Alma Mater Studiorum - Università di Bologna

Department of Statistical Sciences "Paolo Fortunati"

Second Cycle Degree/Two-Year Master in Statistical Sciences

Privacy-Enhancing Technologies for Synthetic Data Creation with Deep Generative Models

Presented by:
Alessio Crisafulli Carpani

0000878169

Supervisor:
Prof. Cinzia Viroli

Co-Supervisor:
Massimo De Cubellis, ISTAT

Academic Year 2022/2023
Session II

Abstract

In light of the recent technological advancements, our society has evolved into a prolific source of data generation, accompanied by the widespread use of machine learning algorithms, particularly deep neural networks. However, these algorithms rely on substantial datasets which often contain sensitive and private information.

Within this context, generative models have emerged to create synthetic samples across various domains. Ideally, these models should prevent the exposure of individual-specific information from the training data. Unfortunately, recent literature has shown that this assumption is not consistently met, particularly with Generative Adversarial Networks (GANs), which lacks of robust privacy guarantees.

Nevertheless, there is a critical need to strike a balance between our responsibilities as data stewards and the importance to advance data mining research. In this regard, Privacy-Enhancing Technologies (PETs) can help mitigate these challenges by imposing privacy constraints on ML models or more generally in algorithms, enabling their use and sharing without compromising the confidentiality of the training data.

This research is dedicated to exploring the latest techniques in the field of privacy, leveraging differentially private synthetic data and investigating the trade-off between data utility and privacy preservation. The outcomes of this study have the potential to:

- Enhance the understanding of the latest techniques for generating synthetic data while respecting the principles of differential privacy.
- Provide insights about the trade-off between data utility and privacy preservation, specifically in the context of generative models.
- Furnish guidance to researchers, organizations, and policymakers on the practical application of differential privacy-enhanced synthetic data.
- Contribute to the development of best practices for leveraging synthetic data in data-driven tasks while adhering to stringent privacy regulations.

This thesis is the outcome of a research project conducted for the Italian National Institute of Statistics (ISTAT).

> *The GDPR has become a synonym for effective data protection legislation worldwide. Now its application will decide on its full success. While the independent authorities are doing enormous work, it is time to ensure that we can act more quickly and decisively, especially in serious cases where a breach can cause many victims across the EU*

— **Věra Jourová**
Vice-President for Values and Transparency,
European Commission

Declaration

I hereby formally declare that I have authored the submitted dissertation independently. I have exclusively relied on the quoted literature and other sources referenced in the paper for support. All literature and other sources used in the creation of this academic work have been distinctly marked and listed separately, whether they were directly quoted or used as content references

Bologna, November 14, 2023

Alessio Crisafulli Carpani

Table of contents

List of Figures

List of Algorithms

List of Tables

Chapter 4

Introduction

1

In recent decades, our global society has experienced a series of remarkable technological advancements that have significantly reshaped various socio-economic aspects of the world. One standout achievement among these milestones is the extensive adoption of the Internet and social networks, which has not only profoundly influenced individuals but also organizations. This digital transformation has given rise to a concept known as *"datafication"* as discussed in Cukier's work [Cukier and Mayer-Schoenberger], where every event or state, whether occurring in the physical or digital realm, is methodically converted into data. These data are then gathered, processed, and subjected to analysis, effectively converting our society, economy, and physical environment into expansive reservoirs of data, resembling what could be termed as *"data fountains"* [Ricciato, Fabio et al., 2020]. Virtually all of our daily activities now serve as opportunities for data collection.

The utilization of data, particularly datasets containing micro-level, individual-specific information, have drawn significant attention in the realm of data mining research. In today's world, numerous real-world systems heavily depend on machine learning (ML) models to carry out a diverse range of tasks, including uncovering novel data patterns, identifying anomalies, and facilitating recommendation systems. However, a significant challenge arises, as many of these ML algorithms have an insatiable demand for data [Cao et al., 2021], often necessitating the inclusion of personal information.

The crux of this challenge lies in the acquisition of these extensive datasets via crowdsourcing platforms, which may contain legally protected information about individuals. This information encompasses various domains, such as medical, financial, behavioral, transactional, and even political preferences [1]. Moreover, it may extend to include location data and images.

Machine learning systems that have been trained on sensitive user data are vulnerable to privacy breaches [Hayes et al., 2018]. The majority of these breaches are associated with a phenomenon known as *overfitting* [Shokri et al.], wherein a model's training error significantly exceeds its test error. Overfitting implies that the model has effectively memorized the sensitive, personal, or private data utilized during training, as demonstrated by various attacks conducted by researchers such as [Shokri et al.], [Song and Shmatikov, 2019], and [Carlini et al.], among others.

[1] Facebook, Cambridge Analytica scandal (CNBC)

For instance, [Shokri et al.] illustrated a method for determining whether a specific record was part of the training dataset for a given ML model. Their technique achieved accuracy rates of 74% and 94% when tested on Amazon and Google Cloud machine learning systems, respectively.

In recent years, major cloud providers like Google[2], IBM[3], Microsoft[4], and Amazon[5] have introduced software solutions aimed at simplifying machine learning tasks within their applications. They offer these capabilities to customers through a suite of APIs, under the concept known as **Machine Learning as a Service (MLaaS)**. This approach has gained popularity among organizations looking to leverage robust ML engines for complex tasks, while avoiding the challenges associated with building such infrastructure from the ground up. However, it's crucial to recognize that if malicious actors were to obtain the data used in training these models, the resultant data leaks could have severe repercussions.

Furthermore, *transfer learning*, heralded as the next frontier in the advancement of machine learning, empowers the utilization of pre-existing, sophisticated models stored on devices, obviating the necessity to initiate model training from scratch. Although this approach offers benefits in terms of reduced latency and enhanced energy efficiency, it gives rise to concerns due to the fact that these models are publicly accessible in model zoo repositories, aligning with open-source principles. Consequently, the exposure of model parameters and training data within these repositories can be susceptible to exploitation in privacy attacks.

1.1 Motivation

In navigating the complex terrain of machine learning applications, the organizations responsible for these technologies must strike a delicate balance. They are tasked with the responsibility of managing their data in a responsible manner, thereby minimizing the risks associated with data loss, theft, and misuse. Simultaneously, they must cater to the needs of the "modeler," whose primary role revolves around the development and refinement of these applications.

In the realm of research, the primary objective is not to replace existing data sources but to augment them with fresh data streams. This scenario has engendered debates and elicited responses from various organizations and public administrations, spurring discussions on effective strategies to address the situation. An illustration of this can be found in the European Union's **General Data Protection Regulation (GDPR)** [Council of the European Union, 2016], which specifically deals with data protection, privacy, and the transfer of personal data, granting users increased

[2] Google Vertex AI
[3] Watson Machine Learning - IBM Cloud
[4] Microsoft Azure ML
[5] Amazon Machine Learning on AWS

control over their personal information. Under the GDPR, businesses are allowed to collect anonymized data without explicit consent, utilize it for diverse purposes, and store it indefinitely. Additionally, the European Commission's White Paper on Artificial Intelligence [Commission, 2020] underscores the transformative potential of artificial intelligence across multiple domains, including healthcare, agriculture, climate change mitigation, production efficiency, and security.

National Statistical Offices (NSOs), alongside other relevant institutions, have the critical responsibility of providing reliable, pertinent, timely, and high-quality data to support evidence-based decision-making. In many instances, NSOs gather sensitive data pertaining to individuals and businesses through surveys and censuses, encompassing information such as population census data or data from household and business surveys. Nevertheless, to respond effectively to emerging issues, NSOs often require supplementary data from secondary sources, including administrative or private sector data.

This scenario calls for a coordinated international response, necessitating timely access to potentially sensitive data shared among multiple partners, some of whom may be located in different countries. However, due to legitimate privacy concerns, unrestricted access to all data cannot be granted to these partners.

NSOs possess data that holds the potential to fuel innovation and improve national services, research endeavors, and societal well-being. However, there has been a notable increase in sustained cyber threats, complex networks of intermediaries motivated to acquire sensitive data, and advancements in techniques for re-identifying and linking data to individuals across multiple sources.

The utilization of micro-data is typically regulated by a range of legal frameworks. One notable example pertains to National Statistical Offices (NSOs), which have long confronted the challenge of safeguarding confidentiality, despite the widely recognized importance of this data. This issue is exemplified in a 1993 White Paper on Open Government in the United Kingdom[6]:

> "Open access to statistics provides the citizen with more than a picture of society. It offers a window on the work and performance of government itself, showing the scale of government activity in every area of public policy and allowing the impact of public policies and actions to be assessed."

To address these challenges, the U.S. Census Bureau, for instance, has implemented the use of differential privacy[7][8] as a means to protect individual privacy while still enabling the release of aggregated population statistics.

[6] White Paper on Open Government
[7] 2020 Decennial Census: Processing the Count: Disclosure Avoidance Modernization
[8] Differential Privacy and the 2020 US Census

In a similar way, the University of California, Berkeley, utilizes differential privacy[9] as a crucial tool in their efforts to study the transmission of infectious diseases, including influenza and COVID-19. This approach allows them to gather valuable insights without compromising the confidentiality and identities of individual patients.

Apple's Differential Privacy Team has been at the forefront of this endeavor, pioneering the development of efficient and scalable local differentially private algorithms [Apple, 2017]. Their goal is to improve user experience and extract valuable insights from data while upholding user privacy. In this approach, each individual user applies privacy measures to their data before transmitting it to a centralized server, ensuring a high level of data protection.

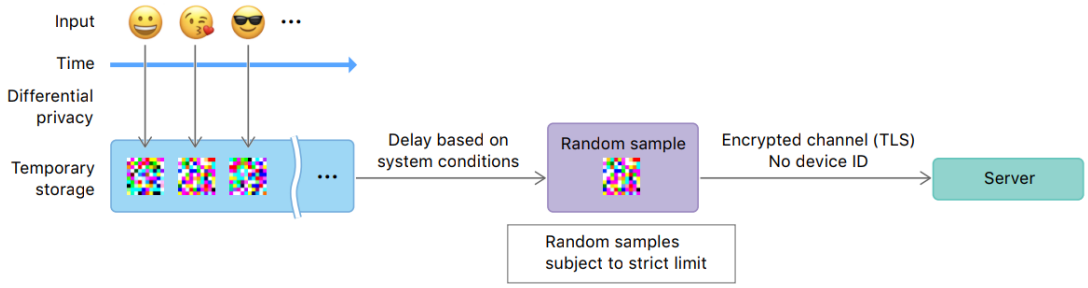


FIGURE 1.1 | Apple uses DP to collect some data from end-user devices running iOS or macOS [Apple, 2017]

This underscores the viability of achieving a delicate equilibrium between data utility, privacy preservation, and server computation, a pivotal consideration within this context.

In this context, a paramount concern is the protection of **personally identifiable information (PII)**, which encompasses data that can be leveraged to identify an individual. In addition to shielding PII from potential breaches, companies must also adhere to a range of data protection regulations, such as Europe's GDPR.

Cyberattacks not only jeopardize the individuals whose information is at risk but also impose legal, financial, and reputational perils upon organizations engaged in data collection. The mere removal of obvious PII, such as names and addresses, is insufficient, as other quasi-identifiers can still be exploited to pinpoint an individual within a dataset.

Pentland [10] revisited this subject in an article for the World Economic Forum, highlighting the advancing intelligence of mobile telephone networks and their function as intelligent, responsive systems equipped with sensors that act as their eyes and ears.

A prevailing paradigm in contemporary machine learning revolves around a centralized authority that holds exclusive access to data sourced from a large user base. Several prominent entities,

[9] University of California Berkeleys - DP
[10] The Global Information Technology Report 2008–2009

including corporations, universities, and government institutions, exert authority over the collection and storage of extensive volumes of sensitive personal data. These entities are then able to create models using this data and deploy them at their discretion. However, this paradigm tends to limit accessibility to machine learning and the spectrum of its potential applications, often leading to the marginalization of data subjects. Users who are unwilling to share their data with these entities may encounter challenges when seeking access to machine learning-powered products and services.

Entities that lack access to extensive data-gathering resources, such as researchers, small businesses, and ordinary individuals, face difficulties in accumulating sufficient data for training specific types of models. Furthermore, once a user shares their data, they may lose control over it, potentially leading to privacy concerns.

1.2 Intuitions

Collecting real-world data can sometimes be cost-prohibitive or logistically challenging. In such situations, generating synthetic data offers a more accessible alternative to acquiring original data. It also facilitates model training across a wide range of scenarios that real-world data may not adequately represent.

There are numerous scenarios in which companies employ synthetic data to make information available for processing, especially when regulations or privacy concerns impose restrictions on accessing the original data. For example, in a post-GDPR world, the processing of customer data involves stringent compliance and governance requirements for companies. In these cases, synthetic data serves as an anonymization technique that provides companies with greater flexibility and freedom to process data in a secure manner.

Generative models offer versatile and adaptable means of data sharing. In this scenario, the data curator initially encodes private data into a generative model. Subsequently, this model is shared with an analyst, who can use it to create data that resembles the training dataset. Importantly, it allows for the accommodation of unforeseen and novel tasks without the need for repeated interactions with the curator, as the analyst can readily generate additional synthetic data as required.

The underlying concept here is that generative models possess the ability to autonomously capture the fundamental characteristics of a dataset, including intricate patterns and valuable correlations among different attributes.

One promising approach in this domain involves modeling the data-generating distribution by training a generative model on the original data using technologies to safeguard privacy, commonly referred to as **Privacy Enhancing Technologies (PETs)**. This privacy-preserving model is then shared along with its private parameters, allowing anyone to generate a synthetic dataset that closely mirrors the original training data without compromising the robust protection of privacy.

1.3 Objectives

This thesis is dedicated to explore the latest techniques in the field of generating synthetic data, adopting one powerful PET: **differential privacy**. Synthetic data generation and employing generative models, gained significant recognition as a method for striking a balance between data utility and privacy preservation. The primary focus of this research is to investigate and evaluate the utility and similarity of synthetic data generated by state-of-the-art differential privacy-enhanced generative models.

The structure of the paper is as follows:

- **Chapter 2:** This chapter provides the theoretical foundation for the paper. It offers a brief introduction to deep learning, generative models, a variety of Privacy Enhancing Technologies (PETs), and includes examples of privacy attacks.
- **Chapter 3:** In this chapter, the methodology of the thesis is explained in detail. It comprehensively reviews and analyzes the existing literature on generative models with Differential Privacy, including models such as DPGAN, DP-CTGAN and PATEGAN. Additionally, this section describes the materials and metrics used to assess the utility of synthetic data generated by these models, which are crucial for the subsequent analysis.
- **Chapter 4:** This chapter explore the experiments conducted in this work. The aim is to underscore the flexibility and effectiveness of the approach in practical scenarios.
- **Chapter 5:** This chapter concludes the paper and provides a summary of the key findings and insights obtained through the research.

By following this organized structure, the thesis aims to provide a comprehensive exploration of the generation of synthetic data using differential privacy, shedding light on its utility and effectiveness.

Background

2

2.1 Machine Learning and Generative Deep Learning

In the realm of machine learning (ML), most models can be conceptualized as parametric functions, denoted as $h_\theta(x)$, where x represents an input, typically presented as a vector of attributes referred to as "features," and θ signifies a parameter vector. The function space, denoted as $\forall\theta, X = x \rightarrow h_\theta(x)$, comprise a collection of potential hypotheses used to approximate the underlying data distribution from which the dataset was originally drawn.

A learning algorithm examines the training data with the aim of determining the optimal values for the parameter(s) θ. Essentially, during the training phase, an ML algorithm strives to catch the inherent characteristics of a dataset in the context of a specified "task." Following this, the model's performance is assessed using an independent test dataset, which must be separate from the training dataset. This separation ensures an evaluation of the model's ability to generalize beyond the data on which it was trained.

Upon the completion of training, the model is ready for deployment to make predictions on previously unseen inputs. At this stage, the parameter values θ are fixed, and the model computes $h_\theta(x)$ for novel input instances x. In the domain of machine learning, tasks are commonly categorized into two primary types based on the underlying data structure:

- *Supervised learning*: This involves establishing a connection between inputs and outputs by using training examples, where inputs are paired with corresponding labeled outputs.
- *Unsupervised learning*: In cases where inputs lack labels, the method's objective remains unsupervised.

Machine learning models can also be categorized based on the probability distributions they learn. Assuming a set of input data denoted as X, with the goal of assigning labels y to it, two fundamental approaches can be employed:

- **Discriminative models**: These models learn the conditional probability distribution $p(Y|X = x)$ of the target variable Y, given an observation x.
- **Generative models**: These models learn the joint probability distribution $p(X, Y)$ over the observable variable X and the target variable Y.

> It's crucial to note that generative models must adhere to a *probabilistic* nature rather than being *deterministic*.

The advent of deep learning has given rise to a novel class of techniques known as **deep generative models**, which combine generative models with deep neural networks. These techniques have made significant contributions to the advancement of **Generative Artificial Intelligence (GenAI)**.

Generative models are specifically engineered to approximate the probability distribution of real-world data. This typically involves defining a parametric family of probability densities and optimizing the associated parameters. The optimization can be carried out either to maximize the likelihood of real data or to minimize the divergence between the distributions of generated data and real data.

