Composition Theorem for differential privacy states that multiple differentially private algorithms can be combined while preserving privacy guarantees. This theorem is essential as it enables the execution of complex privacy-preserving data analysis [8]. In simpler terms, this theorem explains how the privacy guarantees change when a series of operations, each providing a certain level of differential privacy, are performed on a dataset. For instance, if two mechanisms with privacy guarantees of $\varepsilon 1$ and $\varepsilon 2$ are applied sequentially, the Composition Theorem helps us in understanding the overall privacy loss, which is generally a function of $\varepsilon 1$ and $\varepsilon 2$. This theorem is crucial in the field of data privacy because it enables us to quantify and manage the privacy budget in complex analyses with multiple operations. There are two main types of composition in differential privacy including sequential and parallel. Sequential composition applies when the mechanisms depend on each other or operate on the same dataset, whereas parallel composition applies when the mechanisms operate on disjoint subsets of the dataset. As such, Composition Theorem provides a clear framework for balancing the trade-off between utility and privacy in the analysis of sensitive data built from multiple mechanisms [17].

2.6 Applications of Differential Privacy

There are many applications of differential privacy with utilities ranging for tasks like counting queries, classification, clustering and regression, synthetic data generation, to name a few, with differentially private algorithms. For synthetic data generation, an example use case is when we make algorithms that produce synthetic data representations and tabular datasets differentially private by adding noise. The ultimate goal of differential privacy is to provide a rigorous approach to privacy protection to protect individual privacy without sacrificing the overall utility of the data [21]. There are many potential applications of differential privacy in industry domains such as preserving privacy of users of social networks that collect user information for analysis, healthcare data analysis, census and demographics analysis, social science research, online advertising, recommender systems, ride sharing and location data of passengers used for optimizing routes, privacy preserving machine learning techniques used to train models on sensitive user data, financial data analysis to detect fraudulent activities in banking, and genome data analysis for highly sensitive individual genetic data, to name a few [19]. For the particular use case of synthetic data as it is relevant to this paper, we will cover the role of differential privacy in synthetic data generation in the upcoming chapter.

CHAPTER 3

Synthetic Data Generation

Synthetic data is dataset that is produced/generated artificially to mimic a real dataset. There are several techniques and algorithms employed to generate synthetic or artificial data. The primary goal of synthetic data generation is to provide an alternative for sensitive user data, allowing the data analysis without revealing the real individual's sensitive or private information and identity. In this chapter, we will cover the synthetic data generation methodology, tabular datasets which are the typical data structure of synthetic data, overview of statistical techniques and different types of generators such as Generative Adversarial Networks (GANs), LLMs and overview of industry techniques used for synthetic data generation by startups and companies.

3.1 Synthetic Data Generation Methodology

Suppose we have a real dataset with sensitive user information with a set of instances called S. Mathematically, we can define the set of finite instances by the following expression taken from S:

$$\mathcal{D} = \bigcup_{N \in \mathbb{N} \cup \{0\}} \mathcal{S}^N \tag{3.1}$$

A synthetic data generation model or generator would be a random function $F : \Omega \times \mathcal{D} \to \mathcal{D}$ that would take the real dataset as an input and return the synthetic dataset with the same dimensions. This generator function can be mathematically expressed by the following

equation:

$$\mathcal{D}^{(s)} = F(\mathcal{D}^{(r)}) \tag{3.2}$$

where (s) denotes synthetic and (r) denotes real data. The purpose of this function or algorithm is to produce synthetic data that can be used in lieu of the real data but still maintain the same patterns, distribution, correlations and statistical properties of the original dataset. Synthetic data generation algorithms usually have a training step and then, a sampling step. The training step involves a parameter $\theta \in \Theta$ which is learnt based on the real dataset and a sampling step where the synthetic instances are sampled independently and identically distributed (iid) from a distribution p_θ on $\mathcal{S}$. So, the goal is that once the synthetic data generator model is trained, it can produce identical datasets to the real one. As a result, same insights and conclusions can be derived from both datasets [11].

3.2 Tabular Synthetic Dataset

The typical data type for synthetic data is a tabular dataset. A tabular dataset is a data structured as a table with rows and columns, as opposed to an unstructured dataset consisting of images or videos. This data type is one of the most commonly used in data analysis and model training. A challenge with this data type for synthetic data generation is to maintain and retain the original data's statistical properties especially with the correlation and patterns that are dependent or present between the columns. In the next section, we will go over the statistical techniques used to retain the original data's characteristics for a tabular dataset in synthetic data generation. Formally, a tabular dataset $\mathcal{T}$ can be represented as such:

$$\mathcal{T} = \mathcal{T}_1 \times \mathcal{T}_2 \times \cdots \times \mathcal{T}_n \tag{3.3}$$

where $|\mathcal{T}_n|$ is finite.

3.3 Overview of Statistical Techniques in Synthetic Data Generation

Synthetic data generation techniques span a host of different models and methods. However, the ultimate purpose of these generation techniques is the same. The goal is the creation of a dataset that is realistic and mirrors the original dataset without revealing individual information and identity. A large category of models achieve this goal by adding controlled noise or perturbation, and data masking that replace or remove sensitive information. However, there is a trade off between adding noise and how useful the produced dataset will be for analysis. A standard mechanism used in differential privacy to allow us to determine how much noise to add is called Laplace mechanism. Based on the mathematical definition of Laplace mechanism, for a function $l(x)$ which returns a value, the following can be defined $\mathcal{L}(x)$ that satisfies ε-differential privacy:

$$\mathcal{L}(x) = l(x) + \text{Lap}\left(\frac{s}{\varepsilon}\right)$$

where s is the *sensitivity* of l, and $\text{Lap}(s)$ signifies sampling from the Laplace distribution with center 0 and scale s. The sensitivity of a function can simply be defined by how much an output amount changes when its input is modified by a value of 1. Other common techniques to add noise to a distribution is called Gaussian mechanism which adds Gaussian noise to a distribution [17].

In an earlier section, we described the issue with tabular datasets and the difficulty synthetic generators have with keeping the statistical properties of the original dataset. Some popular statistical techniques utilized to capture the data distribution, correlations and inter-dependencies within the real datasets are Gaussian copulas and Bayesian networks. These methods enable the generation of synthetic data to closely mirror the real dataset's properties.

Some other approaches for synthetic data generation include using Generative Adversarial Networks (GANs), Variational Auto-Encoders (VAEs), Diffusion Models, rule-based

approaches based on business policies, entity cloning which involves extracting sensitive data and replicating it, random sampling from a distribution with methods such as Monte Carlo, and another is data masking or anonymization that anonymizes personally identifiable information to generate synthetic and compliant data. Usually, a combination of these techniques are utilized in industry. A newer and more upcoming technique is using Large Language Models (LLMs) such as Generative Pre-trained Transformer (GPT) to generate synthetic data. We will review a few of the generation techniques in detail that we use for experiments in the upcoming sections [6][13][26][5].

3.4 Synthetic Data Generation Models

3.4.1 Generative Adversarial Networks (GANs)

An approach for synthetic data generation is using Generative Adversarial Networks (GANs) that are deep learning models containing a generator and a discriminator. This approach creates synthetic data which is indistinguishable from real dataset through adversarial training. Following is a review of different types of GANs that can be used for synthetic data generation [10], one of which we use for generating synthetic dataset and running experiments on the breast cancer dataset.

An interesting approach to producing tabular synthetic dataset which is an extension of the traditional GAN is Conditional Tabular Generative Adversarial Network (CTGAN). Traditional GANs excel in image and text generation. CTGAN, however, addresses the challenge posed by tabular data such as imbalanced classes and mixed variable types. It functions via a dual-network architecture where one network generates synthetic data and the other discriminates between real and synthetic data. CTGAN is able to handle different data types including continuous, discrete and categorical variables quite well. This is because it uses a conditional generation strategy which allows it to generate data samples conditioned on specific attributes. By training networks in an adversarial manner, CTGAN effectively

learns the complex distributions of real tabular data making the data it generates similar to real datasets. The generated dataset can be used for tasks like data augmentation, privacy preserving data sharing and machine learning model training. This is particularly useful in situations where the real data is scare, biased, private and sensitive [32].

Another extension of GAN is Differentially Private Conditional Tabular Generative Adversarial Network (DPCTGAN). This is one of the approaches we use in our experiments to generate synthetic data with epsilon values of 0.1 and 1. This is an advanced framework designed to generate synthetic tabular data with emphasis on differential privacy and privacy preserving properties. DPCTGAN builds upon CTGAN by adding differential privacy standard that ensure that the output of a data analysis does not reveal the privacy or identity of individual people or instances in the dataset. This is achieved by adding controlled noise to the data and the model parameters during training. The architecture of DPCTGAN is fundamentally similar to a GAN with a generator to generate data samples and a discriminator for evaluation of those samples. The distinguishing factor about DPCTGAN is that both networks are trained with differential privacy constraints [25].

Another special approach to using GAN is Private Aggregation of Teacher Ensembles Generative Adversarial Network (PATEGAN). The objective of PATEGAN is similiar to DPCTGAN such that it generates synthetic data that closely resembles real data which ensures the privacy of individual instances in the dataset. The distinction in the implementation of PATEGAN is that the differential privacy is applied through PATE. This approach involves splitting the original, sensitive dataset into multiple disjoint subsets and training an ensemble of "teacher" models, each on a different subset of the real dataset. Then, the model is trained to predict the output for a given input with any type of classifier. Then, the predictions of each "teacher" model or subset model are aggregated. Subsequently, the "student" model is trained on the aggregated outputs. This aggregating process typically involves techniques such as noisy voting ensuring that the student model learns from the ensemble without gaining access to any specific individual instance in the dataset ensuring

privacy. With the GAN architecture, the student model functions as the generator producing synthetic data, while the discriminator evaluates the validity of the generated data. In essence, the discriminator differentiates between real and synthetic data thereby guiding the generator to produce more realistic synthetic data samples. The key benefit of using PATE-GAN is that it is able to generate highly realistic synthetic data without compromising the privacy of individuals in the original dataset [33].

In the next section, we will cover some market research as well as overview of techniques and use cases of synthetic data generation in industry.

3.4.2 Market Research on Techniques for AI-generated Synthetic Data in Industry

There are many use cases of synthetic data in industry including generating synthetic data for software testing and integrating the data production step into the CI/CD workflows, generating datasets for machine learning model training, privacy regulations compliant data sharing of datasets containing sensitive private information of users, product design to test user flows with data that mimics user data, and generating representative and unbiased data for behavioral simulations, to name a few [20][3][30].

Some industries where synthetic data generation is particularly useful is healthcare with highly sensitive patient datasets, fraud and anomaly detection using synthetic time series data, computer vision and object detection in agriculture and manufacturing, banking and finance models training, disaster prediction and risk management, tech software testing and simulations, automotive and robotics trainings. Several data generation startups and tools have emerged in industry to meet these needs such as mostly.ai, gretel.ai, tonic.ai, Datomize, rendered.ai, Oneview, MDClone, Hazy, K2view, CVEDIA, to name the top players [26]. Synthetic data generation is used by larger companies like Apple Inc. to augment real datasets and obfuscate real user's data with local differential privacy techniques [1]. The common techniques used in industry are deep learning models such as GANs, Variational

Autoencoder (VAE), Diffusion Models, Stochastic Processes and Rules Engines [20].

In the experiments with synthesizing breast cancer data for this paper, we use mostly.ai's platform to generate a dataset from the original data using their proprietary tool (the details of which they refused to reveal) which likely involves an ensemble of generation methods mentioned earlier.

3.4.3 Using LLMs to Generate Tabular Synthetic Data

The emergence of Large Language Models (LLMs) accompanies the possibility of another synthetic data generation method and represents a significant potential in the field of data science and artificial intelligence. Traditionally, generating synthetic tabular datasets have been challenging due their complexity and the need of maintaining statistical properties. LLMs offer a novel solution to applying their natural language processing capabilities to the domain of tabular data since they can analyze and comprehend the underlying patterns and relationships within data with their large textual data processing capability. By leveraging their generative capabilities, LLMs can produce synthetic data that not only mimics the statistical properties of the original dataset but also respects the constraints and correlations inherent in tabular data such as dependencies between columns and data type specificities [30].

Compared to GANs, LLMs have different strengths and weaknesses. GANs are more effective and efficient than LLMs in generating image data and learning complex time-series data. However, difficulty is introduced in GANs for the step of performing optimizations in training. In this case, LLMs can outperform GANs on text generation and other general datasets due to their high performance and scale. Between these two models, LLMs are generally able to generate datasets quicker but at a higher computational cost [30].

We will review the architecture of LLMs and the details of generating synthetic data using LLMs in the next chapter.

