Bachelorarbeit am Institut für Informatik der Freien Universität Berlin,
Arbeitsgruppe ID Management

Privacy preserving synthetic data generation

William Gu

willigu@fu-berlin.de
Matrikelnummer: 4765674
Betreuerin: Franziska Boenisch
1. Gutachter: Prof. Dr. Marian Margraf
2. Gutachter: Prof. Dr. Eirini Ntoutsi

Berlin, den 22. September 2021

Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 22. September 2021

Gu, William

Contents

1 Introduction and motivation **5**

2 Background **7**
2.1 Theory of differential privacy 7
2.1.1 (ϵ, δ)-differential privacy 8
2.1.2 Rényi differential privacy 10
2.2 Generative methods in machine learning 10
2.2.1 Basics of GANs 11
2.2.2 Common GAN variations 12
2.3 Related works on DP and synthetic data generation 14

3 Implementations of DP in GANs **17**
3.1 DPGAN and DP-CGAN 17
3.2 PATE-GAN 19
3.3 Moment accounting 21

4 Experimental setup **23**
4.1 Methodology 23
4.2 Evaluation metrics 24
4.3 Model structures 25

5 Results **29**
5.1 Performance metrics 29
5.2 Variation of parameters ϵ and C 35
5.3 Memorization and disclosure risks 37

6 Discussion **41**
6.1 Applicability of DP in generative models 41
6.2 GAN training issues 42
6.3 Conclusions 44

7 Appendix **45**

List of Figures **53**

Bibliography **55**

Abstract

Privacy related questions with regards to the increasing usage of machine learning algorithms in real-world applications such as medical analysis have repeatedly come up in recent years, yet the answer to the measurement and applicability of privacy guarantees in these application remains imprecise. With amassing of databases of sensitive data for training purposes and the potential risk to user privacy related to such repeated large-scale collections, generative algorithms such as GANs provide a potential solution to minimizing data harvesting by providing models producing unlimited synthetic data. Differential privacy as a way to illustrate and quantify the ability to share public information about data while simultaneously withholding private information of individuals within such a collection has been gaining traction as a measurement of privacy with training machine learning models.

This work investigates the notion of differential privacy in its potential usability in different GANs trained on tabular medical data of patients.

As such, non-differentially private and differentially private GAN models were trained on two datasets (Pima Indian Diabetes and the UCI heart disease dataset collection) and the resulting generated synthetic data was then used as the basis for an ensemble of classifiers to evaluate their classification performance on.

1 Introduction and motivation

With an ever increasing amount of data available within the past decade and the subsequent rise of data analysis methods in various industries, data privacy has gained renewed attention in all parts of society. Notably, as professionals, researchers and companies in the medical field have been trying to accumulate larger amounts of sensitive information from patients for better treatments of common diseases and offer more accurate or personalized diagnoses, there have been multiple issues arising from this, from ethical and legal problems of data collection to practical issues, such as non-digital patient records, the rarity of certain diseases causing insufficient data or more calls for better protections against privacy breaches that have been happening [1].

The WHO reports that cardiovascular diseases (CVD) are the number one cause of deaths globally, with over four out of five CVD-related deaths due to myocardial infarction (MI) and strokes, many times taking lives prematurely [2]. Analyzing patient data for more timely recognition of high-risk CVD patients resulting in early medical intervention can lead to better outcomes. At-risk-of-CVD patients are not the only group that could benefit from more accurate and earlier predictions; the substantial rise in percentage of overweight or obese populations lead to higher chances of adult on-set diabetes for that group [3]; early accurate recognition of diabetic conditions can help set up routes for lifestyle interventions, which can prevent further medical deterioration [4].

Thus, the importance of gaining more knowledge from available data related to CVD and diabetes collected for researchers, professionals, companies and decision makers stands in contrast to concerns about breach of privacy.

Anonymization of data records, while being a more common practice to provide some privacy, does not avoid the risks of information disclosure due to the ability of an adversary to collate already existing information and link them with different databases [5]. A powerful example of this was demonstrated by researchers, who could reveal private health information in a supposed sanitized database about a person in public office by collating and merging information from health and voter registry databases[6].

Early works [7] by statistician Rubin on issues relating to confidentiality concerns

of publicly collated census data established the concept of using synthetic data instead of those of actual individuals. Using imputation methods, they were able to repopulate missing attribute values, allowing the possibility of easing further over-collection of sensitive data and subsequent potential privacy issues. While there was doubt on the feasibility of it initially, the idea has spread since his publication[8].

With the popularity of machine learning models (ML) in recent years, the generation of synthetic data has garnered new attention. Using modern ML models, it is possible to create complete or partially synthetic data out of public demographic census data [8] and population commuter data for mapping purposes [9], paving the road to ease data scarcity issues due to privacy requirements for researchers in certain areas.

However, synthetic data produced by ML models are not an absolute guarantor of privacy. Deliberate membership inference attacks on models have been shown to produce leaks where an adversary can correctly deduce that a data point was in in the private database that was used to train the model [10]. Memorization issues within training of models are another vector of privacy capture; the prospect of accidental leaks due to memorizing training data is a persistent issue that is difficult to avoid if not paid close attention to [11].

Parallel to the idea of using artificial data, the concept of differential privacy (DP) as a means of introducing quantifiable and a mathematically more rigorous privacy definition into ML models has been rising. The promise to provide another angle beside disclosure limitation methods and a potential solution to adversarial attacks and memorization problems has made the topic gain more attention within the last few years [12][13].

A study on the practical usability of synthetic data generation in combination with differential privacy for generative ML models on real-life patient data might offer insights into how to better balance the needs for more accessible, medically accurate data and the chance to early diagnosis of high-risk CVD and at-risk diabetes patients and minimize the negative implications on an individual's privacy.

2 Background

Differential privacy emerged as a field of interest from two separate, yet interlinked problem fields: confidentiality issues of statistical data as seen in censuses and synthesized data as a means of circumventing the problem as well as a need in the computing field to be able to quantify the blurry concept of *privacy*. The earliest statistical works did not yet include a formalized or quantifiable framework of how much disclosure methods of synthetic data generation would allow [14]. In fact, it has been shown that absolute statistical disclosure prevention is impossible when any useful statistical mechanism and non-trivial notions of disclosure will lead to the availability of auxiliary information for a potential adversary, which would make the output of the mechanism ultimately disclosive [15]. Concurrently to the development within the statistical field, the computer science and cryptography community established a formalized notion of ϵ-differential privacy, which Dwork first introduced in 2006 [12]. Differential privacy (DP) addresses the overall paradox of trying to learn nothing about an individual providing data while simultaneously learning useful information about an entire population providing data, moving the view from the study of absolute disclosure prevention to a more relative approach.