Generative AI can be categorized as either *unimodal* or *multimodal*. Unimodal systems exclusively handle a single type of input, such as text, while multimodal systems are capable of processing multiple types of input, such as both text and images. Prominent frameworks for pursuing generative AI include **Generative Adversarial Networks (GANs)**, which is the primary focus of this study, and **Generative Pre-trained Transformers (GPTs)**.

2.1.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs), introduced by [Goodfellow et al., 2014] at the Neural Information Processing Systems (NIPS), have emerged as the state-of-the-art neural network architecture, revolutionizing the field of generative modeling. GANs offer a robust mechanism for generating data samples that accurately capture the characteristics of a desired target distribution. This groundbreaking framework is built upon the intricate interplay between two distinct neural networks, namely the *generator* and the *discriminator*, operating within a dynamic adversarial process (Figure 2.1).

At the core of the GAN framework lies the concept of *adversarial* training, wherein the generator and discriminator engage in a competitive process aimed at improving the quality of generated samples; the former aims to produce samples that are indistinguishable from real data and the latter strives to accurately differentiate between genuine and generated samples.

Mathematically, we denote the generator as (G) and the discriminator as (D). The generator (G) takes an input vector (z) drawn from a latent space, often governed by a random noise distribution ($P_z(z)$), and generates a synthetic sample (x') intended to closely resemble authentic data samples: $x' = G(z)$. Conversely, the discriminator (D) assesses either a real sample X originating from the true data distribution $P\{X\}$, or a generated sample X' produced by G, and estimates the probability that the input is real, which means: $D(X) \rightarrow [0, 1], \quad D(X') \rightarrow [0, 1]$.

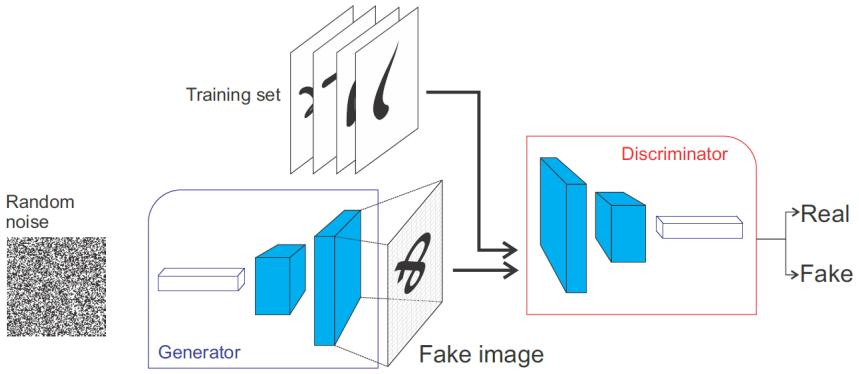


FIGURE 2.1 | Architecture of a GAN. The generator only sees noisy latent representations and outputs a reconstruction. The discriminator gets alternatively real or generated inputs and predicts whether it is real or fake [Md. Rezaul Karim, Java Deep Learning Projects]]

The training regimen alternates between two distinct phases:

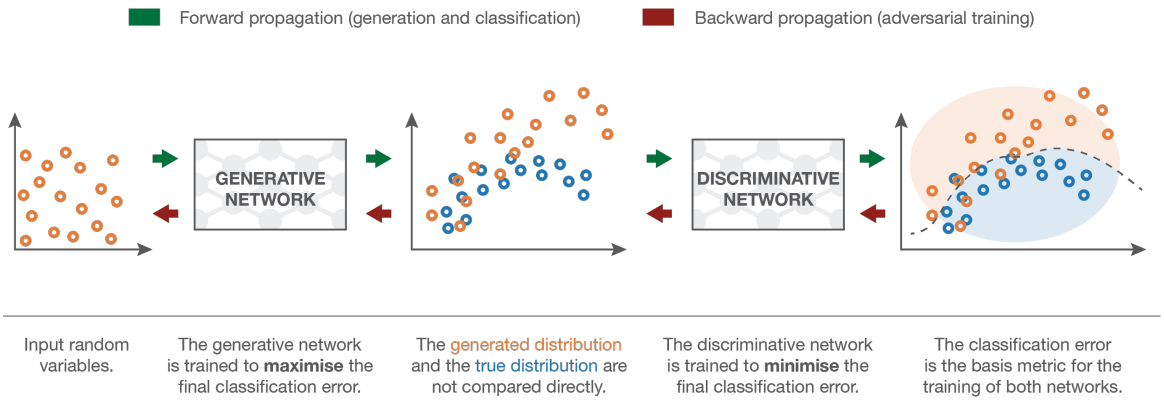
1. **Generator Training**: During this phase, the generator seeks to improve its ability to generate samples that can deceive the discriminator. The generator's objective is to minimize the following expression, compelling the generated samples to be classified as authentic:

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

2. **Discriminator Training**: In this phase, the discriminator is trained to effectively distinguish between real and generated samples. The discriminator aims to maximize its classification accuracy, leading to the maximization of $V(D, G)$.

The iterative optimization process inherent in GANs ultimately reaches an equilibrium state where the generator network produces samples that are hardly distinguishable from real data. In contrast, the discriminator utilizes pure supervised learning to assess whether the samples are real or fake, essentially performing binary classification (see Figure 2.2).

FIGURE 2.2 | Back-propagation of the distribution matching error [Joseph Rocca, TDS]



The growing interest in generative models is well-demonstrated by their versatile applications across various domains, including image synthesis, style transfer, and data augmentation, among many others. However, it's crucial to acknowledge that, while GANs exhibit versatility, achieving effective training of these models requires meticulous hyperparameter tuning and may be susceptible to issues such as mode collapse. Mode collapse occurs when the generator produces a limited range of diverse samples, limiting the variety of generated data.

FIGURE 2.3 | Improvement of GAN models across the years Salehi et al. [2020]

Challenges

[Webster et al., 2021a] found that modern GANs trained on facial images can produce examples that closely resemble their training data, potentially revealing private information. To protect individuals' privacy, it's important to train generative models with privacy constraints. However,

this can be challenging due to training instabilities and the need for careful hyperparameter tuning.

The loss functions of the discriminator and generator can exhibit erratic oscillations instead of showing long-term stability. Mode collapse is another challenge where the generator focuses on a limited set of samples that consistently fool the discriminator. This leads to a near-zero gradient in the loss function.

The generator's loss function may increase over time, even though the quality of the generated images improves. This lack of correlation between the loss and image quality makes monitoring GAN training difficult.

Even with basic GANs, there are many hyperparameters to fine-tune, including the architecture of the discriminator and generator, batch normalization, dropout, learning rate, activation layers, convolutional filters, kernel size, striding, batch size, and latent space size. Finding the right set of parameters often requires an iterative and experimental approach, which in most of the cases turns as an expensive process [Papernot et al., 2019].

2.1.2 Conditional Tabular GAN (CTGAN)

Modeling the probability distribution of rows in tabular data and generating realistic synthetic data can be a complex task for Generative Adversarial Networks. These challenges also include correlated features, a combination of discrete and continuous columns. Continuous columns may exhibit multiple modes, while discrete columns may suffer from imbalances that make modeling challenging. Existing statistical and deep neural network models struggle to properly learn from these highly sparse vectors. In response, **Conditional Tabular GAN** [Xu et al., 2019] introduces several new techniques:

1. **Mode-specific normalization**: To address the non-Gaussian and multimodal distribution of the data, each column is processed independently. A "variational Gaussian mixture model (VGM)" estimates the number of modes, and for each value, it calculates the probability from each mode ($\rho_k = \mu_k * N$), where N represents a Gaussian distribution (see Figure 2.4). The mode with the highest probability is used to normalize the values. Each value is represented as a one-hot vector indicating the mode and a scalar indicating the value within that mode. This approach helps manage the complex distribution of values in tabular data.

2. **Conditional generator and training-by-sampling**: To address the challenges posed by imbalanced discrete columns in tabular data, CTGAN incorporates a conditional generator and a training-by-sampling approach. The key objective here is to efficiently resample data in a manner that ensures all categories within discrete attributes are sampled, aiming

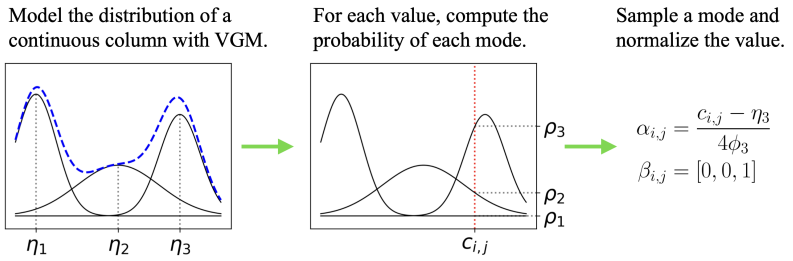


FIGURE 2.4 | An example of mode-specific normalization [Xu et al., 2019]

for balanced representation (though not necessarily uniform) during the training process. Additionally, the model needs to recover the distribution of the original, non-resampled data during testing. To achieve this, the generator penalizes its loss by introducing the cross-entropy between conditional vectors, which is averaged over all instances within a batch. This technique helps address the imbalanced nature of discrete data and ensures that the generator produces diverse and realistic samples.

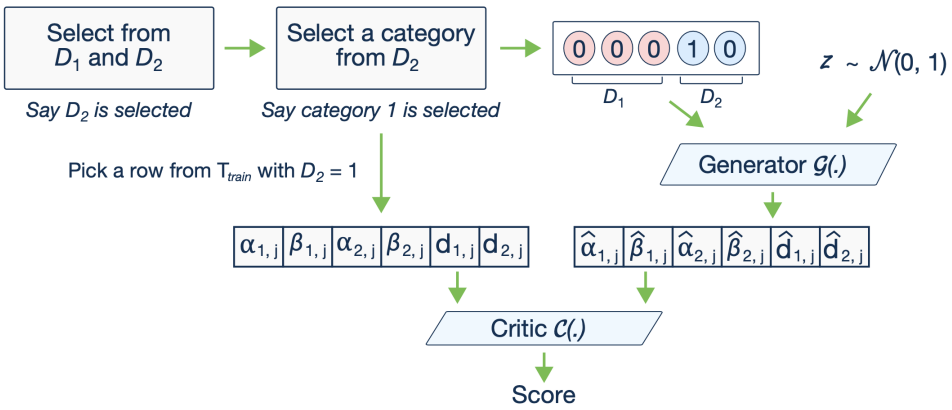


FIGURE 2.5 | CTGAN model. The conditional generator can generate synthetic rows conditioned on one of the discrete columns. With training-by-sampling, the data is sampled according to the frequency of each category, thus CTGAN can evenly explore all possible discrete values [Xu et al., 2019].

The output generated by the conditional generator undergoes evaluation by the discriminator. The discriminator's role is to estimate the distance between the learned conditional distribution $P_G(row|cond)$ and the conditional distribution on real data $P(row|cond)$. Properly sampling the conditional vector and training data is crucial in ensuring that the model adequately explores all possible values within the discrete columns. This balanced exploration of discrete attributes is essential for the generator to produce synthetic data that closely matches the distribution of real data under the given conditions.

3. **Network structure**: two fully-connected hidden layers in both generator and discriminator. In generator, *batch-normalization* and *Relu* activation function are used. After the two hidden

layers, the synthetic row representation is generated using a combination of activation functions. The scalar values α_i are generated using the *Tanh* function, while the mode indicator β_i and the one-hot vector values d_i are generated using the *Gumbel softmax*. In the discriminator, the model employs a *leaky Relu* activation function and incorporates *dropout* on each hidden layer.

2.1.3 Private Aggregation of Teacher Ensembles (PATE)

The PATE framework is designed to protect the privacy of training data during the learning process by transferring knowledge from an ensemble of teacher models to a student model. Privacy guarantees in this framework are intuitively understood and rigorously expressed in terms of differential privacy. The PATE framework comprises three key components:

1. **Ensemble of Teacher Models**:
    - Each teacher is a model trained independently on a distinct subset of the data, with the goal of protecting the privacy of the data.
    - Data is partitioned in a way that ensures no two teachers are trained on overlapping data.
    - Various learning techniques can be applied to train each teacher, resulting in multiple models solving the same task.
    - During inference, teachers make independent predictions.
2. **Aggregation Mechanism**:
    - The aggregation mechanism plays a crucial role in ensuring privacy.
    - When there is a strong consensus among teachers regarding a prediction, the label they mostly agree on does not reveal any specific information about a given training point.
    - To provide rigorous guarantees of differential privacy, the aggregation mechanism counts the votes assigned to each class.
    - It then adds carefully calibrated Laplacian noise to the resulting vote histogram and outputs the class with the most noisy votes as the ensemble's prediction.
3. **Student Model**:
    - The final step in the PATE framework involves training a student model.
    - This student model learns from the ensemble of teacher models by transferring knowledge.
    - Access to public data, which is unlabeled, is used for this training.

- To limit the privacy cost associated with labeling public data, queries are made to the aggregation mechanism for a subset of the public data.
- The student model is trained in a semi-supervised manner using a fixed number of queries.
- It's important to note that each additional ensemble prediction increases the privacy cost, so the number of queries is bounded.

The PATE framework provides a privacy-preserving way to transfer knowledge from a set of teacher models to a student model while maintaining strong privacy guarantees. This approach is particularly valuable when dealing with sensitive data that requires protection during the training process.

The training set is partitioned into k disjoint subsets $D_1, ..., D_k$ and the k teachers classifiers $T_1, ..., T_k$ are trained separately on these k partitions. Let $PATE = \{f_k\},\ k \in T$ be the ensemble of T teacher models Let's denote with x an input, m is number of possible class labels, $j \in [m]$ a label of a given class, and $n_j(x) = |k : k \in T, f_k(x) = j|$ the number of teachers that output class j for x.

When classifying a new instance x, the $PATE$ framework introduces noise during the aggregation process to create ambiguity and achieve a differentially private output. This is necessary because a simple majority-based aggregation could result in a situation where the top choice depends on the voting input of a single teacher.

$$PATE_\lambda(x) = \arg\max_{j \in [m]} \left(n_j(x) + Y_j\right)$$

where $Y_1, ..., Y_m$ are $i.i.d.$ $Lap(0, \lambda)$ random variables following the Laplace distribution on location 0 with scale λ.

The parameter λ is controlling how much noise is added, in turn guaranteeing privacy. A single query to the $PATE_\lambda$ mechanism is $\left(\frac{1}{\lambda}, 0\right)$-differentially private.

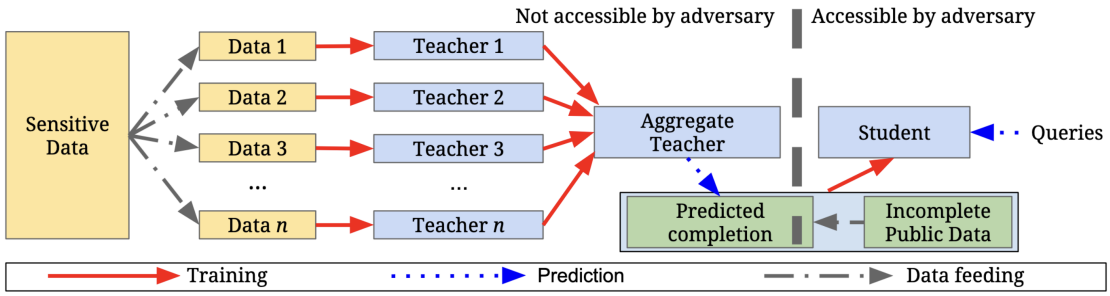


FIGURE 2.6 | Training of ensemble of teachers is trained on disjoint subsets and a student model trained on public data labeled by the ensemble [Papernot et al., 2018].

In the work by [Papernot et al., 2018], a modification to the PATE aggregation mechanism is introduced, where Gaussian noise is sampled instead of Laplacian noise. This change is made because the tails of the Gaussian distribution diminish more rapidly compared to those of the Laplacian distribution, resulting in less noisy aggregation. The reduced noise levels increase the likelihood that the aggregated votes from the teachers lead to the correct consensus answer. This modification is particularly valuable when PATE is applied to learning tasks with a large number of output classes, as it helps improve the quality of the consensus answer.

2.2 Privacy Enhancing Techniques (PETs)

Privacy-Enhancing Technologies (PETs) encompass two primary categories, one focusing on input privacy and the other on output privacy.

1. **Input Privacy**: In the context of input privacy, the Computing Party is unable to access or derive any input values provided by Input Parties. Additionally, the Computing Party cannot access intermediate values or statistical results that are available at the Computing Parties during the data processing phase. This ensures that sensitive input data remains confidential and protected from unauthorized access or exposure.

2. **Output Privacy**: Output privacy, often referred to as "statistical disclosure control," is concerned with altering the results of a computation in such a way that the output data cannot be used to reverse engineer or deduce the original inputs. This safeguards the privacy of individuals or entities who provided the input data, preventing any potential compromise of their sensitive information.

These two categories of PETs play a crucial role in safeguarding the privacy of data throughout its processing and dissemination, ensuring that sensitive information remains secure and confidential.