3.5 Limitations and Challenges with Synthetic Data

A challenge with synthetic data generation involves reliability of the data source. If the real or original data quality is not good or representative, then the quality of the synthetic data will suffer as well with bias or an unrepresentative generated dataset. Another challenge is replicating outliers from the real datasets which synthetic data generators usually miss. Therefore, diversity in data and outliers are critical pieces of information from real datasets we want to gather to avoid data homogenization, unpredictable loss in data utility and variable privacy gain [23]. Other limitation includes the requirements of expertise, time and effort in data science teams to produce robust generator models and data pipelines to generate reliable data. In addition to time and expertise requirements, this change in status quo data workflows would necessitate investments in quality checks and output control to ensure the correctness of the data distributions before passing them into machine learning and deep learning models. Lastly, user acceptance in industry and companies is another gap that needs to be filled by educating employees, executives and decision makers on the use cases and reliability of synthetic data in their companies' normal data workflows and analysis processes [26] [5].

CHAPTER 4

Large Language Models

The traditional synthetic tabular data generation methods are mainly based on probability models, and using LLMs to generate synthetic data is a growing field of research. As such, limited research has gone into the evaluation and auditing of the privacy-preserving aspects of tabular synthetic datasets generated using LLMs. In this paper, the purpose is to compare and contrast the data quality, data distributions and privacy-preserving metrics of the real dataset with synthetically generated datasets from LLMs. Therefore, we will review Large Language Models in this chapter. In this chapter, we introduce LLMs, review their architecture based on Transformer Models as a departure from previous sequence-based models like RNNs (Recurrent Neural Networks) and LSTMs (Long Short-Term Memory Networks) [27]. We review the applications of LLMs particularly in relation to generating synthetic data as well some of their limitations. Then, we review the GReaT (Generation of Realistic Tabular data) framework's method which uses an auto-regressive LLM to sample synthetic data to produce highly realistic data distribution. Finally, we discuss using OpenAI's GPT4 to produce synthetic dataset that mimics a real dataset.

4.1 Introduction to Large Language Models

In the field of artificial intelligence, large language models (LLMs) have emerged as a pivotal technology, marking a significant leap in the ability of machines to process and generate human language. LLMs are a class of deep learning algorithms specifically designed to understand, interpret, and generate text with in-context learning and understanding which is

often indistinguishable from human-written content. These models are trained on a vast corpora of text data, encompassing a wide array of topics, styles and structures, enabling them to capture the complexities, intricacies, colloquialism and nuances of natural language. The development of LLMs represents a paradigm shift in natural language processing (NLP), transitioning from rule-based and statistical methods to more sophisticated, data-driven approaches. As a result, LLMs have found applications in diverse areas including but not limited to language translation, content creation, computational biology, robotics, creative work creation, conversational agents, information extraction as well as synthetic data generation. One of the features of LLMs is their scale, both in terms of the size of the models and the data on which they are trained. These models created from architectures like Transformer are characterized by a large number of parameters in the order of billions and trillions. This massive scale allows LLMs to learn complex patterns and relationships within the text data, leading to more accurate and contextually appropriate outputs [14].

The architecture of LLMs is predominantly based on the Transformer model from the paper called "Attention is All You Need" authored by Vaswani et al. in 2017 [27]. The Transformer architecture marked a departure from previous sequence-based models like RNNs (Recurrent Neural Networks) and LSTMs (Long Short-Term Memory networks) by relying solely on attention mechanisms to weigh the significance in the input data. This architecture allows LLMs to process words in parallel which significantly speeds up training and improves the ability to capture long-range dependencies in textual data. As a result, LLMs demonstrate superior performance in understanding context and generating coherent, contextually relevant text over extended passages. Moreover, the scalability of the Transformer architecture allows for incremental improvements in model performance as more computational resources are employed, leading to the development of increasingly sophisticated and robust models.

There are many applications of LLMs impacting various sectors including technology, healthcare, education, and finance. In the technology sector, LLMs are utilized to improve

efficiency and intuitiveness of search engines, voice assistants and chatbots. In healthcare, LLMs assist in information extraction from medical texts, and aiding in diagnostics and research [15]. For education, they are used to develop personalized learning tools and content generation. Despite their capabilities, LLMs also pose significant challenges and ethical considerations. The scale of these models poses several challenges such as dearth in computational resources and machines needed for training such large models, as well as concerns regarding environmental impact due to high power requirements and accessibility to organizations and areas with limited resources. The quality of the training data can introduce biases and ethical considerations in these models. Therefore, questions about fairness, transparency, trustworthiness and accountability in AI become important [28].

Large Language Models (LLMs) like GPT-4 and BeGReaT framework are new powerful tools to generate synthetic data. LLMs offer a wide range of potential applications across various industries that require synthetic data. This capability is quite valuable in scenarios where real data is scarce, sensitive or expensive to acquire. LLMs can also aid in imputing datasets which includes filling out missing instances or values in a dataset by inference and data augmentation with synthesized variations. As such, LLMs can help enrich the existing datasets by producing realistic synthetic data and improve the robustness of machine learning models with larger and bigger training datasets. In the next two sections, we will review the GReaT framework methodology for producing realistic tabular synthetic datasets and using OpenAI's GPT4 to produce data that mimics real datasets [16].

4.2 GReaT Framework

Tabular datasets are notably challenging to synthesize due to the diverse types of features and size ranges while maintaining statistical properties of the distribution and correlations between columns. Although generative models like variational autoencoders or generative adversarial networks have been adapted for the purpose of synthesizing tabular datasets,

there has been less focus on utilizing the generative capabilities of recent transformer-based large language models (LLMs). The framework GReaT (Generation of Realistic Tabular data) employs an auto-regressive generative LLM to generate synthetic data. This method's effectiveness is proven through a series of rigorous experiments in the paper "Language Models are Realistic Tabular Data Generators" by Vadim et al [4].

The GReaT framework's method leverages advanced pretrained Transformer language models to produce high-quality synthetic tabular data that closely resembles real datasets. The GReaT framework models the distribution of tabular data by conditioning on a subset of data and sampling the rest of the features. The effectiveness of GReaT is validated through various experiments that assess the validity and quality of the generated data. The paper's findings indicate that GReaT achieves high performance on various real-world and synthetic datasets [4].

To utilize the GReaT framework to generate new data samples of the breast cancer dataset, we installed the GReaT framework using `pip install be-great`. We used their API which uses the large language model called 'distilgpt2' with batch size of 16 and set the number of epochs to be 100 for model training. After the model training, we sampled synthetic data from the model's distribution and saved it in a tabular form. This LLM model in the GReaT API was run on a GPU since the model training doesn't work on a Mac CPU. The table 4.1 shows the training loss values from the training phase of the GReaT's model to generate synthetic data distribution. These values of training loss represent the error/difference between the predicted output and the actual target values during the training steps of the model that generated the synthetic data distribution.

4.3 OpenAI's GPT4 to Generate Synthetic Data

OpenAI's GPT-4 is one of the most popular LLMs used widely around the world during the last year. The latest language model GPT-4 represents a significant evolution from its pre-

Step	Training Loss
500	1.126
1000	1.006
1500	0.973
2000	0.951
2500	0.935
3000	0.924
3500	0.917

Table 4.1: Training loss during GReaT's LLM 'distilgpt2' model training

decessor GPT-3 [7]. The evolution from OpenAI's GPT-3 to GPT-4 marks a significant leap in the domain of synthetic data generation as well. GPT-3 with its 175 billion parameters has a remarkable capability of understanding inputted data, performing in-context learning and interpreting language to generate realistic data as an output. However, with synthetic data generation, it possesses limitations around the size of the data it can intake and maintaining statistical properties in the generated datasets [29]. On the other hand, GPT-4 with 1.76 trillion parameters can further produce more contextually accurate and nuanced data while maintaining statistical properties, distributions and correlations. It can also introduce randomness and maintain quality and coherence of the data due to its ability to reflect on and have deeper understanding of the context of data, its subject, text/language, statistical properties and data types. OpenAI's GPT-4 available as a chat which takes in attachments of datasets in csv formats to generate synthetic datasets or their API is also available which allows for the same in programming [18].

A limitation of using GPT-2, GPT-3(.5) and GPT-4 that the paper "The Curse of Recursion: Training on Generated Data Makes Models Forget" highlights is the case of hallucinated distributions. In this paper, they describe how the LLM models trained on synthetic data cause irreversible defects in the resultant learned generative models. A resulting defect is

where the tails of the original data distributions disappear during model training. This effect is referred to as Model Collapse and it only takes place in Variational Autoencoders, Gaussian Mixture Models and LLMs. This issue results in unrepresentative data distributions produced by the LLMs which affect the quality of synthetic data generated from the model. So, the authors emphasize the importance using credible and reliable data sources to train these models and taking this issue seriously to sustain the benefits and credibility of outputs produced by LLMs typically trained from large scale datasets scraped from the web [22].

CHAPTER 5

Privacy Auditing with the TAPAS Toolbox

In this chapter, we will review the adversarial evaluation framework proposed by the paper, "TAPAS: A Toolbox for Adversarial Privacy Auditing of Synthetic Data" [11]. This framework introduces threat modeling with a library of privacy attacks to perform on synthetic datasets. We use their toolbox to evaluate the privacy of the synthetic datasets generated discussed in the experiments in the next chapter. In this chapter, we will go over the threat modeling framework, attacker knowledge and the metrics and reports to evaluate the privacy of the generators.

5.1 Threat Modeling Framework

A threat model or attack model defines a framework in which a privacy attack is conducted on a dataset with the assumption that the attacker has some knowledge of the data and the generator. We can formally define this intuition in computation to evaluate and assess whether a synthetic data generating mechanism meets the privacy guarantees posed by differential privacy. We will review the pseudo-code of a threat modeling framework with attack training and testing in the next Chapter called Experiments.

5.2 Data and Generator Knowledge of the Attacker

While defining a threat model, we can make some assumptions about the knowledge an attacker has of the real dataset. The TAPAS toolbox supports two types of attacker data

knowledge including auxiliary data knowledge and exact data knowledge. Auxiliary data knowledge can be defined as knowledge an attacker has of a random subset of the real dataset, while exact data knowledge means that the attacker has full knowledge of the dataset. Mathematically, we can define exact data knowledge as follows:

$$\mathcal{D}^{(r)} \sim \pi_{\mathcal{D}} \tag{5.1}$$

where $D^{(r)}$ signifies the real dataset and the prior over the dataset and $\pi_{\mathcal{D}}$, denotes the knowledge or information of the dataset the attacker maintains [11].

The TAPAS framework performs attack on the dataset by attacking one record at a time. This is the unit of privacy we defined earlier in the section of Differential Privacy. Formally, this attack can be reduced to one specific record by combining the prior knowledge of the dataset excluding that one record x, and the knowledge of the target x. We can define auxiliary data knowledge as the following:

$$\mathcal{D}^{(r)}_{-x} \sim \pi_{d'} \tag{5.2}$$

Here if the attacker makes an assumption about the record x, it is replaced by another record x' and the remaining data samples in D_{-x} are sampled from the prior $\pi_{d'}$ [11].

For the knowledge of the generator, the TAPAS toolbox presents four possibilities including No-box, Black-box, White-box and Uncertain-box. No-box is when the attacker has no information about the generator. Black-box is when the attacker has the exact knowledge of the generator. White-box is when the attacker not only has the exact knowledge of the generator but also some parameters of the generator model. Uncertain-box is when the attacker has some knowledge of the generator and uncertain or partial knowledge of the generator model parameters [11].

In the experiments for this paper, we assume auxiliary data knowledge and black-box knowledge of the generator by the attacker.

5.3 Goals and Intentions of the Attacker

The attacker's primary goal for launching the attacks is to reveal sensitive and private information from the original dataset. Mathematically, we can define this intention by a function f mapping the data D to a decision/intention of the attacker I as such:

$$f : \mathcal{D} \rightarrow \mathcal{I} \tag{5.3}$$

A type of attacker goal is called Targeted Membership Inference (MIA). In this type, for a target record x, the attacker tries to figure out whether the target x is in the real dataset $\mathcal{D}^{(r)}$. This can be represented formally as such:

$$f : \mathcal{D}^{(r)} \rightarrow I_{\{x \in \mathcal{D}^{(r)}\}} \tag{5.4}$$

Another type of attacker goal is called the Targeted Attribute Inference (AIA). For this type of attack, the attacker's goal is to figure out and reveal the attributes for target instances/records of individuals [9]. Formally, with an attribute y and incomplete knowledge of the target record x_{-y}, the attacker wants to reveal the value v of attribute y in a way that the completed record $x_{-y}|v$ is in the dataset. This can be represented by:

$$f : \mathcal{D}^{(r)} \rightarrow v_{\{x^{v}_{-y} \in \mathcal{D}^{(r)}\}} \tag{5.5}$$

Lastly, the most maliciously intentioned attack is called Reconstruction. In this type of attack, the attacker tries to know the entire original dataset. This can be mathematically denoted by:

$$f : \mathcal{D}^{(r)} \rightarrow \mathcal{D}^{(r)} \tag{5.6}$$

Out of these attackers' goals, MIA and AIA are the most common types of attacker goals and the TAPAS toolbox provides support for testing these two types of attackers [11].

