Improving medical data synthesis with DP-GAN and Deep Anomaly Detection

by

Vojtech, Crha

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday June 11, 2024 at 10:00 AM.

Student number:	5115310	
Project duration:	November 14, 2023 – July 11, 2024	
Thesis committee:	Dr. R. Hai,	TU Delft
	Dr. Z. Erkin,	TU Delft, Thesis Advisor
	MSc. T. Li,	TU Delft, Daily co-supervisor

TU Delft

Preface

Cybersecurity was a field of great interest to me long before I started my studies. After three years of Computer Science & Engineering bachelor, I finally got my opportunity to study it full-time. Since the first lectures of Security and Cryptography I was astonished by the possibilities as well as new ways to think about information and the world. My studies have introduced me to various fields of Cybersecurity, ranging from very low-level mechanisms to high-level theoretical concepts. One of these fields - privacy - has become my sole focus over the last few months.

The journey towards writing my thesis in the last nine months has not been an easy one, however I am glad to have taken it. First and foremost, I would like to thank Professor Zekeriya Erkin for both supporting and supervising me. Secondly, I greatly appreciate the guidance of Tianyu Li in all kinds of matters as well as for his patience and understanding.

I am also grateful to Florinne Dekker, Jelle Vos and Jorrit van Assen for their feedback during the meetings and great ideas for improving my work. Next, I want to thank Chelsea, Davis, Juno, and Prakhar for their support and for generally making the thesis writing process much more pleasant.

I also want to thank Professor Rihan Hai for being part of my thesis committee and for taking the time to read my thesis and my defense.

Lastly, I thank my friends and family for their support, both before and during the writing of this thesis.

Vojtech, Crha
Delft, July 2024

Summary

All modern medical institutions gather data. The use of this data is crucial for their efficient operation and could lead to significant efficiency gains if shared between institutions. However, this sharing poses a substantial risk to privacy, which is critical in the medical domain. To minimize this risk, Differentially Private Generative Adversarial Networks (DP-GANs) are employed to synthesize private data, which can then be shared with significantly reduced risk.

Generating synthetic medical data is a challenging task. The original private data often contains numerous patterns, correlations, and anomalies. Reproducing these patterns and correlations in synthetic data is difficult, and the presence of anomalies can make the training process less efficient. However, by using these patterns to privately train an anomaly detection model, we can reduce the number of anomalies and improve the subsequent quality of the data. Therefore, our research question is: ”How can we improve the utility of synthetic medical data by employing anomaly detection?”.

After researching state-of-the-art approaches in synthetic medical data release, we designed our framework: Generative Adversarial Network Anomaly Detection (GANAD). GANAD involves private training and subsequent usage of an anomaly detection model at three key points in the DP-GAN data synthesis pipeline. GANAD serves as an extension to DP-GANs, further enhancing the utility of synthetic data.

To evaluate the performance of our framework on real datasets and with real DP-GANs, we allocated 10% of the privacy budget (ϵ) to privately train an autoencoder. The autoencoder was utilized at three stages: before the DP-GAN training (Pre-Generation), during the training (Mid-Generation), and after the training (Post-Generation).

These three approaches were tested across three datasets with different privacy budgets. The quality of the produced data was measured using AUROC and AUPRC metrics, averaged across eleven machine learning models. Each test was repeated ten times to account for randomness. Our results indicate that GANAD can lead to a 3-5% increase in utility when applied carefully. Our highest improvement exceeded 10%, indicating promising directions for further research.

Abstract

Ensuring the privacy of medical data in a meaningful manner is a complex task. This domain presents a plethora of unique challenges: high stakes, vast differences between possible use cases, long-established methods that limit the number of feasible solutions, and more. Consequently, an effective approach to ensuring the privacy of medical data must be easy to adopt, offer robust privacy guarantees, and minimize the reduction in data utility.

The unique nature of medical data presents distinct challenges and also opportunities. We must consider various types of correlations that significantly impact privacy guarantees. However, these correlations also provide opportunities to enhance the utility of synthetic data by removing specific anomalies

This thesis proposes a framework compatible with state-of-the-art approaches for differentially private dataset release based on the usage of Generative Adversarial Networks (GANs). Our framework uses a part of the privacy budget to train an unsupervised learning model to detect and remove anomalies. We evaluate the performance of the framework using a variety of machine-learning models and metrics. The final results show an improvement of up 13% compared to approaches not using our framework, under the same privacy budget.

Contents

1 Introduction

According to Statista, the volume of data generated annually has increased more than tenfold over the past decade [32], and this growing trend is expected to continue. Seagate estimates that around 30% of newly generated data is medical in nature [15]. Additionally, there has been a significant reduction in the cost of data storage, with some estimates indicating a thirtyfold decrease over the same period [39]. Despite the substantial growth in data generation and the decreasing cost of storage, a large portion of generated medical data remains underutilized. Estimates suggest that up to 97% of medical data is not used at all [20] although more conservative estimates suggest this figure is closer to 43% [3]. This underutilization is attributed to various factors, with some healthcare leaders highlighting competing financial priorities as a major concern [3]. A critical factor affecting the usage of medical data is privacy. The implementation of stringent guidelines under the General Data Protection Regulation (GDPR) has created significant compliance requirements for data holders [38]. As a result, privacy considerations will be the primary focus of this thesis.

Additionally, improper sharing or usage of data can lead to breaches of patient trust, significant fines, and even patient discrimination in some cases [42]. Consequently, concerns about the social impact of data privacy are a significant factor limiting the utilization of medical data. One potential solution to this issue is the use of synthetic medical data.

1.1. Synthetic medical data

Synthetic medical data is artificially generated, typically by a machine learning model, to mimic real medical data while protecting patient privacy. Ideally, synthetic data retains the utility of real data without disclosing any private individual information. In more realistic terms, it is challenging to completely preserve the utility of the original data while not revealing any information about any individual.

Therefore, it is crucial to define privacy precisely and quantify the level of data protection it offers. The following Section will focus on this aspect. Subsequently, we will analyze the practical aspect of generating synthetic data.

1.1.1. Data synthesis

The theoretical concept behind data synthesis is straightforward. It involves analyzing authentic data to learn its patterns and trends, which are then used to generate synthetic data. In practice, this is most commonly achieved through deep learning techniques. For private data synthesis, the predominant method is the use of Generative Adversarial Networks (GANs). GANs consist of two neural networks trained simultaneously: the discriminator and the generator. The discriminator's role is to distinguish between authentic and synthetic data, while the generator aims to produce data that can deceive the discriminator. Through this adversarial process, the generator eventually learns to produce high-fidelity synthetic data. Further details on evaluating GAN performance, the utility of generated data, and the inner workings of GANs are discussed in Chapters 4 and 2.

1.1.2. Quantifiable privacy

Firstly, consider one of the oldest and most stringent definitions of privacy, formulated by Dalenius in 1977 [14]: No information about an individual that cannot be gained without access to the data should be leaked. However, this notion is impractically strict, as demonstrated by Dwork [17]. In the same paper [17], Dwork proposed a new standard for protecting the privacy of individuals while still preserving meaningful utility - Differential Privacy. Differential Privacy quantifies the increase in the probability that an individual is present in the dataset using the parameter ϵ. This concept has been foundational, inspiring a variety of new approaches to ensure data privacy.

1.1.3. Private data synthesis

In the previous sections, we gave a basic introduction to quantifying privacy as well as data generation. The next logical step is to combine these two elements to synthesize data in a privacy-preserving manner. This is typically achieved by training the model with privacy-preserving techniques, ensuring that any output generated by the model remains private. At this point, a trade-off between utility and privacy becomes evident. Higher privacy requirements reduce the model's ability to learn effectively from the data, resulting in synthetic data with lower utility. For complex classes, such as medical data, the utility of synthetic data with meaningful privacy guarantees can often be insufficient for actual applications of this approach. The following Section gives a high-level overview of how existing implementations address this challenge.

1.2. Privacy-utility tradeoff

The solution to insufficient accuracy is straightforward: to achieve a level of data utility viable for real-world applications, we must either improve the model's performance or relax the privacy constraints. Typically, state-of-the-art solutions employ a combination of both strategies. To understand how these options are balanced, we first need to explore them in more detail.

1.2.1. Approaches for privacy preservation

Before training any model, it is essential to select a precise definition of privacy. Although differential privacy and its various formulations are currently the most widely adopted standards for privacy-preserving data releases, numerous other approaches exist. In this section, we briefly introduce several methods for preserving privacy as well as our final choice - Differential Privacy. We substantiate this choice in detail in Chapter 2.

k-Anonymity

k-Anonymity is achieved when each record is indistinguishable from at least k-1 other records given a group of quasi-identifying attributes.

ℓ-Diversity

For a dataset to be ℓ-Diverse it requires that each group of indistinguishable records contains at least ℓ different values for a given sensitive attribute. As such, it serves as an extension of

k-Anonymity.

t-Closeness

t-Closeness is defined as the maximum distance t between the distribution of a sensitive attribute in a given class and the distribution of said attribute in the whole dataset.

Differential privacy

Lastly, differential privacy is perhaps the most common approach to ensuring the privacy of data. As differential privacy is a vital part of this thesis, we explore the intricacies regarding different definitions, mechanisms, and applications in Section 2.2. Here, a short informal definition is given: A dataset is ϵ differentially private if the presence of an entry is indistinguishable to a certain level based on the ϵ value. This is often achieved by adding noise to the original data from a distribution - often Laplacian or Exponential is used. Differential privacy is an umbrella term for a variety of different approaches however it is chosen the flexibility that is paramount while designing a flexible data synthesis system.

1.2.2. Data generation models

While there is less variety in data generation architecture selection compared to the number of privacy-preservation techniques, there are numerous choices to make regarding optimizers, the number of layers, activation functions, and other parameters. Given a specific goal, such as generating a particular type of synthetic data, we can select the optimal values for these parameters through a process called hyperparameter optimization. Theoretically, any parameter can be optimized. However, in practice, certain parameters, such as neural network depth, neuron type, and optimizer class, often remain constant. To compare our results with existing approaches, we will apply hyperparameter optimization similarly. This process is discussed in more detail in Chapter 5.

1.3. Pattern Guided Anomaly Removal

State-of-the-art approaches utilize highly optimized architectures based on the latest GAN research, along with carefully selected definitions of differential privacy. While these methods can achieve relatively high performance, there is still a clear need for even better-performing data synthesis models. Improving the privacy-utility trade-off could lead to higher adoption of synthetic medical data and enhanced privacy preservation. One aspect of note is that neural networks are often designed for specific data patterns.

Given that medical data is typically highly correlated, intricate networks are required to accurately capture its details. The training of these networks is however negatively affected by another phenomenon present in the data - anomalies. Therefore, the detection and removal of these anomalies, along with an analysis of the associated costs and utility gains, constitute the primary improvements proposed in this thesis. In this Section, we briefly introduce the various patterns that are present in the data, with usage of which we can remove anomalies and subsequently improve the quality of synthetic data.

1.3.1. Correlation between columns

The idea that various measurements of the same patient taken within a short time span are related is intuitive. We can model this as several variables measuring an underlying latent variable—the person's actual physical status. This status could be influenced by activities such as exercising, sleeping, or experiencing a medical emergency. For instance, respiratory and heart rates are usually correlated; both decrease during sleep and increase during exercise. It is important to note that if we add artificial noise to the data, an unauthorized third party with medical expertise may still be able to make accurate predictions about the original data. This property and its privacy implications are discussed in depth in Section 2.2 on differential privacy.

1.3.2. Correlation between rows

While related to the previous type of correlation, correlation across rows is much less immediately obvious. This pattern also indicates an underlying latent variable, although it is not as clear-cut. Practically, we can imagine it as one person influencing the likelihood of another person being in the same dataset. Influencing factors could include geographic location, economic status, or social connections. For instance, consider a family with a rare hereditary condition visiting the same hospital. The presence of one family member in the dataset could significantly increase the probability of other family members being included. This type of correlation is less studied but equally important. We discuss how this affects our privacy guarantee in Section 2.2.

1.3.3. Time series

Another attribute of the data is the presence of time series. Mathematically, a time series is a sequence of data points ordered by time. While this definition does not necessarily imply a time-adjacent correlation, such correlations are almost always present in the medical domain. In practical terms, this means that each data point in a medical time series is likely influenced or partially determined by its predecessors.

Depending on the representation, this can be viewed as either a subset of cross-column or cross-row correlation. In practice, including successive values as columns is more common. For example, consider heart rate measurements taken at five-second intervals. If an initial heart rate measurement is 90 bpm, it is statistically more likely that the next measurement will be close to 95 bpm rather than 180 bpm. The exact correlation depends on the length of the time interval and the nature of the value being recorded. Generally, repeated measurements on the same patient are related, and the same privacy concerns previously mentioned apply. As a practical example, we calculated the Pearson correlation coefficient for the heartbeat data in the Fitbit smartwatch dataset [25] to be 99.4%. This is a perfect example of a simple time series, so we do not expect such high values in more complex data. However, it illustrates that strong correlations can be present in medical data.