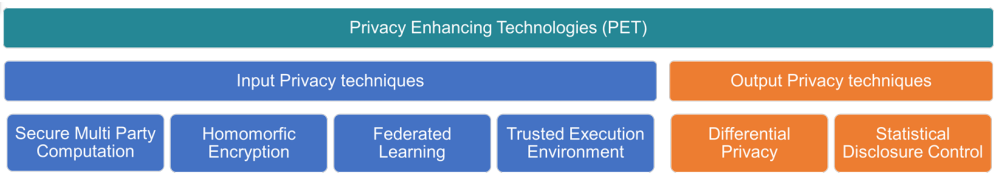


FIGURE 2.7 | Different types of PETs [United Nations, 2023]

Protecting sensitive information and ensuring privacy in various IT scenarios is a complex task that requires a combination of techniques and technologies. Here are some of the key methods and approaches used to safeguard data privacy:

1. **Data Anonymization**: Data anonymization involves modifying or masking identifiers in a dataset to make it difficult to identify individuals. Techniques include removing, substituting, distorting, generalizing, or aggregating data, which aims to protect both direct identifiers (explicit personal information) and indirect identifiers (attributes that, when combined with other data, could identify a user).

2. **Pseudonymization**: Pseudonymization involves replacing private identifiers with fake identifiers or pseudonyms to hide key identifiable information. This technique preserves statistical accuracy and data integrity.

3. **Perturbation**: Perturbation techniques involve adding crafted random noise to sensitive data to hide patterns and prevent privacy data mining attacks.

4. **Synthetic Data**: Synthetic data generation involves using algorithms to create artificial datasets with specific statistical patterns or models, rather than altering the original dataset. This approach can provide a balance between data utility and privacy protection.

5. **k-Anonymity, l-Diversity, t-Closeness**: These are methods used in cloud-based applications to ensure data privacy. They focus on ensuring that each record in a dataset is not distinguishable from at least k-1 other records, adding diversity to sensitive attributes, and ensuring that the distribution of sensitive attributes is similar to a trusted distribution, respectively.

It's important to note that while anonymizing personal data is a valuable privacy protection measure, there can still be risks of re-identification, especially when machine learning applications are involved. As machine learning techniques advance, the potential for re-identification or privacy breaches also increases. Therefore, it's essential to consider a combination of privacy-preserving techniques and stay updated on best practices to protect sensitive information in various IT scenarios.

2.2.1 Differential Privacy

Differential privacy (DP) is a robust and widely accepted framework for ensuring privacy in data analysis. It was first introduced in 2006 by [Dwork et al., 2006b] in their seminal papers "Calibrating Noise to Sensitivity in Private Data Analysis" and "Differential Privacy". DP provides a mathematical foundation for quantifying and achieving privacy in data analysis.

At its core, DP aims to quantify the maximum amount of information about individual records in a database that could potentially be revealed by releasing the results of any computation on that

database. It was designed to address shortcomings in earlier privacy definitions, particularly when dealing with multiple releases of data or when adversaries have access to side knowledge.

One key feature of differential privacy is its reliance on *randomness* in the design of privacy-preserving algorithms. This randomness is essential to ensure that no individual's data can be accurately inferred from the algorithm's outputs. It ensures that attackers cannot retrieve sensitive information about input datasets merely based on the algorithm outputs (indistinguishability). DP is versatile and can be applied to a wide range of data processing scenarios.

> *Differential privacy* constitutes a strong standard guarantees for privacy for algorithms on aggregate databases [Dwork and Roth, 2013].

In the context of differential privacy, a randomized algorithm $A : D \to R$ with domain D and range R, is considered to be (ϵ, δ)-differentially private if it satisfies the following condition:

For every pair of adjacent training datasets, $d, d' \subseteq D$, which differ by at most one training point, and for any subset of outputs $S \subseteq R$, the probability that the algorithm produces an output in S for dataset d is bounded as follows:

$$\mathrm{P}[A(d) \in S] \leq \mathrm{e}^{\epsilon}\, \mathrm{P}[A(d') \in S] + \delta$$

Here's a breakdown of the parameters and their meanings:

1. The parameter $\epsilon : \epsilon > 0$ is often referred to as the **privacy budget**. It quantifies the level of privacy protection provided by the algorithm. A lower ϵ corresponds to stronger privacy guarantees. It controls the trade-off between privacy and utility. Smaller values of ϵ provide stronger privacy but may limit the utility of the algorithm.

2. The parameter δ is the **failure rate**, and it quantifies the tolerance for cases where the privacy bound defined by ϵ does not hold. It allows for a small probability of deviation from the privacy guarantee. In practice, δ is required to be very small ($\delta \in [0, 1]$), and its value should be chosen based on the dataset size and the desired privacy level. [That which we call private, Erlingsson]

The term $\ln\left(\frac{P(A(d) \in S)}{P(A(d') \in S)}\right)$ is known as the **privacy loss**. It quantifies the extent to which the privacy guarantee may be violated when comparing the output distributions of the algorithm on datasets d and d'.

The original definition of ϵ-*differential privacy*, does not include the additive term δ. The variant introduced by [Dwork et al., 2006a] allows for the possibility that pure ϵ-differential privacy is

broken with probability δ, which is preferably a small value (e.g., smaller than $1/|d|$, where $|d|$ is the size of the dataset).

Intuitively, this guarantees that an adversary, provided with the output of A, can draw almost the same conclusions about any individual no matter if this individual is included in the input of A or not, so it is more a privacy standard rather than a single algorithm. In other words, it ensure that even if an attacker has knowledge of the entire dataset, they cannot determine specific information about any individual in the dataset.

Differential privacy is achieved by introducing random noise into the results of data analysis. Various mechanisms can be used to add this noise, such as the *Laplace mechanism*, the *exponential mechanism*, and the *random response mechanism*. The choice of mechanism depends on the specific analysis being conducted and the desired privacy level.

The Laplace mechanism is commonly used for numerical queries in differential privacy. It adds Laplace-distributed noise to the query results, with the magnitude of the noise determined by the privacy budget (ϵ) and the sensitivity of the query. The random response mechanism, on the other hand, is used for scenarios like sensitive surveys, where respondents can plausibly deny their responses. It is widely applied in statistical analysis to obtain population-level information without revealing details about individuals.

Differential privacy differs from other privacy protection methods, like k-anonymity or l-diversity, as it applies to all types of information derived from a database and addresses the challenges of multiple releases and secondary knowledge accessible to attackers.

2.2.2 Synthetic Data

Synthetic data is a privacy protection method that aims to provide a balance between the necessity to share information, in order to perform statistical analysis, and the requirement to preserve the confidentiality of sensitive data. It achieves this by transforming sensitive data into a new dataset that shares similar statistical characteristics with the original data, while avoiding the disclosure of specific details about the original dataset. This approach is particularly valuable for organizations that want to collaborate with external partners while safeguarding the privacy of their sensitive data.

The primary goal of synthetic data generation is to combine two aspects: usefulness for the statistical analysis and the preservation of confidentiality. Synthetic data include features like *data augmentation*, where new data is created or existing data is expanded for validation and verification purposes, although these aspects may fall outside the scope of privacy-enhancing technologies (PETs).

In the context of privacy protection, synthetic data refers to algorithmically generated data that closely resembles the originating source of data. Generative models are used to learn the statistical distribution in the real data and generate artificial samples that mimic the original ones. This synthetic data generation process completely breaks the one-to-one relationship between the original and synthetic records, ensuring that there is no direct way to reverse the synthetic records to their original counterparts. This process is irreversible, providing strong privacy protection.

Deep learning models used for synthetic data generation have the computational capacity to handle complex tasks. However, they may memorize patterns from the training data, which can lead to privacy leaks in the synthetic data. To enhance privacy in the synthetic data generation process, additional layers of privacy, such as **differential privacy**, can be added.

There are different types of synthetic data, each with its benefits and drawbacks:

1. **Fully Synthetic Data:** This type of synthetic data does not contain any original data. It makes re-identification of any individual unit nearly impossible, and all variables are still fully available.

2. **Partially Synthetic Data:** In partially synthetic data, only sensitive data is replaced with synthetic data. This approach relies on imputation models to ensure that the overall structure of the data remains intact.

3. **Hybrid Synthetic Data:** Hybrid synthetic data is derived from a combination of real and synthetic data. It maintains the relationship and integrity between variables while investigating the underlying distribution of the original data. Each data point from the real data is paired with its nearest neighbor in the synthetic data to create a hybrid dataset.

The choice of the type of synthetic data depends on the specific use case, the level of privacy required, and the need to maintain data utility for analysis while protecting sensitive information.

2.2.3 Data Anonymization

Data anonymization (or masking) techniques are a crucial aspect of privacy protection, focusing on the removal or concealment of **Personally Identifying Information (PII)** while leaving other less sensitive attributes in the dataset untouched. PII includes information such as names, phone numbers, or other details that can directly identify individuals. In contrast, the remaining attributes, often referred to as *quasi-identifiers*, are typically less sensitive but can still pose privacy risks when combined with external data sources.

phone	birth year	sex	zip code	medical condition	headache
015940192	1964	f	1203002	chest_pain	10110010110100010
010405919	1964	f	1203505	obesity	100000100000111010
011500159	1964	f	1203106	short_breath	10110010110100010
010192042	1965	m	5403221	heart_disease	1010010110100010
015909191	1965	m	5403221	heart_disease	010010110100010

phone	birth year	sex	zip code	medical condition	headache
[redacted]	1964	f	1203002	chest_pain	10110010110100010
[redacted]	1964	f	1203505	obesity	100000100000111010
[redacted]	1964	f	1203106	short_breath	10110010110100010
[redacted]	1965	m	5403221	heart_disease	1010010110100010
[redacted]	1965	m	5403221	heart_disease	010010110100010

FIGURE 2.8 | Data masking techniques removes PII [O. Fdal]

The vulnerability of masked data to linkage attacks is a key consideration. Even though PII is removed or obfuscated, the presence of quasi-identifiers and the potential for re-identification through external data sources can still pose a significant privacy risk. This is why regulatory frameworks often do not consider masked data to be truly "anonymous" for legal and privacy compliance purposes. Instead, such data is typically still classified as personal data and subject to privacy regulations.

> *"Generally speaking removing directly identifying elements in itself is not enough to ensure that identification of the data subject is no longer possible [Parliament, 2014]."* [1]

2.2.4 k-anonymity, l-diversity and t-closeness

In summary, traditional methods almost systematically present re-identification related risks. To counter these risks and protect privacy, various methods have been developed, including k-anonymity, l-diversity and t-closeness:

1. **k-Anonymity:** k-anonymity is an anonymization technique that ensures that the information about an individual in a published dataset cannot be distinguished from at least k-1 other individuals in the same dataset. In other words, it groups individuals with similar attributes into equivalence classes, and within each class, there are at least k-1 individuals who share the same attributes.

2. **l-Diversity:** l-diversity is another privacy protection method that focuses on the diversity of sensitive attributes within equivalence classes. A dataset is considered l-diverse if each equivalence class, formed based on attributes shared by individuals, has at least l (where l is typically greater than or equal to 2) different values for sensitive attributes. This ensures that each group is diverse in terms of sensitive information, making it harder for an attacker to infer sensitive details about a specific individual.

3. **t-closeness:** t-closeness is a further refinement of the concepts of k-anonymity and l-diversity. In a dataset, t-closeness ensures that the distribution of sensitive attribute values

[1] *European Parliament, Article 29 Data Protection Working Party*

within a group (equivalence class) of records is not significantly different from the overall distribution of sensitive values in the entire dataset.

k-anonymity maintains privacy by editing, via suppression and generalization, quasi-identifiers so that each combination of them is present at k times. Since the same quasi identifiers are shared between different rows, k-anonymity prevents unique joints that expose sensitive attributes. However, research showed [Bellovin et al., 2018] that k-anonymity is subject to attribute inference attacks.

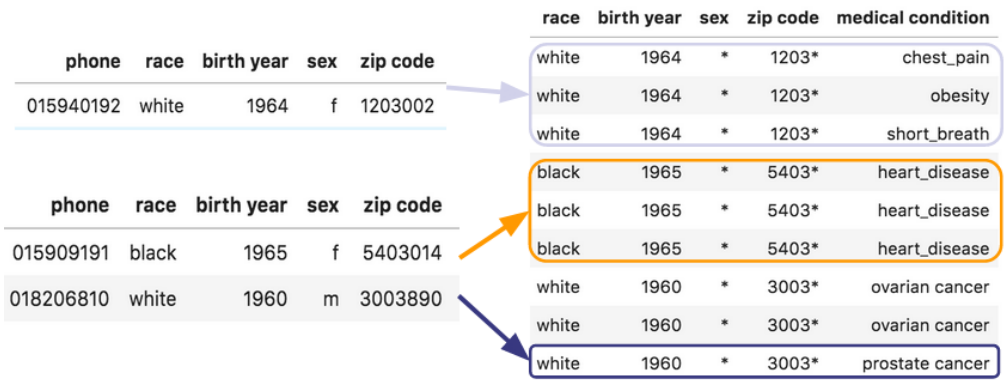


Figure 2.9 | The k-anonymity hides individual records within a group of similar records [O. Fdal]

The other techniques, such as l-diversity or t-closeness, increase the complexity and reduce the utility of the data, assuming that some attributes are more special than others. To completely remove privacy risks, one would need to remove most, if not all data, reducing the data utility to zero.

2.2.5 Secure Multi-Party Computation

Secure multi-party computation (also known as **sMPC**) is a significant cryptographic technological advancement that enables multiple independent entities to perform computations on their private data without the need for data disclosure. The core idea behind sMPC is to facilitate calculations on sensitive data while maintaining its confidentiality.

The fundamental principle of MPC is to limit the knowledge of the participants, ensuring that they only have access to the output of the computation and their individual inputs. This approach helps address the issue of code assurance, where parties involved in the computation need to be confident that the function being computed on their shared data remains the same as agreed upon.

However, it's important to note that both sMPC and homomorphic encryption, which is another technique for secure computation, often come with high communication and computation costs.

2.2.6 Homomorphic Encryption

Homomorphic encryption (HE) is a powerful cryptographic technique that allows for computations to be performed directly on previously encrypted data. This method enables an entity that supplies data to outsource a computation to a third party while keeping the data confidential. The process of homomorphic encryption generally works as follows:

1. The entity that owns the data encrypts it using a homomorphic function, resulting in encrypted data.
2. This encrypted data is then shared with a third party that is responsible for performing calculations or operations on it.
3. The third party performs the desired computation on the encrypted data and returns the result, which is also encrypted.
4. Finally, the original data owner decrypts the result, allowing them to obtain the outcome of the calculation on the original plain-text data.

Homomorphic encryption ensures that no party, other than the data provider who holds the necessary decryption key, gains access to any information about the data during the computation.

While homomorphic encryption offers strong confidentiality guarantees to the data provider, there are practical limitations in terms of code assurance and confidentiality, particularly for the entity responsible for the computation as the algorithm provider may have less assurance about the confidentiality of their algorithm.

Homomorphic encryption has diverse applications in various fields, specially in healthcare, where strict regulations on patient data confidentiality are in place.

2.2.7 Distributed Learning

In distributed learning, the common configuration involves multiple entities owning sensitive data, and a central server that assists them in the training process, preserving the confidentiality of their data. Two of the prominent protocols in this area are **Federated Learning** and **Split Learning**, each with its own variations, advantages, and limitations.

These are not standalone solutions for ensuring data privacy but rather some additional features to consider when designing the confidentiality of output data. It primarily focuses on improving the security of input data exchanged among the participating entities and those responsible for the computations. The level of privacy achieved depends on the specific protocols, techniques, and parameter settings used.

2.2.8 Trusted Execution Environments and Secure Enclaves

A **Trusted Execution Environment (TEE)** is a secure hardware and an isolated software environment built into modern CPUs for executing code and processing data with a high level of confidentiality and security. TEEs offer solutions to the following challenges:

1. **Input Confidentiality:** TEEs ensure the confidentiality of input data, preventing unauthorized access to sensitive information provided as input to a computing process.
2. **Code Confidentiality:** TEEs protect the confidentiality of the code used to perform operations on data. This includes ensuring that the code remains secure and hidden from potential adversaries.
3. **Code Security Assurance:** TEEs provide security assurances for the code executed within the trusted environment, making it resistant to tampering, reverse engineering, and other attacks.

TEEs are designed to be highly resistant to attacks, even by privileged users or attackers with physical access to the hardware. Examples of popular TEE technologies include *Intel Software Guard Extensions (SGX)* and *AMD TrustZone*, which are widely used in cloud and edge computing environments.

2.3 Privacy Attacks

In the following paragraphs, we will provide a brief introduction on the different privacy attack models and the various types of attacks that are directed towards the assets of machine learning (ML) processes. We initially categorize these adversary models based on the level of access that the attacker possesses:

- **White-box adversaries**: These adversaries possess comprehensive knowledge about the model, denoted as $M_\theta(x)$, including its parameters represented as θ. Additionally, they may also have partial access to the raw data, denoted as X.
- **Black-box adversaries**: In contrast, black-box adversaries lack any information concerning the model and its parameters. However, they have the capability to interact with the model by making queries and observing its responses.

Another important consideration is the phase in which the attack is executed, whether it occurs during the **training** or **inference** stage [Oliynyk et al., 2023]:

- **Training Phase**: In this scenario, the adversary's objective is to gain insights into the model, which might involve accessing a summary, partial, or even the entirety of the training data.
- **Inference Phase**: Here, the adversary collects information about the model's characteristics by observing the inferences made by the model during its operational phase.