2.1 Theory of differential privacy

An intuitive understanding of the idea behind DP can be summarized as a promise of a data holder, who holds data of all individuals in a database D to an individual data subject: "You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis"[13]. The most commonly cited formal definition of ϵ-DP ultimately stems from Dwork [12]:

Let $\epsilon \in \mathbb{R}^+$, M be a randomized algorithm, S be all the subsets of the output space of M and D, D' be two neighboring datasets, then M is said to be ϵ-DP if for all neighboring datasets

$$Pr(M(D) \in S) \leq \exp(\epsilon) \cdot Pr(M(D') \in S) \tag{2.1}$$

Two datasets D, D' are said to be neighboring if

$$\exists x \in D \text{ such that } D \backslash \{x\} = D' \tag{2.2}$$

holds.

In other words, this definition assures that adding or removing a data point of an individual in a dataset does not significantly change the behavior of the algorithm M and the conclusion that can be drawn from the outcome of a request for information. A practical example to illustrate the desired effect can be cited from [15]: Suppose there is a database full of female patients with data including height and nationality which will be inquired by a health insurance. In the (contrived) case that height would be a deciding factor for qualification for insurance coverage, height data of an individual would be a sensitive piece of information. Suppose there is an inquiry into the database about an individual *Terry Gross*. DP mechanisms guarantee that whether *Terry Gross'* data point is or is not in the database, the outcome of an inquiry will not significantly affect her chances for insurance coverage.

The factor ϵ is often referred to as the *privacy budget*, *privacy guarantee*, *privacy cost* or *privacy loss* which we will use synonymously in this work, and can be seen as a quantifiable measurable amount certain mechanisms are allowed to incur. From 2.1, it is deductible that $\epsilon = 0$ guarantees perfect privacy for the singular individual data point as adding or removing a data point has no influence on the overall plausibility of being either dataset.

In practice however, such strict notions of pure ϵ-DP are very hard to achieve, since adding noise, sometimes even small amounts, might lead to a reduction of the usability of information. Therefore, a more popular and widely used generalized definition has been proposed, where we allow a certain weakening of ϵ-DP.

2.1.1 (ϵ, δ)-differential privacy

Let M be a randomized algorithm, $\epsilon, \delta \in \mathbb{R}^+$, S be all the subsets of the output space of M and D, D' neighboring datasets. M is said to be (ϵ, δ)-differentially private if

$$Pr(M(D) \in S) \leq \exp(\epsilon) \cdot Pr(M(D') \in S) + \delta \tag{2.3}$$

The δ parameter lets us define a space where a certain amount of privacy degradation is acceptable as δ represents the probability of the privacy loss not being bounded by such ϵ. For $\delta = 0$, we get the specific case of ϵ-DP. Heuristically good values are where $\delta < \frac{1}{|D|}$ holds [16].

One of the more notable properties of (ϵ, δ) mechanisms is their robustness towards post-processing methods: Let $M : D \to R$ be an (ϵ, δ)-DP randomized algorithm

and $f : R \to R'$ an arbitrary randomized mapping:

$$f \circ M : D \to R' \text{ is } (\epsilon, \delta)\text{-differentially private} \tag{2.4}$$

This in fact means that an external actor cannot manipulate the output of such a mechanism M to be *less differentially private* without additional information about the private dataset itself [13].

Another notable property is the closedness of (ϵ, δ)-DP mechanisms regarding different composition methods. It is important to ensure that privacy guarantees are not degraded too much under mechanisms which are built out of a combination of simpler privacy mechanisms [13].

Let M_1 be an ϵ_1-differentially private algorithm and M_2 be an ϵ_2-differentially private algorithm. The combination of both, defined as $M_{1,2} = (M_1(x), M_2(x))$ is $\epsilon_1 + \epsilon_2$-differentially private [13].

This basic composition property is generalizable to the relaxed (ϵ, δ) definition as well, leading to the following:

Let M_i be an (ϵ_i, δ_i)-differentially private algorithm for $i \in [k]$.

$$\text{The combination } M_{[k]}(x) = (M_1(x), \ldots, M_k(x)) \text{ is } \left(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i\right)\text{-DP} \tag{2.5}$$

For more complex compositional ideas of chaining different DP mechanisms on the same database or even on different input databases containing information relating to the same data point of an individual as in many real-world applications, the boundary analysis of advanced compositions provide a guarantee that the privacy guarantee is only degrading in limited ways. It has been shown that under such k-fold adaptive compositions, the privacy guarantee can be stated as at least $(\epsilon', k\delta+\delta')$-DP with $\epsilon' = \mathcal{O}(k\epsilon^2 + \epsilon\sqrt{k \log(1/\delta')})$ [17][18][19].

Despite the (ϵ, δ) characterization of differential privacy being very widely used, it has its drawbacks with two specific usage cases: analysis of Gaussian noised mechanisms, which are commonly used and the analysis under repeated usage of advanced compositions. While the privacy costs under Laplace noise-based mechanisms can be described tightly by the (ϵ, δ) definition, choosing a single Gaussian-noised algorithm results in a range of $(\epsilon(\delta), \delta)$ privacy guarantees which it all satisfies. Selecting a single point on this curve would not give a complete picture of the privacy mechanism, making it impossible to provide a straight forward ϵ-DP analysis. Multiple successive applications of such algorithms in composition would lead to a combinatorial explosion of parameters and $(\epsilon(\delta), \delta)$ privacy guarantees [20].

2.1.2 Rényi differential privacy

Considering these points, Mironov [20] [21] has proposed another way to address the shortcomings of the (ϵ, δ) definition.

Let P, Q be two probability distributions over R, the Rényi divergence of order $\alpha > 1$ is defined as

$$\mathcal{D}_\alpha(P||Q) = \frac{1}{\alpha - 1} \log E_{x \sim Q} \left(\frac{P(x)}{Q(x)} \right)^\alpha \tag{2.6}$$

Based on the definition of this divergence, a new definition of differential privacy can be established.

A randomized mechanism $f : D \to R$ is said to have ϵ-Rényi differential privacy of order α or (α, ϵ)-RDP if for any neighboring datasets D, D', the following statement holds:

$$\mathcal{D}_\alpha(f(D)||f(D')) < \epsilon \tag{2.7}$$

Like the ϵ-DP definition, mechanisms having (ϵ, α)-RDP are also robust against post-processing, meaning given any such mechanism M and any arbitrary mapping $f : R \to R'$, $f \circ M$ also has (α, ϵ)-RDP.

For a Gaussian-noised mechanism $G_\sigma M(x) = M(x) + \mathcal{N}(0, \sigma^2)$, RDP analysis provides an easy to use and tight privacy bound, in which G_σ has $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP. Any (α, ϵ)-RDP implies $(\epsilon_\delta, \delta)$-DP for any given $0 < \delta < 1$, making it possible to show a privacy guarantee conversion: If M has (α, ϵ)-RDP, then the mechanism also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-DP for any $0 < \delta < 1$ [20][22].