5.4 Library of Attacks in the TAPAS Toolbox

Some examples of attacks provided by the TAPAS toolbox include Shadow Modeling attack, Groundhog attack, Probability Estimation attack, Synthetic Predictor attack, Closest Distance AIA attack, Closest Distance MIA attack, Local Neighborhood attack and Direct Linkage attack.

Shadow Modeling attack simulates the data generation process using auxiliary data knowledge of the attacker and trains a classifier to predict a property of the training dataset from the synthetic dataset. A Groundhog attack is a derivative of the Shadow Modeling attack where a Random Forest Classifier is used with n number of estimators and Gini Impurity split for the decision trees in the random forest [23]. Probability Estimation attack falls under the class of Membership Inference Attack. In this attack, a statistical model p_x of the distribution of synthetic records is estimated and then $p_x(target\_record)$ is computed as the score. Intuitively, the probability distribution of the synthetic data is defined by the generator trained on the real data and the probability is likely to be high for the records in the real data. Synthetic Predictor attack is a type of attribute inference attack that trains a classifier C on the synthetic data to predict the value v of record x, then uses $C(target\_record)$ to predict the target record. The Closest Distance AIA is an attack that finds the closest record to the target record and uses the value of a sensitive attribute of that closest record as the outcome for this attack. The Closet Distance MIA is an attack that looks for the closest record to a given target in the synthetic data to determine if the target is in the training set. Local Neighborhood Attack makes a decision based on records similar to the target record. It specifically takes into account records in a given radius for a specific value of distance. Direct Linkage attack, as we went over it in section for Differential Privacy, is an attack that checks if a target record is present or not in the synthetically generated data [11].

In the experiments for this paper, we assume the goal of the attacker to be Targeted

Membership Inference (MIA) with the Groundhog attack.

5.5 Evaluation Metrics for the Attacks

The TAPAS Toolbox provides several evaluation metrics to assess and audit the privacy preservation of target records for the synthetic data generators in the experiments. The metrics include:

- **Accuracy**: Differential Privacy guarantees safety against membership inference attacks in which an attacker tries to figure out whether a target record x is present in the real dataset, $D^{(r)}$. The accuracy metric signifies the success rate for the attacker correctly classifying a record in or not in the training dataset [11].

- **True Positive Rate**: The True Positive Rate (TPR) is the proportion of the actual positive target records that are correctly identified by the attack. Mathematically, a random mechanism is defined as $\mathcal{M} : \Omega \times \mathcal{D} \to \mathcal{O}$ that satisfies (ε, δ)-differential privacy guarantees and $d, d' \in \mathcal{D}$ as part of the neighboring datasets ($d \approx d'$). Then, the True Positive Rate can be defined by the randomized mechanism $\mathcal{M}$, and a randomized attacker $\mathcal{A} : \Omega \times \mathcal{O} \to \{d, d'\}$ as such:

$$TP_{\mathcal{A}} = \Pr[\mathcal{A}(\mathcal{M}(d)) = d] \tag{5.7}$$

- **False Positive Rate**: The False Positive Rate (FPR) is the probability of the records that are incorrectly identified as positive target records by the attack. Formally, this can be defined as the following:

$$FP_{\mathcal{A}} = \Pr[\mathcal{A}(\mathcal{M}(d')) = d] \tag{5.8}$$

- **Effective Epsilon**: Based on the definitions of TPR and FPR, we can establish that:

$$e^{\varepsilon} \geq \max\left(\frac{TP_{\mathcal{A}} - \delta}{FP_{\mathcal{A}}}, \frac{1 - FP_{\mathcal{A}} - \delta}{1 - TP_{\mathcal{A}}}\right) \tag{5.9}$$

Based on this inequality, we can further define the effective $\varepsilon^{\text{eff}}(\delta; d, d')$ as:

$$\varepsilon^{\text{eff}}(\delta; d, d') = \log \sup_{\mathcal{A}:\Omega\times\mathcal{O}\to\{0,1\}} \max\left(\frac{TP_{\mathcal{A}} - \delta}{FP_{\mathcal{A}}}, \frac{1 - FP_{\mathcal{A}} - \delta}{1 - TP_{\mathcal{A}}}\right) \tag{5.10}$$

In simpler terms, effective epsilon ε^{eff} value is a measure of the actual privacy loss that occurs in practice when a differentially private mechanism is used, as opposed to a theoretical measure of privacy loss. As we reviewed in Chapter 2, epsilon, ε, is used to quantify the probability of an outcome when one instance or record of an individual is added or removed from the dataset. Smaller the ε^{eff} value, stronger the privacy guarantee of that mechanism. In the real world, when differential privacy techniques are applied, and randomness and noise are added to the mechanisms in composition, the DP theoretical guarantees may not be fully upheld. Therefore, ε^{eff} value is a measure for a more realistic and quantitative privacy guarantee [11].

The TAPAS toolbox calculates the ε^{eff} by first greedily performing an attack on 10% of the testing samples and performs estimation for statistically significant lower bound on the ε^{eff} using the 90% of the remaining samples [12].

- **MIA Advantage**: MIA refers to the value of Membership Inference Attack (MIA) Advantage. This metrics evaluates the success of the membership inference attacks. The advantage refers the ability of the attacker to classify and reveal the target record better than a random chance. Formally, MIA advantage is measured by the distance between the attacker's success probability and the probability of a random guess [11]. As an example, if a random guess has a 50% chance of being correct, and the attacker has a success rate of 70%, the MIA advantage value is 10%. The MIA advantage can be represented as:

$$\text{Advantage}_{\text{MIA}} = \text{Accuracy}_{\text{Attack}} - \text{Baseline}_{\text{Random Guess}} \tag{5.11}$$

where $\text{Accuracy}_{\text{Attack}}$ is the attacker's success rate for correctly identifying if a record is in the training dataset and $\text{Baseline}_{\text{Random Guess}}$ is the success rate of random guessing. In the context of DP, a lower MIA value is preferred since it indicates that the model does not leak much information about its data.

- **Privacy Gain**: Privacy gain is a metric to quantify the gain or improvement in privacy after applying DP to a mechanism. In other words, this allows us to understand how hard is it for an attacker to infer or reveal sensitive information after privacy-enhancing methods have been applied. The gain refers to the increased uncertainty for the attacker in revealing individual information. We can quantify privacy gain by taking the difference between initial risk, which is the risk of privacy loss before applying DP, and residual risk, which is the risk of privacy loss after applying DP to a mechanism. In relation to the epsilon (ε) parameter, privacy gain is inversely proportional. For instance, lower the value of epsilon, higher the privacy gain, since the output of the mechanism is less dependent on in individual record [11].

- **AUC**: AUC refers to area under the ROC (Receiver Operating Characteristic) curve. The ROC curve plots the TPR against the FPR at various thresholds for binary attacks. The threshold denotes the point above which a data point is classified as positive in a threat model. The area under the ROC curve signifies the quality of the classifier by plotting FPR on the x-axis and TPR on the y-axis and the points on the curve represent sensitivity (TPR) and specificity (1-TNR) corresponding to a particular threshold. If the curve has an arc closer to the top-left corner, which indicates good performance of the inference attacks and closer to the diagonal of the plot space indicates the lower effectiveness of the attack. The AUC summarizes this using a single value which is computed by calculating the area under the ROC curve. So, higher the value of the

AUC suggests better performance of the attack, and thus, lower privacy preservation. Lower the value of the AUC suggests worse performance of the attack, and thus, higher privacy preservation [11].

CHAPTER 6

Experiments

6.1 Breast Cancer Data

Here are a list of unique variables in the Breast Cancer Wisconsin Diagnostic data [31] used to generate the summary reports. Every instance of patient ID has three readings as separate columns for each of the unique variables and one corresponding diagnosis as the dependent variable.

Variable Name	Description
ID	ID of the patient
Diagnosis	M=malignant, B=benign
radius	Distances from center to points on the perimeter
texture	Gray-scale values for the texture of the cell nucleus
perimeter	Perimeter values of the cell nucleus
area	Area of the cell nucleus
smoothness	Smoothness values of the cell nucleus
compactness	compactness values are calculated as $perimeter^2/area - 1.0$
concavity	Values of concavity for the concave portions of the contour
concave points	Number of concave portions of the contour
symmetry	Symmetry values of the cell nucleus
fractal dimension	"coastline approximation" - 1

Table 6.1: Breast Cancer Data.

6.2 About the Data

Breast cancer is a malignant type of cancer and has been life threatening for women around the world. With early detection as a non-metastatic disease, breast cancer is curable. Health-care datasets such as these are highly confidential, but important for analysis and model training that help with early detection. Therefore, exploring the use cases of synthetic data generation for such a dataset is quite beneficial.

The real dataset from The Breast Cancer Wisconsin Diagnostic Data is used to generate synthetic data with 569 record instances and 32 columns. For the dataset, the features or each instance are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. Ten real-valued features computed from the raw data for each cell nucleus are as follows:

- radius (mean of distances from the center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area – 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" – 1)

6.3 Exploratory Data Analysis

Exploratory data analysis (EDA) is conducted on the real dataset and the synthetically generated datasets from several sources (including GReaT, OpenAI's GPT4, Mostly.AI). The purpose is to understand the data quality and data distributions, and compare and contrast between the real dataset and the synthetically generated ones. EDA includes checking for data types, data previews, missing values, constant occurrences, duplicate rows, conducting univariate analysis, bivariate analysis and multivariate analysis. An open-source Python library called "Edvart" is used to explore datasets and generate EDA reports. The EDA notebooks, summary plots and html files for the EDA reports can be found in the Appendices section [2].

For the original dataset, the EDA revealed that there are no missing instances, rows with missing values or duplicate rows. The univariate analysis revealed the most frequent values for each of the categorical and numeric columns using descriptive and quantile statistics such as the number of unique values, sum of the distribution, mean, mode, standard deviation, men absolute standard deviation, coefficient of variation, kurtosis, skewness, minimum, maximum, Q1, median, Q3, range and interquartile range. Overall, for each data column in the original data set, we see a unimodal distribution with a trend of the distribution skewed towards the left with some outliers for variables such as area, perimeter and radius. The bivariate analysis utilized correlation plots such as Pearson Correlation, Spearman Correlation, Kendall Correlation, Pairplot and Contingency table for analysis. Looking at the Pearson Correlation in further detail which measures the strength of the linear relationship between two variables with -1 signifying total negative linear correlation, 0 being no correlation and +1 meaning a total positive correlation. We notice a significant positive correlation amongst several numeric variables/columns in figure 9.14. For multivariate analysis, the library performs principal component analysis, calculates explained variance ratio, parallel coordinates and parallel categories. Examining the explained variance ratio, which signifies

the percentage of variance that is attributed for principal components with 80% considered high variance, 68.18% represents high data variance.

Examining the EDA of mostly.ai dataset reveals no missing instances, rows with missing values or duplicate rows. The univariate analysis reveals similar trends to the original dataset of unimodal left skewed distributions with outliers. The bivariate analysis reveals an interesting observation through the correlation plots that show all correlation values amongst variables to be 0, signifying a drop in the correlation information after applying the synthetic data generator to the original dataset. We can also notice that the explained variance ratio has dropped to 9.58% signifying low data variance in the synthetic data compared to the 62.18% in original dataset. So, through this synthetic data generation process, we lost data variability and correlations present in the original data.

Examining the EDA of the dataset generated by the GReaT framework, we notice in the univariate analysis a trend of unimodal left skewed distribution with outliers quite similar to the original dataset. The bivariate analysis reveals the observation of the correlation values being between 0 to 1, with the correlations being conserved from the original dataset, which is in contrast to the other generator models. We notice the explained variance ratio to be around 46% which is quite higher than the other generators and display a medium data variance compared to the original dataset's value of around 68.18%.

Examining the EDA of the dataset generated through OpenAI's GPT4, we see no missing values or duplicate rows. We see unimodal distributions with zero skew with low numbers of outliers. The correlation plots reveal no correlation amongst the columns signifying a drop in correlation information after applying the synthetic generator. The explained variance ratio is 9.68% which is similar to the mostl.ai's value, signifying a drop in data variance. Comparing with the GReaT framework's generated dataset, we see a higher loss in representative information such as outliers, correlations and data variance with GPT4's data. So, it is less representative of the original dataset than the data produced by the GReaT framework.

Overall, comparing the EDA of the original dataset with the EDA from the generators

of GPT4, mostly.ai and the GReaT framework, we can observe that the GReaT framework produced the most representative dataset conserving the outliers, skewness of the unimodal distributions, correlations as well as data variance.

6.4 Experiment Design

6.4.1 Generators

The synthetic data generators for experiments in this paper come from several sources such as LLMs from the GReaT framework and OpenAI's GPT4, Generative Adversarial Networks (GANs) as well as an AI-generated dataset produced using a proprietary technique from an industry startup, mostly.ai.