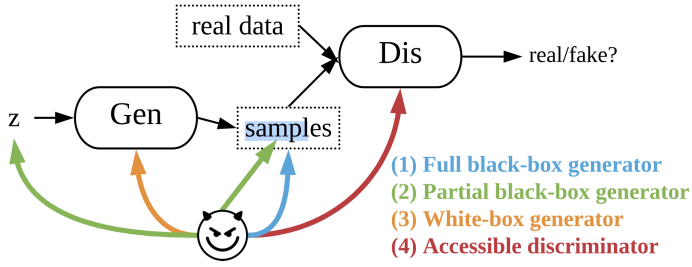


FIGURE 2.10 | A taxonomy of attack models against GANs [Chen et al., 2020]

2.3.1 Membership Inference Attacks

In this particular type of attack, the adversary's objective is to ascertain whether a given data point, denoted as x_i, was originally part of the raw dataset or if it was included in the training set, denoted as X.

The attack strategy involves several key steps:

1. **Data Generation**: The attacker begins by generating data that closely resembles the distribution of the original dataset. This is accomplished by making queries to machine learning models and capturing their responses.
2. **Local Model Training**: Subsequently, the attacker employs the generated data to train local models. These local models are designed to mimic the behavior of the original machine learning models.
3. **Classifier Development**: The attacker then utilizes the data generated by these local models to train a classifier. The purpose of this classifier is to determine whether a given data record belongs to the original training dataset.

Membership inference attacks can have implications for both the privacy of the raw dataset and the privacy of the feature datasets. For instance, an adversary could employ this attack to infer whether a specific individual's record was used in training a machine learning model designed to detect the presence of a particular medical condition [Hayes et al., 2018, Shokri et al., Carlini et al.].

It is important to note that this attack can be executed by a *black-box adversary*, meaning that the attacker does not possess any knowledge about the internal structure or parameters of the machine learning models but can still perform these inference attacks by querying the models and observing their responses.

2.3.2 Model Inversion Attacks

Model inversion attacks and attribute inference attacks infer class features and/or construct class representatives about the training dataset X, given that the adversary has some access (either *black-box* or *white-box*) to a model $M_\theta(x)$. [Fredrikson et al., 2015, Zhang et al., 2020].

Model inversion attacks enable the adversary to leverage the model's output to infer the values of sensitive attributes that were utilized as input to the model. This type of attack is not limited to the level of access an attacker has, as even a limited black-box attacker, who can interact with the model by making queries and collecting its responses, can perform it.

Furthermore, it is worth noting that it is not uncommon for adversaries to employ a two-step approach. They may initially execute a **model extraction** attack to obtain a copy of the model $M_\theta(x)$. Subsequently, they utilize this extracted model to carry out a model inversion attack, thereby inferring information about the training dataset.

Figure 2.11 | An image recovered using a model inversion attack (left) and a training set image of the victim (right) [F. Mireshghallah, 2020].

2.3.3 Model Extraction Attacks

The primary objective of a model extraction attack is to create an approximate model, denoted as $M'\theta(x)$, that closely mimics the behavior of the original model, $M\theta(x)$. This is typically carried out by an adversary with black-box access, who lacks any prior knowledge about the machine learning model's parameters or training data. The ultimate goal of this attack is to effectively steal the real model's parameters.

The attack relies on a fundamental intuition, namely, the utilization of information-rich outputs provided by ML prediction APIs. These outputs often include high-precision confidence values in addition to class labels. By analyzing and leveraging these rich output details, the adversary endeavors to construct an approximation that accurately reflects the behavior of the original model [Takemura et al., 2020, Reith et al., 2019]. The success of the attack is directly related to the accuracy of the constructed approximation.

2.3.4 De-anonymisation Attacks

The objective of this attack is to uncover the identity of an individual who has provided their data to a dataset. Even in cases where model owners have initially anonymized the dataset, either by publishing the raw dataset Z or feature datasets $X1$ and $X2$, this attack has the potential to compromise the privacy of data contributors.

To execute this attack successfully, the adversary typically requires white-box access to the dataset. However, it is important to note that it can also be performed with black-box access, through a series of chained attacks [Qian et al., 2016, Gambs et al., 2014].

2.3.5 Reconstruction Attacks

The objective of this attack is to reconstruct a raw dataset, denoted as Z, by employing a process of reverse engineering on the feature training dataset, $X1$, or the validation dataset, $X2$. In most cases, this type of attack necessitates *white-box* access to a model that explicitly embeds the feature datasets within its structure. The ability to access the internal workings and parameters of the model is crucial for successfully executing this attack [Al-Rubaie and Chang, 2016].

Methodology

3

In recent years, significant advancements in *neural networks* have yielded remarkable achievements across a diverse range of applications, such as image classification and language representation, among many others. These breakthroughs owe a substantial part of their success to the existence of extensive and representative datasets used for training neural networks.

These datasets are frequently collected from crowdsourced contributions and may encompass sensitive information. The utilization of such data necessitates the development of techniques that can satisfy the performance requirements of these applications while simultaneously providing sound and principled privacy safeguards.

3.1 Deep Learning with Differential Privacy

In the context of privacy considerations, generative models offer a distinct advantage by introducing noise within the latent space, rather than directly altering the data. This approach allows us to ensure privacy while minimizing the overall loss of information.

Various approaches have been proposed for integrating differential privacy into the generation of data using Generative Adversarial Networks (GANs). One of the more intuitive methods is to introduce sampled noise at the end of the generation process, thereby applying obfuscation directly to the output data to enhance privacy. However, this approach frequently results in a trade-off with utility, where the data's usefulness may be compromised.

In this work, the two primary approaches examined focus on incorporating differential privacy directly into the training process. These methods aim to strike a balance between privacy protection and preserving the utility of the generated data.

The first approach involves integrating differential privacy (DP) into the training process itself. This is achieved by adding a small amount of noise, typically sampled from a Gaussian or Laplacian distribution, to the gradients that are already clipped. Notable implementations of this approach include **DPGAN**, proposed by [Xie et al., 2018], and **DP-CTGAN** by [Ling et al., 2022].

The second approach explores the use of the **Private Aggregation of Teacher Ensembles (PATE)** mechanism, initially introduced in the work by [Papernot et al., 2018, 2017]. This mechanism

provides guarantees of (ϵ, δ)-privacy and is being considered within GAN training, with the aim to enhance the privacy-preserving aspects of the process.

Here are some notable approaches for achieving differential privacy in data generation and machine learning tasks:

- **Multiplicative Weights Exponential Mechanism (MWEM):** This approach combines Multiplicative Weights and Exponential Mechanism techniques to achieve differential privacy. It is relatively straightforward yet effective and requires fewer computational resources, resulting in shorter runtime.
- **Differentially Private Generative Adversarial Network (DPGAN):** DPGAN introduces noise to the discriminator of a GAN to enforce differential privacy, through a Differentially Private Stochastic Gradient Descent (DPSGD). It has been used with various data types, including images and electronic health records (EHRs).
- **Differentially Private Conditional Tabular GAN (DP-CTGAN):** DP-CTGAN utilizes the state-of-the-art CTGAN for synthesizing tabular data and applies DP-SGD, as in DPGANs, to ensure differential privacy. It is well-suited for tabular data, helps address issues like mode collapse, but may require extensive training time.
- **Private Aggregation of Teacher Ensembles Generative Adversarial Network (PATEGAN):** This is a modification of the PATE framework applied to GANs to preserve differential privacy in synthetic data generation. It is an improvement over DPGAN, particularly for classification tasks. PATEGAN ensures privacy by transferring knowledge from an ensemble of "teacher" models trained on disjoint data partitions to a "student" model.
- **PATE-CTGAN:** Similar to DP-CTGAN, PATE-CTGAN utilizes CTGAN for tabular data synthesis but applies the Private Aggregation of Teacher Ensembles (PATE) mechanism for ensuring differential privacy. It is suitable for tabular data and mitigates problems associated with mode collapse.
- **Qualified Architecture to Improve Learning (QUAIL):** QUAIL is an ensemble method designed to enhance the utility of synthetically generated differentially private datasets for machine learning tasks. It combines a differentially private synthesizer with an embedded differentially private supervised learning model to produce flexible synthetic datasets with high machine learning utility.

To create differentially-private synthetic records, models are trained using a DP algorithm to learn the original data distribution. Consequently, the synthetic data inherits the theoretical privacy guarantees offered by DP.

These multiple layers of privacy protection serve to significantly enhance the privacy of the synthetic data. Nevertheless, it's important to recognize that no technique can ensure perfect privacy while still maintaining utility. Regulations such as the General Data Protection Regulation (GDPR) mandate that companies must evaluate the residual risks of re-identification, acknowledging that complete privacy may necessitate a trade-off with data utility.

3.1.1 DP-GANs

Differential privacy can be seamlessly integrated into the deep learning framework by introducing random noise into the *Stochastic Gradient Descent (SGD) algorithm*. This novel approach, known as **DP-SGD**, was introduced and outlined in [Abadi et al., 2016].

In DP-SGD, the standard mini-batch gradient estimate used in SGD is replaced with a privatized version. In this modified version, the gradient of each training example is clipped to a maximum norm.

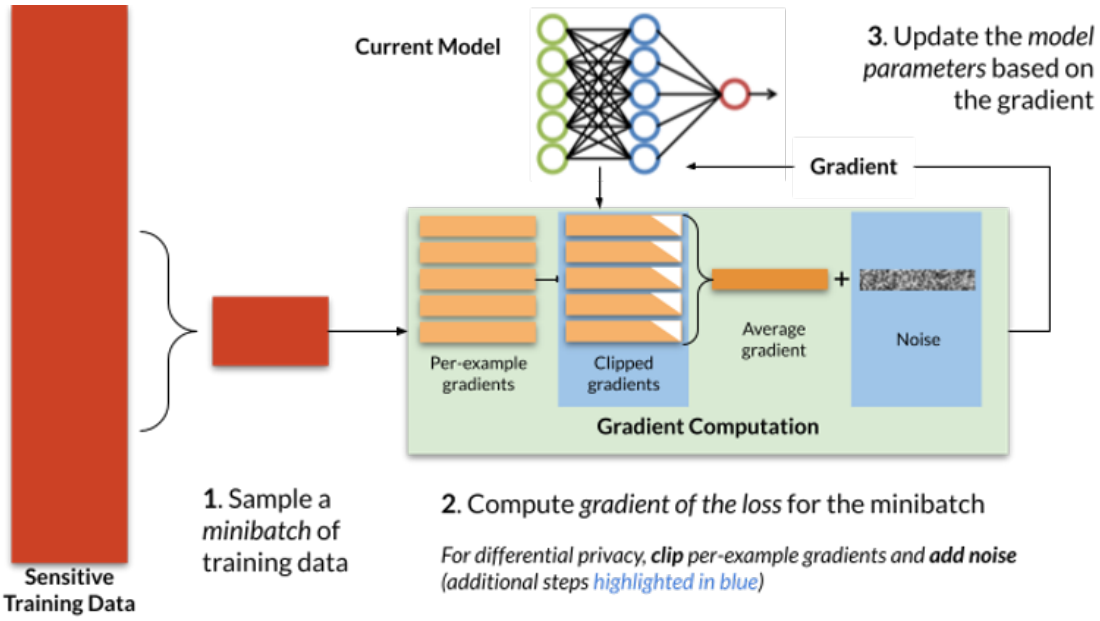


Figure 3.1 | Stochastic Gradient Descent (SGD) and Differentially Private SGD (DP-SGD). To achieve differential privacy, DP-SGD clips and adds noise to the gradients, computed on a per-example basis, before updating the model parameters. Steps required for DP-SGD are highlighted in blue; non-private SGD omits these steps [Papernot and Thakurta]]

This clipping operation constrains the sensitivity of the learning process to each individual training example. Additionally, Gaussian noise, with a standard deviation proportional to this sensitivity, is added to the sum of the clipped gradients. This noise effectively conceals the contribution of any single example to the sum, thus providing privacy guarantees. Note that raising the bound for gradient clipping also increases the variance of the Gaussian noise [Ryffel et al., 2018].

DP-SGD offers a robust and practical mechanism for implementing differential privacy in deep learning, ensuring that individual training examples do not overly influence the learning process while preserving privacy.

Algorithm 1: Differentially Private SGD

Data: Testing set x

Input : Training examples $[x_1 ... x_n]$
Loss function $\mathscr{L}(\theta) = \frac{1}{N}\sum_i \mathscr{L}(\theta, x_i)$,
Learning rate η_t,
Group size L,
Gradient norm bound C,
Total privacy budget ρ_{total}

Initialize θ_0
for $t = 1 : T$ **do**
  Take a batch of data samples $\mathscr{B}_t$ from training, with sampling probability $\frac{L}{N}$
  $B = |\mathscr{B}_t|$
  ▷ **Compute gradient**
  for $\forall i \in L_t$ **do**
    compute $g_t(x_i) \leftarrow \nabla_{\theta_t}\mathscr{L}(\theta_t, x_i)$
  end
  ▷ **Clip gradient**
  $\bar{g}_t(x_i) \leftarrow \bar{g}_t(x_i)/max(1, \frac{||\bar{g}_t(x_i)||_2}{C})$
  ▷ **Add noise**
  $\tilde{g}_t \leftarrow \frac{1}{L}(\sum_i g_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$
  ▷ **Descent**
  $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{g}_t$
end
Output: θ_t
Compute the overall privacy cost $(\epsilon,\ \delta)$ with a privacy accounting method

One of the primary challenges encountered in training models with differential privacy is the issue known as the "curse of dimensionality," as highlighted by [Bassily et al., 2014]. It refers to the fact that the accuracy of models trained with privacy protection tends to degrade as the number of dimensions increases. Regrettably, lower bounds in this field suggest that this dependency on dimensionality is an inherent constraint.

To effectively bound the impact of any training example, DP-SGD introduces two key alterations in every gradient step. First, it restricts each example's gradient contribution by imposing a fixed limit, often accomplished by *capping the l2 norm* of individual gradients. Second, it introduces random Gaussian noise with a magnitude proportional to the clipping norm into the combined gradient of each batch before propagating it backward to update the model parameters.

These modifications collectively create a new noise floor at each step of gradient descent. Consequently, the unique signal contributed by any individual example remains below this noise

floor. This fundamental change allows differential privacy to be guaranteed for all training examples, ensuring that no single training example can influence excessively the privacy of the model. This approach is explained in more details in *"The Algorithmic Foundations of Differential Privacy"* [Dwork and Roth, 2013].

3.1.2 DP-CTGAN

The necessity of this model arise from the unique properties of tabular data, such as correlated features, blended data types such as discrete or continuous features, difficulty in learning from highly sparse vectors and potential mode collapse due to high class imbalance. To mitigate these issues, [Ling et al., 2022] choosed CTGAN [Xu et al., 2019] as the underlying generative model.

Both the DPGAN and the DP-CTGAN frameworks are designed to introduce noise during the optimization step of the network. A noteworthy property of any (ϵ, δ)-differential privacy mechanism is its post-processing robustness, which ensures that any mapping operation performed after an already differentially private mechanism will also preserve differential privacy. In the context of GANs, this property is crucial, as it is frequently employed on the discriminator, contributing to the overall privacy guarantees of the model.

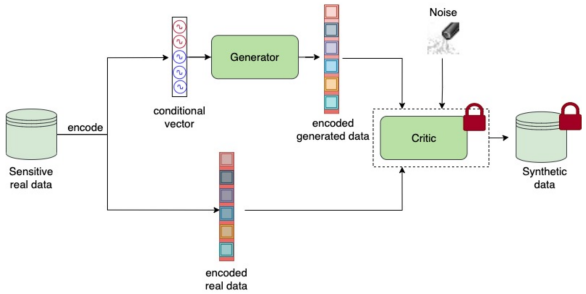


Figure 3.2 | DP-CTGAN. Sensitive training data is fed into a conditional generator. At the same time, random perturbation is added to the critic to enforce privacy [Ling et al., 2022].

The integration of a differentially private discriminator in the optimization process of the generator guarantees that the resulting generative network is also differentially private. This ensures that privacy preservation is maintained throughout the generation of synthetic data.

In the case of DP-CTGAN, the privacy guarantee is meticulously tracked using a privacy accountant. This approach incorporates a differential privacy framework within a CTGAN model to capture correlated feature patterns and complex data distributions while preserving privacy.

It's worth noting that the standard vanilla GAN lacks the ability to generate label-conditional data points. Therefore, a conditional GAN (CGAN), a well-established variant of the standard GAN, is employed to generate non-differentially private synthetic data.