2.2 Generative methods in machine learning

It's possible to categorize two distinct classes of models within the fields of statistics and machine learning: *Discriminative* models and *generative* models of analyzing and processing data. An intuitive understanding might be that discriminative models are able to predict given an amount of observations, while generative models are learning the joint distribution over all the variables concerned. The generative process has many attractive features for why it is interesting. One of its advantages for example is that it can naturally express more subtle relations between variables and lead to better generalizations than mere correlations. We can test the resulting models on real world data observations and can confirm or reject the theories

and regularities of how we understand the world by having built the model on that understanding [23].

For the purposes of generating synthetic data, there are a variety of options of generative models to produce a desired outcome. Two of the most popular methods recently have been Variational Autoencoders (VAE) and generative adversarial networks (GANs), which we will focus on in this thesis.

2.2.1 Basics of GANs

Generative adversarial networks were popularized by Goodfellow et al., although similar theoretical ideas of adversarial networks have been circulating in the past[25]. The idea of the original GAN (here referred to as *vanilla GAN*) is to train two neural networks in adversity towards each other, one generator G who tries to generate good new samples out of an input vector in a latent space (usually random noise) and one discriminator D, who tries to differentiate between the newly generated samples and the actual real samples that are fed. A GAN is therefore creating a situation

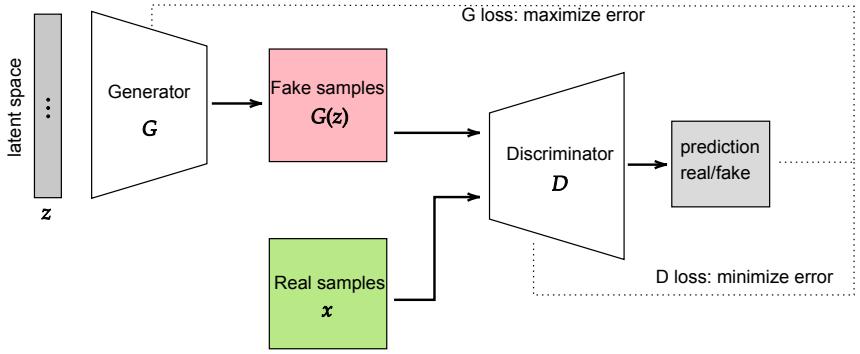


Figure 2.1: The model structure of a vanilla GAN as proposed by Goodfellow et al. with generator G and adversary D as discriminator [24]

comparable to a zero-sum game, where one player (the generator) tries to fool the other player (the discriminator) into not being able to see the difference between real and fake samples as much as possible while the other player tries to avoid being fooled and be able to pick apart what is real and what is not real. An often cited analogy is the dynamic between a forger and a detective being competitive adversaries [24].

The resulting generator will ideally improve within time trying to outcompete the discriminator, enabling us to use it to create synthetic datasets. This game can be

seen as a formalized 2-player minimax game with a value function $V(D, G)$[24]

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}(x)}}(\log D(x)) + \mathbb{E}_{z \sim p_{z(z)}}(1 - \log D(G(z))) \tag{2.8}$$

where $p_{\text{data}(x)}$ is the source data distribution to learn and $p_z(z)$ is a prior probability distribution on noise variables. D and G are both multi-layer perceptrons with parameters θ_d and θ_g respectively. G outputs into source data space while the output of D as a binary classifier is a singular value representing the probability that its input x came from source data (so from the real distribution) and not from the generator distribution p_g. The optimal case therefore is when both distributions as identical

Algorithm 1 Main algorithm for vanilla GAN using SGD. Parameters E training iterations, k discriminator update steps (in this work $k = 1$), b batch size [24]

for $e < E$ **do**
  for k discriminator steps **do**
    Collect b samples $\{z_0, \dots z_b\}$ from random noise distribution $p_z(z)$
    Collect b samples $\{x_0, \dots x_b\}$ from the data source distribution $p_{\text{data}}(x)$
    ▷ Update D weights by ascending its stochastic gradient
    $\nabla\theta_d \frac{1}{m} \sum_{i=1}^{b} [\log D(x_i) + \log(1 - D(G(z_i)))]$
  end for
  Collect b samples $\{z_0, \dots z_b\}$ from random noise distribution $p_z(z)$
  ▷ Update G weights by descending its stochastic gradient
  $\nabla\theta_g \frac{1}{b} \sum_{i=1}^{b} \log(1 - D(G(z_i)))$
end for

and D cannot sufficiently decide which distribution the sample comes from. For a given generator G fixed, the optimal discriminator is

$$D^*_G(x) = \frac{p_{\text{data}}}{p_{\text{data}} + p_g(x)} \tag{2.9}$$

and will always output 0.5 as a probability. Goodfellow et al. shows in the theoretical optimal case, that given enough capacity and D is optimal in each step of the algorithm, then $p_g \to p_{\text{data}}$ will converge.

2.2.2 Common GAN variations

Over the years, there have been many proposed variations on the original vanilla GAN, suited for different tasks. One of the most common problems facing vanilla GANs is their inability to control the modes of data that are being generated. Conditional data generation by using class labels for example are crucial in real-life classification applications where it is desirable to want to generate a large amount of specific classes of points.

Mirza et al. proposed a popular solution to this problem by using the labels y as a secondary input channel into the both the generator network as well as the discriminator network [26].

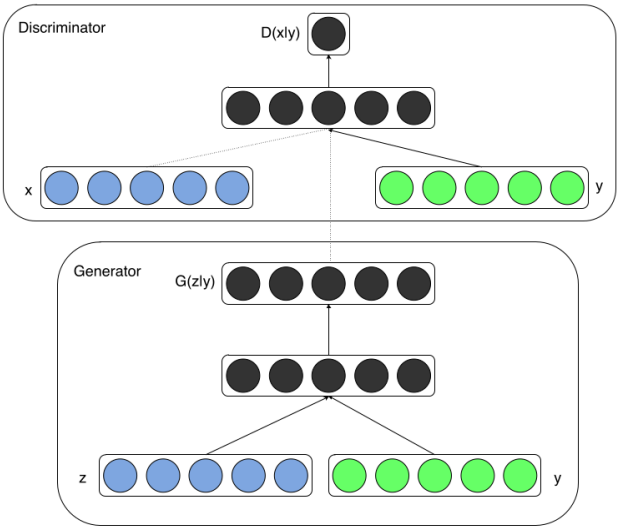


Figure 2.2: Conditional GAN (CGAN) structure proposed in [26]

By joining input noise and labels, the GAN equation changes slightly to

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}(x)}}(\log D(x|y)) + \mathbb{E}_{z\sim p_{\text{z}(z)}}(1 - \log D(G(z|y))) \quad (2.10)$$

Another popular variant of a GAN replaces the basic discriminator with a general critic, where the loss function changes from a binary cross entropy function to the Wasserstein-1 or Earth mover (EM) distance and has been first introduced by Arjovsky et al. [27] as WGAN. The Wasserstein-1 distance $W(p,q)$ is a metric on probability distributions p, q that can be interpreted as measuring the minimum amount of energy needed to moving a mass between one point and another (or 'transforming' one distribution p into another q), hence EM distance [28]. Under an optimal critic, minimizing this distance minimizes the value function of the generator in a WGAN, where the equation is now restated as

$$\min_G \max_{D\in\mathcal{D}} V(D,G) = \mathbb{E}_{x\sim p_{\text{data}(x)}}(D(x)) - \mathbb{E}_{z\sim p_{\text{z}(z)}}(D(G(z))) \quad (2.11)$$

with $\mathcal{D}$ being the set of all 1-Lipschitz continuous functions here 7.