1.3.4. Class Imbalance

Another crucial aspect to discuss, though not unique to medical data, is class imbalance. This issue is particularly relevant in medical anomaly detection. For instance, one of the datasets we analyze focuses on diagnosing epilepsy based on several medical readings. In this binary classification task, the vast majority of the data does not indicate epilepsy. Therefore, it is essential to take special care in developing and evaluating our approach to ensure that the class imbalance in the original data does not distort our results.

1.4. Research Question

After all the considerations explained in the previous sections, we have formulated our research question as follows: ”How can we improve the utility of synthetic medical data by employing anomaly detection?”

1.5. Our Contribution

We introduce a framework that enhances the utility of data generated by differentially private generative adversarial networks without increasing the privacy budget. Our framework is compared to baseline and state-of-the-art approaches by evaluating the quality of several machine learning models. This quality is measured using two common machine learning metrics AUROC and AUPRC, explained in detail in Chapter 5. Finally, we demonstrate the epsilon cost of our method, comparing it to the utility gain achieved by simply increasing the original privacy budget.

1.6. Overview

In Chapter 2 we introduce the reader to the concepts necessary for the understanding of our work. In Chapter 3, we discuss the existing works that explore our topic or topics adjacent to ours. In the fourth Chapter - 4, we introduce our approach in detail as well as calculate the privacy cost. Chapter 5 explains our testing methodology and evaluates the improvement of our approach against existing state-of-the-art approaches. Lastly, in Chapter 6, we discuss the limitations and merits of our approach as well as consider possible further improvements.

2

Preliminaries

In this Chapter, we first introduce several attack models useful for establishing the limitations of existing privacy preservation approaches. This serves to substantiate the choice of Differential Privacy as present in Chapter 1. Subsequently, we explore intricacies behind Differential Privacy as well as GANs and Autoencoders, the Anomaly Detection method of choice.

2.1. Attack Models

First, we define four attack models and provide a short practical example of each.

Record Linkage

In a record linkage attack, a specific value (qid) on the Quasi-Identifier (QID) in the released table T identifies a small group of records. If the victim's QID matches this value (qid), the victim's information can potentially be linked to this small group of records. The attacker then has a limited set of records to consider for the victim's information. With some additional knowledge, the attacker might be able to uniquely identify the victim's record within this group.

As an example, we can imagine an exchange of information between a hospital and a census office. The hospital provides Table A with information about Job, Marital Status, Sex and disease diagnosis. The census office also has access to Table B and knows that each person in Table A also has an entry in Table B. Joining these tables on both the Job, Sex and Marital Status, supposedly public information, the census office might be able to deanonymize the entries of few select individuals and link name and diagnosis.

Attribute Linkage

In an attribute linkage attack, the attacker might not be able to pinpoint the exact record of the target victim, but can infer the victim's sensitive information from the published data (T). This is done by analyzing the sensitive values associated with the group that the victim belongs to. If certain sensitive values are common within a group, it becomes easier for the attacker to make accurate inferences, even if k-anonymity is in place.

Assuming the same Table A as in the previous examples and a random attacker who is curious to know the diagnosis of a person he knows the Job and Sex of. Depending on the exact size of the dataset, there is a possibility that all entries with the same Job and Sex combination share the same diagnosis, leaking sensitive information.

Table Linkage

Both record linkage and attribute linkage assume that the attacker already knows the victim's record is in the released table T. However, sometimes just the presence or absence of the victim's record in T can reveal sensitive information. For instance, if a hospital releases a data table containing records of patients with a specific disease, simply knowing whether the victim's record is in that table can be harmful. Table linkage occurs when an attacker can confidently determine whether the victim's record is included in the released table.

Assume we have a Table C containing ten entries, five of which are both Female and Singers. An attacker also has access to public Table D where C $\subseteq$ D. If Table D contains in total six entries of Female Singers, the attacker can say with $\frac{5}{6}$ certainty that any Female Singer from D is present in C.

Probabilistic Attacks

Another type of attack focuses not on linking records, attributes, or tables directly to a person but instead on modifying the attackers' probabilistic belief on the sensitive information of a victim, purely by access to public data.

Comparison of techniques

We have given a simplified notion of the various privacy-preservation techniques as well as the attack models. Now, a more comprehensive comparison is given to justify the prevalence of Differential Privacy as the widespread method for private data release.

Table 2.1: Vulnerability to various attack models - Privacy Models. Adapted from Benjamin C. M. Fung et al.[24]

Privacy Techniques	Attack Model			
	Record Linkage	Attribute linkage	Table Linkage	Probabilistic Attack
k-Anonymity	✓			
ℓ-Diversity	✓			
t-Closeness		✓		
ϵ-Differential Privacy			✓	✓

2.2. Differential privacy

Differential Privacy [17] is a framework that allows for a quantification of privacy using a simple formula. In recent years, Differential Privacy has become a standard for state-of-the-art anonymization and subsequent private data release. This can attributed to the flexibility, wide academic support and great ability to conduct statistical analysis to determine the exact cost of releasing data in various practical terms. In this section, we will explore the common definitions of Differential Privacy, the interactive vs the noninteractive approach, composition property and various common mechanisms.

Definitions

Since Differential Privacy was first proposed by Dwork in 2006 [17], a number of relaxations to the quite restrictive original definition have been made. These new definitions seek to better measure the actual risk. In theory, a more accurate measurement of said risk allows us to set the privacy parameter lower, subsequently increasing the utility of the released data. In practice, there is no clear best definition and all definitions are used. As such, we introduce the reader to a few most common ones that are relevant to this thesis. For an even more in-depth exploration, we recommend the excellent work by Jayaraman and Evans [33]

Epsilon Differential Privacy

The most common definition of Differential Privacy is the one originally proposed by Dwork in [17]. This formula uses a single parameter ϵ to quantify the risk of a potential leakage.

Definition 1 *(ϵ-Differential Privacy) Mechanism $\mathcal{M}$ is ϵ-differentially private for all neighboring datasets D and D' and S $\subseteq$ Range($\mathcal{M}$) differing by only one record given the privacy budget ϵ*

$$Pr(\mathcal{M}(D) \in S) \leq Pr(\mathcal{M}(D') \in S) * e^{\epsilon}. \tag{2.1}$$

Epsilon-Delta Differential Privacy

Another common definition is a slight relaxation of the previously defined ϵ-Differential Privacy. This relaxation introduces a variable δ which signifies the probability of information being leaked.

Definition 2 *(ϵ,δ-Differential Privacy) A mechanism $\mathcal{M}$ is ϵ,δ-Differentially Private for all neighboring datasets D and D', differing by only one record given the privacy budget ϵ and the failure probability δ*

$$Pr(M(D) \in S) \leq Pr(M(D') \in S) * e^{\epsilon} + \delta. \tag{2.2}$$

The value of δ depends greatly on the specifics of the dataset but commonly, a value below $\frac{1}{datasetSize}$ is picked.

Rényi Differential Privacy

Rényi Differential Privacy [41] is another relaxation that uses order α. This definition is based on the concept of Rényi divergence.

Definition 3 *(Rényi Divergence) For two probability distributions P and Q defined over R, the Rényi divergence of order $\alpha > 1$ is*

$$D_{\alpha}(P||Q) \triangleq \frac{1}{\alpha - 1} \ln E_{x \sim Q} \left(\frac{P(x)}{Q(x)} \right)^{\alpha}. \tag{2.3}$$

Given the definition of Rényi divergence, we can now give a precise definition of Rényi Differential Privacy:

Definition 4 *(Rényi Differential Privacy) A mechanism $\mathcal{M}$ is ϵ-Rényi Differentially Private in order of α, if the following formula holds for any two neighboring dataset D and D':*

$$D_{\alpha}(\mathcal{M}(D)||\mathcal{M}(D')) \leq \epsilon. \tag{2.4}$$

Composition property

While the intuitive definition of differential privacy as ”Adding noise from Laplacian or Exponential distribution to make the original data more private” is easy to understand, several questions arise when considering the application of differential privacy to correlated data or even repeated application of it. Composition property is also one of the most risky elements when implementing differential privacy as there is a plethora of pitfalls that can lead to underestimation of the final privacy budget which in turn gives a false sense of security. An example from practice could be the sensitivity doubling when using microbatching [48] or floating-point implementations being subject to several vulnerabilities [8].

Definition 5 *(Composition Property) For both repeated queries or releases of the same dataset D, the resulting privacy budget ϵ_{total} can be defined as*

$$\epsilon_{final} = \epsilon_1 + \epsilon_2 \tag{2.5}$$

where ϵ_1 and ϵ_2 are the privacy budgets of the queries/releases.

It is also worthwhile to consider the epsilon budget when applying differential privacy to individual columns. As an example, let's imagine a simple dataset consisting of two columns of continuous correlated data. If we apply the Laplacian mechanism with ϵ-Differential Privacy, a naive computation will underestimate the amount of information leakage. This is the product of the correlation between the columns. The higher bound of the actual value is simply 2ϵ, as the added value of ϵ for individual columns.

Sensitivity

Sensitivity is another crucial aspect of Differential Privacy. Sensitivity determines how much the presence of a single value can modify the result of a query on the dataset. A more formal definition is given:

Definition 6 *(Sensitivity) For two neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$, the sensitivity of a mechanism $\mathcal{M}$ can be defined as:*

$$\triangle \mathcal{M} = \max_{\mathcal{D},\mathcal{D}',||\mathcal{D}-\mathcal{D}'||_1=1} ||\mathcal{M}(\mathcal{D}) - \mathcal{M}(\mathcal{D}')||. \tag{2.6}$$

Laplace Mechanism

The Laplace Mechanism [18] is one of the two most common techniques for achieving differential privacy. Laplace Mechanism works by adding Laplacian noise to an output of a given function. As such, it is ideal for adding noise to non-discrete values. First, we need to define the scale parameter b.

Definition 7 *(Laplace Mechanism Scale parameter) Laplacian scale parameter determines the rate at which the probability density function decays away from the mean*

$$Lap(x|b) = \frac{1}{2b} e^{-\frac{|x|}{b}}. \tag{2.7}$$

Definition 8 *(Laplace Mechanism) Given a dataset D, domain of the dataset $\mathcal{D}$, dimension of the dataset d and a function $f : \mathcal{D} \to \mathbb{R}^d$, the Laplace Mechanism $\mathcal{A}$ can be defined as:*

$$\mathcal{A} = f(D) + Lap(O|b)^d. \tag{2.8}$$

Exponential Mechanism

The Exponential Mechanism [40] is the second widely used approach for ensuring differential privacy of data. This mechanism is usually used for discrete data, a case where the Laplace mechanism is usually not applicable. A commonly given example is that of product pricing. Assume we sell products of type A and have several buyers willing to purchase exactly one copy for a maximum price of 1$, 1$, and 3.01$ respectively. The maximum gain is when setting the price at 3.01$ and the second maximum is at either 1$ or at 3$. However adding a bit of random noise to these values, setting the price as 3.02 or 1.01 significantly reduces the gain. As such, a mechanism rather needs to evaluate the "goodness" of the value and replace it with a value with similar "goodness".

Definition 9 *(Exponential Mechanism) Assume a dataset $\mathcal{D} \in \mathcal{D}^n$, a set of objects $\mathcal{R}$ and a scoring function $f : (\mathcal{D}^n \times \mathcal{R}) \to \mathbb{R}$. An Exponential Mechanism M_E with inputs $\mathcal{D}$, $\mathcal{R}$ and f selects an output $r \in \mathcal{R}$ with probability $exp\left(\frac{\epsilon f(\mathcal{D},r)}{2\triangle}\right)$.*

Interactivness

The mechanisms in the previous Section can generally be divided into two categories - interactive and non-interactive. This division defines the behavior of the mechanisms concerning the usage of the data by a data analyst. This Section will explore the two approaches in detail.

Interactive Differential Privacy

Interactive differentially private mechanism incorporates an alternative approach to the usage of the data. Such a mechanism does not allow direct access to the data but instead manages interaction by a offering only predefined set of differentially private queries. As an example of a differentially private query, we can imagine calculating the average of a column. The average will be calculated on the unmodified (non-private) data and then adding amount of noise depending on the privacy parameter and the value of the data. The Differential Privacy library by Google [29] provides implementations for the most common queries with the implementation details found in [37].

The benefit of an interactive approach to Differential Privacy is a higher utility with regard to data leakage. In other words, if the data analyst is interested only in specific values (i.e. the average), interactive mechanisms can provide higher utility, as opposed to calculating the value post-release (from a dataset processed by a non-interactive differentially private mechanism). Assuming a more complex system that allows the user to select among several types of queries as well as define the budget of each one, a user can potentially extract significantly more utility from the data suited to their specific needs given the same privacy budget. The drawback of this approach is the inherent assumptions made about the usage of the data and the limitations with regards to the usage of other tools. This can be thought of as a trade-off between the flexibility and utility of the data. It is therefore nearly impossible to create an interactive system that will permit the use case of each potential user. For these reasons, we will be discussing only non-interactive approaches as they best fit with the existing workflows as well as the varied nature of the data.