Algorithm 2: Training DP-CTGAN

Data: Training data $\mathscr{D}_{train}$

Input : Conditional generator parameters ϕ_G,
Critic parameters ϕ_C,
Step size s,
Batch size m,
Gradient clipping bound $\mathscr{C}$,
Noise scale σ,
Privacy budget (ϵ_0, δ_0)

while $\epsilon \leq \epsilon_0$ **do**
  for $1 \leq j \leq m$ **do**
    $N_d \leftarrow$ number of discrete column from $\mathscr{D}_{train}$
    $d_i \leftarrow$ one-hot discrete vector
    Create masks $\{m_1, ..., m_i, ..., m_{N_d}\}_j$
    Create conditional vectors $cond_j$ from masks
    ▷ **Sample from multi-variable dist.**
    $z_j \leftarrow MVN(0, \mathbf{I})$
    ▷ **Generate synthetic data**
    $\hat{r}_j \leftarrow Generator(z_j, cond_j)$
    ▷ **Get real data**
    $r_j \leftarrow Uniform(\mathscr{D}_{train}, cond_j)$
    for $1 \leq k \leq s$ **do**
      sample $cond_k^j$, fake data $\hat{r}_k^j$, and real data r_k^j
    end
    $\mathscr{L}_C \leftarrow \frac{1}{s}\sum_{k=1}^{s}\left(Critic(\hat{r}_k^j, cond_k^j) - Critic(r_k^j, cond_k^j)\right) + \mathscr{L}_G$
    ▷ **Generate noise**
    $\xi \leftarrow \mathscr{N}^s\left(0, (\sigma C)^2 \mathscr{I}\right)$
    $\phi_C \leftarrow \phi_C - 0.0002 \cdot ADAM\left(\nabla_{\phi_C}(\mathscr{L}_C + 10\mathscr{L}_{\mathscr{G}}\mathscr{P} + \xi)\right)$
    $\mathscr{L}_G \leftarrow \frac{1}{m}\sum_{j=1}^{m} CrossEntropy(\hat{d}_{i,j}, m_i) - \frac{1}{m/s}\sum_{k=1}^{m/s} Critic\left(\hat{r}_k^s, cond_k^s\right)$
    $\phi_G \leftarrow \phi_G - 0.0002 \cdot ADAM\left(\nabla_{\phi_G}\mathscr{L}_G\right)$
    $\epsilon \leftarrow$ query $\mathscr{A}$ with δ_0
  end
end

Output: Parameters ϕ of a differentially private generator $\mathscr{G}$

Ling et al. [2022]

3.1.3 PATE-GAN

The Private Aggregation of Teacher Ensembles (PATE) of GAN, introduced in [Jordon et al., 2022b], is a novel approach designed to generate synthetic multivariate data while preserving the privacy of the training data. It represents a variation of the standard GAN architecture, where the traditional discriminator is replaced with the PATE mechanism.

In this modified setup, instead of a single discriminator, denoted as D, that is trained in a conventional adversarial manner with the generator G, there are k teacher-discriminators, namely, $T_1, T_2, ..., T_k$, along with a student discriminator S. A notable deviation from the conventional GAN training is the asymmetrical nature of the adversarial training process. In PATE-GAN, the teachers are trained to improve their loss concerning the generator G, while G is trained to enhance its loss with respect to the student S, which, in turn, is trained to minimize its loss concerning the teachers. This asymmetric training process constitutes a unique feature of the PATE-GAN framework.

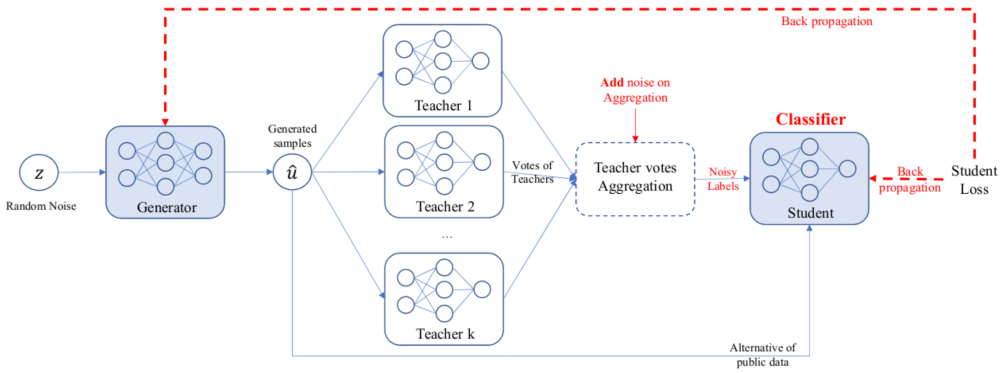


FIGURE 3.3 | Training procedure for the student-discriminator and the generator [Jordon et al., 2022b].

In the PATE-GAN framework, the generator G, as in the standard GAN architecture, is trained to minimize its loss in relation to the student discriminator. Formally, it is represented as a function $G(\cdot;\theta_G)$, parametrized by θ_G, and accepts random noise $\mathbf{z} \sim \text{Unif}([0,1]^d)$ as its input.

For a given set of n independent and identically distributed (i.i.d.) samples, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$, drawn from the uniform distribution $\text{Unif}([0,1]^d)$, the empirical loss of the generator G at parameter configuration θ, with the student discriminator S held constant, is defined as follows:

$$\mathscr{L}_G(\theta_G; S) = \sum_{j=1}^{n} \log\left(1 - S\left(G\left(\mathbf{z}_j; \theta_G\right)\right)\right).$$

Then each teacher-discriminator is trained to perform classification tasks, much like in a standard GAN network. However, there is a crucial distinction: each teacher-discriminator only has access to its own partition of the real data, which is derived from a disjoint subset of size $|D_i| = \frac{|D|}{N}$, where N represents the total number of teachers. Formally, these teachers are represented as functions $T_1(\cdot;\theta_T^1), T_2(\cdot;\theta_T^2), \ldots, T_k(\cdot;\theta_T^k) : \mathscr{U} \to [0,1]$, with each function parametrized by θ_T^i.

For a given set of n independent and identically distributed $(i.i.d.)$ samples drawn from $\text{Unif}([0,1]^d)$, denoted as $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$, the empirical loss of teacher i, with weights θ_T^i, is defined as follows when the generator G is held constant:

$$\mathscr{L}_T^i\left(\theta_T^i\right) = -\left[\sum_{\mathbf{u}\in\mathscr{D}_i} \log T_i\left(\mathbf{u};\theta_T^i\right) + \sum_{j=1}^{n} \log\left(1 - T_i\left(G\left(\mathbf{z}_j\right);\theta_T^i\right)\right)\right].$$

The student classifier, denoted as S, is trained by an unlabeled dataset $\mathscr{P} = \{x_i\}_{i=1}^K$. From this dataset, each sample, x_i is subjected to the standard PATE mechanism, which provides a differentially private label $\hat{y}_i$. This process results in the creation of a new dataset $\mathscr{P}' = \{x_i, \hat{y}i\}_{i=1}^K$, where each sample is associated with its corresponding noisy label. The student classifier S is then trained on this dataset $\mathscr{P}'$ to make predictions based on the noisy labels.

> The student, denoted as S, which has been trained on the dataset $\mathscr{P}'$ where labels were generated in accordance with the $PATE_\lambda$ mechanism using $\lambda = \frac{K}{2\epsilon}$, satisfies $(\varepsilon, 0)$-differential privacy with respect to the original data D.

This property of the differentiable student, which can be implemented using any classifier such as a neural network, results in it incurring no privacy costs when processing input and producing an output. The sole privacy cost is associated with acquiring the data used for training the student.

The student discriminator is introduced with the goal of emulating the behavior of the ensemble of teachers, as it is trained on teachers-labeled samples. Consequently, the student model can be trained privately without the need for public data, and the generator can leverage this process to enhance the quality of the generated samples.

The training data for the student is obtained by taking n independent and identically distributed (i.i.d.) samples from the uniform distribution $\mathrm{Unif}\left([0,1]^d\right)$, denoted as $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$. Next, n samples are generated using the generator, resulting in $\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_n$, where $\hat{\mathbf{u}}_j = G\left(\mathbf{z}_j\right)$. These generated samples are then labeled by the teachers using the PATE_λ mechanism, and the labels are represented as $r_j = \mathrm{PATE}_\lambda\left(\hat{\mathbf{u}}_j\right)$.

Subsequently, the student, denoted as $S(\cdot;\theta_S) : \mathscr{U} \to [0,1]$, is trained with the objective of maximizing the standard cross-entropy loss on this teacher-labelled data:

$$\mathscr{L}_S\left(\theta_S\right) = \sum_{j=1}^{n} r_j \log S\left(\hat{\mathbf{u}}_j;\theta_S\right) + \left(1 - r_j\right) \log\left(1 - S\left(\hat{\mathbf{u}}_j;\theta_S\right)\right)$$

For a fixed value of λ, it's important to note that having more teachers in the ensemble results in the teacher-labelled dataset being less noisy. When more teachers are involved, the noise added becomes smaller relative to the individual teacher's counts, n_j. This trade-off introduces a delicate balance.

On one hand, with a small number of teachers, the noise may be too prominent, rendering the output meaningless or unreliable due to the excessive perturbations introduced by the privacy

mechanism. On the other hand, if there are a large number of teachers, there is less data available to train each teacher, which can also compromise the quality of the output. Despite the reduction in noise, the limited training data may lead to inaccurate or unreliable results.

This can be traduced in the following algorithm:

Algorithm 3: PATE-GAN Algorithm

Data: Partition dataset in k subsets $\mathscr{D}_1, ..., \mathscr{D}_k$ of size $\frac{|\mathscr{D}|}{k}$
Input : Number of teachers k,
Noise λ,
Number of training steps for each teacher/student n_T, n_S,
Batch size b

$\forall\, l = 1, ..., L : \alpha(l) = 0$
while $\hat{\epsilon} < \epsilon$ **do**
  for $t_1 = 1, ..., n_T$ **do**
    Collect n samples $\{\boldsymbol{z_1}, ..., \boldsymbol{z_n}\} \rightarrow \boldsymbol{Z_a}$, from random noise distribution P_Z
    for $i = 1, ..., k$ **do**
      Collect n samples $\{\boldsymbol{u_1}, ..., \boldsymbol{u_n}\} \rightarrow \boldsymbol{U_a}$, from disjoint set $\mathscr{D}_j$
      ▷ **Update teacher** T_i **using SGD**
      $\nabla_{\theta_T^i} - \left[\sum_{j=1}^{d} \log\left(T_i\left(\mathbf{u}_j\right)\right) + \log\left(1 - T_i\left(G\left(\mathbf{z}_j\right)\right)\right)\right]$
    end
  end
  for $t_2 = 1, ..., n_S$ **do**
    Collect n samples $\{\boldsymbol{z_1}, ..., \boldsymbol{z_n}\} \rightarrow \boldsymbol{Z_a}$ from random noise distribution $\mathscr{P}_Z$
    for $j = 1, ..., n$ **do**
      $\hat{\boldsymbol{u}}_{\boldsymbol{j}} \leftarrow G(\boldsymbol{z_j})$
      $\hat{r}_j \leftarrow PATE_\lambda(\boldsymbol{u_i})$ for $j = 1, ..., n$
      ▷ **Update moments accountant**
      $q \leftarrow \frac{2+\lambda|n_0 - n_1|}{4\exp(\lambda|n_0 - n_1|)}$
      for $l = 1, ..., L$ **do**
        $\alpha(l) \leftarrow \alpha(l) + min\left\{2\lambda^2 l(l+1), \log\left((1-q)\left(\frac{1-q}{1-e^{2\lambda}q}\right)^l + qe^{2\lambda l}\right)\right\}$
      end
    end
    ▷ **Update student using SGD**
    $\nabla_{\theta_S} - \sum_{j=1}^{n} r_j \log\left(S\left(\hat{\boldsymbol{u}}_{\boldsymbol{j}}\right) + (1 - r_j)\log\left(S\left(\hat{\boldsymbol{u}}_{\boldsymbol{j}}\right)\right)\right)$
  end
  Collect n samples $\{\boldsymbol{z_1}, ..., \boldsymbol{z_n}\} \rightarrow \boldsymbol{Z_a}$ from random noise distribution $\mathscr{P}_Z$
  ▷ **Update the generator using SGD**
  $\nabla_{\theta_G}\left[\sum_{i=1}^{n} \log\left(1 - S(G(\boldsymbol{z_i}))\right)\right]$
  $\hat{\epsilon} \leftarrow min_l \frac{\alpha(l) + \log(\frac{1}{\delta})}{l}$
end
Output: G

Jordon et al. [2022b]

3.2 Privacy and Utility Metrics

Data quality is highly dependent on the context. When evaluating the quality of differentially private synthetic data, it is mandatory to keep in mind the specific needs and requirements of those who work with this type of data. This could help developing appropriate measures to tune better the models according to specific use cases and goals for the project. In this context, balancing privacy and utility is very challenging and therefore most balanced metrics were implemented to investigate the different trade-offs when generating new private data.

3.2.1 TRTR, TSTR and TSTS Settings

First, to evaluate the quality of the generated synthetic datasets, a set of ML models for classification are performed in three different training-testing settings:

- Setting **TRTR**: the predictive models are trained on the real training set and assessed their performance on the real testing set.
- Setting **TSTR**: the predictive models are trained on the synthetic training set and assessed their performance on the real testing set.
- Setting **TSTS**: the predictive models are trained on the synthetic training set and assessed their performance on the synthetic testing set.

The first one is the standard setting, and therefore it is considered as starting point when benchmarking the different results. Then, by assessing the predictive performance on real data for models trained on synthetic data (TSTR), we can determine how well the synthetic data is able to capture the relationship between the features and labels. Intuitively, synthetic data that performs well in this setting can be used to train models without the need for real data.

Another important aspect is to evaluate the consistency of relative performance between two algorithms when trained and tested on synthetic data (TSTS), compared to when trained and tested on real data (TRTR). A basic requirement is that if model 1 outperforms model 2 on real data, the same should hold true for synthetic data. This ensures that researchers can use the synthetic data to select the best method for their application to the real data or to provide it to the data-holder for real data testing.

3.2.2 Classification Algorithms and Evaluation Metrics

The performance of the datasets generated by these differentially private generative models is assessed by training a set of binary classifiers, from the *scikit-learn* python library [Pedregosa et al., 2011], namely:

1. Logistic Regression
2. Random Forest
3. AdaBoost
4. Bagging of Decision Trees
5. Gaussian Naive Bayes
6. Gradient Boosting

To obtain an overall score for the datasets, their single scores were then averaged for each type of setting. In addition the same evaluation was considered using the **repeated k-fold cross-validation,** a resampling method that iterate over different portions of the data to test and train each single model. In this way, it is possible to obtain a more stable result since training and testing is performed on different parts of the dataset and a more accurate estimate of the true unknown underlying mean performance of the model on the dataset, as calculated using the standard error.

Once the model were fitted, these have been evaluated using the **accuracy**, the area under the receiver operating characteristics curve (**AUC**), the **recall** and the **F1 score** (also named **Dice similarity coefficient**).

1. **Accuracy**: calculates the ratio of correctly predicted instances to the total instances in the dataset. Higher accuracy means that the model makes a large proportion of correct predictions.

2. **AUC**: it measures the area under the Receiver Operating Characteristic (ROC) curve, which quantifies how well the model distinguishes between positive and negative classes. A higher AUC value indicates better discrimination.

3. **Recall**: also known as *Sensitivity* or *True Positive Rate (TPR)*, is a metric that evaluates a model's ability to identify all positive instances in a dataset. It measures the proportion of actual positive instances correctly predicted as positive by the model.

4. **F1 Score**: is the harmonic mean of *precision* and *recall*. It combines both measures to provide a balanced assessment of a model's performance, as considers both false positives and false negatives.

3.2.3 Propensity Mean Squared Error Ratio Score (pMSE)

To assess the overall similarity of the joint distribution and determine the general quality of the synthetic copies [Arnold and Neunhoeffer, 2021], the **propensity score mean squared error (pMSE) ratio score** for synthetic data is computed [Snoke et al., 2018]. The pMSE score requires training a discriminator, in charge of distinguishing between real and synthetic examples. High general data quality in a synthetic dataset is indicated when the discriminator cannot differentiate between real and synthetic instances.

This score is calculated by dividing the pMSE by the expectation of the null distribution, so it is a way to assess how well the model is performing relative to its expected performance. The ratio provides a measure of how well the model's predictions align with expectations, a higher value of this metric suggests better performance.

3.2.4 Synthetic Ranking Agreement (SRA)

In line with the importance of ensuring that a synthetic dataset respects the ranking of models in terms of their prediction performances, another metric known as the **Synthetic Ranking Agreement (SRA)** [Jordon et al., 2018] is considered. Let's take the scenario where we have L predictive models, denoted as $f_1, f_2, \ldots, f_L$[2]. Furthermore, let's assume that the performance of model i when trained and tested on the real data (Setting TRTR) is represented as $A_i \in \mathbb{R}$, and the performance of the same model when trained and tested on the synthetic data (Setting TSTS) is denoted as $C_i \in \mathbb{R}$. We define the synthetic ranking agreement as follows:

$$\text{SRA}\left(\{A_i\}_{i=1}^{L}, \{C_i\}_{i=1}^{L}\right) = \frac{1}{L(L-1)} \sum_{j=1}^{L} \sum_{k \neq j} \mathbb{I}\left((A_j - A_k) \times (C_j - C_k) > 0\right)$$

where $\mathbb{I}$ is an indicator function. It is important to note that the summation results in a value of 1 when the ordering of algorithms j and k is the same in both settings (real and synthetic), and it equals 0 when the ordering in one setting differs from the ordering in the other.

The SRA can be conceptualized as the empirical probability of a comparison on the synthetic data being "correct" meaning that the comparison on the synthetic data matches the comparison that would be observed on the real data [Jordon et al., 2018]. This provides a measure of how well the rankings of predictive models are preserved when transitioning from real data to synthetic data.