A further improvement on that was a replacement of the clipping by a gradient penalty factor [29]. Gulrajani et al. found that weight clipping alone seemed to be problematic as it encounters either vanishing or exploding gradients. Instead, a slight modification by addition of a penalty factor to the loss function was proposed

$$L = \mathbb{E}_{x\sim p_{\text{z}(z)}}(D(G(z))) - \mathbb{E}_{x\sim p_{\text{data}(x)}}(D(x)) + \lambda \mathbb{E}_{\tilde{x}\sim p_{\tilde{x}(\tilde{x})}}\left[(||\nabla_{\tilde{x}} D(\tilde{x})||_2 - 1)^2\right] \quad (2.12)$$

with $p_{\tilde{x}}$ being defined as the random uniform sampling between points sampled from a straight line from p_z to p_{data} and λ as the penalty coefficient. The various combinations of WGAN with or without gradient penalty with the aforementioned CGAN yields models like WCGAN-GP, which have been used to show improved convergence speed and sample quality to the standard GAN [30][31].

2.3 Related works on DP and synthetic data generation

Besides early works of Rubin introducing concepts such as fully synthesized data records, there have been studies by Little proposing partially synthetic data[32] and systematizing the imputation process more.

A GAN as introduced in 2014 in [24] produces a very useful framework tool for learning features of datasets; this has been commonly applied to standard image datasets like ImageNet, MNIST or CIFAR-10 [33][34] but can also be used with other types of structured data [35][36].

There have been multiple approaches over the last years to include various concepts of DP into GAN, notably the differentially private SGD method in [42] and DP-GAN [37], which both work by introducing Gaussian noise to the gradient in the discriminator network while providing (ϵ, δ)-DP guarantees. DP-CGAN as published in [16] and PATE-GAN [38] using the PATE framework introduced by Papernot et al. [39] provide two alternative approaches, in which both further improve the privacy-loss/data accuracy balance and coming closer to the effectiveness of training on real data. DP-CGAN replaces moment accounting with the concept of RDP accounting to refine the calculation of the privacy budget from DP-GAN while PATE-GAN replaces the classic GAN discriminator with the so-called "Private aggregation of Teachers ensembles (PATE)", which is a framework for effectively doing semi-supervised training tasks using the principle of 'teacher' models training on disjointed, sensitive private information while using a 'student' model to train on a non-sensitive, non-labeled public data set, which has been labeled by the noised aggregation of the 'teacher' models outputs. It thus provides passed on concepts of the private, sensitive data without revealing the source dataset to the 'student' discriminator model.

Choi et al. has proposed medGAN [40] as a framework to using a non-privacy-preserving GAN to generate often difficult to access patient health data from electronic health records (EHR) of existing patients to which Xie et al. has further examined the usefulness of DP-GAN on one of the EHR datasets used in [40] (MIMIC-III) and reports encouraging and usable results while still minimizing privacy loss [37]. A comprehensive study by Goncalves et al. [41] has further shown the usefulness of

various generative methods on EHR from subsets of the data provided by the SEER program from the National Cancer Institute of the NIH in the US and discussed potential disclosure and privacy issues in context.

3 Implementations of DP in GANs

There are many approaches on how to include a differential privacy aspect into GAN generated data. One of the more intuitive ones is to add sampled noise at the end of the generation process, thus taking obfuscation directly to the output data and making them more private. However, this approach often comes at a price of suffering utility loss [37]. The two main approaches that will be looked at in this work mainly deal with including a differentially private aspect directly during the training process:

- The first approach is to include the DP aspect into the training process itself: Adding a small amount of noise (usually sampled from a Gaussian or Laplacian distribution) onto the already clipped gradients. Xie et al. have proposed DPGAN as a way of realizing this approach as have Torkzadehmahani et al. with DP-CGAN.

- The second approach that is being looked at harnessing the PATE *(private aggregation of teacher ensembles)* mechanism as introduced in [39], which guarantees (ϵ, δ)-privacy and trying to include it into GAN training.

3.1 DPGAN and DP-CGAN

Both the DPGAN and the DP-CGAN framework work on adding noise during the optimization step of the network. The post-processing robustness of any (ϵ, δ)-DP mechanism guarantees that any mapping operation after an already differentially private mechanism will also be differentially private, which in the case of GANs is used on the discriminator. Since the optimization calculation of the generator will depend on a differentially private discriminator, it guarantees that the resulting generative network is also differentially private.

In 2, we can see the addition of noise sampled from the normal distribution to the sum of gradients before they are applied.

The tracking of privacy guarantee during the noisy gradient calculations is done using an implementation of the RDP accountant in [16] which is also used as default

Algorithm 2 DP-CGAN algorithm using SGD. Parameters E training iterations, k discriminator update steps (in this work $k = 1$), b batch size, C gradient clip norm, lr learning rate after [16]

for $e < E$ **do**
  for k discriminator steps **do**
    Collect b samples $Z_b \leftarrow \{z_0, \dots z_b\}$ from random noise distribution $p_g(z)$
    Collect b samples $(X_b, Y_b) \leftarrow \{(x_0, y_0), \dots (x_b, y_b)\}$ from the data source distribution $p_{\text{data}}(x)$
    ▷ First calculate the discriminator loss on both real and fake data
    $\text{dloss}_{\text{real}} \leftarrow \log(D(X_b, Y_b))$
    $\text{dloss}_{\text{fake}} \leftarrow \log(1 - D(G(Z_b), Y_b))$
    ▷ Calculate and clip gradients
    $\text{grad}_{\text{real}} \leftarrow \nabla_{\theta_d} \text{dloss}_{\text{real}}(\theta_d, X_b)$
    $\text{grad}_{\text{fake}} \leftarrow \nabla_{\theta_d} \text{dloss}_{\text{fake}}(\theta_d, Z_b)$
    $\text{grad}_{\text{real}} \leftarrow \frac{\text{grad}_{\text{real}}}{\max(1, \frac{||\text{grad}_{\text{real}}||}{C})}$
    $\text{grad}_{\text{fake}} \leftarrow \frac{\text{grad}_{\text{fake}}}{\max(1, \frac{||\text{grad}_{\text{fake}}||}{C})}$
    ▷ Calculate total noisy gradient by adding Gaussian noise to sum
    $\text{grads} \leftarrow \theta_d \frac{1}{b} \sum_{i=1}^{b} [\text{grad}_{\text{real}} + \text{grad}_{\text{fake}}] + \mathcal{N}(0, \sigma^2 C^2 I)$
    ▷ Update D weights by applying gradients and taking the gradient descent
    $\theta_d \leftarrow \text{SGD}(\text{grads}, \text{lr})$
    ▷ Keep track of privacy costs
  end for
  Collect b samples $\{z_0, \dots z_b\}$ from random noise distribution $p_g(z)$
  ▷ Update G weights by descending its stochastic gradient
  $\nabla \theta_g \frac{1}{b} \sum_{i=1}^{b} \log(1 - D(G(z_i)))$
end for