6.4.2 Experimental Attack

For defining the attack, auxiliary data knowledge is assumed which means that the attacker knowledge assumes access to some auxiliary dataset from which training datasets are sampled as a random subset of the auxiliary data. For attacker knowledge on generator, the recommended assumption by the TAPAS toolbox of the black-box knowledge is made. The threat model is defined as a targeted MIA (membership inference attacks) on random records with the defined attacker. The randomized target record indices are determined by an isolation forest model and they are combined with an array of outlier indices. The attacker of Groundhog attack with standard parameters is initialized with Random forest classifier declared as the feature set classifier. With this attack setup, the threat model is trained and tested on different generators, and the TAPAS toolbox produces resultant summary reports of privacy metrics for each generator. For the differentially private mechanism of DP-CTGAN, we use the ε values of 0.1 and 1 to compare the looser and stricter constraints to the ε^{eff} value. The industry standard for ε is 1. The pseudo-codes for the attack function

and the main threat model function are as follows. The GitHub link for the Jupyter Notebook with Privacy Auditing of Synthetic Data using the TAPAS Toolbox can be found in the Appendices.

Algorithm 1 Pseudo-code for the **Attack Function**

1: Initialize the **attacker knowledge on data**
2: Initialize the **knowledge on generator**
3: Create an array of **target indices** to target combining **random indices** selection using an isolation forest model and **outliers**
4: Define a threat model for Membership Inference Attack on target records using TAPAS's **threatModels.TargetedMIA(dataKnowledge, getRecords([targetIndex]), sdgKnowledge, standard parameters)**
5: Initialize an attacker of Groundhog attack **Attack.GroundhogAttack** with standard parameters
6: Train the attack with **attacker.train(threatModel, numSamples)**
7: Test the attack with **attacker.test(threatModel, numSamples)**
8: Return metrics with **summary.getMetrics()**

Algorithm 2 Pseudo-code for the **main Threat Model Function**

1: Initialize empty data frames for storing **metrics** and **all summaries**
2: Define an **array of generators**
3: Loop **for generator in generators**:
4: Nested loop **for target in targets**:
5: Call the attack function with **attack(dataset, targetIndex, generator)**
6: Return **summary metrics**

6.5 Results

In Figure 6.1, we can analyze the plot comparing different generators on random targets for the real Cancer dataset with DP-CTGAN with epsilon values of $\varepsilon = 0.1$ and $\varepsilon = 1$. We can visually compare metrics like the effective epsilon, the classification accuracy, area under the receiver operating characteristic curve (AUC) and the privacy gain (PG). Some points are above the threshold of epsilon of 1 signaling that the privacy guarantee was not upheld. For some of the target records, the values are not showing in the plot because the epsilon value of $\varepsilon = \infty$ due to effective epsilon being beyond bounds or numerical issues with floating point precision. Based on the AUC and privacy gain values, we can state that DP-CTGAN (eps=0.1) performed the best for privacy preservation against the Groundhog attack in comparison to DP-CTGAN (eps=1) and the real dataset.

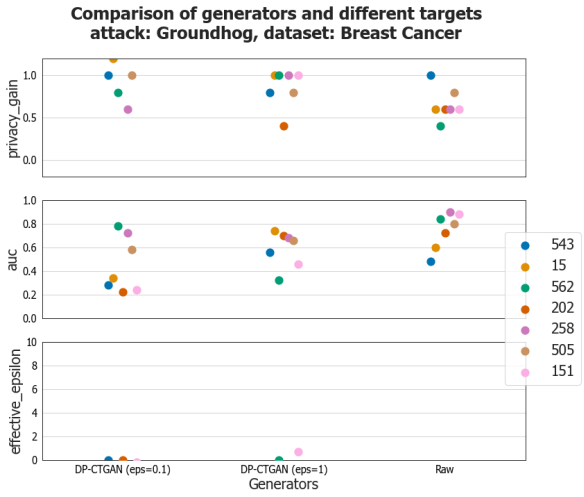

Figure 6.1: Comparing different generators on random targets from real Cancer dataset.

Figure 6.2 shows comparison of metrics between random vs outlier targets for real Cancer dataset. Generally, we see a trend of higher privacy for outliers than random targets. For the real dataset, the AUC value and privacy gain values have a higher disparity between random and outliers than DP-CTGAN (eps=0.1 and eps=1), which intuitively makes sense

since synthetic data generators usually miss generating representative outliers and this is still an active area of research and development.

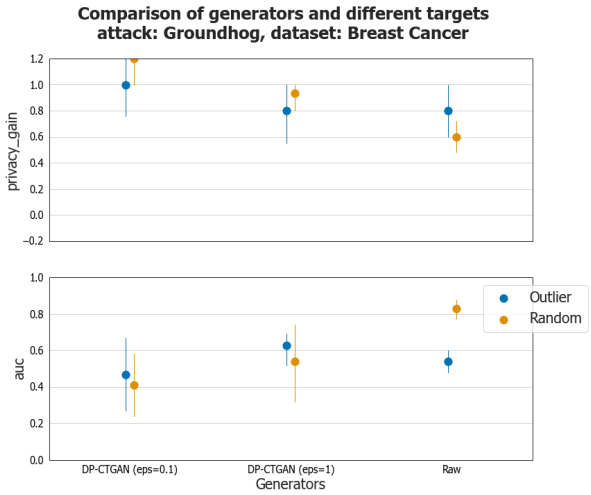

Figure 6.2: Random vs. outlier targets from real Cancer dataset.

Figure 6.3 shows comparison between synthetically generated dataset from BeGReaT and DP-CTGAN (eps=0.1 and eps=1). Here, we can see that the privacy gain of 0.8 and auc value of 0.7 are similar to the results from DP-CTGAN. Overall, the averaged values for auc and privacy gain are similar for the GAN based generator and the LLM based GReaT framework generator.

Figure 6.4 portrays random vs outlier targets from synthetic data generated from the BeGreaT framework. Here, the disparity between the outlier and random values for different generators follow a similar trend to the values we see for the real dataset, with outlier targets having higher privacy preservation than random target records.

Figure 6.5 shows comparison between synthetically generated dataset from GPT4 and DP-CTGAN (eps=0.1 and eps=1). Here again, the LLM generator, GPT4, performs quite similarly to the DP-CTGAN generator with privacy gain value around 0.7, AUC value around 0.6 and effective epsilon under the value of 1.

Figure 6.6 shows random vs outlier targets from synthetic data generated from GPT4.

Figure 6.3: Comparing different generators on random targets from BeGReaT Cancer dataset.

We see that outlier and random targets do not display high disparity in privacy gain and AUC values, potentially indicating outliers records not being too representative of the real dataset.

Figure 6.7 shows comparison between synthetically generated dataset from mostly.ai and DP-CTGAN (eps=0.1 and eps=1). Here, we see privacy gain and auc in the ranges of 0.6 and 0.8, with DP-CTGAN performing slightly better in privacy preservation than the counterpart.

Figure 6.8 shows random vs outlier targets from synthetic data generated from mostly.ai. In this plot, we see an interesting trend with outlier targets having lower privacy gain and higher AUC value indicating worse privacy preservation than random target records, which is quite different and opposite from the trends in the plot for the original dataset. Further examination and comparison amongst the actual outlier target record values would shed light into this anomalous observation.

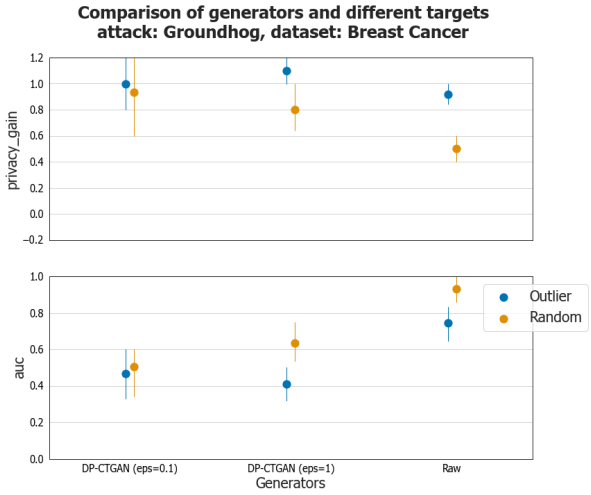

Figure 6.4: Random vs. outlier targets from BeGReaT Cancer dataset.

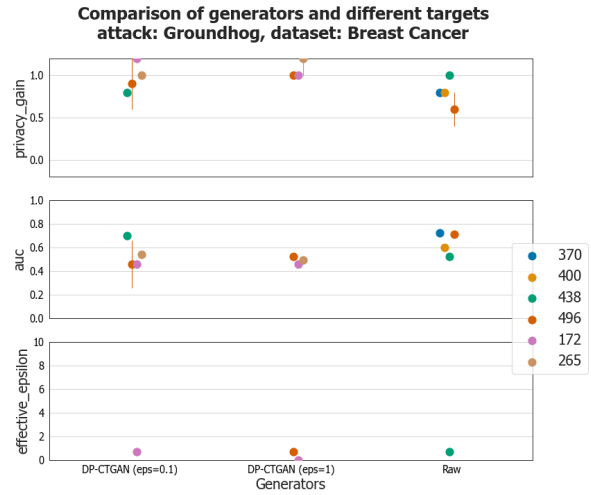

Figure 6.5: Comparing different generators on random targets from GPT4 Cancer dataset.

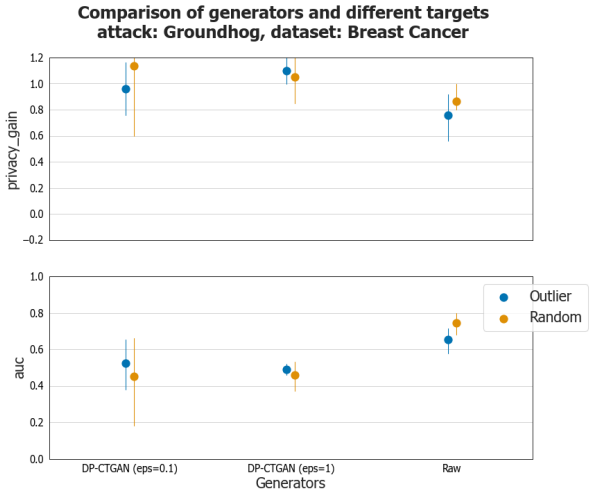

Figure 6.6: Random vs. outlier targets from GPT4 Cancer dataset.

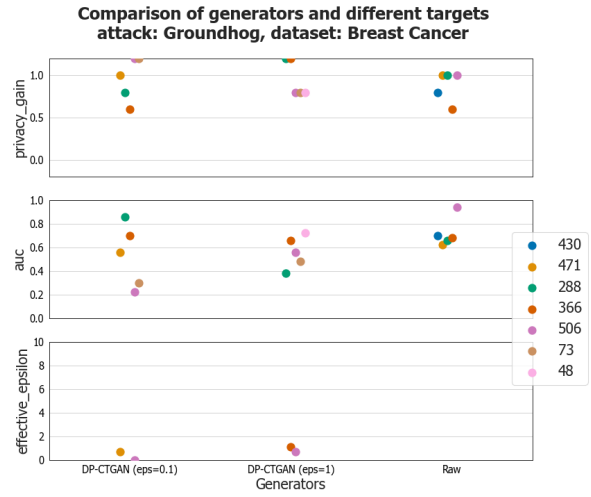

Figure 6.7: Comparing different generators on random targets from mostly.ai Cancer dataset.

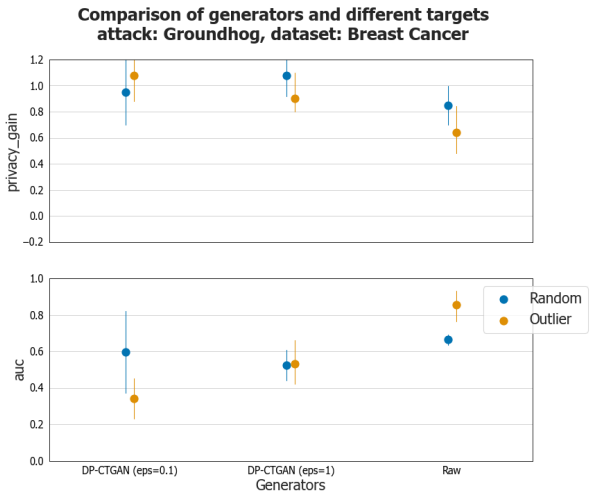

Figure 6.8: Random vs. outlier targets from mostly.ai Cancer dataset.

CHAPTER 7

Summary of the Results

The table 7.1 summarizes the result metrics to compare the averaged parameters for privacy preservation between the real dataset and synthetic data from different generators including DP-CTGAN (eps=0.1), DP-CTGAN (eps=1), GReaT, GPT4 and mostly.ai. Starting with comparison of the attack accuracy values, we see that the attack accuracy values for all the synthetic data generators are relatively the same, around 0.6. We see that the attack accuracy of the original dataset is the highest with a value of 0.7 signifying that it has the lowest privacy preservation, which makes sense since it is the real dataset without a differential privacy guarantee mechanism applied to it. The lowest accuracy value lies with DP-CTGAN (eps=0.1), which again makes sense because it has a strict epsilon value of 0.1, guaranteeing higher privacy preservation. Based on comparison of the accuracy values for synthetic data generator DP-CTGAN with LLM generators such as GReaT, GPT4 and mostl.ai, we can state that LLMs are comparable in performance with the traditional GAN based generator methods.