A particularly interesting property of SRA is that it does not necessarily require the synthetic data to have the same distribution as the real data in order to be considered high-quality. This means

that even if the generated data differs from the original data in terms of its distribution, it can still be valuable for comparison purposes.

3.2.5 Privacy Risk Assessment

To evaluate the overall privacy level of the generated datasets, multiple membership inference attacks (MIA) were conducted at each privacy budget level.

This privacy risk assessment for synthetic datasets is based on Black-Box MIA attack using distances of members (training set) and non-members (holdout set) from their *nearest neighbors* in the synthetic dataset. By default, the *Euclidean distance* is used (*L2* norm). The privacy risk measure is the **share** of synthetic records closer to the training than the holdout dataset.

The member and non-member query probabilities are calculated based on their distance to the nearest neighbors among synthetic samples. This distance is referred to as the "distance to the closest record (DCR)," as defined by [Park et al., 2018].

3.3 Materials

3.3.1 Data

For the experimental setup, two publicly available datasets in particular were considered:

- the Adult Census Income[1] dataset, which is an extraction done by Barry Becker from the 1994 US Census database, were the task here is to predict whether income exceeds $50K/yr. Here the variables include most demographic information such as the age, gender, race, marital status, education level, and then the occupation and the class of work (private/public);

- the City Employee Payroll[2], which is an open data source of payroll information for all Los Angeles City Employees, updated bi-weekly. The variables include some PII columns such as the ID, full name, gender, race and postal code (which have been removed) and information about their occupation, such as job type, department, type of contract (full/part time), job status, the MOU classification code and information about their salaries and benefits;

[1] UCI Machine Learning Repository - Adult
[2] City of Los Angeles - City Employee Payroll

The choice of considering an open data source reflects the motivation of testing these applications on real world data, reflecting this scenario in which this sensitive type of data is made publicly available and could also be used as input for different privacy attacks.

Both the datasets accounted of thousands of observations, therefore a sample of 10.000 observations was considered in order to accelerate the models computation time. Some preprocessing procedures were applied before fitting the model and these include treating the missing values, removing some influential outliers, scaling the numerical variables, recoding factors and removing useless groups among categorical data, creating dummy variables and finally encoding the relative dependent variables.

3.3.2 Repositories

The models in this project were implemented using Python and various open-source libraries. One of the key components for data modeling was the **SmartNoise Synthesizers**[3] package, developed by OpenDP[4], a collaborative community focused on building trustworthy open-source software tools for statistical analysis of sensitive private data. The synthesizers in this package are designed to run on PyTorch integrated with **Opacus**[5], a scalable and optimized library for incorporating differential privacy into PyTorch neural networks and machine learning models.

In addition, other important open-source python packages used in this project come from the **AI Privacy 360 Toolbox**[6], in development by IBM. This framework includes a range of tools designed to assist in evaluating the privacy risks associated with AI-based solutions, allowing for the exploration of trade-offs between privacy, accuracy, and performance throughout the different stages of the machine learning (ML) lifecycle. Among these:

- **diffprivlib**[7] [Holohan et al., 2019] is specifically designed for training differentially private models. It follows a similar logic to the widely-used *Scikit-learn* framework.
- **apt**[8], in particular the *apt.risk* submodule was used to perform and asses membership inference attacks on both the real and synthetic datasets.

[3] SmartNoise Synthesizers
[4] Open Source Tools for Differential Privacy
[5] Opacus - train PyTorch models with Differential Privacy
[6] AI Privacy 360 Toolbox
[7] IBM Differential Privacy Library
[8] APT package

Results and Discussion

4

Training several generative adversarial networks, specially on large data, require large computational power. Indeed, this has been the greatest limitation of these analysis, as it was not even possible to execute these models on the payroll dataset due to its complexity, except for some DP-CTGANs trained over few epochs ($\approx$ 80).

Training times took many hours to complete the different fittings, so each model was executed within different MS Azure machines (CPU only), each with a batch size relatively small. A larger batch size trains faster but may result in the model not capturing the nuances in the data, and as our final goal is to generate good synthetic data, this could have led to more generalized results.

4.1 Experiments Results

The object was to train differentially private generative models, by investigating the effects of the privacy constraints (ϵ, δ). The choice of the ϵ parameter has been investigated by varying its values, whereas the δ was fixed to be small, namely $\delta = 1\ /(n * \epsilon)$.

Table 4.1 | DPGAN synthetic dataset results

		DPGAN							
		Baseline Scores				Cross Validation Scores			
Setting	ε	Accuracy	AUC	F1	Recall	Accuracy	AUC	F1	Recall
TRTR		82.05%	79.54%	71.09%	74.53%	77.39%	88.10%	60.90%	63.85%
TSTR		74.88%	65.89%	39.70%	48.10%	65.01%	50.11%	11.57%	20.26%
TSTS	0.1	93.86%	86.67%	80.69%	75.55%	90.45%	92.78%	69.47%	64.05%
	10	93.60%	82.33%	96.04%	93.65%	93.43%	85.62%	95.96%	93.60%
	100	95.36%	82.24%	58.15%	67.91%	94.85%	92.35%	22.45%	32.08%

The table 4.1 presents the accuracy metrics results, which were computed using the datasets generated by the DPGAN synthesizer under different evaluation settings. The TSTS ("train synthetic, test synthetic") setting is then expanded according to the various values of ϵ fixed.

From the results, it is possible to notice that the classification scores for synthetically generated datasets (TSTS) are close to those of non private datasets (TRTR), for some parameters these are even greater than the performance of classifiers on the original data. However, as outlined in section 2, GANs tends to overfit training data, therefore this could be due to a generalization of the synthetic data, which was not able to capture the whole variability of source data.

Table 4.2 | DP-CTGAN synthetic dataset results

		DP-CTGAN							
		Baseline Scores				Cross Validation Scores			
Setting	ε	Accuracy	AUC	F1	Recall	Accuracy	AUC	F1	Recall
TRTR		83.38%	79.78%	70.64%	73.09%	79.98%	87.56%	60.61%	62.02%
TSTR		75.55%	65.98%	39.32%	47.70%	66.92%	49.55%	9.69%	18.19%
TSTS	0.1	76.93%	77.44%	74.56%	79.30%	69.33%	78.47%	58.46%	57.38%
	10	89.39%	89.29%	83.02%	88.94%	85.28%	92.65%	74.96%	80.64%
	100	84.84%	82.58%	88.74%	92.27%	79.50%	88.23%	83.77%	85.18%

This is also suggested by the TSTR ("train synthetic, test real") scores. The performances for these settings are significantly smaller compared with the standard TRTR setting, implying that the synthetic set is not a good candidate to substitute the original set. For example, the TSTR recall and F1 score dropped by more that one third from the TRTR one.

Instead, this condition is not seen within the DP-CTGAN or PATEGAN metrics, as shown in table 4.2 and table 4.3 respectively. Here, the results show really close values between the TRTR and TSTS settings; particularly in the PATEGAN generated set the most of these values do not surpass the real settings. This suggests a more trustworthy model, compared to the DPGAN.

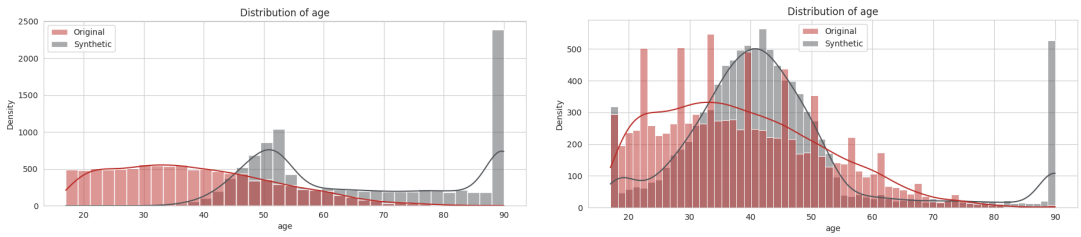
Table 4.3 | PATEGAN synthetic dataset results

		PATEGAN							
		Baseline Scores				Cross Validation Scores			
Setting	ε	Accuracy	AUC	F1	Recall	Accuracy	AUC	F1	Recall
TRTR		82.18%	79.76%	71.44%	75.28%	77.79%	88.50%	61.48%	64.68%
TSTR		83.15%	65.90%	34.36%	32.56%	66.23%	50.23%	10.20%	18.91%
TSTS	0.1	84.52%	71.83%	53.48%	49.73%	76.91%	70.19%	27.29%	22.05%
	10	85.62%	75.06%	91.13%	95.44%	76.99%	72.72%	85.70%	91.15%
	100	77.53%	73.56%	61.97%	65.96%	74.70%	76.39%	43.25%	39.73%

Another evidence in support of the reliability of PATEGAN is evidenced by the variables distributions, which showed a better approximation of the real counterparts, in particular when

increasing the privacy loss, as shown in Figure 4.1. This could depend on the PATE mechanism, which allows tighter bounds on the influence of a single sample on the discriminator, and hence tighter differential privacy guarantees, when the differential privacy guarantee is fixed.

Although sometimes for the same privacy budget, differential privacy techniques that produce tighter bounds and result in lower noise requirements come with increased concrete privacy risks [Chen et al., 2020].



(a) Variable generated with ϵ = 0.1 **(b)** Variable generated with ϵ = 10

FIGURE 4.1 | Synthetic variables densties generated with different values of epsilon agains real data densities

When we consider the privacy budget of a model, we can observe that greater values of ϵ imply fewer privacy constraints being imposed. This, in turn, suggests a more transparent generation of data. Consequently, if we increase the privacy budget parameter, we can expect an improvement in the model's performance on utility metrics.

This can be clearly seen at Figure 4.2, where the accuracy, F1 and recall of the DP-CTGAN model are shown for the different models fitted, varying the ϵ parameter and using different validation techniques.

The overall score aligns with the theoretical expectations, regardless of the validation technique implemented.

Another piece of evidence that further supports this idea can be found in the PATE-CTGAN approach. Similar to the PATEGAN method, the PATE-CTGAN technique incorporates the concept of private aggregation of teacher ensembles over a conditional tabular GAN.

Due to its complexity, this model was poorly trained over few teachers discriminators but still the results

TABLE 4.4 | ε-tuning synthetic datasets from PATE-CTGAN

	PATE-CTGAN			
ε	**Accuracy**	**AUC**	**F1**	**Recall**
0.1	72.64%	67.35%	72.64%	67.35%
0.5	75.14%	66.79%	75.14%	66.79%
1.0	82.12%	66.31%	82.12%	66.31%
5.0	83.02%	78.01%	83.02%	78.01%
10	86.19%	85.59%	86.19%	85.59%
25	85.65%	85.68%	85.65%	85.68%
100	86.04%	85.52%	86.04%	85.52%

are in line with previous findings. These can be seen in Figure 4.3, which shows TSTS accuracy and value of the area under the ROC curve over different values of ϵ.

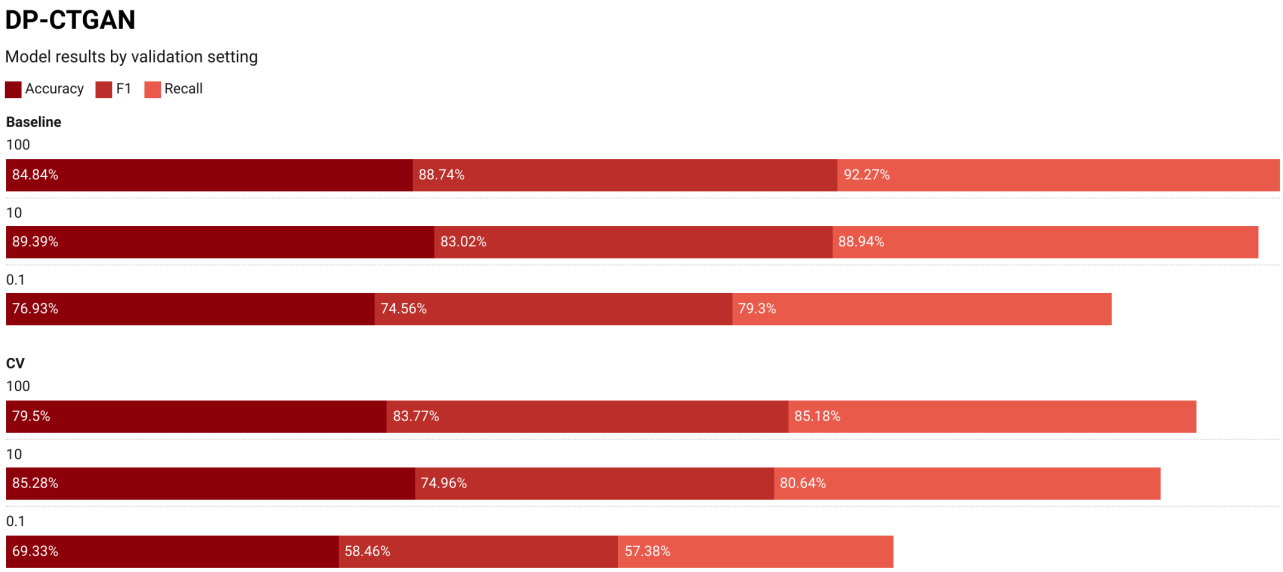
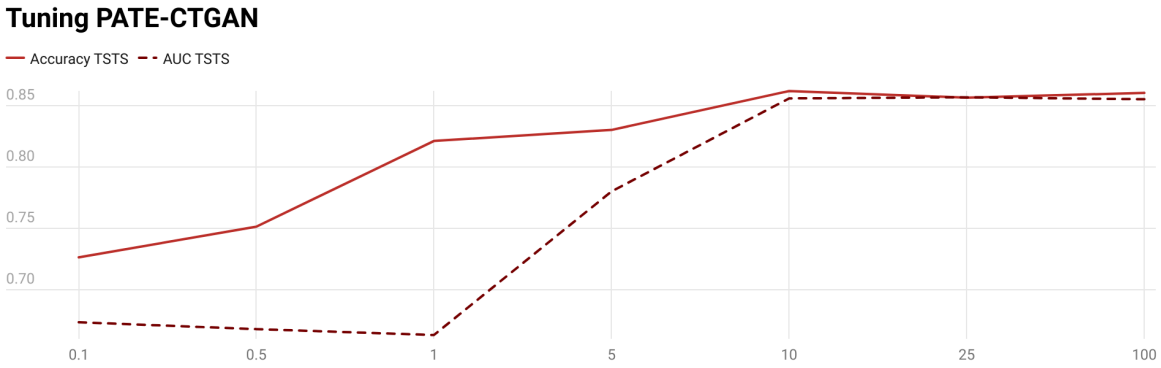


Figure 4.2 | Accuracy, F1 and recall of tuning ϵ over different DP-CTGAN synthetizers

To investigate this relationship further, a comprehensive measure of association between categorical variables was computed for both the original dataset and the different ϵ-DP synthetic datasets.

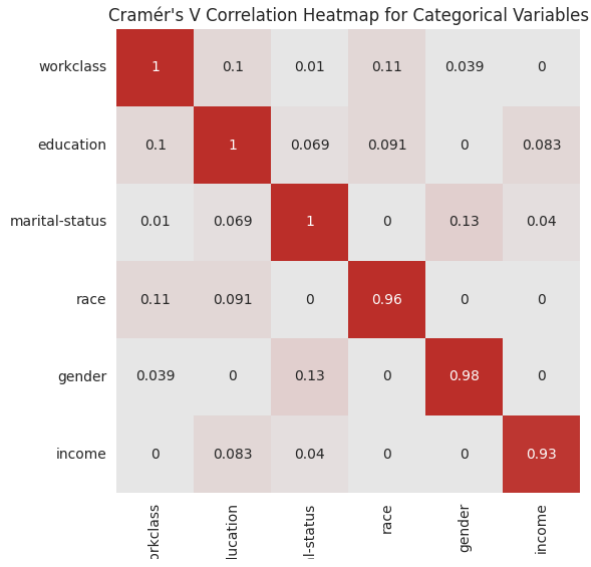
Figure 4.3 | Tuning results for ϵ in PATE-CTGAN



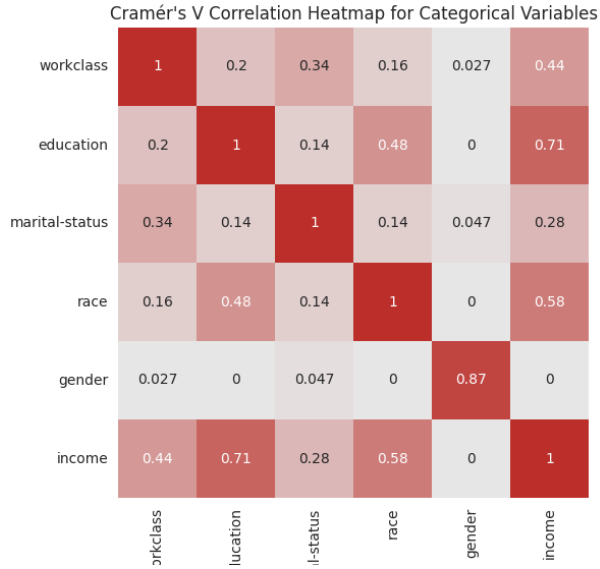
The measure employed for this analysis was the Cramer's V statistic, which serves as an indicator of the strength of correlation and ranges between 0 (indicating no association) and 1 (indicating perfect association). The results of this analysis are presented in Figure 4.4, which displays the four pairwise correlation matrices for Cramer's V. Specifically, Figure 4.4a illustrates the dataset

generated with the highest privacy budget, while Figure 4.4d depicts the original dataset without any privacy constraints.