in the TF privacy framework used in this work. The RDP accountant is based on the workings of the moment accounting method [37][21].

3.2 PATE-GAN

Since training models on sensitive data always poses a certain kind of privacy risk, the idea behind PATE is to separate sensitive and non-sensitive training.

The PATE mechanism uses the general principle that if a collection (*ensemble*) of independently trained, non-overlapping ML models (called *teachers*) collectively agree on a similar outcome, then training a secondary *student* model who is only fed non-sensitive data with labels will minimize the risk of memorization and therefore privacy leakage issues with training directly on the source.

The input fed into the student model is usually a public, unlabeled part of a dataset. All the teacher models are then queried for how they would classify the unlabeled set and their votes aggregated, where then the top-most prediction would be the one with the highest tally and ultimate dictate what the student model gets to see as the final input.

Let $f = \{f_i\}, i \in N$ be the ensemble of N teacher models, $[m]$ the amount of class labels, $j \in [m]$ a label of a given class and $\vec{x}$ an input:

The label tally at the end is the number of teacher models that assigned j to $\vec{x}$:

$$n_j(\vec{x}) = |i : i \in N, f_i(\vec{x}) = j|$$

Since simple majority aggregation could lead to a situation where the top-most choice depends on a single teachers voting input, PATE adds noise during aggregation to add ambiguity[39]:

$$f(x) = \arg\max_j \left\{ n_j(\vec{x}) + \text{Lap}\left(\frac{1}{\lambda}\right) \right\} \tag{3.1}$$

where $\text{Lap}(b)$ is the Laplace distribution on location 0 with scale b. The parameter λ is controlling how much noise is added, in turn guaranteeing privacy.

$$\text{A single query to the PATE mechanism is } \left(\frac{1}{\lambda}, 0\right)\text{-differentially private} \tag{3.2}$$

PATE-GAN in the proposal work [38] is a type of GAN which replaces the standard discriminator in a vanilla GAN with the PATE mechanism, effectively creating a gap

of which discriminator model backpropagates to the generator. Instead of a single discriminator, the network now has N teacher discriminators. Combined with the generator, this part behaves like a regular vanilla GAN. Each teacher only sees a disjoint subset of size $|D_i| = \frac{|D|}{N}$ of the original dataset as input, making them train on independent datasets.

Algorithm 3 PATE-GAN Algorithm with parameters k number of teachers, λ noise, n_T, n_S number of training steps for teacher/student, b batch size [38]

Partition dataset into k subsets $D_1 \dots D_k$ of size $\frac{|D|}{k}$
$\forall l : \alpha(l) = 0$
for $e < E$ **do**
  if $\epsilon < \epsilon_t$ **then**
    for $t_1 = 1 \dots n_T$ **do**
      for $i = 1 \dots k$ **do**
        Collect b samples $Z_b \leftarrow \{z_0, \dots z_b\}$ from random noise distribution $p_g(z)$
        Collect b samples $(X_b, Y_b) \leftarrow \{(x_0, y_0), \dots (x_b, y_b)\}$ from disjoint set D_j
        ▷ Update teacher T_i using SGD
        $\nabla_{\theta_T^i} - \Big[\sum_{j=1}^{b} \log(T_i(x_j, y_j)) + \log(1 - T_i(G(z_j)))\Big]$
      end for
    end for
    for $t_2 = 1 \dots n_S$ **do**
      Collect b samples $Z_b \leftarrow \{z_0, \dots z_b\}$ from random noise distribution $p_g(z)$
      for $j = 1 \dots b$ **do**
        $u_j \leftarrow G(Z_b)$
        $r_j \leftarrow \text{pate}(u_j, \lambda)$
        $q \leftarrow \frac{2+\lambda|n_0 - n_1|}{4\exp(\lambda|n_0 - n_1|)}$ ▷ Update moments accountant
        for $l = 1 \dots L$ **do**
          $\alpha(l) \leftarrow \alpha(l) + \min(2\lambda^2 l(l+1), \log((1-q)(\frac{1-q}{1-e^{2\lambda}q})^l + qe^{2\lambda l}))$
        end for
      end for
      ▷ Update student using SGD
      $\nabla_{\theta_S} - \sum_{j=1}^{b} \Big[\log(S(u_j)) + (1 - r_j)\log(1 - S(u_j))\Big]$
    end for
    Collect b samples $Z_b \leftarrow \{z_0, \dots z_b\}$ from random noise distribution $p_g(z)$
    $\nabla_{\theta_G} \sum_{i=1}^{b} \Big[\log(1 - S(G(u_i)))\Big]$ ▷ Update generator G using SGD
    $\epsilon \leftarrow \min \frac{\alpha(l) + \log(\frac{1}{\delta})}{l}$
  end if
end for

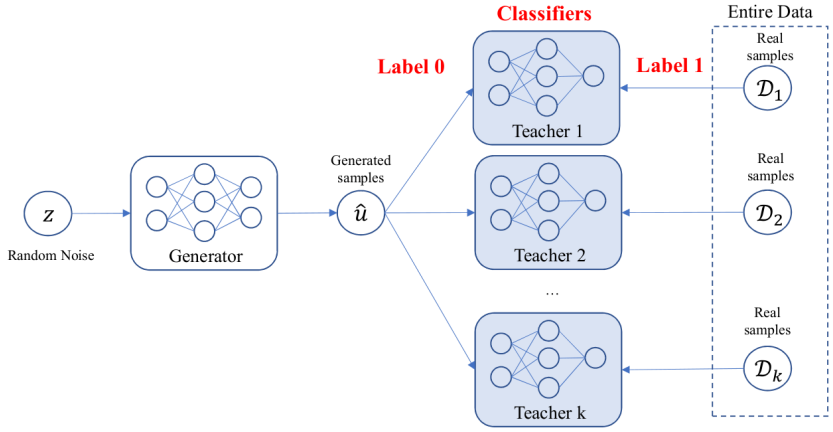


Figure 3.1: Ensemble of teacher part of PATE-GAN with a fixed generator G

PATE now adds another discriminator, the student, which tries to mimic the behavior of the ensemble of teachers by training on non-sensitive data. But since GANs are generative by definition, the student discriminator usually has no further publicly unlabeled data available as input, so a solution proposed was to directly use the generator content as input, querying the ensemble of teacher to label them as either real/unreal and then use the query result to determine the loss of the generator. As Jordon et al. have shown, PATE-GAN as a whole is indeed (ϵ, δ)-DP [38].