Non-Interactive Differential Privacy

Non-Interactive mechanisms apply a more static approach to ensuring the privacy of the data. Compared to Interactive approaches, the noise is added to the data itself which is subsequently released. From a theoretical perspective, the user loses control over which elements of the data he wishes to learn more information. This however allows greater flexibility with regards to the application of the data, integration of existing tools and generally allowing the data to be seen in ways not yet developed at the time of release.

2.3. Autoencoders

In simple terms, anomaly detection is a process of identifying anomalies. For this thesis, we are only considering unsupervised anomaly detection within datasets. Unsupervised means that we assume no labeled data. In more abstract terms, we know nothing of the anomalies beforehand and our detection works purely by finding irregular patterns in the data. In this section, we briefly introduce the reader to the anomaly detection method used in this thesis - Autoencoders. Autoencoders are currently widely used in the state-of-the-art DP-GAN approaches and are also therefore our method of choice.

Definition 10 *(Neural Networks) Neural Networks (NNs) are a class of machine learning models inspired by the structure and function of biological neural systems. A neural network consists of interconnected layers of nodes, or neurons, where each node processes input signals through weighted connections and activation functions to produce an output.*

The fundamental unit of a neural network is the perceptron, which computes a weighted sum of input features, applies a bias, and passes the result through an activation function. This *can be mathematically expressed as:*

$$y = \phi\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.9}$$

where y is the output, x_i represents the input features, w_i denotes the weights, b is the bias, and ϕ is the activation function.

Multilayer neural networks, also known as multilayer perceptrons (MLPs), extend the perceptron by incorporating multiple layers of neurons, typically including an input layer, one or more hidden layers, and an output layer. Each layer performs a nonlinear transformation of its inputs, enabling the network to model complex, non-linear relationships in data. This makes neural networks suitable for a wide range of tasks such as classification, regression, and pattern recognition.

Autoencoders are a type of neural network, used primarily for unsupervised learning. The goal of an autoencoder is to learn efficient representation/encoding of the data. An autoencoder is composed of two components: an encoder and a decoder. The encoder first compresses the input into a smaller dimension representation. The encoder then tries to reconstruct the original data from this smaller dimension.

Architecture

While there are many possible variations of autoencoders, the vast majority of implementations are based on these few simple definitions:

Definition 11 *(Encoder) Encoders transform the input $\mathbf{x}$ into a compressed representation $\mathbf{z}$. The encoder function can be modeled as:*

$$\mathbf{z} = f_{enc}(\mathbf{x}). \tag{2.10}$$

Definition 12 *(Decoder) Decoders reconstruct the original $\mathbf{x}'$ from the compressed representation (z). The decoder function can be expressed as:*

$$\mathbf{x}' = f_{dec}(\mathbf{z}). \tag{2.11}$$

Definition 13 *(Objective Function) The primary goal of an autoencoder is to minimize the reconstruction error between the input data $\mathbf{x}$ and the reconstructed data $\mathbf{x}$'. The mean squared error (MSE) is often the metric used for measuring the reconstruction loss:*

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = |\mathbf{x} - \mathbf{x}'|^2. \tag{2.12}$$

Reconstruction error minimalization

The autoencoder training process tries to minimize reconstruction errors. As such, it is inherently adept at anomaly detection. Intuitively, the most anomalous entries will be the ones the model is worst at reconstructing. Therefore, we can simply get N entries with the highest reconstruction error.

Data Quality Improvement

Additionally, autoencoders can also be used to improve the performance of GANs. For example, in [52] data is first encoded, then the encoded data is the basis for the training of the GAN and lastly, the output of the GAN is again decoded. This is done to help ”better capture correlations between neighboring features, represent a new compact feature space, transform possible discrete records to a new continuous space and simultaneously model discrete and continuous phenomena” [52].

2.4. Generative Adversarial networks

Generative Adversarial Networks (GANs) [28] are a class of deep learning models designed for generating synthetic data. GANs commonly consist of two main components: A generator and a discriminate trained in an adversarial manner.

Generator

The generator is a neural network that takes random noise as input and generates samples as close to the target real data distribution as possible. The generator aims to learn a mapping from (random) noise space to the real data space, capturing the latent structure of the real data. During training, the performance is measured as the ability to fool the discriminator by creating synthetic data indistinguishable from real data.

Discriminator

The discriminator is also a neural network that is instead focused on binary classification of real and synthetic data. The discriminator takes a sample of data as an input and outputs the probability of the data being synthetic or genuine. The performance of the discriminator is measured by its ability to correctly classify samples from both the synthetic and authentic datasets.

Adversial Training

The training process of GANs involves a minimax game between the generator and the discriminator. The generator is trying to minimize the difference between the distribution of the real and the synthetic data, while the discriminator is trying to maximize its ability to distinguish between real and synthetic data. The training of these two networks is done in an alternating fashion, with the generator being able to generate continuously more realistic samples and the discriminator becoming better at identifying fake samples.

Loss Functions

As for any neural network, the selection of a loss function plays a crucial role in the training process. In the original paper proposing GANs back in 2014 [28], the binary cross-entropy loss was used for both the discriminator and the generator. Since then, a variety of modifications and improvements have been proposed. These improvements, such as Wasserstein distance loss functions or architecture changes will be discussed in detail in Chapter 3.

3
Related Works

While we recognize the need for high-utility methods for ensuring the privacy of medical as well as the importance of anomalies in this data, we are not the first to delve into these topics. This Chapter first introduces the various advancements in the area of Differentially Private Generative Adversarial Networks (DP-GANs) and compares them. This comparison is subsequently used to select three candidates for the evaluation of our framework. Additionally, we also introduce several techniques towards efficient anomaly removal using Autoencoders (AE).

3.1. DP-GANs

In this Section we introduce the reader to latest advances in the field of Differentially Private Generative Adversarial Networks. We place special focus on solutions focused on the generation of data for the medical domain.

3.1.1. DP-GAN

The major widely adopted approach to training Generative Adversarial Networks privately was originally proposed by Xie et al. [57] in 2018. The paper achieves this by adding noise to gradients, clipping the gradients, and also by the usage of Wasserstein distance as a means of measuring the distance between probability distributions. Their approach brings several key advances. Firstly, the size of the training data does not influence the privacy parameter in any way which has been an issue for one of the previously proposed approaches [45]. Secondly, they provide rigorous proof of privacy which makes the privacy of the training data independent from the generator (although the discriminator and computation of the generator are still private). They also explore the relation between the privacy budget ϵ, the number of iterations, and the quality of the data. While their selection of the metric for quality of data, Wasserstein Distance, is not as advanced as the metrics used in later papers, they still provide valuable insight for hyperparameter tuning. As such, DP-GAN is now widely used as the basis for the implementation of more advanced approaches introduced further in this Section. It is however worth noting that that this GAN is able to handle only labelled data, a shortcoming that is addressed by some of the more advanced later approaches.

3.1.2. Medical DP-GANs

In this Section, we introduce the reader to the latest advancements in the specific field of medical data focused DP-GANs. These advancements will be explained in detail as they are the most relevant for our framework.

PATE-GAN

PATE-GAN [34] improves upon the original GAN architecture [28] by utilizing the PATE framework. Private Aggregation of Teacher Ensembles, first proposed in [45] and improved in [44] divides a dataset into n disjoint subsets $\mathbf{D}_1, .., \mathbf{D}_n$. Then n Teachers (Classifiers) $\mathbf{T}_1, ..\mathbf{T}_n$, are trained separately on the n disjoint subsets. After the Teachers are trained, a sample to be evaluated is then passed to each of the Teachers. The results of the Teachers are then aggregated in a private/noisy manner. First, given a set of possible classes c, we calculate how many Teachers gave each class as the result. Lastly, noise from the Laplace distribution is added to each of these counts and afterward, the maximum is selected. The output of a single query to the PATE mechanism is $(\frac{1}{\lambda})$-differentially private where $Lap(\lambda)$ corresponds to c i.i.d. noise variables [34]. The key issue for using Teachers, however, is the private nature of their parameters and the increased cost for repeated queries. As such, only the usage of Teachers is not sufficient for implementing a DP-GAN. Therefore an extension originally proposed in [45] is used. This extension adds a concept of Students, classifiers trained on public (unlabelled) data, first labeled by the private PATE mechanism. Compared to Teachers, no parameters of Students are public and they can also be queried repeatedly without any additional privacy cost.

In PATE-GAN, the combination of Teachers and Students replaces the discriminator. In the standard GAN model, the generator is trying to minimize its loss with respect to the discriminator and the discriminator is trying to minimize it with respect to the generator. However, in PATE-GAN, Teachers are trying to minimize their loss with respect to the generator, the generator is trying to minimize it with respect to the Students and the Students are minimizing loss with respect to the Teachers.

Lastly, it is important to mention that the authors of PATE-GAN have slightly modified the training process for Students. As we cannot depend on any similar data being publicly available, Xavier initialization [26] is used to initialize the Neural Networks on the Students. Afterward, the Students are trained on data generated by the Discriminator (labeled by the Teachers). In each iteration of training of the GAN, first, the Teachers and then the Students are updated.

RDP-CGAN

Rényi Differential Privacy - Convolutional Generative Adversarial Network [57] proposes several changes upon the original GAN model [28]. Firstly, a relaxation of DP, Rényi Differential Privacy [2.2], is used to compute the privacy bounds in a tighter manner, allowing for better performance under the same privacy budget. Secondly, the usage of Convolutional Autoencoders allows the RDP-CGAN to work on discrete and continuous data at the same time. Additionally, temporal and correlational dependencies are included in the generated data by employing one-dimensional convolutional neural networks. Lastly, the Wasserstein GAN [4] is used to train the CGAN model to avoid mode collapse [30].

The main novelty of RDP-CGAN, apart from combining several existing approaches is the innovative usage of the Autoencoders within the architecture. Part of the privacy budget is first used to train the autoencoder. This is done both by noise addition as well as gradient clipping. The generator is then trained to produce encoded data. As such, before being either released or passed to the discriminator for further training, the data must be again decoded by the autoencoder.

This combination of dimensionality reduction which simplifies the training process for the GAN, the ability to easily include various types of data and the utility of generated data [5] make RDP-CGAN one of the leading approaches in the field of medical data synthesis.

Privacy-Preserving Generative Deep Neural Networks Support Clinical Data Sharing

The work conducted by Beaulieu-Jones [5] takes the DP-GAN model proposed in [57] and tests it against various types of medical data. Unlike the other discussed papers this one comes from a purely medical background and therefore takes a slightly different approach. The emphasis is on the medical aspect, focusing on various healthcare metrics to measure the quality of the data, rather than proposing new improvements to the original model. As such, the paper shows great promise for clinical data sharing in practice.

3.1.3. General use DP-GANs

The following GANs are only partially related to our research and are therefore introduced to showcase their various improvement, rather than to directly compare them in detail.

DP-auto-GAN

DP-auto-GAN [50] proposes an approach using autoencoders to improve the way the generator learns the latent distribution. Functionally, we can draw similarities between RDP-CGAN [57] and DP-auto-GAN [50] as both approaches use Rényi Differential Privacy [41] and Autoencoders to make the training process more effective. The difference lies in the structure of the GAN, as DP-auto-GAN does not use the CGAN architecture, instead focusing on the traditional approach. Another difference is the analysis of results, where the authors of DP-auto-GAN also focus on metrics such as 1-Way Marginal and Diversity Divergence.

DP-GAN for Time Series, Continuous, and Discrete Open Data

The GAN model proposed in [23] uses a Long Short Term Memory (LSTM) model inside the generator to handle streaming data and a Multi Layer Perceptron (MLP) to handle discrete data. There is also a proposed improvement of reducing the clipping parameter over time, which should reduce the variance in the noise and improve the convergence of the model. Using these methods and a generous privacy budget (ϵ=7), the authors were able to achieve performance close to non-private approaches. It should however be noted that the model was not tested on more complicated, medical data which tends to increase the gap between private and nonprivate model performance.

DP-AeGM & DP-VaeGM

Differentially Private Autoencoder-Based Generative Model and Differentially Private Variational Autoencoder-Based Generative Model are two DP-GAN variations proposed in [10]. The main contribution is the usage of Autoencoder and Variational Autoencoder respectively, the introduction of which should increase the quality of the data for the same privacy budget. The core focus of the paper is proposing a model that is protected against various types of attacks - Model Inversion, Membership Inference and GAN-based Attacks. The model is indeed robust against the mentioned attacks, however, this comes at a cost of performance compared to the other state-of-the-art approaches.

Differentially Private Releasing via Deep Generative Model

The approach presented in [58] shares many similarities with the DP-GAN model [57]. The key contributions comes in the form of several performance improvements. Parameter grouping and stratified clipping of different groups of parameters provide a tradeoff between convergence rate and privacy cost. Additionally, similarly to [23], the degree of gradient clipping is continuously adjusted. Lastly, the model is ”warm started” by conducting the first few training iterations on public data in a non-private manner. This assumption of the existence of public data is perhaps the main drawback of this approach. In practice, publicly available data similar to some specific private data can be more of an exception rather than a rule.