Examining the TPR values, we can notice a similar trend in the distribution of the metrics to the accuracy values, with the real dataset having the highest TPR value of 0.8 signifying high rate of correct classification by the attacker of target records, followed by GReaT, DT-CTGAN (eps=1), GPT4, mostl.ai, and finally, the lowest value of 0.4 for DT-CTGAN (eps=0.1). For the FPR which denotes values that are incorrectly classified as target records, the highest values of 0.5 belong to the real dataset and DP-CTGAN (eps=0.1), followed by similar values between 0.4-0.3 for DP-CTGAN (eps=1), GReaT, GPT4 and mostly.ai. Due

to similarity in the range of these values, we can conclude that LLMs perform comparably and similarly to other GAN-based generators. The MIA advantage quantifies the success of the attacks based on the attacker's success probability calculated based on the difference between the baseline accuracy of a random guess and accuracy of the attack. The highest MIA advantage value belongs to the real dataset and mostl.ai synthetic dataset, and there is comparable advantage for DP-CTGAN (eps=1), GReaT, GPT4 and finally, the lowest advantage value for DP-CTGAN (eps=0.1) due to its stricter epsilon value.

Looking at the Privacy Gain value that quantifies the gain and improvement in privacy after applying DP to a mechanism, the lowest privacy gain of 0.7 is associated with the real dataset and mostly.ai's data. Highest privacy gain belongs to DP-CTGAN (eps=0.1) because of its stricter epsilon value and similar values can be noticed in the range of 0.8-0.9 for DP-CTGAN (eps=1), GReaT and GPT4. Next, examining the AUC value, higher the value of AUC suggests better performance of the attack, and lower the value of AUC suggests worse performance of the attack. Looking at the AUC values, the real dataset, GReaT and mostly.ai have values in the range of 0.75-0.80. DP-CTGAN (eps=0.1), DP-CTGAN (eps=1) and GPT4 have similar AUC values with the range of 0.45-0.69, suggesting worse performance of the attack compared to the group mentioned earlier, meaning higher privacy preserving qualities. The effective epsilon value, ε^{eff}, portrays and quantifies a practical measure of privacy guarantee in experiments, as opposed to the theoretical guarantee of ε. Smaller the value of ε^{eff}, stronger the privacy guarantee of that generator. For the some of the generators, the ε^{eff} does not exist signaling that the privacy promise is not being fully upheld or numerical computation errors with floating point precision resulting in $\varepsilon^{\text{eff}} = \text{inf}$. DP-CTGAN (eps=0.1) has the smallest value of ε^{eff} signifying a strong privacy guarantee, followed by DP-CTGAN (eps=1) and GPT4 with values lower than 1, which is an acceptable value according to DP standards.

Based on the observations of the summary table of metrics from privacy attacks on different generators, we can conclude that the LLM-generated synthetic data performed

Metrics	Real	DP-CTGAN (eps=0.1)	DP-CTGAN (eps=1)	GReaT	GPT4	mostly.ai
Accuracy	0.7	0.5	0.6	0.6	0.6	0.6
True Positive Rate	0.8	0.4	0.5	0.6	0.5	0.5
False Positive Rate	0.5	0.5	0.3	0.4	0.3	0.2
Mia Advantage	0.3	-0.1	0.1	0.2	0.2	0.3
Privacy Gain	0.7	1.1	0.9	0.8	0.8	0.7
Auc	0.75	0.45	0.59	0.80	0.69	0.77
Effective Epsilon	inf	-0.07	0.35	inf	0.70	inf

Table 7.1: Summary Metrics for Different Generators

relatively similarly, and at times, better to the other traditional generator of DP-CTGAN and the synthetic data generator startup, mostly.ai. The best performing generator was DP-CTGAN (eps=0.1) with a stricter epsilon value. Amongst the LLMs, GReaT and GPT4 performed relatively and overall better in privacy preservation than the alternatives [1]. Both LLM generators performed comparably well to DP-CTGAN (eps=1). In summary, we can conclude that LLMs are a strong alternative to the traditional techniques of synthetic data generation since they exhibit acceptable privacy guarantees based on the privacy gain, AUC, MIA advantage, TPR/FPR, attack accuracy and ε^{eff} values.

[1]As can be observed in Table 7.1, for GReaT and mostly.ai, the fact that the ε^{eff} does not exist signals that the privacy promise is not being fully upheld or numerical computation errors in the TAPAS model with floating point precision resulting in $\varepsilon^{\text{eff}} = \text{inf}$, leads to concerns on the potential privacy leakage risks. Further investigation is necessary to evaluate risks.

CHAPTER 8

Conclusion

8.1 Conclusion and Future Work

In this paper, we used a healthcare dataset with sensitive user information of Breast Cancer to evaluate the differential privacy preserving metrics with adversarial attacks from the TAPAS toolbox. The paper compares and contrasts the data quality, data distributions and privacy-preserving metrics of the real dataset with synthetically generated datasets from several sources including LLMs from the GReaT framework and OpenAI's GPT4, Generative Adversarial Networks (GANs), and an AI-generated dataset produced using a proprietary technique from an industry startup, mostly.ai. The EDA comparing the original dataset's distribution with the distributions of the generators revealed the GReaT framework to have produced the most representative dataset conserving the outliers, skewness of the unimodal distributions, correlations as well as data variance. In conclusion, the experimental findings reveal that synthetic data generated from LLMs such as the GReaT framework and GPT4 is on par with the differential privacy guarantees of other traditional generator methods such as GANs.

For future work in privacy auditing techniques used in this paper, there is room for advancement and permutations in threat modeling such as black box auditing with data point canary, white box auditing with gradient canary and white box auditing with data point canary. Furthermore, for privacy auditing of more realistic synthetically generated data from LLMs, trying different attacks and different datasets with varying dimensions and

data types to generate synthetic data is a potential area of future research as well. For differential privacy as a field, there are many open questions and potential directions for advancement such as improving the trade-off in utility-preservation by efficient allocation of the privacy budget across multiple mechanisms in composition. For synthetic data generation techniques and models, future investigation and research would be beneficial on producing a representative set of outliers to generate highly realistic datasets. Moreover, research at the interaction of public policy, privacy regulations, ethics, AI and fairness to bridge the gap between theory and practical deployment of DP and synthetically generated datasets in industry applications is another growing field of future work.

CHAPTER 9

Appendices

- MASDS Thesis GitHub repository/README: link
- Privacy Auditing of Synthetic Data using TAPAS toolbox notebook: link
- Datasets directory: link
- Privacy Auditing Experiments reports: link
- BeGReaT framework data generation notebook: link
- EDA notebooks for real and synthetic datasets: link
- EDA HTML files: link

iter	target id	generator	acc	true pos rate	false neg rate
0	543	Raw	0.5	1.0	1.0
1	15	Raw	0.7	0.6	0.2
2	562	Raw	0.8	1.0	0.4
3	202	Raw	0.7	0.8	0.4
4	258	Raw	0.7	0.4	0.0
5	505	Raw	0.6	1.0	0.8
6	151	Raw	0.7	1.0	0.6
7	543	DP-CTGAN (eps=0.1)	0.5	0.2	0.2
8	15	DP-CTGAN (eps=0.1)	0.4	0.2	0.4
9	562	DP-CTGAN (eps=0.1)	0.6	0.6	0.4
10	202	DP-CTGAN (eps=0.1)	0.3	0.4	0.8
11	258	DP-CTGAN (eps=0.1)	0.7	0.8	0.4
12	505	DP-CTGAN (eps=0.1)	0.5	0.0	0.0
13	151	DP-CTGAN (eps=0.1)	0.3	0.6	1.0
14	543	DP-CTGAN (eps=1)	0.6	0.4	0.2
15	15	DP-CTGAN (eps=1)	0.5	0.4	0.4
16	562	DP-CTGAN (eps=1)	0.5	0.2	0.2
17	202	DP-CTGAN (eps=1)	0.8	0.8	0.2
18	258	DP-CTGAN (eps=1)	0.5	0.6	0.6
19	505	DP-CTGAN (eps=1)	0.6	0.2	0.0
20	151	DP-CTGAN (eps=1)	0.5	0.6	0.6

Table 9.1: Experiment metrics for the real Breast Cancer dataset with Groundhog attack

iter	mia advantage	privacy gain	auc	effective epsilon
0	0.0	1.0	0.48	inf
1	0.4	0.6	0.60	inf
2	0.6	0.4	0.84	inf
3	0.4	0.6	0.72	inf
4	0.4	0.6	0.90	inf
5	0.2	0.8	0.80	inf
6	0.4	0.6	0.88	inf
7	0.0	1.0	0.28	0
8	-0.2	1.2	0.34	inf
9	0.2	0.8	0.78	inf
10	-0.4	1.4	0.22	0
11	0.4	0.6	0.72	inf
12	0.0	1.0	0.58	inf
13	-0.4	1.4	0.24	-0.223144
14	0.2	0.8	0.56	inf
15	0.0	1.0	0.74	inf
16	0.0	1.0	0.32	0
17	0.6	0.4	0.70	inf
18	0.0	1.0	0.68	inf
19	0.2	0.8	0.66	inf
20	0.0	1.0	0.46	0.693147

Table 9.2: Experiment metrics for the real Breast Cancer dataset with Groundhog attack (continued)

iter	target id	generator	acc	true pos rate	false neg rate
0	430	Raw	0.6	0.6	0.4
1	471	Raw	0.5	0.4	0.4
2	288	Raw	0.5	0.0	0.0
3	366	Raw	0.7	1.0	0.6
4	506	Raw	0.5	0.0	0.0
5	73	Raw	0.8	0.8	0.2
6	48	Raw	0.7	0.4	0.0
7	430	DP-CTGAN (eps=0.1)	0.3	0.2	0.6
8	471	DP-CTGAN (eps=0.1)	0.5	0.6	0.6
9	288	DP-CTGAN (eps=0.1)	0.6	0.4	0.2
10	366	DP-CTGAN (eps=0.1)	0.7	0.6	0.2
11	506	DP-CTGAN (eps=0.1)	0.4	0.2	0.4
12	73	DP-CTGAN (eps=0.1)	0.4	0.0	0.2
13	48	DP-CTGAN (eps=0.1)	0.6	0.4	0.2
14	239	DP-CTGAN (eps=0.1)	0.6	0.8	0.6
15	430	DP-CTGAN (eps=1)	0.5	0.6	0.6
16	471	DP-CTGAN (eps=1)	0.4	0.0	0.2
17	288	DP-CTGAN (eps=1)	0.4	0.6	0.8
18	366	DP-CTGAN (eps=1)	0.4	0.8	1.0
19	506	DP-CTGAN (eps=1)	0.6	0.4	0.2
20	73	DP-CTGAN (eps=1)	0.6	0.8	0.6

Table 9.3: Experiment metrics for the synthetic Breast Cancer dataset from mostly.ai with Groundhog attack

iter	mia advantage	privacy gain	auc	effective epsilon
0	0.2	0.8	0.70	inf
1	0.0	1.0	0.62	inf
2	0.0	1.0	0.66	inf
3	0.4	0.6	0.68	inf
4	0.0	1.0	0.94	inf
5	0.6	0.4	0.82	inf
6	0.4	0.6	0.96	inf
7	-0.4	1.4	0.26	0
8	0.0	1.0	0.56	0.693147
9	0.2	0.8	0.86	inf
10	0.4	0.6	0.70	inf
11	-0.2	1.2	0.22	0
12	-0.2	1.2	0.30	inf
13	0.2	0.8	0.46	inf
14	-0.4	1.4	0.20	-0.223144
15	0.0	1.0	0.44	0
16	-0.2	1.2	0.58	0.693147
17	-0.2	1.2	0.38	inf
18	-0.2	1.2	0.66	1.098612
19	0.2	0.8	0.56	0.693147
20	0.2	0.8	0.48	inf

Table 9.4: Experiment metrics for the synthetic Breast Cancer dataset from mostly.ai with Groundhog attack (continued)

iter	target id	generator	acc	true pos rate	false neg rate
0	370	Raw	0.6	0.2	0.0
1	400	Raw	0.6	0.8	0.6
2	438	Raw	0.5	0.2	0.2
3	496	Raw	0.6	0.6	0.4
4	496	Raw	0.8	1.0	0.4
5	172	Raw	0.6	0.8	0.6
6	265	Raw	0.6	0.2	0.0
7	370	DP-CTGAN (eps=0.1)	0.7	0.8	0.4
8	400	DP-CTGAN (eps=0.1)	0.2	0.2	0.8
9	438	DP-CTGAN (eps=0.1)	0.6	0.4	0.2
10	496	DP-CTGAN (eps=0.1)	0.4	0.4	0.6
11	496	DP-CTGAN (eps=0.1)	0.7	0.6	0.2
12	172	DP-CTGAN (eps=0.1)	0.4	0.8	1.0
13	265	DP-CTGAN (eps=0.1)	0.5	0.2	0.2
14	265	DP-CTGAN (eps=0.1)	0.4	0.2	0.4
15	370	DP-CTGAN (eps=1)	0.5	0.4	0.4
16	400	DP-CTGAN (eps=1)	0.5	0.8	0.8
17	438	DP-CTGAN (eps=1)	0.3	0.2	0.6
18	496	DP-CTGAN (eps=1)	0.6	0.2	0.0
19	496	DP-CTGAN (eps=1)	0.5	0.8	0.8
20	172	DP-CTGAN (eps=1)	0.5	0.4	0.4