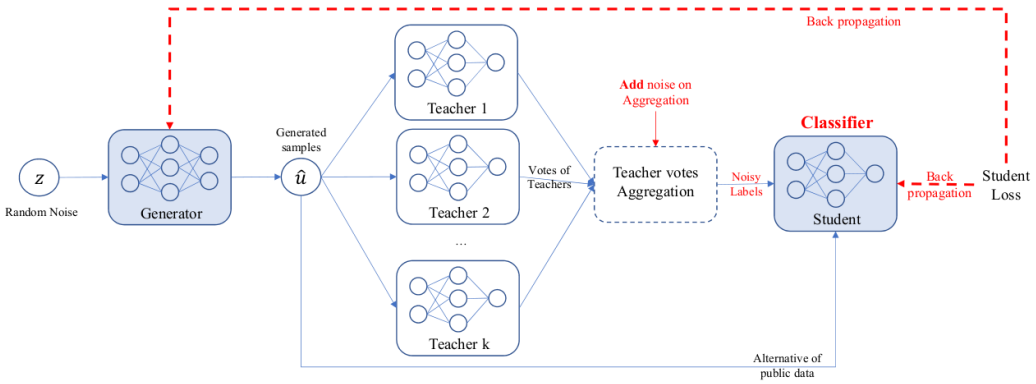


Figure 3.2: Overall structure of PATE-GAN[38]

3.3 Moment accounting

The moment accounting method introduced in [42] uses the composition property of DP to compute the overall privacy loss by calculating the loss in each iteration

and accumulating it like an accountant. Abadi et al. states that privacy loss is a random variable depending on the noise added and a mechanism M being (ϵ, δ)-DP is equivalent to a certain tail bound on the privacy loss random variable of M. For neighboring databases D, D', an auxiliary input a and an outcome $o \in R$, the privacy loss at o is defined as

$$c(o, M, a, D, D') = \log \frac{P(M(a, D) = o)}{P(M(a, D') = o)} \tag{3.3}$$

This privacy loss itself is a random variable, for which we can compute the log moments:

$$\alpha_M(\lambda, a, D, D') = \log \mathbb{E}_{o \sim M(a,D)}(\exp(\lambda c(o, M, a, D, D'))) \tag{3.4}$$

$\alpha_M(\lambda, a, D, D')$ here is called the λ^{th} moment, in which we compute the log of the moment generating function evaluated at λ. For $\lambda = 1$ this is just the expectation value.

For the overall privacy guarantee of M, we define a bound

$$\alpha_M(\lambda) = \max_{a,D,D'} \alpha_M(\lambda, a, D, D') \tag{3.5}$$

[39] has shown that in the case of the PATE mechanism, we are also able to bound the privacy loss of a single step to more specific values, thus using these for the λ^{th} moments.

In the practical implementation of $\alpha(\lambda)$, numerical integration was used in the implementation within TF as well as in PATE-GAN. The order of moments within the TF privacy framework defaults to the list `orders = ([1.25, 1.5, 1.75, 2., 2.25, 2.5, 3., 3.5, 4., 4.5] + list(range(5, 64)) + [128, 256, 512])`[1] for the Gaussian mechanism and we have used an ordered list of moments until 32 for PATE-GAN as was suggested in [42] as well as moments up to 8 as by [39].

[1] `github.com/tensorflow/privacy/blob/master/tensorflow_privacy/privacy/analysis/compute_noise_from_budget_lib.py` line 49ff

4 Experimental setup

4.1 Methodology

For the experimental setup [1], two publicly available datasets in particular were of importance:

- the Pima Indians diabetes (PID) dataset, a collection of data points of female-only patients of Pima Indian heritage originally published by the National Institute of Diabetes and Digestive and Kidney Diseases. The dataset has 8 featural attributes with an additional binary predictable label telling if a particular patient was indeed considered diabetic.[2]

- the Heart Disease (HD) dataset published on the UCI Machine learning webpage. This dataset combines patient data from four clinics across three countries related to heart disease diagnostics. Although the original dataset contains a large amount of attributes, published papers and sources only refer to 14 specific attributes.[3]

A significant challenge both datasets presented are not just that the amount of data for training was relatively small but also missing or incomplete data points. Missing values were either inferred if possible (missing values replaced by the mean average of the relevant attribute) or if a data point/feature was heavily incomplete, it was taken out of the dataset entirely. For the HD dataset, the prediction attribute *(num)* was modified into a binary form, where any value above 1 indicated a 50%+ narrowing of a major vessel, thus indicating a serious heart disease.

Since the HD dataset includes multiple categorical integer attributes, it was necessary to one-hot encode these specific features. All datasets were then scaled to a $(0, 1)$ value space and later reverse scaled back for consumption output; the prepared source datasets had final sizes of $n_{\text{PID}} = 768$ and $n_{\text{HD}} = 920$. Generated datasets were significantly larger than the source datasets, with $n = 3000$ fixed for synthetic data.

[1] Code to this work at `https://git.imp.fu-berlin.de/private_secure_ml/gu-synthetic-data`
[2] `https://www.kaggle.com/uciml/pima-indians-diabetes-database`
[3] `https://archive.ics.uci.edu/ml/datasets/heart+disease`

To assess the viability of differentially private data generation, different datasets generated by training different models on each base dataset were compared against each other by their classification performance on a collection of fixed classifiers.

4.2 Evaluation metrics

The baseline classification on the modified, cleaned-up datasets was used as a general reference point for performing on publicly available limited data points. The performance of datasets generated by non-DP GANs were then compared to their differentially private counterparts on their usability by training a range of binary classifiers and measuring the Area under the Receiver Operating Characteristics (AUROC) curve of the trained models as well as the accuracy and the F1 score. The AUROC score is a way to measure the ability of a binary classifier to properly distinguish and classify samples and was first looked at as a metric for machine learning algorithms in the mid 90s [43].