GS-WGAN
Gradient Sanitized Wasserstein Generative Adversarial Network is one of the more recent approaches toward private data synthesis. The authors of GS-WGAN [9] propose an improvement by applying the gradient sanitization mechanism to only a select group of parameters, exploiting the fact that only the generator is released. This enhances the general utility of the generated data and increases the reliability of discriminator training. Additionally, the use of Wasserstein Distance to bound sensitivity is proposed, citing that the new loss function provides lower variance in gradient norms during training. GS-WGAN also employs subsampling to reduce the chance of information about any given individual being leaked. This is done by dividing the private dataset into N subsets and creating the N discriminators, each discriminator training on a random subset each iteration. Lastly, the subsampling allows GS-WGAN to be used in a federated learning scenario.

3.1.4. Comparison of DP-GANs

Table 3.1: Comparison of analyzed DP-GAN approaches

Name	Evaluated Data	GAN structure	Uses AE
DP-GAN [57]	MNIST & Medical	Standard	No
PATE-GAN [34]	Medical	PATE	No
RDP-CGAN [57]	Medical	CGAN	Yes
PPGDNN [5]	Medical	Standard	No
DP-auto-GAN [50]	Mixed & Medical	Standard	Yes
DP-GAN for TS, C and DOD [23]	Synthetic & IOT & Medical	Standard	No (LSTM)
DP-AeGM [10]	MNIST & Medical	Standard	Yes
DPR via DGM [58]	MNIST & Faces	Standard	No
GS-WGAN [9]	MNIST & FASHION Mnist	Wasserstein	No

3.2. AutoEncoders

Autoencoders are our method of choice for improving the utility of generated data using anomaly detection. In this Section, we explore the variants and improvements of autoencoders. It is important to note that these approaches are not differentially private and would require additional modifications to use them with our framework.

Variational AE based Anomaly Detection using Reconstruction Probability

An et al. [1] propose the usage of variational autoencoders combined with the use of reconstruction probability as an improvement over the usually used reconstruction error.

Variational autoencoder [35] is an adaptation of the traditional autoencoder model with several changes. Firstly, the output of a variational autoencoder is a distribution, represented in terms of variance and mean. As such, the variance can be used as one of the parameters to determine the reconstruction probability. This is an improvement of classical autoencoders, as their deterministic mapping cannot represent the variability of data reconstruction. Additionally, the same can be said for the latent variables which are stochastic variables within the variational autoencoder context.

Lastly, the usage of reconstruction probability compared to reconstruction error allows for a much more logical selection of an anomaly threshold. In classical autoencoders, a specific threshold of reconstruction error must be chosen to remove specific anomalies. Due to the black-box nature of neural networks in general, selecting a specific threshold is inherently an arbitrary process and the selection of the value of the threshold is often done on a trial-and-error basis. The usage of reconstruction probability however allows us to select a specific consistent threshold that can be reasoned for regardless of the specific data.

Using the modified structure and the new reconstruction probability metric, the authors report significant improvement in the efficiency of the removal of anomalies compared to classic autoencoders. This efficiency is measured as the performance gain of ML models trained on datasets in which anomalies were removed with a given method.

Deep Autoencoding Guassian Mixture Model for Unsupervised AD

The approach in [62], named Deep Autoencoding Gaussian Mixture Model (DAGMM) first utilizes an autoencoder to generative low-dimensional representation of data as well as the corresponding representation error. These values are then used as input to the Gaussian Mixture Model (GMM). The output of GMM is several means, variances, and weights of Gaussian distributions corresponding to several clusters within the data. Lastly, after the GMM is fully trained, the density of each data point can be estimated. Lower density, meaning the data point belongs to lower-density regions within the Gaussian Distributions, indicates a higher chance of being an anomaly. The final probability of being an anomaly is then a composite of the density as well as the reconstruction error. Using this metric, the authors show an up to 14% improvement in f-1 scores compared to basic autoencoder-based approaches.

Robust Autoencoders

Zhoue et al. [59] propose an improvement inspired by a robust Principal Component Analysis (PCA) approach. The new approach focuses on robustness, stating performance increases, especially in cases with no clean training data. The main difference from a standard autoencoder model comes in the form of two main improvements. Firstly, an improvement upon the older model of denoising autoencoders [56] is proposed - computing entries freely based on the performance of the autoencoder algorithm. Second, unlike in [36], the sparsity of the hidden layer is not penalized, allowing for a sparse set of exceptions, for which the autoencoder loss function will not be used. Using these approaches, the authors have demonstrated a significant increase in performance. When dealing with highly noisy data, an improvement of up to 30% was achieved compared to the baseline autoencoder model.

Memory-Augmented Autoencoders

Memory-Augmented Autoencoder (MemAE), proposed by Gong et al. in [27] is a variant of autoencoders that aims to increase the efficiency of autoencoder-based anomaly detection by decreasing the overall generalization ability, fixing the issue of the autoencoder ”learning too well” and being able to reconstruct even anomalies. To amend this issue, a memory module is used. The first step is encoding the data. The encoded data is then passed to the memory module which uses it for a query, returning the most relevant memory item. This item is subsequently decoded, serving as the output of MemAE. During training the memory module is also improving, the contents are updated to present the core elements of the non-anomalous data. After the training is finished, the memory module is static. Using this approach, the authors were able to create a highly generalizable and effective method for anomaly detection. Applied to the cybersecurity dataset KDDCUP, MemAE is able to outperform the previously mentioned DAGMM [62].

Autoencoders with Nonlinear Dimensionality Reduction

In [49], Sakurada et al. compare classical autoencoders, denoising autoencoders, and linear and kernel PCA. During evaluation, they establish the performance benefits that both classical and denoising autoencoders provide. The authors also report a class of anomalies that autoencoders detect that other approaches were not able to. The performance of the tested approaches is evaluated experimentally on both synthetic and authentic non-medical data, showing a significant increase in performance for both the autoencoder models.

Context-Encoding Variational Autoencoder

CeVAE [60] uses an approach similar [62] and [1], using both the reconstruction error as well as the density to score the anomalies. The novelties come in the form of a combination of a Context Encoder [46] and a variational Autoencoder [35]. With this combination, the authors also include the KL-divergence of the posterior from the prior of the latent variable distributions. The KL-divergence, serving as a rough estimator combined with the reconstruction error can then achieve high performance, especially in the field of medical image data. To test the performance, the model was evaluated on several public challenges [12]. During the experimentation, the model was able to outperform the common autoencoder implementations, such as the variational or denoising variant.

4

GAN Anomaly Detection - GANAD

In this chapter, we introduce our framework for embedding differentially private autoencoder into various steps of a DP-GAN pipeline. Our framework, GANAD, proposes three anomaly removal approaches. Pre-Generation, Mid-Generation, and Post-Generation each apply anomaly detection at different points in the data synthesis pipeline. We first introduce the reader to the two possible approaches to ensuring the privacy of our anomaly detection model and then the following sections will explain each of our anomaly removal approaches in detail, considering the design philosophy, practical details, and lastly the privacy guarantee.

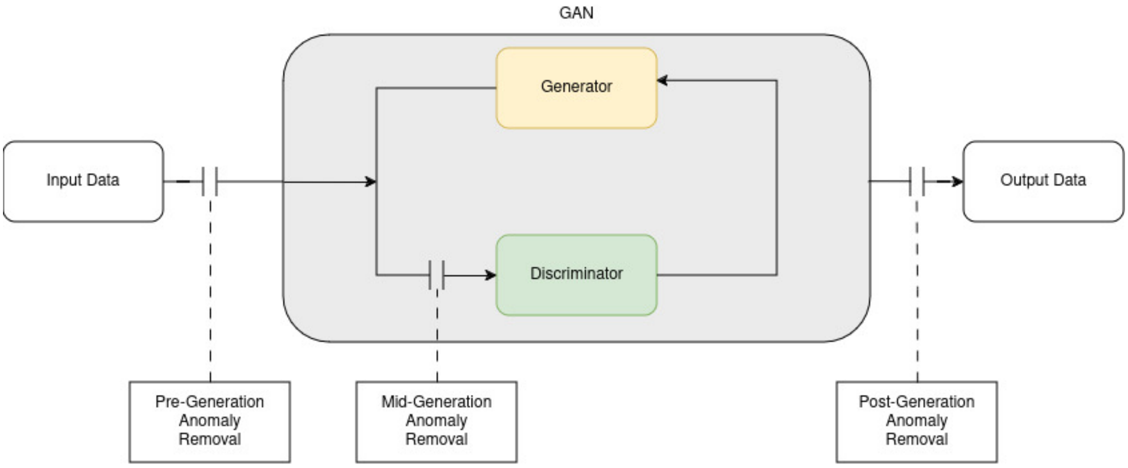
GAN
Generator
Input Data
Output Data
Discriminator
Pre-Generation Anomaly Removal
Mid-Generation Anomaly Removal
Post-Generation Anomaly Removal

Figure 4.1: The Data Synthesis Pipeline with three approaches with Anomaly Removal

4.1. Privacy of autoencoder

In this section, we explore two possible approaches to ensuring the privacy of the anomaly model of choice - autoencoders. These two privacy-preserving approaches have their own benefits and specifics which are crucial to consider when selecting one for usage in our framework.

Private Release

The simpler and more versatile approach is the private release of information by an unprivately trained model. The process starts with the standard training of a model. Afterward, the dataset in which we wish to detect anomalies is processed by the model, and the related outputs are analyzed. Using the predetermined privacy budget and the sensitivity calculated as the maximum distance between any two elements in the outputs, we apply laplacian noise to the outputs. Afterward, the highest values selected correspond to the most anomalous entries. We can repeatedly select any number of anomalies from this with no additional cost however classifying new data will require more of the privacy budget. The main advantage of this approach is that it is very simple and model-agnostic. The privacy guarantee is very easy to understand and as we are only modifying the output of a model, this approach does not require rigorous proof of the private training. With respect to the three presented options for anomaly detection, this method is not well suited for Mid-Generation removal, as we would have to split our allocated privacy budget into n parts where n is the number of GAN training iterations.

Private Training

The more advanced approach is ensuring the privacy of the output of the anomaly model by making the training process private. A model trained this way can process an unlimited amount of non-private data and leak no information with the output. This is an advantage achieved by a complex training approach. For basic autoencoders, similarly to GANs, gradient clipping, and gradient noise addition can be used to train a model privately [47]. As stated, the biggest drawback of this approach is the limitation with regard to the various improvements to the standard autoencoder model. These improvements, discussed in Chapter 3, would require additional analysis to determine whether this training approach does not leak any information, however, that is beyond the scope of this thesis. Additionally, private training of autoencoders can often lead to higher performance compared to the previous naive approach. For this reason, we believe privately trained models to be the superior option to use with our framework.

4.2. Pre-Generation Anomaly Removal

The first approach is Pre-Generation data anomaly removal. This is the simplest approach and is akin to standard data preprocessing.

Design philosophy

As stated before, medical data contains a variety of intricate patterns. This naturally proposes a challenge, as generating data incorporating these patterns is difficult. However, we can also exploit the presence of said patterns to train a highly efficient anomaly removal model.

The most immediate use of this fact is simply removing the anomalies from the data by training and applying the anomaly detection model to the private authentic data. This is done before giving a GAN access to the data, which should allow the GAN to be trained to produce significantly fewer anomalies, leading to utility gains.

Practical details

One of the design goals of our approach is flexibility about the type of input data. Therefore, our approach works in an unsupervised manner, assuming no labeling of anomalies in the private data. The private, unprocessed data is then passed to an anomaly detection deep learning model, an Autoencoder in our case. As training is done in a non-privacy-preserving manner, we may simply train the model till we reach the desired loss. Afterward, a number or percentage of anomalies is removed. This number highly depends on the dataset and therefore becomes another variable to hyperoptimize. The GAN is then trained on this anomaly-free data.

Privacy guarantee

First, to understand the potential privacy cost we need to state the situation and our assumptions: The anomaly detection model is trained on private data in a nonprivate manner. The model and its output are never published. A model can be trained with regard to its own loss, however, it is not to be optimized with regard to the output of the GAN. Under these assumptions, there is no additional privacy cost associated with the anomaly removal. As differential privacy quantifies the chance of an individual's information being leaked and there is no way to way for an attacker to determine whether an individual's data was removed from the dataset via anomaly removal, there is no additional privacy leakage relating to pregen GANAD.

4.3. Mid-Generation Anomaly Removal

The second approach involves the continuous removal of anomalies during the GAN training process. This is the most unique of the approaches proposed and aims to increase the efficiency of the discriminator to generate higher utility data.