Table 9.5: Experiment metrics for the synthetic Breast Cancer dataset from OpenAI's GPT4 with Groundhog attack

iter	mia advantage	privacy gain	auc	effective epsilon
0	0.2	0.8	0.72	inf
1	0.2	0.8	0.60	inf
2	0.0	1.0	0.52	0.693147
3	0.2	0.8	0.70	inf
4	0.6	0.4	0.72	inf
5	0.2	0.8	0.68	inf
6	0.2	0.8	0.76	inf
7	-0.6	1.6	0.18	0
8	0.2	0.8	0.70	inf
9	-0.2	1.2	0.26	0
10	0.4	0.6	0.66	inf
11	-0.2	1.2	0.46	0.693147
12	0.0	1.0	0.54	inf
13	-0.2	1.2	0.52	inf
14	0.0	1.0	0.52	0.693147
15	0.0	1.0	0.54	0.693147
16	-0.4	1.4	0.32	0
17	0.2	0.8	0.46	inf
18	0.0	1.0	0.52	0.693147
19	0.0	1.0	0.46	0
20	-0.4	1.4	0.46	0.287682

Table 9.6: Experiment metrics for the synthetic Breast Cancer dataset from OpenAI's GPT4 with Groundhog attack (continued)

iter	target id	generator	acc	true pos rate	false neg rate
0	331	Raw	0.5	0.4	0.4
1	59	Raw	0.5	1.0	1.0
2	178	Raw	0.6	0.4	0.2
3	357	Raw	0.5	1.0	1.0
4	114	Raw	0.6	0.2	0.0
5	197	Raw	0.7	0.4	0.0
6	150	Raw	0.8	1.0	0.4
7	331	DP-CTGAN (eps=0.1)	0.3	0.6	1.0
8	59	DP-CTGAN (eps=0.1)	0.7	0.8	0.4
9	178	DP-CTGAN (eps=0.1)	0.6	1.0	0.8
10	357	DP-CTGAN (eps=0.1)	0.3	0.0	0.4
11	114	DP-CTGAN (eps=0.1)	0.5	0.4	0.4
12	197	DP-CTGAN (eps=0.1)	0.5	0.6	0.6
13	150	DP-CTGAN (eps=0.1)	0.7	0.6	0.2
14	331	DP-CTGAN (eps=1)	0.4	0.6	0.8
15	59	DP-CTGAN (eps=1)	0.4	0.4	0.6
16	178	DP-CTGAN (eps=1)	0.6	0.8	0.6
17	357	DP-CTGAN (eps=1)	0.7	0.6	0.2
18	114	DP-CTGAN (eps=1)	0.7	0.6	0.2
19	197	DP-CTGAN (eps=1)	0.6	0.2	0.0
20	150	DP-CTGAN (eps=1)	0.5	0.4	0.4

Table 9.7: Experiment metrics for the synthetic Breast Cancer dataset from GReaT framework with Groundhog attack

iter	mia advantage	privacy gain	auc	effective epsilon
0	0.0	1.0	0.58	inf
1	0.0	1.0	0.80	inf
2	0.2	0.8	0.66	inf
3	0.0	1.0	0.86	inf
4	0.2	0.8	0.82	inf
5	0.4	0.6	1.00	inf
6	0.6	0.4	0.86	inf
7	-0.4	1.4	0.34	0
8	0.4	0.6	0.58	inf
9	0.2	0.8	0.60	inf
10	-0.4	1.4	0.34	0
11	0.0	1.0	0.54	0.693147
12	0.0	1.0	0.32	inf
13	0.4	0.6	0.66	1.098612
14	-0.2	1.2	0.50	inf
15	-0.2	1.2	0.48	inf
16	0.2	0.8	0.74	inf
17	0.4	0.6	0.80	1.609438
18	0.4	0.6	0.60	1.098612
19	0.2	0.8	0.56	0.693147
20	0.0	1.0	0.32	0

Table 9.8: Experiment metrics for the synthetic Breast Cancer dataset from GReaT framework with Groundhog attack (continued)