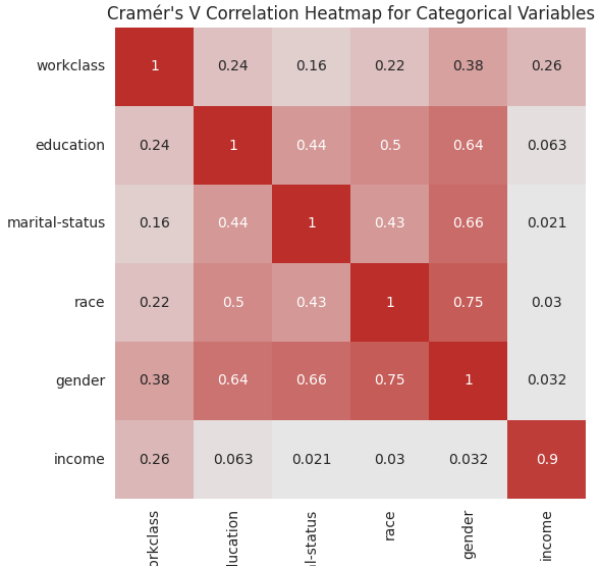
Upon examining the heatmaps of these correlation matrices, we can observe an interesting trend. As we transition from Figure 4.4a to Figure 4.4d, the pairwise Cramer's V statistics become progressively more similar to the results obtained from the original dataset. Additionally, it is worth noting that the least differentially private dataset, displayed in Figure 4.4c, exhibits a correlation heatmap that accurately reflects the original pattern in Figure 4.4d.



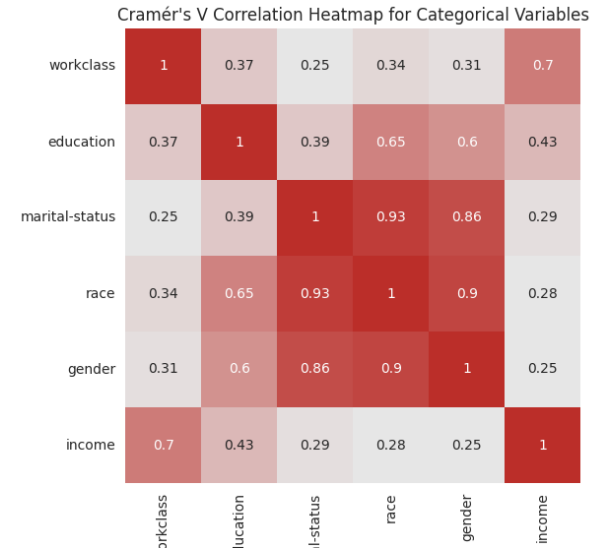
(a) Synthetic data with $\epsilon = 0.1$



(b) Synthetic data with $\epsilon = 10$



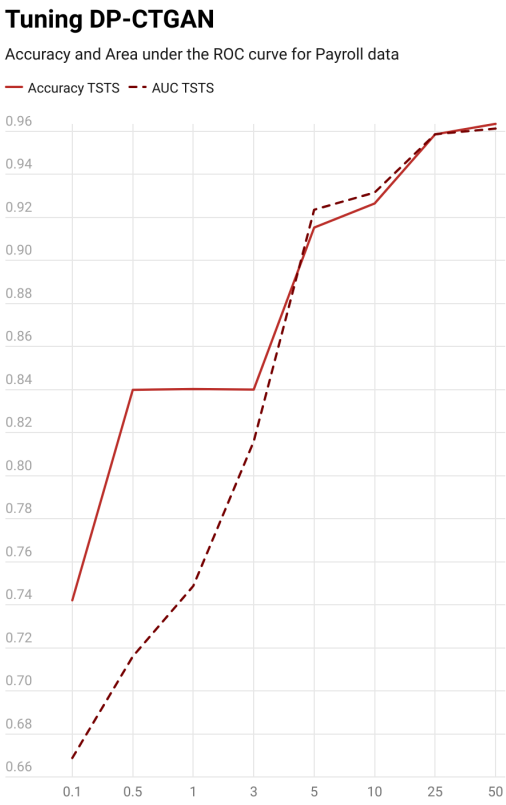
(c) Synthetic data with $\epsilon = 100$



(d) Original data without privacy constraints

FIGURE 4.4 | Heatmaps of Cramer's V pairwise correlations matrices

Unfortunately, the results shown could be assesed only on census dataset. As for the other dataset, the required computational resources exceeded our limited power. Only a DP-CTGAN, trained with smaller number of epoch, successfully converged. The results are shown on the side figure, where we see accuracy and AUC scores (TSTS), which still confirmed the linear pattern between greater privacy budget and model accuracies.



However, the use of differential privacy comes at a cost, primarily in terms of reduced output accuracy. This cost depends on factors such as the desired level of privacy, the size of the dataset, and the range of possible values for each individual. It is also influenced by the amount of information being released, so it is important to previously define the variable types or their bounds and preprocess the data previously, instead of adapting a solution which infer this information or automatically preprocess the data.

For this reason, also other metrics to asses the utility of this data have been considered. Among these, the pMSE ratio scores, which can be seen as the ratio between observed utility and expected utility, are displayed below. Here, PATE-synthesizers reported the highest scores; the best results were achieved with larger privacy budgets, aligning with increased utilities.

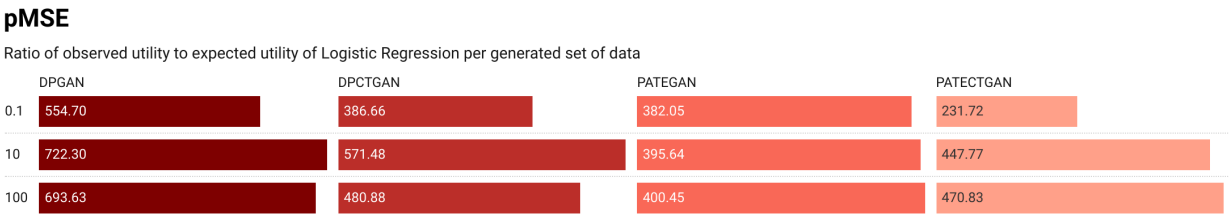


Figure 4.5 | Evaluation of pMSE across different models generated data

Another metric useful for data comparison is the synthethic ranking agreement, which on figure on the side is displayed for the recall. This metric serves as in indicator of consistency in the rankings of predictive models when transitioning from real data to synthetic data. The results are in line with previous findings; the PATEGAN was found as the most consistent synthesizer.

SRA

Recall agreemeent heatmap by model

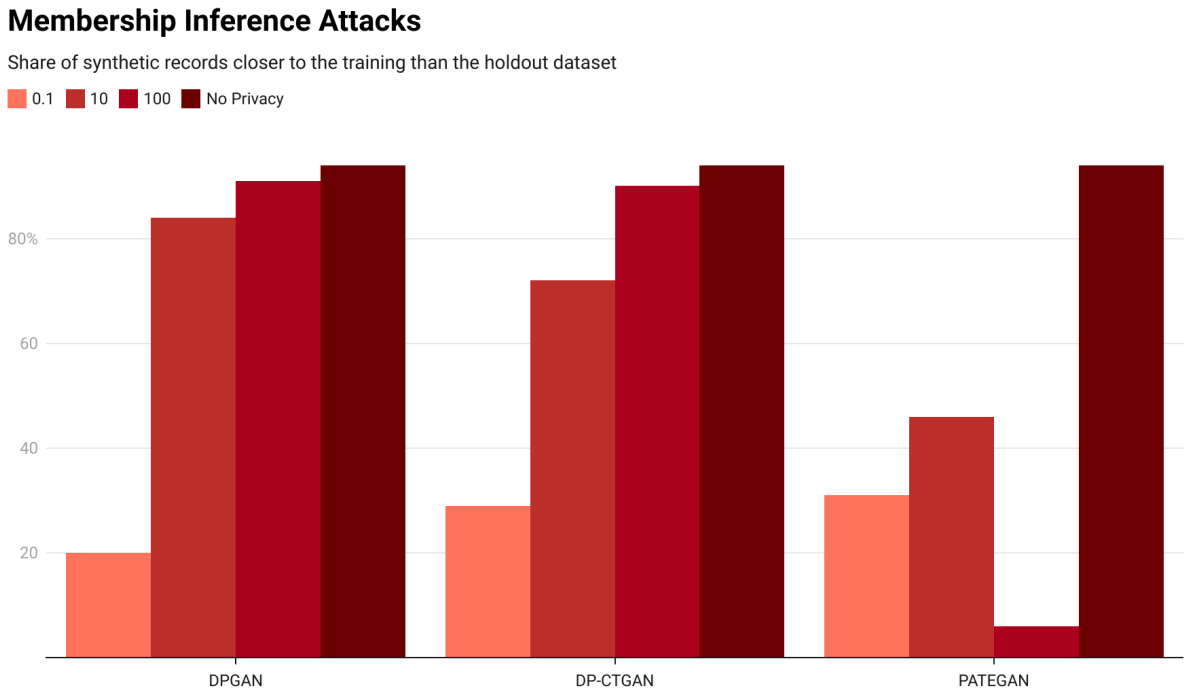
ε	DPGAN	DP-CTGAN	PATEGAN
0.1	53%	67%	67%
10	60%	60%	80%
100	87%	80%	100%

4.2 Privacy Attacks

Although tighter cumulative privacy loss bounds, which are offered by different forms of differential privacy, greatly improve the model's usefulness within a specific privacy budget, it is important to understand that the decrease in noise also increases the vulnerability to privacy attacks. These attacks take advantage of the reduced amount of noise to gain unauthorized access to sensitive information, putting the overall privacy at risk.

Figure 4.6 show the results of each membership inference attack performed against the original dataset and the different datasets generated with different privacy budgets, namely $\epsilon = 0.1$, 10, 100. More precisely, it displays the percentage share of records which were found closer to the training than the holdout dataset. In other words, this is the proportion of observations which were likely being part of the real data.

Figure 4.6 | Membership inference attack over synthetic datasets generated with different values of ε



The darkest red column corresponds to the NP dataset, whereas the lightest one shows the most privatized synthetic dataset. There is a significant jump (from 67% to 78% percentage decrease) in the shares of records reveled by this attack, which is remarkably evidenced in all the three generative models. However, this reduction is particularly high when setting low values of ϵ, but still reflects the efficiency of applying DP.

Membership inference attack calculates probabilities of member and non-member samples to be generated by the synthetic data generator.

The assumption is that since the generative model is trained to approximate the training data distribution then the probability of a sample to be a member of the training data should be proportional to the probability that the query sample can be generated by the generative model.

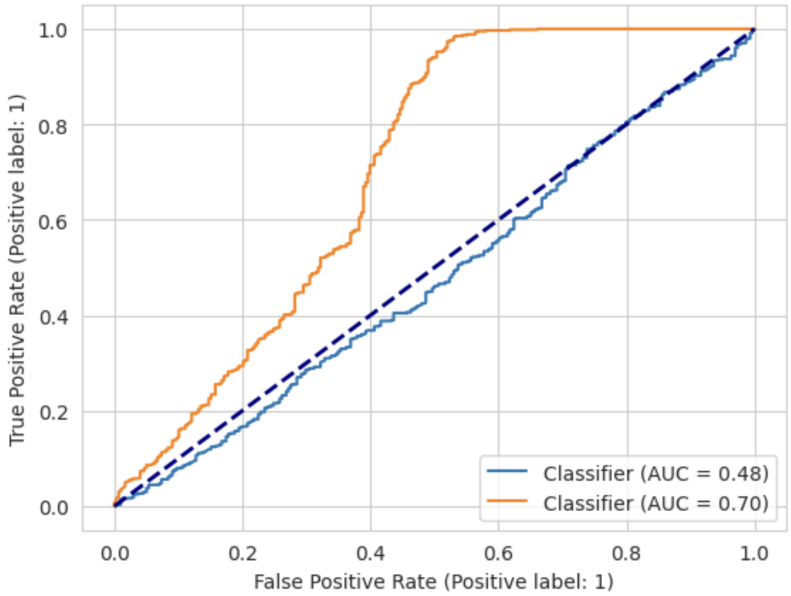
TABLE 4.5 | Membership inference attacks results

	Share Observations MIA Attack		
ε	DPGAN	DP-CTGAN	PATEGAN
0.1	20%	29%	31%
10	84%	72%	46%
100	91%	90%	6%
-	94%	94%	94%

So, if the probability that the query sample is generated by the generative model is large, it is more likely that the query sample was used to train the generative model. This probability is approximated by the Parzen window density estimation, computed from the NN distances from the query samples to the synthetic data samples.

The area under the receiver operating characteristic curve (AUC ROC), displayed in Figure 4.7, gives another privacy risk measure. In the figure, the AUCs of membership inference attacks over not private data (yellow) and 0.1 DP data (blue) provide further evidences on the efficacy of applying DP to protect against privacy attacks. For the former the attack obtained a success score of 70%, compared to the latter which was significantly lower, precisely it obtained an AUC score of 48%.

FIGURE 4.7 | ROC curves of MIA attacks over NP data (yellow) and 0.1 DP data (blue)



4.3 Differentially Private ML Models

Differential privacy, among its several applications, can also be applied within common machine learning algorithms. This section demonstrate how DP can be achieved also within supervised learning on a tabular dataset. This ensures that the contribution of the individuals' data to the resulting machine learning model is masked out. Consequently it is not possible that information of individuals may be leaked from the trained machine learning model.

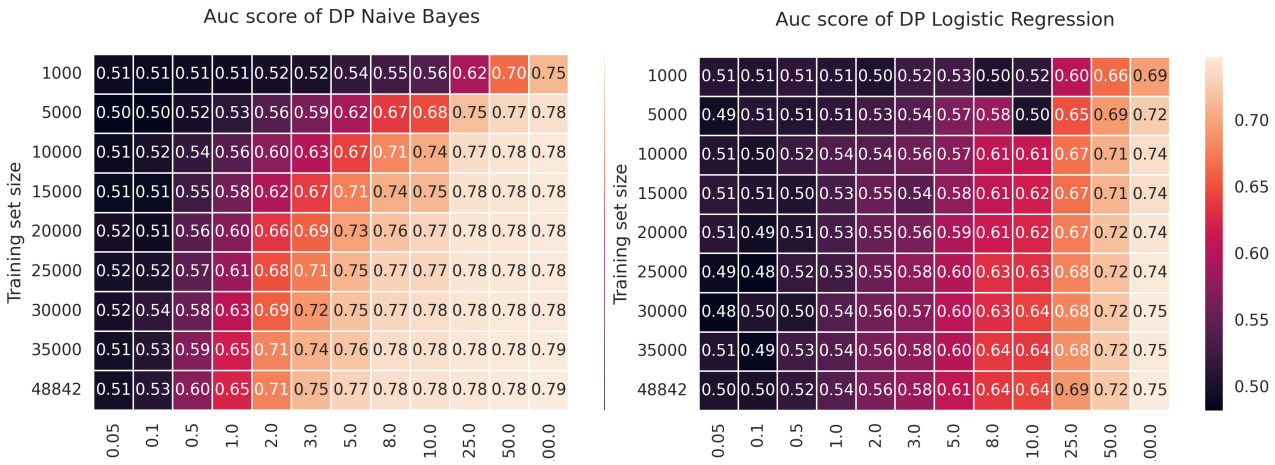
Table 4.6 | Classification accuracies metrics of differentially private ML models

		Evaluating DP-ML Models			
		No Privacy Constraints		Differential Privacy *(ε = 1)*	
Setting	Class	GaussianNB	Logistic Regression	GaussianNB	Logistic Regression
Accuracy		64%	85%	75%	79%
AUC		74%	75%	57%	71%
Precision	0	96%	88%	79%	86%
	1	39%	72%	44%	56%
Recall	0	56%	93%	91%	86%
	1	92%	58%	23%	57%
F1 Score	0	70%	90%	85%	86%
	1	55%	64%	30%	56%

In table 4.6, the results of the analysis are shown. The models considered are the logistic regression and the gaussian naive bayes, both computed with a privacy budget of 1 and without privacy constraints. We can see that the overall performance of differentially private models was found to be good. As expected, the DP scores were found slightly lower but still very competitive with the NP counterparts as their averages ranged between 60%-70%. Particularly, the score of DP logistic regression decreased in accuracy and AUC of just 6 and 4 basis points respectively.

Furthermore, another point of interest was to assess the impact of the dataset size and its complexity on resulting privacy levels and accuracy of the machine learning model. The performance of machine learning models can generally be improved by providing more training data. The following part assesses to which extend the accuracy impact, due to the noise ingestion, can be compensated by increasing the size of the training set. It is important to recall that lower ε values are related with higher statistical privacy guarantees.

The results, displayed in Figure 4.8, provide AUC scores for different levels of privacy on the x-axis, controlled by the parameter epsilon ϵ, and decreasing training set sizes on the y-axis.