Design philosophy

The core idea is again related to the complex, correlated nature of medical data. In a regular training process, we expect the generator to try to reproduce the patterns present in the training data. Especially at the beginning of this process, the generator performs poorly, leading to slow training of the discriminator. In the later stages, the discriminator might reach a local maximum by overly focusing on improperly reproduced anomalies. Mid-Generation removal addresses this by eliminating anomalies from the data passed to the discriminator, both authentic and synthetic.

Ideally, this forces the discriminator to consider less immediately apparent patterns instead of focusing on improperly reconstructed anomalies. In other words, with fewer ”low-hanging fruits,” the discriminator must learn to reproduce more subtle patterns, ultimately leading to the generation of higher utility data.

Practical details

In practice, the inner flow of data between the generator and discriminator is the most likely to differ in various improved GAN implementations. As such, we provide the general approach for classical GANs and guidelines for more advanced approaches. The first step is again privately training the anomaly detection model on private data. Afterwards, during the GAN training process, anomalies are removed from both the original as well as generated data before being passed to the discriminator. We make sure to remove the same number of anomalies from both of these classes to prevent data imbalance in the training. In case of multiple sets of data being passed to discriminators, we make sure that each of these sets ends up the same size to again prevent imbalance.

Privacy Guarantee

In this approach, unlike from the other ones, we make repeated queries to the anomaly detection model. As such, it is crucial that the model is trained privately, instead of only its results being privately released. With a model trained in such a way, the privacy loss can be quantified as only the privacy budget allocated to the training of the anomaly detection model, meaning the number of training iterations, and queries to the private model, does not have an impact on the final privacy budget. The training process which ensures the privacy of the output of the anomaly detection model will be explored further in Section 4.1.

4.4. Post-Generation Anomaly Removal

The last presented option is removing the anomalies after the data is generated. This approach is similar to standard data processing however the key difference is in the usage of part of the budget to get access to the private data for training the anomaly detection model.

Design Philosophy

The core idea is to carefully select a part of the budget and use it to train an anomaly detection model in a differentially private manner on authentic private data. This trained model can then be used on the final output of the GAN to remove anomalies and improve the utility of the synthetic data. The theory is that a Differentially Private learning process, especially one done on highly complex, highly correlated data, will introduce anomalies that can be effectively removed using information from the private dataset. As such, we hope that allocating part of the budget to the anomaly detection model provides higher total utility than just increasing the budget for the GAN training. This will be tested in depth in Chapter 5.

Practical Details

As we are proposing a flexible framework, the data can again be in any format. Afterwards, an anomaly detection model is trained privately, with a preset privacy budget. Typically, this is only a small fraction of the overall privacy budget, although the exact figures are reported in Chapter 5. The anomaly detection model can be trained simultaneously with the GAN. In practice, we expect the training of the GAN to take significantly more time than that of the anomaly detection model. This combined with the negligible anomaly classification time means that the additional runtime of this approach is close to zero.

Another aspect to consider is the amount of data points released. Compared to the other approaches, the generator will give us a preset number of data points to which we subsequently apply anomaly detection. As an example, if we assume that we want to release n data points, we need to decide in advance how many anomalous points we want to remove and generate $n+$ that many extra points. This is the scenario we assume and it ensures that no extra information is leaked by optimizing on the outputs of the generator.

Privacy guarantee

Post-Generation anomaly removal works on the basis of using private information to directly modify the data that a third party will have access to. As such, the anomaly removal leaks information in a way that is quite easy to exploit by the attacker.

Imagine the classical inference attack scenario, where the attacker has access to two neighboring (private) datasets and to the data released by the GAN. In such a scenario, the attacker might see a difference in the output of Post-Generation using GAN depending on the anomality of the one entry. While this is an informal example, there is certainly information leakage present. As such, we have chosen to quantify the higher bound of this risk with the part of the privacy budget ϵ. The allocated privacy budget then limits the learning of an anomaly detection model so that the effect it has on the outputted data by the GAN and any subsequent privacy leakage is limited.

5 Evaluation and Results

To evaluate our framework in practice, we conducted a variety of experiments with different DP-GAN implementations, several relevant medical datasets, number of different machine learning models and changing epsilon values. These experiments were repeated 10 times to account for the randomness of the learning process. This Chapter first provides general reproducibility details which allow anyone to easily verify the results as well as expand on the work done. In the next Section, we consider the specific implementation details related to the three tested models - DPGAN, RDPCGAN, and PATEGAN. Following, we introduce the three datasets used for the testing, placing focus on the specifics of the data present in the datasets, as well as general metrics such as dataset size or the number of features. Lastly, we apply the robust testing setup to reliably evaluate the three proposed anomaly removal approaches across a variety of settings.

5.1. Reproducibility specifics

This Section provides all the details necessary for reproducing the results.

Codebase

We have published the code used for testing at a public GitHub repository [54]. We thank the authors of [52] for their provided implementation of RDP-CGAN and the authors of [34] for the codebase for PATE-GAN. Both of these implementations, as well as our own implementation of the standard DP-GAN model [57] were used in conjunction with our framework.

Hardware and OS

The experiments were run simultaneously on 2 systems - A System with 16 GB of RAM and 1165G7 CPU and another System with 32 GB of Ram, 7600X CPU and a 3080 GPU. Both systems ran on Ubuntu. We believe our code to be machine agnostic however Linux operating system is most likely required due to the required libraries and drivers.

Environment and Libraries

For our environment, we used Python version 3.7, with Tensorflow version 1.15.0 and the Nvidia version of Tensorflow [43]. For the implementation of the Laplace Mechanism, the Diff-PrivLib by IBM was used [31]. All the other required packages are present in a requirements.txt [54] file.

5.1.1. Hyperparameters

In this Section, we provide the hyperparameters as well as explain the rationale behind their selection.

Unless stated explicitly stated otherwise, the following values were used as hyperparameters:

Table 5.1: Hyperparameters used for running experiments

Parameter Name	PATEGAN	RDPCGAN	DPGAN
Epsilon	0.1/1/10	0.1/1/10	0.1/1/10
Training Iterations	10	1/10/30	NA
Number of Experiments per setup	10	10	10
Delta	10^{-5}	10^{-5}	10^{-5}
Teachers	1000	NA	NA
Batch Size	64/16	64/16	64/16
Noise Rate	1.0	10/4/1	1.0
Mid/Post Generation ϵ budget	10%	10%	10%

Epsilon

Three distinct values were selected for the privacy budget ϵ. These specific values were chosen to allow for easy comparison to existing research, using the same values. Additionally, they correspond to values used in the real world by both companies and researchers [16].

Training Iterations

Due to the implementation specifics, each of the considered models handles the number of epochs a bit differently. In PATEGAN, we can simply specify the number of epochs. In RDP-CGAN, the privacy budget ϵ is not directly selectable and depends partially on the number of epochs. As such, 1/10/30 epochs were used, corresponding to 0.1, 1, and 10 ϵ respectively. Lastly, our implementation of DPGAN simply trains in epochs till the privacy budget is reached so no epoch selection is required/necessary.

The selection of the number of epochs for RDP-CGAN was an empirical process. By far the two most deciding factors of the calculated epsilon value are the number of epochs and the Noise Rate. Therefore, we have attempted to select values that achieve a balance between these two parameters.

Delta

The value for delta was selected to again coincide with values used in existing research. With value of 10^{-5}, delta corresponds to a small enough chance to not meaningfully impact the data release while still improving the utility of generated data due to relaxed security constraints.

Teachers

The values regarding the number of Teachers for the implementation of PATEGAN were picked due to findings in the original paper [34] regarding the efficiency of PATEGAN related to various amounts of teachers.

Batch Size

The batch size used when training on both the UCI Epileptic as well as the Cardio Vascular disease dataset was 64. For compatibility reasons and due to the smaller amount of records, 16 was used for the Cervical Cancer dataset.

The value of 64 was used due to other works on the same dataset selecting it.

Noise Rate

Again, due to implementation details regarding the privacy budget ϵ selection, RDP-CGAN uses 10, 4, and 1 as the noise ratio corresponding to 0.1, 1, and 10 ϵ respectively.

The values for Noise Rate in general were selected to achieve between the number of epochs as well as the noise rate - the two deciding factors for the privacy budget computation.

Mid/Post Generation ϵ budget

This corresponds to the part of the overall privacy budget used for the training of the anomaly removal autoencoder. We have briefly tested several different percentages and settled on 10% as the best-performing amount. To truly determine the ideal percentage, additional research that is outside the scope of this work would be required.

5.2. Utility Measurements

To measure the utility of generated data, we adapted the techniques developed by Jordon and Yoon [34]. The reason for this is twofold - firstly, adapting the same utility metrics allows us to directly compare our results to the original paper. Secondarily, we believe the technique to be versatile and of high quality.

In this Section, we give a brief introduction to the individual models used and the way their performance is measured. For performance measurements, we have chosen to use AUROC and AUPRC instead of metrics such as F1 score or accuracy. We believe that AUROC and AUPRC are advantageous in our specific context as they perform better with unbalanced data and provide a more comprehensive evaluation of a model's performance. We measured both the AUROC and AUPRC scores, however for the conciseness only AUROC scores are present in the Section. The additional data can be found in Appendix A.

Logistic Regression

Logistic regression is a simple linear model used for binary classification that predicts the probability of a binary outcome based on input features.

Gaussian Naive Bayes

Gaussian Naive Bayes [61] variant of Naive Bayes that assumes features are normally distributed and uses Bayes' theorem to predict the probability of each class.

Bernoulli Naive Bayes

Bernoulli Naive Bayes [61] is another variant of Naive Bayes suitable for binary feature data, where features are assumed to be independent boolean variables.

Linear Discriminant Analysis

Linear Discriminant Analysis [6] is a method used for dimensionality reduction and classification. It models the distribution of the predictors separately in each class and uses Bayes' theorem to estimate the probability of each class.

Random Forest

Random Forest [7] is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes or the average prediction of the individual trees.

Extra Randomized Trees

Extra Randomized Trees is model similar to Random Forests, but the splits are chosen randomly instead of optimally.

Adaptive Boosting

Adaptive Boosting [21] is a learning method that iteratively trains weak classifiers by focusing on instances that were previously misclassified.

Bootstrap Aggregating

Bootstrap Aggregating, also known as Bagging is an approach that improves the accuracy and stability of machine learning algorithms by averaging multiple models trained on different subsets of the data.

Support Vector Machine

Support Vector Machine [13] is a supervised learning model used for classification and regression analysis. It can find an optimal hyperplane that best separates classes in a high-dimensional space.

Gradient Boosting Machine

Gradient Boosting Machine [22] is a technique where new models are sequentially added, each correcting errors made by the previous models, thereby minimizing the overall error.

XGBoost

XGBoost [11] is an optimized implementation of gradient boosting that is highly efficient, flexible, and scalable.

5.2.1. AUROC (Area Under the Receiver Operating Characteristic Curve)

AUROC is a metric commonly used to evaluate the performance of binary classification models. It plots the true positive rate against the false positive rate at various threshold settings. The area under this curve (AUROC) provides a measure of the model's ability to distinguish between positive and negative classes across all possible thresholds, with values closer to 1 indicating better discrimination capability and a value of 0.5 suggesting performance comparable to random chance.

5.2.2. AUPRC (Area Under the Precision-Recall Curve)

AUPRC evaluates the precision-recall trade-off of a binary classifier. Unlike AUROC, which focuses on the true positive rate against the false positive rate, AUPRC considers the precision and recall of the classifier. It is particularly useful in settings where the class distribution is highly imbalanced, as it provides a comprehensive measure of the classifier's performance across all thresholds. A higher AUPRC indicates better precision-recall trade-off, with a maximum value of 1 indicating perfect performance and a baseline value influenced by the proportion of positive examples in the dataset.

5.3. Datasets

After careful analysis of the performance of state-of-the-art DPGAN approaches, we have selected three datasets that have been widely used for experiments in existing works. This selection allows us to directly compare our results as well as test our approach in a variety of medical settings.

5.3.1. UCI-Epileptic

The version of the UCI-Epileptic dataset [2] that we used contains 178 features, each indicating a one-second brain activity recording. The dataset contains six classes, with 1 indicating that the measurements were recorded during an epileptic seizure and the other corresponding to other factors - such as eyes being closed or the exact area of the brain being recorded. We merged the five other classes, transforming this into a binary classification problem. Overall, this modified version of the dataset contains 11500 rows. As such, it serves as an example of a highly correlated highly complex dataset with a decent number of samples.

5.3.2. Kaggle Cardiovascular Disease

The Cardiovascular dataset [53] provides 11 features, ranging from age, and weight, to glucose levels and smoking status. The goal of this dataset is to determine whether the person has a cardiovascular disease. This dataset contains the fewest features of the three used datasets however it also contains over 70000 rows of data. As such, it provides the most complex setting concerning both time and memory. This allows us to test how well our proposed approaches scale with these parameters.

5.3.3. Kaggle Cervical Cancer

The Cervical Cancer dataset [19] is the third dataset we used for our experiments. With 35 features, it serves as a nice middlepoint between the UCI and the Cardiovascular dataset. It also only has around 800 records, presenting a unique opportunity to test the performance of GANAD in a setting with a limited amount of training data. As this data contains some missing values and is already limited in size, we used preprocessing to extend the utility of existing data as much as possible.