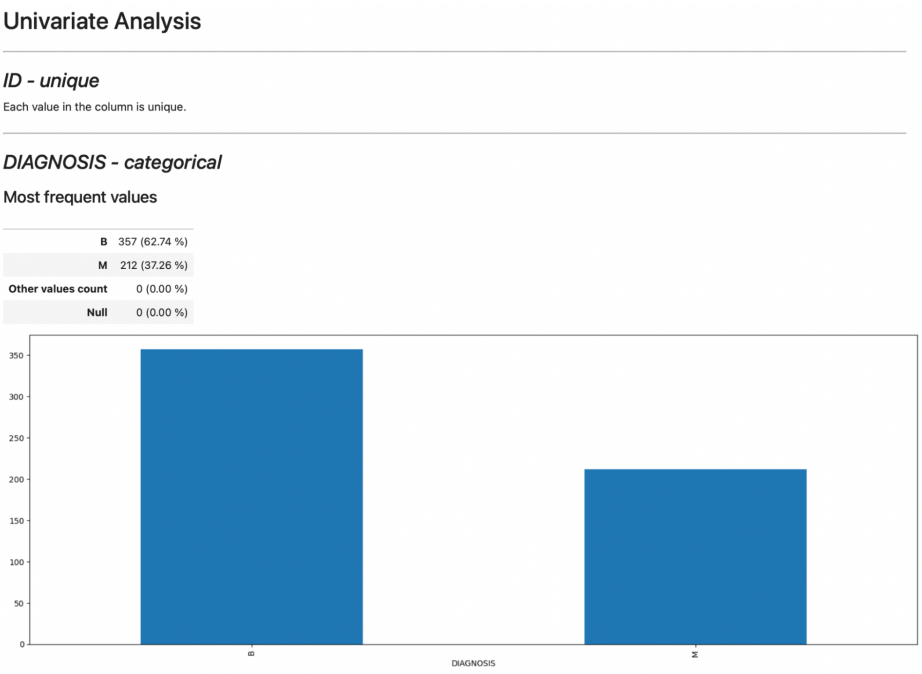

Figure 9.1: Univariate Analysis for Real Dataset

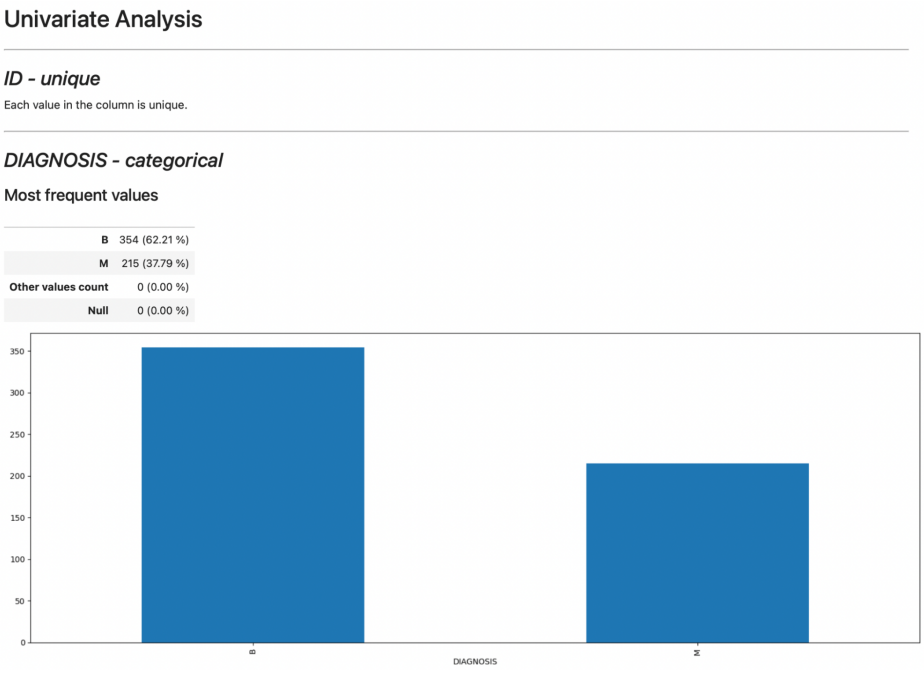

Figure 9.2: Univariate Analysis for Synthetic Dataset from GPT4

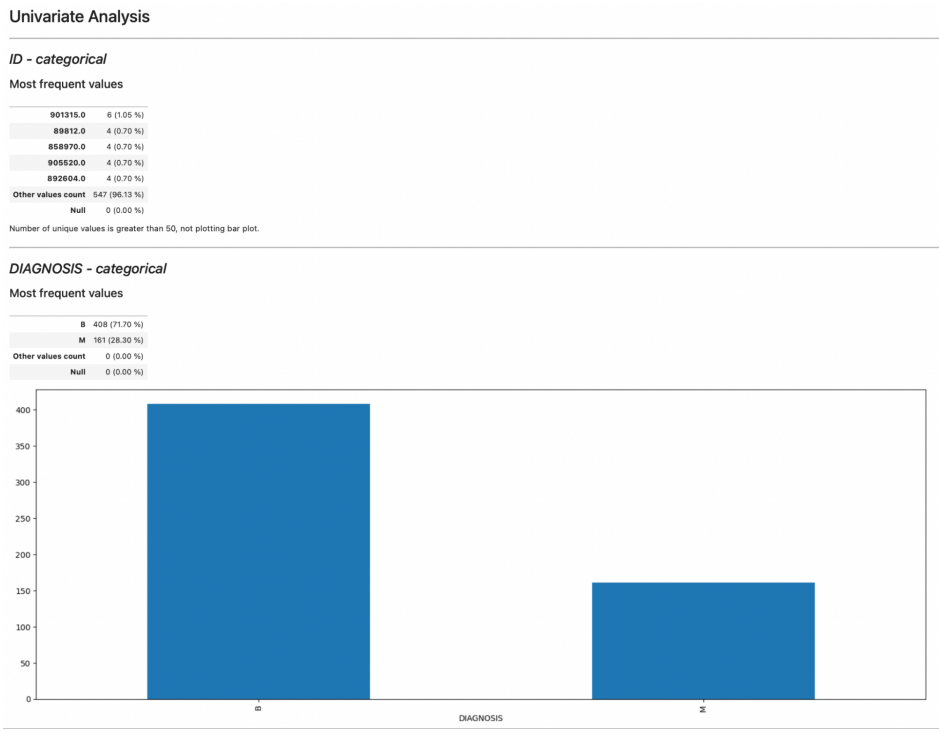

Figure 9.3: Univariate Analysis for Synthetic Dataset from GReaT framework

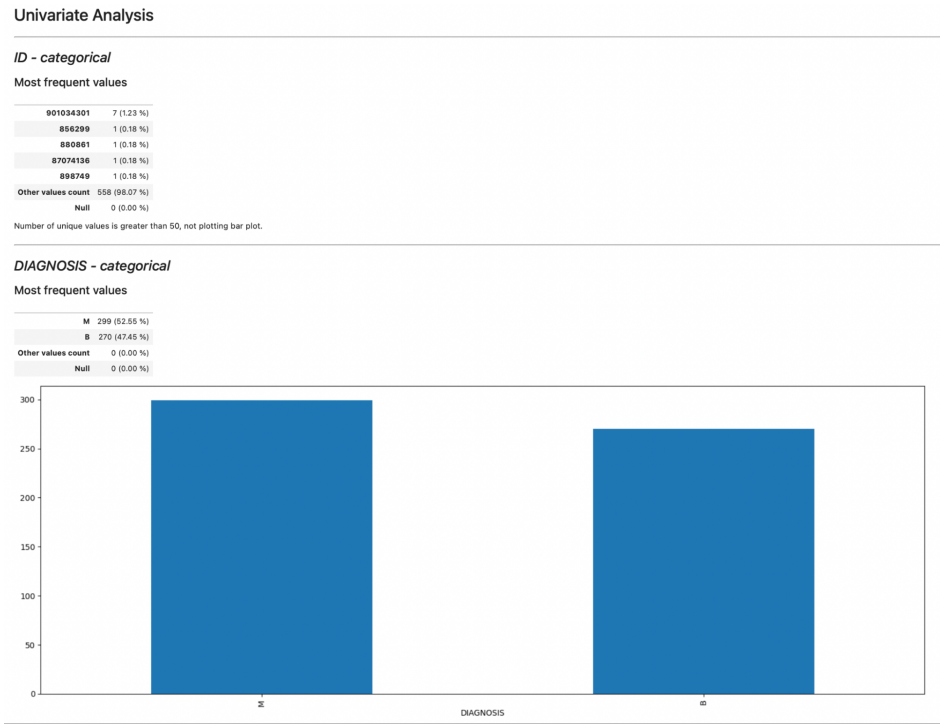

Figure 9.4: Univariate Analysis for Synthetic Dataset from mostly.ai

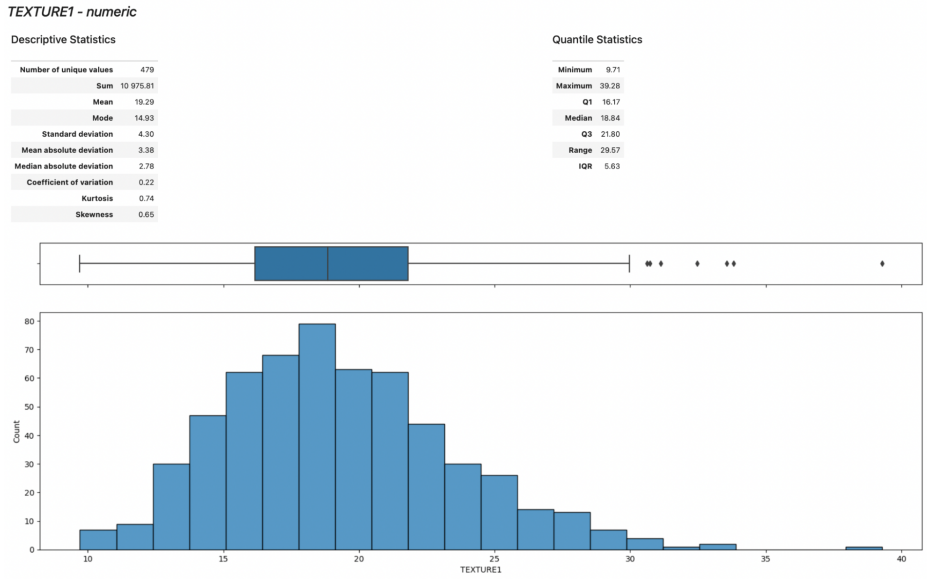

Figure 9.5: Texture1 Summary Statistics for Real Dataset

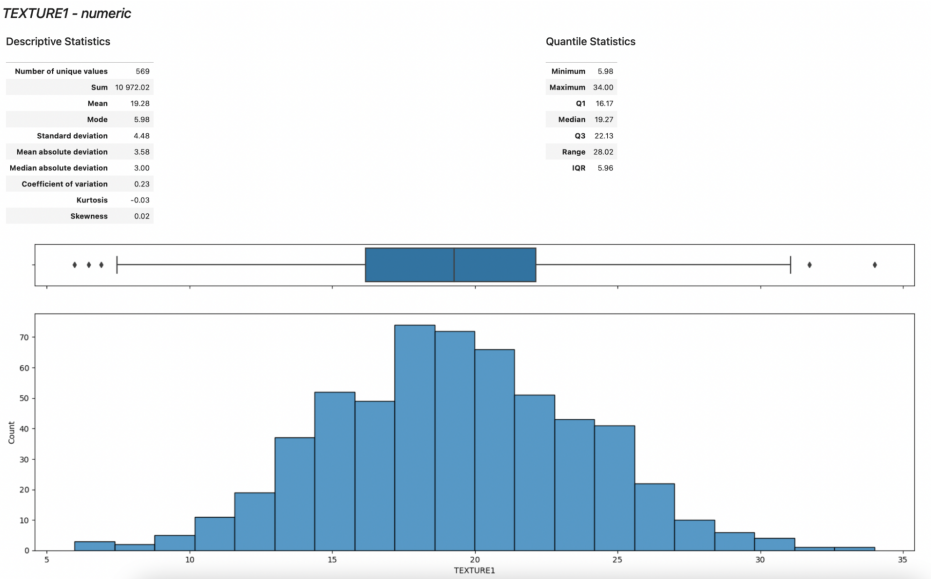

Figure 9.6: Texture1 Summary Statistics for Synthetic Dataset from GPT4

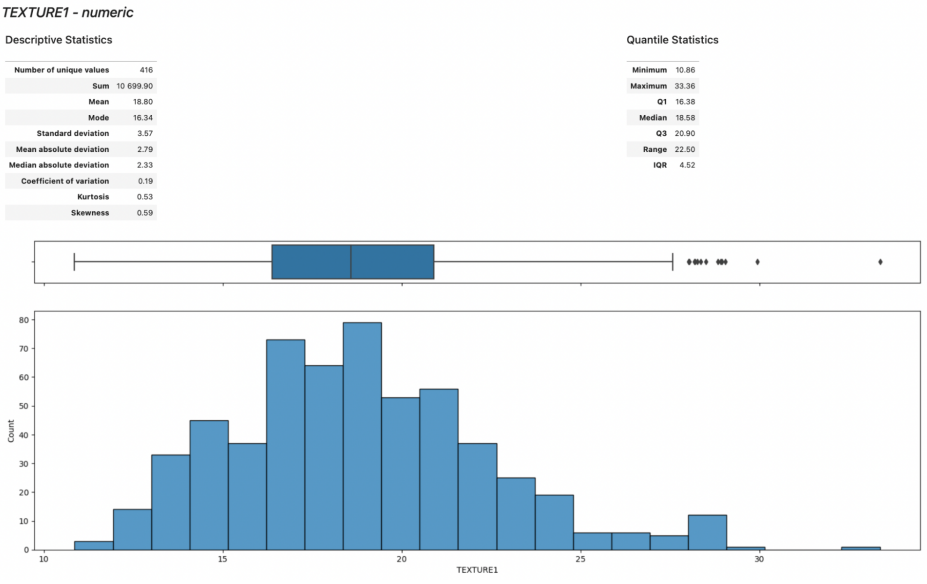

Figure 9.7: Texture1 Summary Statistics for Synthetic Dataset from GReaT framework

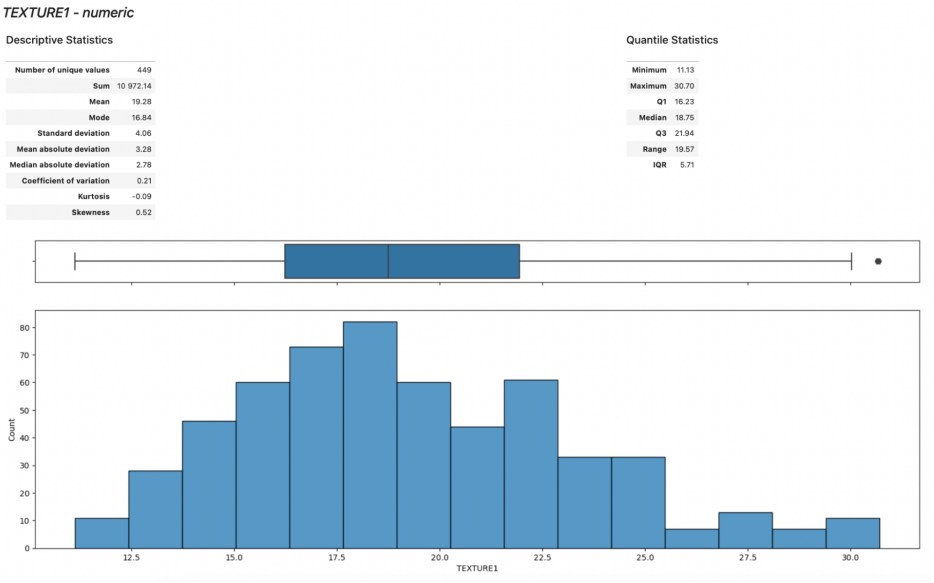

Figure 9.8: Texture1 Summary Statistics for Synthetic Dataset from mostly.ai

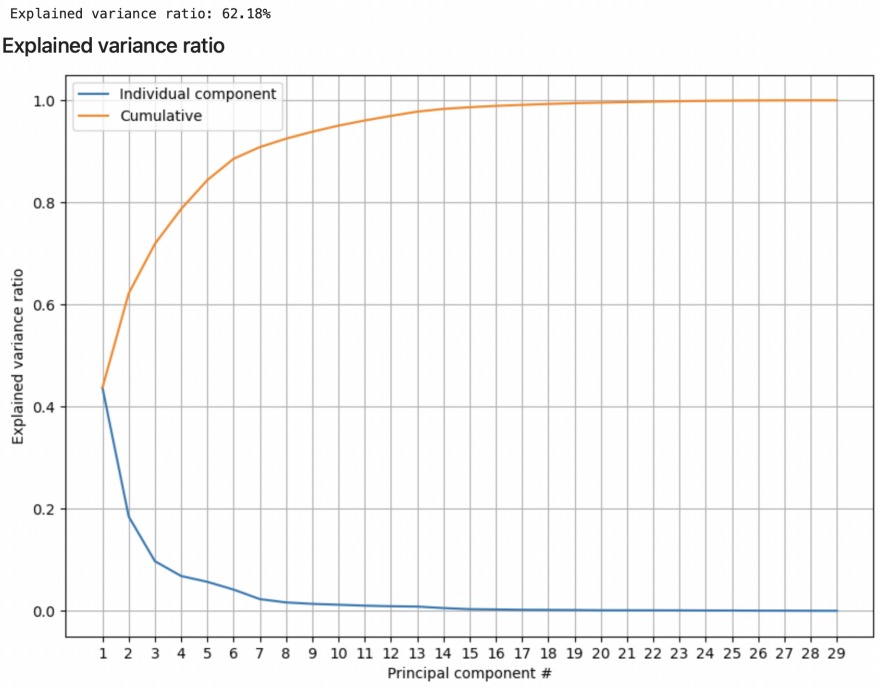

Figure 9.9: Explained variance ratio vs. principal components for Real Dataset

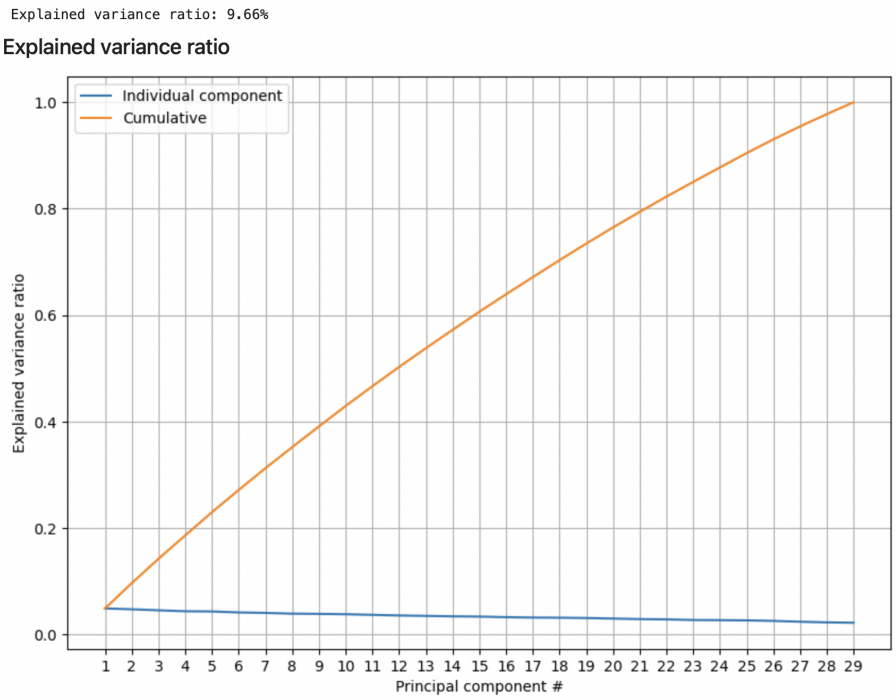

Figure 9.10: Explained variance ratio vs. principal components for Synthetic Dataset from GPT4