The AUC is then often calculated by the trapezoidal rule as follows

$$\mathrm{AUC} = \sum_i (1 - \beta_i \Delta\alpha) + \frac{1}{2}(\Delta(1-\beta)\Delta\alpha) \tag{4.1}$$

where $\Delta\alpha, \Delta(1-\beta)$ are defined as follows:

$$\Delta\alpha = \alpha - \alpha_{i-1} \qquad \Delta(1-\beta) = (1-\beta_i) - (1-\beta_{i-1})$$

α and β here are the rates of false-positives $P(F_p)$ or *fpr* and true-positives $P(T_p)$ or *tpr*, usually acquired by putting the elements in a confusion matrix into relation with each other.[4]

$$\begin{array}{lcc} & \text{Predicted positive} & \text{Predicted negative} \\ \text{Ground truth positive} & T_p & F_n \\ \text{Ground truth negative} & F_p & T_n \end{array}$$

$$P(T_p) = \frac{T_p}{T_p + F_n} \quad P(F_p) = \frac{F_p}{F_p + T_n} \quad \mathrm{PPV}^5 = \frac{T_P}{T_p + F_p}$$

The F score (also named the *Dice coefficient score*) is capturing a way of measuring the quality of the model with a single number composed of recall and precision.

$$F_\alpha = (1+\alpha)^2 \frac{\mathrm{PPV} \cdot P(T_p)}{\alpha \mathrm{PPV} + P(T_p)} \tag{4.2}$$

[4] Synonyms of TPR include: Sensitivity, recall or hit rate, FPR is also known as a type I error or 'false alarm' or fall-out

[5] PPV is also called precision

In its most common usage, $\alpha = 1$, the F_1 score is the equally weighted harmonic mean between the two scores.

The accuracy metric in a classifier captures how well the model correctly predicts the outcome

$$\text{acc} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{4.3}$$

In this work, all technical implementations were done in Python 3 with TensorFlow (v2.5.0) and the TF privacy library in its current version (v0.5.1) as well as the numerical libraries of sklearn, numpy and visualization libraries seaborn and matplotlib.

4.3 Model structures

The standard vanilla GAN is not capable of generating label-conditional data points, thus a conditional GAN (CGAN), an established derivation of the standard GAN was used to generate the non-DP synthetic data. A CGAN uses not just the attributes to train on, but also the labels as input both for the generator and the discriminator respectively.

Hyperparameter	PID		HD	
	Value CGAN	Value DP-CGAN	Value CGAN	Value DPCGAN
Generator learning rate	$2E^{-4}$			
Discriminator learning rate	$2E^{-3}$		$1E^{-4}$	
Latent dimension	64		128	
Batch size	32		40	
Micro batch size	-	8	-	10
Training epochs	400	400	200	200
$\beta_{1,\text{d}}$	0.5	0.5	0.1	0.1
$\beta_{1,\text{g}}$	0.5			
β_2	0.99			

Figure 4.1: Hyperparameter used for training CGAN and DP-CGAN

The differentially private equivalent was modelled after the architecture and algorithm for DP-CGAN shown in Torkzadehmahani et al.[16]. Slight changes to fit the datasets and training situation in this work were made (i.e. changing the differentially private optimizer from a vanilla SGD with added noise and clipping to the DP equivalent of the Adam algorithm with parameters listed in 4.1). The background tracking of spent privacy within the TF privacy framework is done using the RDP

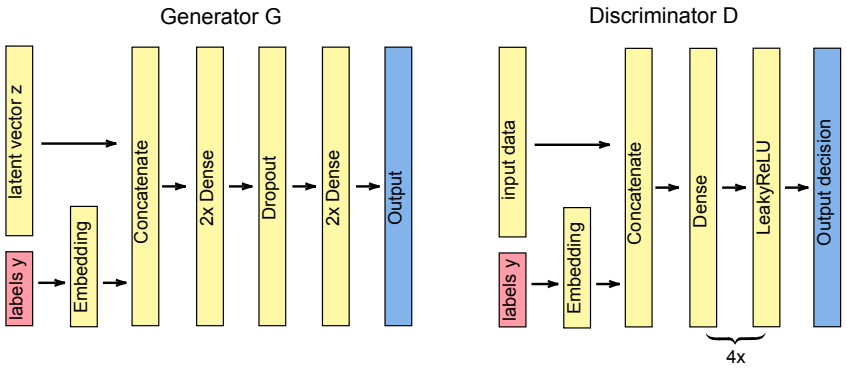


Figure 4.2: Schematic model architecture for our CGAN and DP-CGAN after the design in [16]

Generator layers		
	Layer type	parameters
	Dense	$2^7 + 2N_{\text{features}}$, ReLU
	Dense	$2^6 + 2N_{\text{features}}$, ReLU
	Dropout	rate= 0.2
	Dense	$2^5 + 2N_{\text{features}}$, ReLU
	Dense	$2^4 + 2N_{\text{features}}$, ReLU
	Output	N_{features}
Discriminator layers		
	Layer type	parameters
i=7,6,5,4	Dense	$2^i + 2N_{\text{features}}$, ReLU
	LeakyReLU	$\alpha = 0.2$
	Output	size=1

Figure 4.3: Parameters for the layers in the CGAN and DP-CGAN models

accountant by the more tight RDP estimation of privacy and then converted back to the conventional (ϵ, δ)-DP equivalent values later.

The PATE-GAN model uses the same model layers as the CGAN and DP-CGAN but with copies of D as teachers as well as one D as the student discriminator and with parameters outlined in 4.4.

To evaluate the quality of the datasets, we used an ensemble of six binary classifying models (Linear SVM, RBF SVM, MLP Classifier, AdaBoost, Naive Bayes classifier and quadratic discriminant analysis (QDA)) to train on the datasets and evaluating their performance on the ability to correctly predict the labels and the corresponding reliability of their predictions.

Hyperparameter	PID	HD
Generator learning rate	$1E^{-4}$, $1E^{-5}$	$1E^{-4}$
Teacher$_i$ learning rate	$1E^{-4}$,$1E^{-5}$	$1E^{-4}$
Student learning rate	$1E^{-4}$,$1E^{-5}$	$2E^{-4}$
Number of teachers	12	20
Latent dimension	64	128
Batch size	32	40
Teacher batch size	32	40
Training epochs	100, 200	100
Teacher training epochs	1	
Student training epochs	1	
$\beta_{1,\mathrm{s}}$	0.9	0.9
$\beta_{1,\mathrm{t}}$	0.6	0.9
$\beta_{1,\mathrm{g}}$	0.9	0.9
β_2	0.99	

Figure 4.4: Hyperparameter used for training PATE-GAN

The models were trained using a 5-fold cross validation split and the prediction performance was measured for two different settings:

- Setting A: Training on a portion of the generated dataset, testing on another portion of the dataset
- Setting B: Training on a portion of the generated dataset, testing on the original dataset

A baseline performance was established by running the ensemble on the original PID and HD datasets. We then conducted a series of runs using this method and varying the privacy loss ϵ as well as the clipping norm value C for both datasets.