5.4. GANs

While we already explored various state-of-the-art approaches towards DP-GANs in detail in Chapter 3, here we focus more on the implementation details and decisions that were made while implementing GANAD within specific GANs. We also report on the performance of GANAD in these scenarios.

5.4.1. PATE-GAN

For testing of PATE-GAN [34], we modified the code provided by the authors at [55].

For the implementation of the pre-generation anomaly removal, we first load the data and apply any necessary preprocessing. Subsequently, we train an autoencoder with an unlimited privacy budget, based on the rationale explained in Chapter 4, and remove several anomalies. The rest of the process does not differ from the original PATEGAN implementation.

Our mid-generation anomaly removal approach is a bit more involved, as it consists of removing anomalies within each of the training loops of the students. In our implementation, we performed anomaly removal on each of the batches during each training epoch using a differential-privately trained autoencoder.

The post-generation anomaly removal again involves the private training of an autoencoder. The autoencoder is then used on the output of the GAN to remove a preset number of anomalies. Otherwise, the data generation and the PATE-GAN training process are not modified.

Experimental Results

Table 5.2: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - RDPCGAN UCI-Epileptic.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	0.402 ± 0.001	0.409 ± 0.002	0.465 ± 0.018
Pre-Generation	**0.519 ± 0.006**	0.403 ± 0.001	**0.519 ± 0.014**
Mid-Generation	0.493 ± 0.007	0.427 ± 0.003	0.431 ± 0.006
Post-Generation	0.454 ± 0.002	**0.476 ± 0.008**	0.456 ± 0.014

Table 5.3: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - PATEGAN Cardiovascular Disease.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	**0.575 ± 0.001**	0.597 ± 0.001	0.627 ± 0.000
Pre-Generation	0.557 ± 0.000	0.586 ± 0.000	**0.650 ± 0.000**
Mid-Generation	0.559 ± 0.001	0.597 ± 0.001	0.637 ± 0.001
Post-Generation	0.558 ± 0.001	**0.599 ± 0.001**	0.647 ± 0.001

Table 5.4: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - PATEGAN Cervical Cancer.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	**0.632 ± 0.005**	**0.654 ± 0.004**	0.686 ± 0.004
Pre-Generation	0.599 ± 0.014	0.621 ± 0.008	0.693 ± 0.008
Mid-Generation	0.594 ± 0.009	0.636 ± 0.005	0.672 ± 0.004
Post-Generation	0.605 ± 0.009	0.642 ± 0.009	**0.697 ± 0.002**

Intepreting the results

Across the tested datasets and the various ϵ values, we can see that the usage of GANAD is the most effective at high-privacy budgets, yielding above-baseline performance in all three tested datasets. The Pre-Generation anomaly removal also stands out as the best-performing variant of GANAD. Lastly, we observe the highest improvements related to GANAD in UCI-Epileptic, the dataset with the highest number of features.

5.4.2. RDP-CGAN

For our testing of RDP-CGAN [57], we used the base implementation provided by the authors at [51]. For ease of testing, we created a single file, combining the data synthesis pipeline provided by RDP-CGAN with the data quality testing methods used in [55]. It is worth noting that RDP-CGAN takes a bit of a different approach to generating labeled data. It is trained only on a specific class of data which it then reproduces. As such, generating data with two classes involves two trainings of the model. However, as the training is always only accessing one class of the data, the overall privacy budget is not increased compared to other GANs.

The pre-generation anomaly removal involves training the autoencoder on the whole dataset once and then applying it on the data before the start of each of the respective RDP-CGAN training processes. No further changes were required to implement pre-generation anomaly removal with RDP-CGAN.

To remove anomalies using the mid-generation approach, we start by differential-privately training the autoencoder. Afterward, we use the trained model to remove an equal number of synthetic and authentic entries in each training batch in each training epoch.

Post-generation anomaly removal consists of first using the predefined privacy budget to train both the RDP-CGAN as well at the autoencoder. The data is then generated using the trained RDP-CGAN and subsequently, the autoencoder is used to remove anomalies from this data. Lastly, the quality of the data can be evaluated.

Experimental Results

Table 5.5: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - RDPCGAN UCI-Epileptic.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	0.402 ± 0.001	0.409 ± 0.002	0.465 ± 0.018
Pre-Generation	**0.519 ± 0.006**	0.403 ± 0.001	**0.519 ± 0.014**
Mid-Generation	0.493 ± 0.007	0.427 ± 0.003	0.431 ± 0.006
Post-Generation	0.454 ± 0.002	**0.476 ± 0.008**	0.456 ± 0.014

Table 5.6: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - RDPCGAN Cardiovascular Disease.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	0.548 ± 0.002	0.539 ± 0.001	0.555 ± 0.001
Pre-Generation	**0.563 ± 0.001**	**0.545 ± 0.001**	**0.559 ± 0.000**
Mid-Generation	0.553 ± 0.003	0.536 ± 0.001	0.523 ± 0.002
Post-Generation	0.555 ± 0.001	**0.545 ± 0.001**	0.553 ± 0.000

Table 5.7: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - RDPCGAN Cervical Cancer.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	0.544 ± 0.004	**0.659 ± 0.001**	0.602 ± 0.012
Pre-Generation	**0.560 ± 0.010**	0.638 ± 0.002	**0.731 ± 0.008**
Mid-Generation	0.556 ± 0.006	0.618 ± 0.001	0.695 ± 0.012
Post-Generation	0.555 ± 0.001	0.636 ± 0.004	0.580 ± 0.016

Interpreting the results

The results of RDP-CGAN show a clear benefit of using GANAD in practice. In only one of the nine total tested scenarios, we see a baseline returning the best results. We again see good results in high privacy budget setting, with RDPCGAN Cervical Cancer returning the highest improvement out of all tested scenarios. We do not see any point at which Mid-Generation anomaly removal performs well, suggesting it may not be the best option to use with RDPCGAN. Lastly, while we observed the highest performance in the Cervical Cancer dataset, there does not seem to be a significant difference regarding the complexity of the data.

5.4.3. DP-GAN

As a test of our approach with the most basic settings, we implemented our own version of DP-GAN. This version closely follows the standard GAN architecture and only implements Differential Privacy by means of gradient perturbation and clipping.

The pre-generation anomaly removal approach again involves (nonprivately) training the autoencoder and using it to remove anomalous entries from the original dataset. This cleansed dataset is then used for the DP-GAN training. Afterward, the data is simply generated and evaluated.

For the implementation of the mid-generation anomaly removal, we use part of the privacy budget to train the autoencoder. Then, after the generation of the synthetic data but before forwarding it to the discriminator in each iteration, we employ the autoencoder and remove anomalies from both authentic and synthetic data. After the training process, the data generation and evaluation is done.

Lastly, the post-generation anomaly removal first involves privately training both the DP-GAN and the autoencoder. We generate the data and subsequently remove a preset number of anomalies using the autoencoder. The quality of anomaly-free data is then tested.

Experimental Results

Table 5.8: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - DPGAN UCI-Epileptic.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	0.503 ± 0.001	**0.512 ± 0.001**	**0.499 ± 0.000**
Pre-Generation	**0.525 ± 0.001**	**0.512 ± 0.001**	0.483 ± 0.001
Mid-Generation	0.508 ± 0.002	0.498 ± 0.001	0.491 ± 0.001
Post-Generation	0.507 ± 0.001	0.498 ± 0.002	0.483 ± 0.001

Table 5.9: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - DPGAN Cardiovascular Disease.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	0.483 ± 0.001	0.505 ± 0.001	0.514 ± 0.001
Pre-Generation	**0.508 ± 0.001**	**0.516 ± 0.002**	0.489 ± 0.002
Mid-Generation	0.506 ± 0.002	0.506 ± 0.001	0.503 ± 0.001
Post-Generation	0.492 ± 0.001	0.493 ± 0.001	**0.515 ± 0.001**

Table 5.10: ML Utility comparison of different AR methods using AUROC metric with varying epsilon values - DPGAN Cervical Cancer.

AR Methods/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
Control	0.542 ± 0.016	0.428 ± 0.018	**0.515 ± 0.012**
Pre-Generation	0.505 ± 0.008	0.509 ± 0.004	0.488 ± 0.014
Mid-Generation	**0.544 ± 0.008**	**0.515 ± 0.007**	0.498 ± 0.013
Post-Generation	0.446 ± 0.007	0.507 ± 0.021	0.476 ± 0.026

Interpreting the results

The testing of DPGAN and GANAD revealed two things. Firstly, the application of GANAD with DPGAN can lead to utility gains mostly in lower-complexity datasets. Secondly, the DP-GAN seems to struggle with synthesizing complex medical data. This can be observed in the fluctuating AUROC scores measured in the Cervical Cancer dataset, with a non-positive correlation between the utility and the privacy budget.

5.5. Anomaly detection usage

Pre-Generation Anomaly Removal

Table 5.11: The difference between the AUROC score of control and Pre-Generation AR - averaged across models for each dataset

Dataset/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
UCI Epileptic	0.040 ± 0.005	-0.013 ± 0.000	0.009 ± 0.002
Cervical Cancer	-0.018 ± 0.001	0.009 ± 0.004	0.036 ± 0.007
Cardio Vascular	0.007 ± 0.001	0.002 ± 0.000	0.001 ± 0.001

Mid-Generation Anomaly Removal

Table 5.12: The difference between the AUROC score of control and Mid-Generation AR - averaged across models for each dataset

Dataset/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
UCI Epileptic	0.028 ± 0.003	-0.010 ± 0.001	-0.014 ± 0.000
Cervical Cancer	-0.008 ± 0.001	0.009 ± 0.005	0.021 ± 0.004
Cardio Vascular	0.004 ± 0.000	-0.001 ± 0.000	-0.011 ± 0.000

Post-Generation Anomaly Removal

Table 5.13: The difference between the AUROC score of control and Post-Generation AR - averaged across models for each dataset

Dataset/Epsilon	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 10$
UCI Epileptic	0.022 ± 0.001	0.013 ± 0.002	-0.009 ± 0.000
Cervical Cancer	-0.037 ± 0.003	0.015 ± 0.003	-0.017 ± 0.001
Cardio Vascular	0.000 ± 0.000	-0.001 ± 0.000	0.007 ± 0.000

5.6. Intepreting the results

Defined in the first Chapter, we ask the question ”How can we improve the utility of synthetic medical data by employing anomaly detection?”. While we have observed utility gains across a variety of datasets and data-generation approaches, the true answer is a bit more complex. We start by analyzing cases where the model yielded significant results, either positively or negatively.

5.6.1. RDPCGAN performance gains

Across datasets and different values of epsilon, we have observed the highest increase of utility when applying GANAD to RDP-CGAN. Not only have we observed the highest individual gain compared to baseline - around 0.13 AUROC increase with Cervical Cancer $\epsilon = 10$ dataset, we additionally observed performance increase related to the usage of GANAD compared to the baseline in 8 out of 9 tested scenarios.

We believe this performance gain is related to the RDP-CGAN training process. As RDP-CGAN uses an autoencoder itself to reduce the dimension of the training data, the anomaly removal provided by the GANAD framework can lead to more efficient training and subsequent higher utility of data. While this is just a possible explanation, it is also corroborated by the fact that the high RDP-CGAN results were often achieved by pre-generation anomaly removal.

5.6.2. Efficiency based on complexity of datasets

Based on the results, it seems that the efficiency of GANAD seems to depend more on the size of the dataset rather than the number of features. In the dataset with over 70000 entries, 8 out of 9 test configurations again returned an AUROC score higher than the baseline using one of the GANAD removal techniques.

On the other side of the spectrum is the Cervical Cancer dataset with only 800 records. In this dataset, apart from the outlier related to the performance of RDP-CGAN at $\epsilon = 10$, we have observed the smallest utility increase related to GANAD usage compared to the other datasets.

6
Conclusion, Discussion & Future Work

In this Chapter we analyze the results, discussing what they mean and how they answer the question stated in Chapter 1. We conclude by presenting several directions in which this work can be expanded upon and summarizing our contribution.

6.1. Discussion

In this Section, we discuss the real-world usefulness and applicability of our framework.

6.1.1. Use-case scenarios

Both our expectations as well as our results show that the GANAD framework works best in highly correlated and complex datasets. Such datasets both make the data synthesis more difficult as well as allow the autoencoder to be trained more efficiently. The dataset should also not be preprocessed in terms of removing anomalies as this is crucial for the efficiency of GANAD.

6.1.2. GANAD practicality

Our results, analyzed in the previous Chapter, indicate that while the blind application of GANAD yields mixed results, careful and informed configuration can lead to significant gains in utility. We estimate that with proper application, a 3-5% increase in AUROC score is achievable in most scenarios. Key parameters to optimize include the number of anomalies removed, the specific partition of the privacy budget allocated for the private training of the anomaly removal model, and the exact stage at which anomaly removal is performed. However, it is crucial to ensure that the optimization process does not inadvertently leak additional information about the private dataset. While we do not provide a rigorous proof that the selection of GANAD hyperparameters itself reveals sensitive information, caution is warranted given the nature of the data. Thus, we recommend erring on the side of caution to protect data privacy.