(a) DP Naive Bayes **(b)** DP Logistic Regression

Figure 4.8 | AUC scores of DP-ML models by ϵ and training set size

The different scores follow a clear pattern, reducing the privacy constraints on the model and increasing the number of observations result in higher accuracy scores. Another interesting finding is that for very small values of ϵ the score was found to be always the same, despite the set size. Similarly, increasing the privacy budget without feeding the model with large amount of data will result in low scores. This result provides a valuable evidence to consider when tuning hyperparameters among with privacy constraints in a machine learning model.

All in all, results show that performing machine learning based on a differentially private algorithm can lead to comparable performance like when using a normal classifier. However, research shows that this is not always the case depending on the dataset's size and characteristics, and other factors.

The synthesizer approach's main advantage is that the resulting dataset can be shared and used for analytical purposes any number of times without increasing the risks associated with privacy loss. Another advantage is that the synthesizer allows producing any arbitrary amount of data derived from the original dataset's distribution. This could be a promising approach for data augmentation to improve the resulting machine learning model's quality.

Conclusion

5

This thesis, in addition to serving as a comprehensive guide on the latest state-of-the-art Privacy-Enhancing Techniques to address the important issues related to privacy attacks, aimed at providing evidence in the field of differential privacy, by evaluating the latest techniques discovered in most recent years. In particular, this work focused on assessing privacy and utility levels within synthetic data created from differentially private generative adversarial networks, for different values of privacy budgets and datasets and by executing concrete attacks against these models.

The applicability of DP in generative methods has been outlined and results show that for certain generative models with DP, the performance was competitive and sometimes better than their non-private counterparts while addressing a reasonably good privacy guarantee.

Furthermore, it was demonstrated that the performance of produced models strongly depends on the chosen privacy budget, with low values reducing model utility. Nevertheless, the use of higher privacy budgets allows training models that show only a moderately reduced performance in comparison to the non-private baseline models.

This should motivate ML engineers to include DP in their systems, as the perfect trade-off between privacy and utility could be met. Data custodians must be aware of the potential risks and employ appropriate techniques to mitigate them while maintaining data usability. These findings also provided evidence on reduced privacy-related risks when applying DP. Although understanding and assessing deeper the residual privacy risks associated with different data protection methods is essential to ensure that individuals' privacy is adequately protected.

Finally, it was shown that performing machine learning based on a differentially private algorithm can lead to comparable performance to a normal classifier. In this line, could be interesting investigating in whether a DP-ML model is comparable to a ML model applied to a DP dataset.

5.1 Limitations and Future Work

Training generative adversarial networks, specially when integrating the PATE framework, comes with several challenges. First of all, this include the necessity of working with a very powerful machine to meet the intense computational costs required to correctly train these networks. This limitation affected the results as it was not possible to perform hyperparameters tuning within

the models, more than the crucial ones for the scope of this work. Also, it was not possible to evaluate these results on a more complex data set, to be able to generalize more these findings.

In addition, it is well documented in the literature that GANs are unstable during training. The large bias produced by the critic in the gradient of the generator, when mixed with the imposed gradient noise by the differential privacy, can increase training instabilities. These issues are reported by the GANs-ML community, therefore next work should focus mostly in finding the best practices to overcome these limitations.

GANs' instability is present also when performing hyperparameters tuning, as these tend to be highly sensitive to slight changes in parameters, affecting the quality of synthetic datasets [Frigerio et al., 2019]. Future work could focus on implementing DP within other generative models and compare the results, by trying over different datasets used in the industry with higher complexity and number of classes.

Moreover, the variety of privacy attacks used to evaluate the synthetically generated datasets was limited due to the constrained scope of this work, so this study could be expanded by including more attack types in the future. Although, it is worth mentioning that the resources available on the web are still limited. For example, most of the libraries used for these analysis are still on the making, with latest commits dated to some week before.

Bibliography

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS16*, 2016. doi: 10.1145/2976749.2978318. URL https://arxiv.org/abs/1607.00133.

Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks, 07 2018. URL https://arxiv.org/abs/1709.04514.

Mohammad Al-Rubaie and J. Morris Chang. Reconstruction attacks against mobile-based continuous authentication systems in the cloud. *IEEE Transactions on Information Forensics and Security*, 11:2648–2663, 12 2016. doi: 10.1109/tifs.2016.2594132.

Differential Privacy Team Apple. Learning with privacy at scale, 2017. URL https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf.

Christian Arnold and Marcel Neunhoeffer. Really useful synthetic data: A framework to evaluate the quality of differentially private synthetic data. *arXiv:2004.07740 [cs, stat]*, 10 2021. URL https://arxiv.org/abs/2004.07740.

Sean Augenstein, H. Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, and Blaise Aguera y Arcas. Generative models for effective ml on private, decentralized datasets, 02 2020. URL https://arxiv.org/abs/1911.06679.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds, 10 2014. URL https://arxiv.org/abs/1405.7085.

Steven M. Bellovin, Preetam K. Dutta, and Nathan Reitinger. Privacy and synthetic datasets. *SSRN Electronic Journal*, 2018. doi: 10.2139/ssrn.3255766.

Justin Brickell and Vitaly Shmatikov. The cost of privacy. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*, 2008. doi: 10.1145/1401890.1401904.

Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence, 2021. URL https://arxiv.org/abs/2111.01177.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. URL https://arxiv.org/pdf/2112.03570.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. *arXiv:2012.07805*, 06 2021. URL https://arxiv.org/abs/2012.07805.

Kamalika Chaudhuri and Staal Vinterbo. A stability-based validation procedure for differentially private machine learning, 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/e6d8545daa42d5ced125a4bf747b3688-Paper.pdf.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, page 343–362, 10 2020. doi: 10.1145/3372297.3417238. URL https://arxiv.org/abs/1909.03935.

Dongjie Chen, Sen-ching Samson Cheung, Chen-Nee Chuah, and Sally Ozonoff. Differentially private generative adversarial networks with model inversion. *2021 IEEE International Workshop on Information Forensics and Security (WIFS)*, 12 2021. doi: 10.1109/wifs53200.2021.9648378. URL https://arxiv.org/pdf/2201.03139.pdf.

Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. Differentially private data generative models, 12 2018. URL https://arxiv.org/abs/1812.02274.

European Commission. White paper on artificial intelligence: a european approach to excellence and trust, 2020. URL https://commission.europa.eu/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en.

European Parliament Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj.

Kenneth Cukier and Viktor Mayer-Schoenberger. The rise of big data how it's changing the way we think about the world. URL https://cs.brown.edu/courses/cs100/lectures/readings/riseOfBigData.pdf.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale, 06 2022. URL https://arxiv.org/abs/2204.13650.

Emiliano De Cristofaro. An overview of privacy in machine learning. *arxiv.org*, 05 2020. doi: 10.48550/arXiv.2005.08679. URL https://arxiv.org/abs/2005.08679.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407, 2013. doi: 10.1561/0400000042.

Cynthia Dwork, Krishnaram Kenthapadi, Frank Mcsherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation, 2006a. URL https://www.iacr.org/archive/eurocrypt2006/40040493/40040493.pdf.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis, 2006b.

P. Vepakomma A. Singh R. Raskar H. Esmaeilzadeh F. Mireshghallah, M. Taram. Privacy in deep learning: A survey, 04 2020.

Liyue Fan. A survey of differentially private generative adversarial networks. URL https://www2.isye.gatech.edu/~fferdinando3/cfp/PPAI20/papers/paper_9.pdf.

Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10:2733, 08 2022. doi: 10.3390/math10152733.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, 2015. doi: 10.1145/2810103.2813677. URL https://rist.tech.cornell.edu/papers/mi-ccs.pdf.

Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data, 03 2019. URL https://arxiv.org/abs/1901.02477.

Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. De-anonymization attack on geolocated data. *Journal of Computer and System Sciences*, 80:1597–1614, 12 2014. doi: 10.1016/j.jcss.2014.04.024.

Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv:1712.07557*, 03 2018. URL https://arxiv.org/abs/1712.07557.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets, 2014. URL https://arxiv.org/pdf/1406.2661.pdf.

William Gu. Privacy preserving synthetic data generation. URL https://www.mi.fu-berlin.de/inf/groups/ag-idm/theseses/2021_Gu_BSc.pdf.

Frederik Harder, Kamil Adamczewski, Mijung Park, and Equal Contribution. Differentially private mean embeddings with random features for synthetic data generation. URL http://proceedings.mlr.press/v130/harder21a/harder21a.pdf.

Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:133–152, 12 2018. doi: 10.2478/popets-2019-0008. URL https://arxiv.org/pdf/1610.05820.

Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Rebecca N. Wright. Privacy-preserving machine learning as a service. *Proceedings on Privacy Enhancing Technologies*, 2018:123–142, 06 2018. doi: 10.1515/popets-2018-0024.

Naoise Holohan, Stefano Braghin, Pol Mac Aonghusa, and Killian Levacher. Diffprivlib: The ibm differential privacy library, 2019. URL https://www.arxiv-vanity.com/papers/1907.02444.

Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon, 2014. URL https://arxiv.org/pdf/1402.3329.pdf.

Aryan Jadon and Shashank Kumar. Leveraging generative ai models for synthetic data generation in healthcare: Balancing research and privacy. URL https://arxiv.org/pdf/2305.05247.pdf.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. URL https://arxiv.org/pdf/1902.08874.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. *arXiv:1902.08874 [cs, stat]*, 08 2019. URL https://arxiv.org/abs/1902.08874.

Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. 12 2014. doi: 10.48550/arxiv.1412.7584.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Measuring the quality of synthetic data for use in competitions, 06 2018. URL https://arxiv.org/abs/1806.11345.

James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. Synthetic data – what, why and how? *arXiv:2205.03257 [cs]*, 05 2022a. URL https://arxiv.org/abs/2205.03257.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees, 02 2022b. URL https://openreview.net/forum?id=S1zk9iRqF7.

Mei Ling, Devendra Singh, Dhami Devendra, and Kristian Kersting. Dp-ctgan: Differentially private medical data generation using ctgans, 2022. URL https://ml-research.github.io/papers/fang2022dpctgan.pdf.

Claire Little, Mark Elliot, Richard Allmendinger, Shariati Sahel, and Samani. Generative adversarial networks for synthetic data generation: A comparative study. URL https://arxiv.org/pdf/2112.01925.pdf.

David Liu and Nathan Hu. Gan-based image data augmentation stanford cs229 final project: Computer vision. URL http://cs229.stanford.edu/proj2020spr/report/Liu_Hu.pdf.

S. Merugu and Joydeep Ghosh. Privacy-preserving distributed clustering using generative models, 2003.

Alejandro Mottini, Alix Lheritier, and Rodrigo Acuna-Agost. Airline passenger name record generation using generative adversarial networks, 07 2018. URL https://arxiv.org/abs/1807.06657.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets, 2008.

OECD. Emerging privacy-enhancing technologies. 2023. doi: https://doi.org/https://doi.org/10.1787/bf121be4-en. URL https://www.oecd-ilibrary.org/content/paper/bf121be4-en.

Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. 04 2023. doi: 10.1145/3595292.

Nicolas Papernot, Brain Google, M. Lfar, Erlingsson Google, Ian Goodfellow, Google Brain, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data, 2017. URL https://arxiv.org/pdf/1610.05755.pdf.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate, 02 2018. URL https://arxiv.org/abs/1802.08908.

Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Ulfar Erlingsson. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy. *openreview.net*, 09 2019. URL https://openreview.net/forum?id=rJg851rYwH.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11:1071–1083, 06 2018. doi: 10.14778/3231751.3231757.

European Parliament. Article 29 data protection working party, 2014. URL https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jianwei Qian, Xiang-Yang Li, Chunhong Zhang, and Lin-Lin Chen. De-anonymizing social networks and inferring private attributes using knowledge graphs. *IEEE International Conference Computer and Communications*, 04 2016. doi: 10.1109/infocom.2016.7524578.

Robert Nikolai Reith, Thomas Schneider, and Oleksandr Tkachenko. Efficiently stealing your machine learning models. *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society - WPES'19*, 2019. doi: 10.1145/3338498.3358646.

Wang Ren, Xin Tong, Jing Du, Na Wang, Shancang Li, Geyong Min, and Zhiwei Zhao. Privacy enhancing techniques in the internet of things using data anonymisation. *Information Systems Frontiers*, 05 2021. doi: 10.1007/s10796-021-10116-w. URL http://dx.doi.org/10.1007/s10796-021-10116-w.

Ricciato, Fabio et al. Trusted smart statistics: Motivations and principles. 2020. URL https://cros-legacy.ec.europa.eu/content/trusted-smart-statistics-motivations-and-principles_en.

Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: Applied evaluations and enhancements, 11 2020. URL https://arxiv.org/abs/2011.05537.

Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning, 11 2018. URL https://arxiv.org/abs/1811.04017.

Pegah Salehi, Abdolah Chalechale, and Maryam Taghizadeh. Generative adversarial networks (gans): An overview of theoretical model, evaluation metrics, and recent developments. *arXiv:2005.13178*, 05 2020. URL https://arxiv.org/abs/2005.13178.

Siani Shen, Yun; Pearson. Privacy enhancing technologies: A review. 2011. URL https://www.hpl.hp.com/techreports/2011/HPL-2011-113.html.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, 2015. doi: 10.1145/2810103.2813687. URL https://www.cs.cornell.edu/~shmat/shmat_ccs15.pdf.

Reza Shokri, Cornell Tech, Marco Stronati, and Vitaly Shmatikov. Membership inference attacks against machine learning models. URL https://arxiv.org/pdf/1610.05820.

Joshua Snoke, Gillian M Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society*, 181, 03 2018.

Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models, 05 2019. URL https://arxiv.org/abs/1811.00513.

L. Sweeney. k-anonymity: A model for protecting privacy, 2002. URL https://epic.org/wp-content/uploads/privacy/reidentification/Sweeney_Article.pdf.

Tatsuya Takemura, Naoto Yanai, and Toru Fujiwara. Model extraction attacks against recurrent neural networks, 01 2020. URL https://arxiv.org/abs/2002.00123.

Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms, 02 2022. URL https://arxiv.org/abs/2112.09238.

Amirsina Torfi, Edward A. Fox, and Chandan K. Reddy. Differentially private synthetic medical data generation using convolutional gans. *Information Sciences*, 586:485–500, 03 2022. doi: 10.1016/j.ins.2021.12.018. URL https://people.cs.vt.edu/~reddy/papers/IS22.pdf.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation, 01 2020. URL https://arxiv.org/abs/2001.09700.

United Nations. *UN Handbook on Privacy-Preserving Computation Techniques*. United Nations Committee of Experts on Big Data and Data Science for Official Statistics, New York, 2019. URL https://unstats.un.org/bigdata/task-teams/privacy/UN%20Handbook%20for%20Privacy-Preserving%20Techniques.pdf.

United Nations. *United Nations Guide on Privacy-Enhancing Technologies for Official Statistics*. United Nations Committee of Experts on Big Data and Data Science for Official Statistics, New York, 2023. URL https://unstats.un.org/bigdata/task-teams/privacy/guide/index.cshtml.

Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. This person (probably) exists. identity membership attacks against gan generated faces, 2021a. URL https://doi.org/10.48550/arXiv.2107.06018.

Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. This person (probably) exists. identity membership attacks against gan generated faces, 07 2021b. URL https://arxiv.org/abs/2107.06018.

Matthew Wilchek and Yingjie Wang. Synthetic differential privacy data generation for revealing bias modelling risks. *2021 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 09 2021. doi: 10.1109/ispa-bdcloud-socialcom-sustaincom52081.2021.00211. URL https://www.cloud-conf.net/ispa2021/proc/pdfs/ISPA-BDCloud-SocialCom-SustainCom2021-3mkuIWCJVSdKJpBYM7KEKW/264600b574/264600b574.pdf.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv:1802.06739 [cs, stat]*, 02 2018. URL https://arxiv.org/abs/1802.06739.

Lei Xu, Maria Skoularidou, Alfredo Cuesta Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan, 2019. URL https://arxiv.org/abs/1907.00503.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting, 05 2018. URL https://arxiv.org/abs/1709.01604.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. *arXiv:2110.06500 [cs, stat]*, 07 2022. URL https://arxiv.org/abs/2110.06500.

Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning, 05 2019. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8835283.

Xiang Yue, Huseyin A. Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. Synthetic text generation with differential privacy: A simple and practical recipe, 05 2023. URL https://arxiv.org/abs/2210.14348.

Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model (technical report), 03 2018. URL https://arxiv.org/abs/1801.01594.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. *arXiv:1911.07135 [cs, stat]*, 04 2020. URL https://arxiv.org/abs/1911.07135.

Jingwen Zhao, Yunfang Chen, and Wei Zhang. Differential privacy preservation in deep learning: Challenges, opportunities and solutions. *IEEE Access*, 7:48901–48911, 2019. doi: 10.1109/access.2019.2909559.

Zilong Zhao, Aditya Kunar, Van Der Scheer Hiek Zhao, Tudelft, Robert Birke, and Lydia Chen. Ctab-gan: Effective table data synthesizing. URL http://www.datascienceassn.org/sites/default/files/CTAB-GAN%20Effective%20Table%20Data%20Synthesizing.pdf.