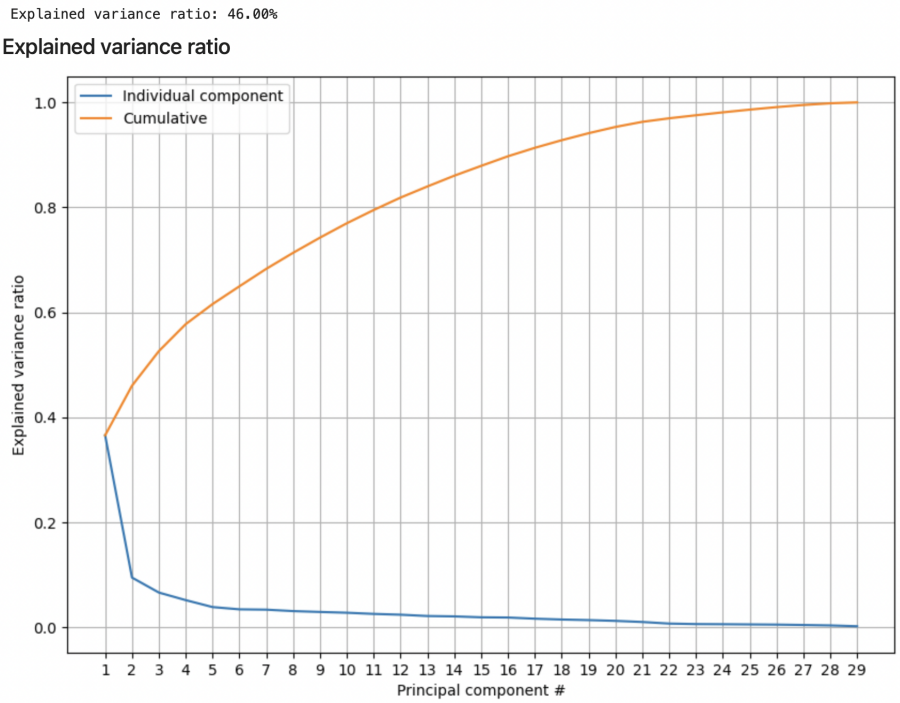

Figure 9.11: Explained variance ratio vs. principal components for Synthetic Dataset from GReaT framework

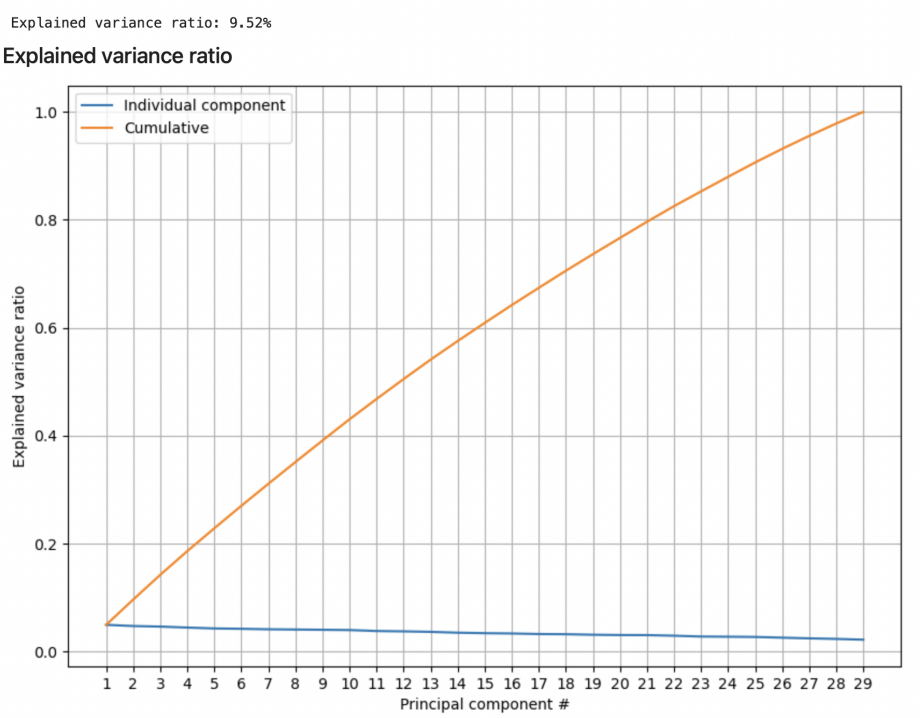

Figure 9.12: Explained variance ratio vs. principal components for Synthetic Dataset from mostly.ai

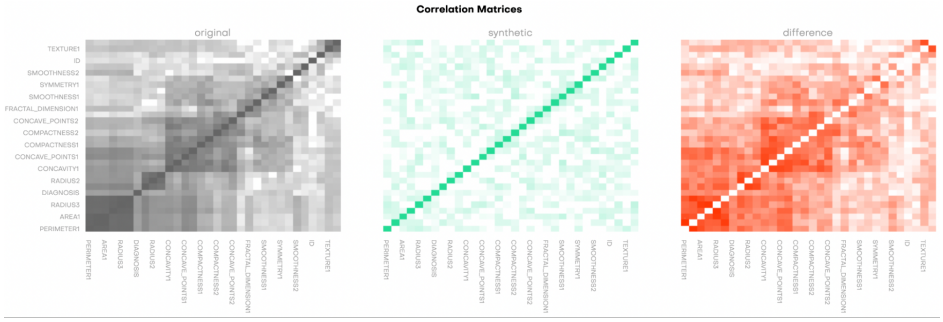

Figure 9.13: Correlation Matrices between original and synthetic dataset for Synthetic Dataset from mostly.ai

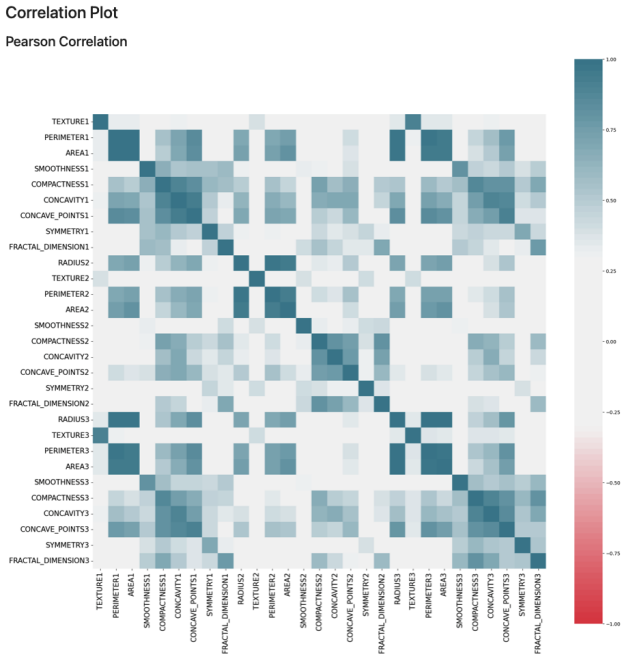

Figure 9.14: Pearson Correlation Plot of the Real Dataset

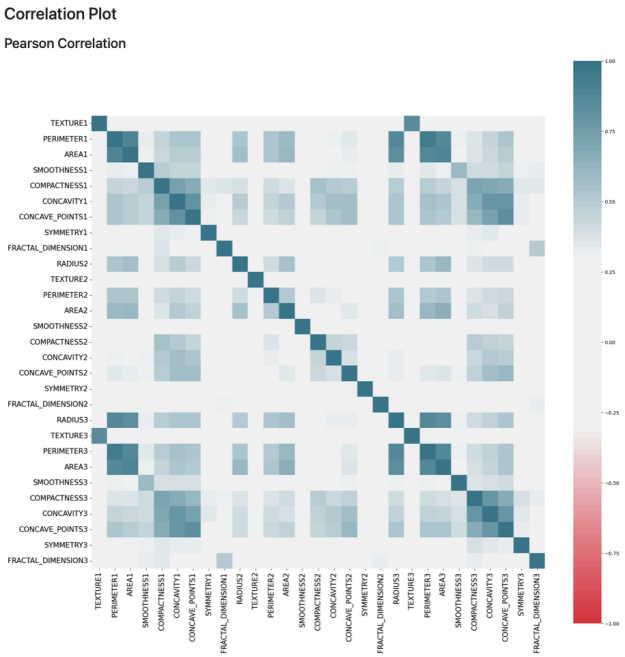

Figure 9.15: Pearson Correlation Plot of the GReaT Dataset

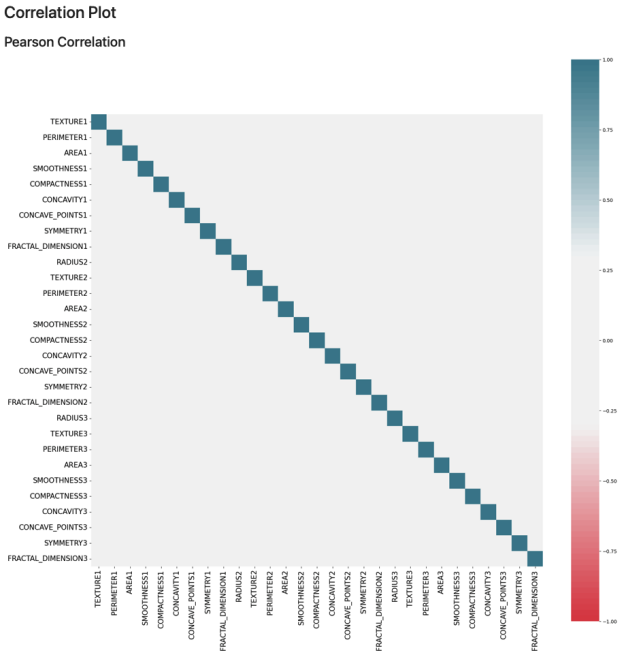

Figure 9.16: Pearson Correlation Plot of the mostly.ai Dataset

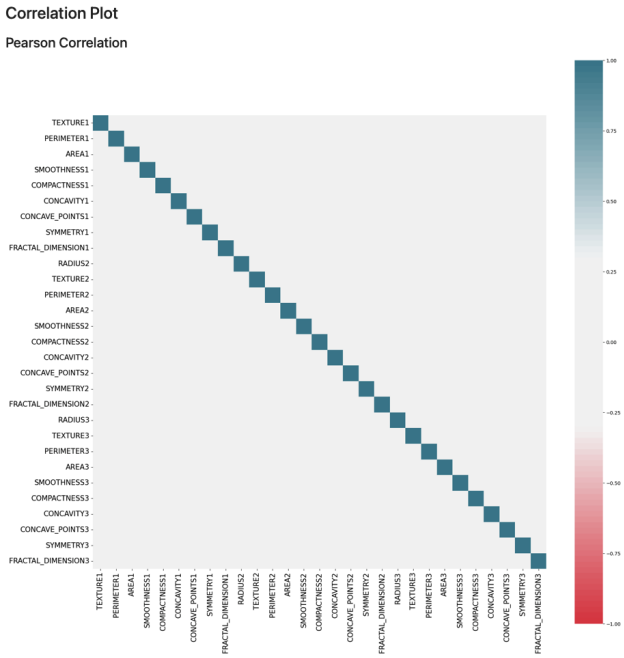

Figure 9.17: Pearson Correlation Plot of the GPT4 Dataset

REFERENCES

[1] Apple inc. differential privacy technical overview. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf/, 2023.

[2] Edvart python library for eda. https://datamole-ai.github.io/edvart/index.html, 2023.

[3] Test data automation for ci/cd workflows, tonic.ai, 2024.

[4] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators, 2023.

[5] datagen.tech. Techniques for generating synthetic data. https://datagen.tech/guides/synthetic-data/synthetic-data-generation/, 2023.

[6] Cem Dilmegani. Synthetic data generation: Techniques, best practices and tools, 2024.

[7] Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Shafiq Joty, Boyang Li, and Lidong Bing. Is gpt-3 a good data annotator?, 2023.

[8] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[9] Neil Zhenqiang Gong and Bin Liu. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 979–995, Austin, TX, August 2016. USENIX Association.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[11] F Houssiau, J Jordon, SN Cohen, O Daniel, A Elliott, J Geddes, C Mole, C Rangel-Smith, and L Szpruch. Tapas: a toolbox for adversarial privacy auditing of synthetic data. 2022.

[12] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd?, 2020.

[13] K2ViewEBook. Synthetic data generation: The complete handbook, 2023.

[14] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models, 2023.

[15] Rumeng Li, Xun Wang, and Hong Yu. Two directions for clinical data generation with large language models: Data-to-label and label-to-data. *PubMed*, 2023:7129–7143, 2023.

[16] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*, 2023.

[17] Joseph P. Near and Chiké Abuah. *Programming Differential Privacy*, volume 1. 2021.

[18] OpenAI. Innovations in ai: Openai's recent developments. `https://www.openai.com/research/`, Jan 2023. Accessed: 2024-01-15.

[19] OpenMined.org. Use cases of differential privacy, 2020.

[20] Manuel Pasieka. A comparison of synthetic data generation methods, mostly.ai, 2023.

[21] IEEE Digital Privacy. Differential privacy and applications, 2024.

[22] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget, 2023.

[23] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data – anonymisation groundhog day, 2022.

[24] Latanya Sweeney. Simple demographics often identify people uniquely, 2000.

[25] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation, 2020.

[26] Turing. Synthetic data generation: Definition, types, techniques, and tools, 2023.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[28] Akhil Arora Martin Josifoski Ashton Anderson Robert West Veniamin Veselovsky, Manoel Horta Ribeiro. Generating faithful synthetic data with large language models: A case study in computational social science. *arXiv preprint arXiv:2305.15041*, 2023.

[29] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? gpt-3 can help, 2021.

[30] Alex Watson. How to generate synthetic data: Tools and techniques to create interchangeable datasets, gretel.ai, 2022.

[31] Mangasarian Olvi Street Nick Wolberg, William and W. Street. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C5DW2B.

[32] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, 2019.

[33] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.