6.2. Future Work

While we tried to design as diverse and objective testing of DP-GAN-related anomaly removal as possible, there is some space for improvement. In this section, we outline possible directions for further effort on this front.

6.2.1. Usage of more sophisticated anomaly removal methods

For our testing, we have employed a rather basic differentially trained autoencoder. This was done simply for the solid theoretical basis behind the differential privacy of autoencoders, allowing us to both easily calculate the related information leakage as well as ensure that no additional information is leaked due to unexplored intricacies of the anomaly removal model. However, as we explored in Chapter 3, there is a plethora of highly advanced options for anomaly removal which could in practice yield even better results, provided we can accurately calculate the privacy cost. Therefore, we believe focusing on more sophisticated differentially private anomaly removal is likely the most promising improvement in terms of achievable utility gains.

6.2.2. Number of anomalies to remove

As the focus of this thesis was on a rather diverse setting - synthetic medical data - we have been unable to focus on any specific learning scenario in too much detail with regard to the number of anomalies removed from the data. More in-depth research, experimenting with various settings, and different numbers of anomalies removed could lead to a substantial utility increase in the final data. Additionally, observing the tradeoff between the number of removed anomalies and the utility of the data could reveal some interesting trends across datasets and could lead to good heuristics for the selection of the number of anomalies to remove, usable in practice.

6.2.3. Application of GANAD to non-medical data

There is a solid reasoning behind our selection of a group of data to analyze. As stated in Chapter 1, we believe that due to several reasons such as high stakes or high level of correlation, medical data is a prime candidate for differentially private data synthesis as well as the anomaly removal we propose. However, it may be possible that GANAD performs well on various other types of data. Therefore, we believe applying GANAD on new, highly correlated domains may yield promising results and could serve as a promising direction for research.

6.2.4. Exploration of low-dimensional datasets

During our testing, we focused on testing our framework with datasets commonly used in existing medical DP-GAN research. This however meant all of the explored datasets were rather complex concerning the number of features they have - even the most simple dataset had over 30 features. Therefore, there is a lack of data regarding the performance of GANAD over low-dimensional datasets. As GANAD works by exploiting the correlated nature of high-dimensional datasets, we hypothesize it performs worse with a lower number of dimensions however further research is needed.

6.3. Conclusion

In conclusion, we have demonstrated that with careful application, it is possible to enhance the utility of synthetic medical data through anomaly detection and removal. ”While our ultimate goal—maximizing data utility—aligns with several existing approaches, to the best of our knowledge, we are the first to propose a method specifically tailored to removing anomalies. Although our approach does not increase utility in every scenario, it can be a valuable tool for optimizing the final utility of generated data when applied thoughtfully. Moreover, our method incurs no additional privacy costs and introduces a negligible time overhead, less than 1% of the overall runtime.

References

[1] Jinwon An and Sungzoon Cho. “Variational autoencoder based anomaly detection using reconstruction probability”. In: *Special lecture on IE* 2.1 (2015), pp. 1–18.

[2] Ralph Andrzejak et al. “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state”. In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 64 (Jan. 2002), p. 061907. DOI: 10.1103/PhysRevE.64.061907.

[3] Arcadia. “Report: Only 57 percent of Healthcare Organizations’ Data is Used to Make Decisions”. In: *HIMSS and Arcadia Market Insights Pulse Survey Research* (2023).

[4] Martín Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein GAN”. In: *CoRR* abs/1701.07875 (2017). arXiv: 1701.07875. URL: http://arxiv.org/abs/1701.07875.

[5] Brett Beaulieu-Jones et al. *Privacy-preserving generative deep neural networks support clinical data sharing*. Dec. 2018. DOI: 10.1101/159756.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent Dirichlet Allocation”. In: *J. Mach. Learn. Res.* 3 (2003), pp. 993–1022. URL: http://jmlr.org/papers/v3/blei03a.html.

[7] Leo Breiman. “Random Forests”. In: *Mach. Learn.* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[8] Sílvia Casacuberta et al. “Widespread Underestimation of Sensitivity in Differentially Private Libraries and How to Fix It”. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*. Ed. by Heng Yin et al. ACM, 2022, pp. 471–484. DOI: 10.1145/3548606.3560708. URL: https://doi.org/10.1145/3548606.3560708.

[9] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. “GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/9547ad6b651e2087bac67651aa92cd0d-Abstract.html.

[10] Qingrong Chen et al. “Differentially Private Data Generative Models”. In: *CoRR* abs/1812.02274 (2018). arXiv: 1812.02274. URL: http://arxiv.org/abs/1812.02274.

[11] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. Ed. by Balaji Krishnapuram et al. ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[12] Xiaoran Chen et al. “Deep Generative Models in the Real-World: An Open Challenge from Medical Imaging”. In: *CoRR* abs/1806.05452 (2018). arXiv: 1806.05452. URL: http://arxiv.org/abs/1806.05452.

[13] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Mach. Learn.* 20.3 (1995), pp. 273–297. DOI: 10.1007/BF00994018. URL: https://doi.org/10.1007/BF00994018.

[14] Tore Dalenius. "Towards a methodology for statistical disclosure control". In: *Statistik Tidskrift* 15 (1977), pp. 429–444.

[15] R David, G John, and R John. "Data age 2025: the digitization of the world, from edge to core". In: *IDC White Paper. Seagate* (2018).

[16] Damien Desfontaines. *A list of real-world uses of differential privacy*. https://desfontain.es/blog/real-world-differential-privacy.html. Ted is writing things (personal blog). Oct. 2021.

[17] Cynthia Dwork. "Differential Privacy". In: *Automata, Languages and Programming*. Ed. by Michele Bugliesi et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12. ISBN: 978-3-540-35908-1.

[18] Cynthia Dwork et al. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *J. Priv. Confidentiality* 7.3 (2016), pp. 17–51. DOI: 10.29012/JPC.V7I3.405. URL: https://doi.org/10.29012/jpc.v7i3.405.

[19] Kelwin Fernandes, Jaime S. Cardoso, and Jessica Fernandes. "Transfer Learning with Partial Observability Applied to Cervical Cancer Screening". In: *Pattern Recognition and Image Analysis - 8th Iberian Conference, IbPRIA 2017, Faro, Portugal, June 20-23, 2017, Proceedings*. Ed. by Luís A. Alexandre, José Salvador Sánchez, and João M. F. Rodrigues. Vol. 10255. Lecture Notes in Computer Science. Springer, 2017, pp. 243–250. DOI: 10.1007/978-3-319-58838-4_27. URL: https://doi.org/10.1007/978-3-319-58838-4%5C_27.

[20] World Economic Forum. "4 ways data is improving healthcare". In: *Global Innovation Index* (2019).

[21] Yoav Freund and Robert E. Schapire. "Experiments with a New Boosting Algorithm". In: *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*. Ed. by Lorenza Saitta. Morgan Kaufmann, 1996, pp. 148–156.

[22] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

[23] Lorenzo Frigerio et al. "Differentially Private Generative Adversarial Networks for Time Series, Continuous, and Discrete Open Data". In: *ICT Systems Security and Privacy Protection - 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings*. Ed. by Gurpreet Dhillon et al. Vol. 562. IFIP Advances in Information and Communication Technology. Springer, 2019, pp. 151–164. DOI: 10.1007/978-3-030-22312-0_11. URL: https://doi.org/10.1007/978-3-030-22312-0%5C_11.

[24] Benjamin C. M. Fung et al. "Privacy-preserving data publishing: A survey of recent developments". In: *ACM Comput. Surv.* 42.4 (2010), 14:1–14:53. DOI: 10.1145/1749603.1749605. URL: https://doi.org/10.1145/1749603.1749605.

[25] R. Furberg et al. *Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016*. 2016. DOI: 10.5281/zenodo.53894.

[26] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*. Ed. by Yee Whye Teh and D. Mike Titterington. Vol. 9. JMLR Proceedings. JMLR.org, 2010, pp. 249–256. URL: `http://proceedings.mlr.press/v9/glorot10a.html`.

[27] Dong Gong et al. “Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection”. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 1705–1714. DOI: `10.1109/ICCV.2019.00179`. URL: `https://doi.org/10.1109/ICCV.2019.00179`.

[28] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: *CoRR* abs/1406.2661 (2014). arXiv: `1406.2661`. URL: `http://arxiv.org/abs/1406.2661`.

[29] Google. “Differential Privacy - Google Library”. In: *Google’s differential privacy libraries* (2024). URL: `https://github.com/google/differential-privacy`.

[30] Ishaan Gulrajani et al. “Improved Training of Wasserstein GANs”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 5767–5777. URL: `https://proceedings.neurips.cc/paper/2017/hash/892c3b1c6dccd52936e27cbd0ff683d6-Abstract.html`.

[31] Naoise Holohan et al. “Diffprivlib: the IBM differential privacy library”. In: *ArXiv e-prints* 1907.02444 [cs.CR] (July 2019).

[32] IDC and Statista. *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 (in zettabytes) [Graph]*. `https://www.statista.com/statistics/871513/worldwide-data-created/`. Statista, 2021.

[33] Bargav Jayaraman and David Evans. “Evaluating Differentially Private Machine Learning in Practice”. In: *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*. Ed. by Nadia Heninger and Patrick Traynor. USENIX Association, 2019, pp. 1895–1912. URL: `https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman`.

[34] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. “PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: `https://openreview.net/forum?id=S1zk9iRqF7`.

[35] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: `http://arxiv.org/abs/1312.6114`.

[36] Honglak Lee et al. “Efficient sparse coding algorithms”. In: *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*. Ed. by Bernhard Schölkopf, John C. Platt, and Thomas Hofmann. MIT Press, 2006, pp. 801–808. URL: `https://proceedings.neurips.cc/paper/2006/hash/2d71b2ae158c7c5912cc0bbde2bb9d95-Abstract.html`.

[37] Ninghui Li et al. *Differential Privacy: From Theory to Practice*. Synthesis Lectures on Information Security, Privacy, & Trust. Morgan & Claypool Publishers, 2016. ISBN: 978-3-031-01222-8. DOI: 10.2200/S00735ED1V01Y201609SPT018. URL: https://doi.org/10.2200/S00735ED1V01Y201609SPT018.

[38] Isabel Maria Lopes and Pedro Oliveira. “Implementation of the general data protection regulation: A survey in health clinics”. In: *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*. 2018, pp. 1–6. DOI: 10.23919/CISTI.2018.8399156.

[39] John C. McCallum. *Disk Drive Prices 1955+*. 2023. URL: https://jcmit.net/diskprice.htm.

[40] Frank McSherry and Kunal Talwar. “Mechanism Design via Differential Privacy”. In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*. IEEE Computer Society, 2007, pp. 94–103. DOI: 10.1109/FOCS.2007.41. URL: https://doi.org/10.1109/FOCS.2007.41.

[41] Ilya Mironov. “Rényi Differential Privacy”. In: *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*. IEEE Computer Society, 2017, pp. 263–275. DOI: 10.1109/CSF.2017.11. URL: https://doi.org/10.1109/CSF.2017.11.

[42] Sharyl Nass, Laura Levit, and Lawrence Gostin. *Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research*. Feb. 2009. ISBN: 978-0-309-12499-7. DOI: 10.17226/12458.

[43] NVIDIA. “NVIDIA Tensorflow”. In: (2023). URL: https://github.com/NVIDIA/tensorflow.

[44] Nicolas Papernot et al. “Scalable Private Learning with PATE”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rkZB1XbRZ.

[45] Nicolas Papernot et al. “Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=HkwoSDPgg.

[46] Deepak Pathak et al. “Context Encoders: Feature Learning by Inpainting”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 2536–2544. DOI: 10.1109/CVPR.2016.278. URL: https://doi.org/10.1109/CVPR.2016.278.

[47] NhatHai Phan et al. “Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction”. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. Ed. by Dale Schuurmans and Michael P. Wellman. AAAI Press, 2016, pp. 1309–1316. DOI: 10.1609/AAAI.V30I1.10165. URL: https://doi.org/10.1609/aaai.v30i1.10165.

[48] Natalia Ponomareva et al. “How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy”. In: *J. Artif. Intell. Res.* 77 (2023), pp. 1113–1201. DOI: 10.1613/JAIR.1.14649. URL: https://doi.org/10.1613/jair.1.14649.

[49] Mayu Sakurada and Takehisa Yairi. "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction". In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, Gold Coast, Australia, QLD, Australia, December 2, 2014*. Ed. by Ashfaqur Rahman, Jeremiah D. Deng, and Jiuyong Li. ACM, 2014, p. 4. DOI: `10.1145/2689746.2689747`. URL: `https://doi.org/10.1145/2689746.2689747`.

[50] Uthaipon Tao Tantipongpipat et al. "Differentially private synthetic mixed-type data generation for unsupervised learning". In: *Intell. Decis. Technol.* 15.4 (2021), pp. 779–807. DOI: `10.3233/IDT-210195`. URL: `https://doi.org/10.3233/IDT-210195`.

[51] Amirsina Torfi. "RDPCGAN Github Implementation". In: *Github* (2020). URL: `https://github.com/astorfi/differentially-private-cgan`.

[52] Amirsina Torfi, Edward A. Fox, and Chandan K. Reddy. "Differentially private synthetic medical data generation using convolutional GANs". In: *Inf. Sci.* 586 (2022), pp. 485–500. DOI: `10.1016/J.INS.2021.12.018`. URL: `https://doi.org/10.1016/j.ins.2021.12.018`.

[53] Svetlana Ulianova. "Cardiovascular Disease Dataset". In: *www.kaggle.com* (2019).

[54] V.Crha. "GANAD - Improving medical data synthesis with DP-GAN and Deep Anomaly Detection". In: (2024). URL: `https://github.com/VojtechCrha`.

[55] vanderschaarlab. "PATEGAN Github Implementation". In: *ML For HealthLab Publications* (2021). URL: `https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/pategan`.

[56] Pascal Vincent et al. "Extracting and composing robust features with denoising autoencoders". In: *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*. Ed. by William W. Cohen, Andrew McCallum, and Sam T. Roweis. Vol. 307. ACM International Conference Proceeding Series. ACM, 2008, pp. 1096–1103. DOI: `10.1145/1390156.1390294`. URL: `https://doi.org/10.1145/1390156.1390294`.

[57] Liyang Xie et al. "Differentially Private Generative Adversarial Network". In: *CoRR* abs/1802.06739 (2018). arXiv: `1802.06739`. URL: `http://arxiv.org/abs/1802.06739`.

[58] Xinyang Zhang, Shouling Ji, and Ting Wang. "Differentially Private Releasing via Deep Generative Model". In: *CoRR* abs/1801.01594 (2018). arXiv: `1801.01594`. URL: `http://arxiv.org/abs/1801.01594`.

[59] Chong Zhou and Randy C. Paffenroth. "Anomaly Detection with Robust Deep Autoencoders". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 2017, pp. 665–674. DOI: `10.1145/3097983.3098052`. URL: `https://doi.org/10.1145/3097983.3098052`.

[60] David Zimmerer et al. "Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection". In: *CoRR* abs/1812.05941 (2018). arXiv: `1812.05941`. URL: `http://arxiv.org/abs/1812.05941`.

[61] Andrzej Zolnierek and Bartlomiej Rubacha. "The Empirical Study of the Naive Bayes Classifier in the Case of Markov Chain Recognition Task". In: *Computer Recognition Systems, Proceedings of the 4th International Conference on Computer Recognition Systems, CORES'05, May 22-25, 2005, Rydzyna Castle, Poland*. Ed. by Marek Kurzynski et al. Vol. 30. Advances in Soft Computing. Springer, 2005, pp. 329–336. DOI: `10.1007/3-540-32390-2\_38`. URL: `https://doi.org/10.1007/3-540-32390-2%5C_38`.

[62] Bo Zong et al. “Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: `https://openreview.net/forum?id=BJJLHbb0-`.

A

Detailed results

PATEGAN

Table A.1: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - PATEGAN UCI-Epileptic.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	0.56 ± 0.001	0.426 ± 0.001	**0.573** ± **0.001**	**0.436** ± **0.001**	**0.553** ± **0.001**	0.414 ± 0.001
Pre-Generation	0.544 ± 0.002	0.412 ± 0.002	0.539 ± 0.004	0.405 ± 0.004	0.543 ± 0.001	0.405 ± 0.001
Mid-Generation	0.550 ± 0.003	0.415 ± 0.002	0.538 ± 0.002	0.397 ± 0.002	**0.553** ± **0.001**	0.407 ± 0.002
Post-Generation	**0.574** ± **0.002**	**0.443** ± **0.002**	0.560 ± 0.004	0.425 ± 0.003	0.552 ± 0.001	**0.422** ± **0.001**

Table A.2: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - PATEGAN Cardiovascular Disease.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	**0.575** ± **0.001**	**0.568** ± **0.001**	0.597 ± 0.001	0.582 ± 0.001	0.627 ± 0.000	0.598 ± 0.001
Pre-Generation	0.557 ± 0.000	0.552 ± 0.000	0.586 ± 0.000	0.568 ± 0.000	**0.650** ± **0.000**	0.583 ± 0.000
Mid-Generation	0.559 ± 0.001	0.557 ± 0.001	0.597 ± 0.001	0.577 ± 0.000	0.637 ± 0.001	**0.608** ± **0.001**
Post-Generation	0.558 ± 0.001	0.561 ± 0.001	**0.599** ± **0.001**	**0.583** ± **0.001**	0.647 ± 0.001	0.604 ± 0.001

Table A.3: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - PATEGAN Cervical Cancer.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	**0.632** ± **0.005**	0.203 ± 0.009	**0.654** ± **0.004**	**0.212** ± **0.006**	0.686 ± 0.004	0.196 ± 0.004
Pre-Generation	0.599 ± 0.014	**0.224** ± **0.009**	0.621 ± 0.008	0.167 ± 0.003	0.693 ± 0.008	0.227 ± 0.011
Mid-Generation	0.594 ± 0.009	0.178 ± 0.006	0.636 ± 0.005	0.195 ± 0.004	0.672 ± 0.004	0.225 ± 0.007
Post-Generation	0.605 ± 0.009	0.183 ± 0.006	0.642 ± 0.009	0.186 ± 0.008	**0.697** ± **0.002**	**0.232** ± **0.005**

RDPCGAN

Table A.4: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - RDPCGAN UCI-Epileptic.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	0.402 ± 0.001	0.237 ± 0.001	0.409 ± 0.002	0.249 ± 0.001	0.465 ± 0.018	0.292 ± 0.011
Pre-Generation	**0.519 ± 0.006**	**0.334 ± 0.005**	0.403 ± 0.001	0.238 ± 0.000	**0.519 ± 0.014**	**0.339 ± 0.010**
Mid-Generation	0.493 ± 0.007	0.310 ± 0.004	0.427 ± 0.003	0.253 ± 0.002	0.431 ± 0.006	0.256 ± 0.002
Post-Generation	0.454 ± 0.002	0.284 ± 0.001	**0.476 ± 0.008**	**0.294 ± 0.005**	0.456 ± 0.014	0.282 ± 0.009

Table A.5: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - RDPCGAN Cardiovascular Disease.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	0.548 ± 0.002	0.547 ± 0.001	0.539 ± 0.001	0.537 ± 0.001	0.555 ± 0.001	0.543 ± 0.001
Pre-Generation	**0.563 ± 0.001**	**0.558 ± 0.001**	**0.545 ± 0.001**	**0.544 ± 0.001**	**0.559 ± 0.000**	**0.551 ± 0.000**
Mid-Generation	0.553 ± 0.003	0.547 ± 0.002	0.536 ± 0.001	0.530 ± 0.000	0.523 ± 0.002	0.528 ± 0.001
Post-Generation	0.555 ± 0.001	0.548 ± 0.001	**0.545 ± 0.001**	0.541 ± 0.001	0.553 ± 0.000	0.549 ± 0.000

Table A.6: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - RDPCGAN Cervical Cancer.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	0.544 ± 0.004	0.075 ± 0.001	**0.659 ± 0.001**	0.135 ± 0.002	0.602 ± 0.012	0.167 ± 0.019
Pre-Generation	**0.560 ± 0.010**	0.091 ± 0.002	0.638 ± 0.002	0.113 ± 0.001	**0.731 ± 0.008**	**0.295 ± 0.014**
Mid-Generation	0.556 ± 0.006	**0.120 ± 0.001**	0.618 ± 0.001	0.103 ± 0.001	0.695 ± 0.012	0.282 ± 0.019
Post-Generation	0.555 ± 0.001	0.095 ± 0.000	0.636 ± 0.004	**0.140 ± 0.003**	0.580 ± 0.016	0.150 ± 0.012

DPGAN

Table A.7: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - DPGAN UCI-Epileptic.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	0.503 ± 0.001	0.358 ± 0.001	**0.512 ± 0.001**	0.357 ± 0.001	**0.499 ± 0.000**	0.337 ± 0.000
Pre-Generation	**0.525 ± 0.001**	**0.371 ± 0.001**	**0.512 ± 0.001**	**0.360 ± 0.001**	0.483 ± 0.001	0.329 ± 0.001
Mid-Generation	0.508 ± 0.002	0.360 ± 0.001	0.498 ± 0.001	0.351 ± 0.001	0.491 ± 0.001	**0.338 ± 0.001**
Post-Generation	0.507 ± 0.001	0.350 ± 0.001	0.498 ± 0.002	0.352 ± 0.001	0.483 ± 0.001	0.326 ± 0.001

Table A.8: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - DPGAN Cardiovascular Disease.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	0.483 ± 0.001	0.495 ± 0.000	0.505 ± 0.001	0.514 ± 0.001	0.514 ± 0.001	0.520 ± 0.001
Pre-Generation	**0.508 ± 0.001**	**0.516 ± 0.001**	**0.516 ± 0.002**	**0.523 ± 0.001**	0.489 ± 0.002	0.501 ± 0.001
Mid-Generation	0.506 ± 0.002	0.514 ± 0.001	0.506 ± 0.001	0.515 ± 0.001	0.503 ± 0.001	0.513 ± 0.000
Post-Generation	0.492 ± 0.001	0.505 ± 0.001	0.493 ± 0.001	0.504 ± 0.000	**0.515 ± 0.001**	**0.524 ± 0.000**

Table A.9: ML Utility comparison of different AR methods using AUROC & AUPRC metric with varying epsilon values - DPGAN Cervical Cancer.

AR Methods/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Baseline	0.542 ± 0.016	**0.167** ± **0.009**	0.428 ± 0.018	0.106 ± 0.004	**0.515** ± **0.012**	**0.141** ± **0.007**
Pre-Generation	0.505 ± 0.008	0.150 ± 0.005	0.509 ± 0.004	0.127 ± 0.001	0.488 ± 0.014	0.129 ± 0.003
Mid-Generation	**0.544** ± **0.008**	0.137 ± 0.002	**0.515** ± **0.007**	0.141 ± 0.006	0.498 ± 0.013	0.135 ± 0.002
Post-Generation	0.446 ± 0.007	0.097 ± 0.002	0.507 ± 0.021	**0.142** ± **0.004**	0.476 ± 0.026	0.137 ± 0.005

Pre-Generation Anomaly Removal

Table A.10: The difference between the AUROC & AUPRC score of baseline and Pre-Generation AR - averaged across models for each dataset

Dataset/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
UCI Epileptic	**0.040** ± **0.005**	**0.032** ± **0.003**	-0.013 ± 0.000	-0.013 ± 0.000	0.009 ± 0.002	0.010 ± 0.001
Cervical Cancer	-0.018 ± 0.001	0.007 ± 0.000	**0.009** ± **0.004**	-0.015 ± 0.001	**0.036** ± **0.007**	**0.049** ± **0.005**
Cardio Vascular	0.007 ± 0.001	0.005 ± 0.000	0.002 ± 0.000	**0.001** ± **0.000**	0.001 ± 0.001	-0.009 ± 0.000

Mid-Generation Anomaly Removal

Table A.11: The difference between the AUROC & AUPRC score of baseline and Mid-Generation AR - averaged across models for each dataset

Dataset/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
UCI Epileptic	**0.028** ± **0.003**	**0.021** ± **0.002**	-0.010 ± 0.001	-0.013 ± 0.000	-0.014 ± 0.000	-0.014 ± 0.000
Cervical Cancer	-0.008 ± 0.001	-0.003 ± 0.002	**0.009** ± **0.005**	-0.005 ± 0.001	**0.021** ± **0.004**	**0.046** ± **0.004**
Cardio Vascular	0.004 ± 0.000	0.003 ± 0.000	-0.001 ± 0.000	**-0.003** ± **0.000**	-0.011 ± 0.000	-0.004 ± 0.000

Post-Generation Anomaly Removal

Table A.12: The difference between the AUROC & AUPRC score of baseline and Post-Generation AR - averaged across models for each dataset

Dataset/Epsilon	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
UCI Epileptic	**0.022** ± **0.001**	**0.019** ± **0.001**	0.013 ± 0.002	**0.010** ± **0.001**	-0.009 ± 0.000	-0.005 ± 0.000
Cervical Cancer	-0.037 ± 0.003	-0.023 ± 0.002	**0.015** ± **0.003**	0.005 ± 0.001	-0.017 ± 0.001	**0.005** ± **0.001**
Cardio Vascular	0.000 ± 0.000	0.001 ± 0.000	-0.001 ± 0.000	-0.002 ± 0.000	**0.007** ± **0.000**	**0.005** ± **0.000**