5 Results

5.1 Performance metrics

Dataset	Source	AUROC	Setting A		Setting B	
			F1 score	Accuracy	F1 score	Accuracy
PID based	original	0.81±0.02	-	-	0.63±0.03	0.75±0.02
	CGAN	0.91±0.01	0.84±0.02	0.84±0.02	0.68±0.01	0.74±0.01
	$(50, 10^{-4})$ DP-CGAN	0.88±0.02	0.79±0.03	0.79±0.03	0.67±0.02	0.74±0.02
	$(10, 10^{-4})$ DP-CGAN	0.92±0.02	0.84±0.02	0.84±0.01	0.67±0.01	0.74±0.01
	$(1, 10^{-4})$ DP-CGAN	0.88±0.02	0.77±0.02	0.80±0.02	0.67±0.01	0.74±0
	$(0.1, 10^{-4})$ DP-CGAN	0.86±0.02	0.79±0.03	0.79±0.02	0.66±0.02	0.73±0.01
HD based	original	0.87±0.01	-	-	0.82±0.02	0.79±0.01
	CGAN	0.95±0.01	0.86±0.02	0.87±0.02	0.78±0.01	0.77±0.01
	$(50, 10^{-4})$ DP-CGAN	0.92±0.02	0.82±0.02	0.81±0.02	0.77±0	0.76±0.01
	$(10, 10^{-4})$ DP-CGAN	0.96±0.01	0.89±0.03	0.89±0.02	0.76±0.01	0.73±0.02
	$(1, 10^{-4})$ DP-CGAN	0.91±0.02	0.83±0.02	0.82±0.02	0.68±0.03	0.67±0.02
	$(0.1, 10^{-4})$ DP-CGAN	0.92±0.01	0.87±0.02	0.86±0.02	0.75±0.01	0.74±0.01

Figure 5.1: Comparative performance results: Classification using CGAN, DP-CGAN-generated datasets vs original on $\epsilon = 50, 10, 1, 0.1$ and $\delta = 10^{-4}$[1]

From the results, it is evident that training classification models on synthetically generated data without particular concern for privacy is close in performance metrics to training on the original data, in some parameters even surpassing the performance of classifiers on the original dataset. This holds true for both settings as well as both datasets that were looked at. For comparison with differentially private equivalents, we have chosen runs with higher privacy costs and lower privacy costs.

Looking at the given runs, we can see that the AUC score is higher across all runs with synthetic data compared to the baseline. The AUC scores of the differentially private models are slightly worse compared to the non-DP CGAN. The classifiers are perfectly capable of using synthetic data to classify points. Performance of classifiers trained on DP-CGAN generated datasets have achieved slightly lower accuracy scores than the baseline run. The difference in performance loss is more noticeable

[1] All numbers are rounded to two decimal places. δ was chosen according to heuristic $\delta < \frac{1}{|D|}$

in the runs involving the HD dataset. The F1 scores have seen a negative impact in the DP-CGAN runs on the HD dataset, but have been better on average in the PID runs.

Adding a mechanism for DP therefore does indeed impact certain parameters ever so slightly, making them perform worse than the non-DP CGAN as well as the baseline model.

It suggest two things: Performance by datasets generated by differentially private GANs can be on par with their non-DP counterparts and are comparable to baseline; however there seems to be a link between performance and the underlying dataset on which the model learns the distributions of attributes. The runs on the HD dataset have shown that the drop in both the F1 score and the accuracy is larger compared to both baseline and non-DP generation.

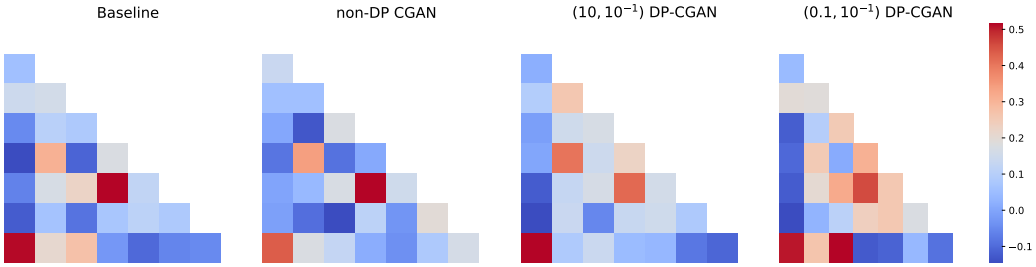


Figure 5.2: Correlation between features of the generated datasets in different runs on the PID dataset after [31]

As seen in the visual comparison between the correlations of the original and generated features in the datasets, the synthetically generated datasets generally mirror the correlations in baseline scenario both in the non-DP and DP runs, meaning generation does indeed produce similar and usable datasets. However, we can also

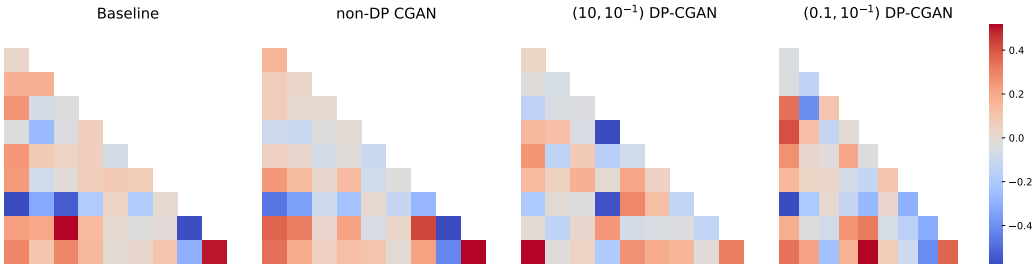


Figure 5.3: Correlation between features of the generated datasets in different runs on the HD dataset

see that feature correlation in datasets generated on the PID dataset was captured

slightly better across runs than in the HD dataset. We can also observe a degradation of correlation capture between different privacy parameters, especially within the PID generation runs: Higher privacy constraints (lower ϵ) results in a worse capture of feature correlations compared to a non-DP run or the baseline.

CGAN generated data

	Pregnancies	Glucose	BP	ST	Insulin	BMI	DPF	Age	Outcome
1	4.0	138.29	63.09	20.78	123.23	24.70	0.66	42.70	0.0
2	-1.0	143.26	78.34	34.82	206.74	31.30	0.41	22.42	0.0
3	1.0	92.66	50.56	25.02	122.28	27.37	0.53	23.26	0.0
4	4.0	149.82	87.40	41.80	138.52	37.12	1.10	45.16	1.0
5	5.0	158.77	76.05	62.12	197.58	43.76	1.34	28.10	1.0
6	2.0	136.96	77.04	26.69	329.69	31.62	0.74	24.65	1.0
7	1.0	104.38	68.05	23.08	97.98	37.93	1.11	31.92	0.0
8	2.0	143.52	85.90	32.64	98.78	30.47	1.01	25.99	1.0
9	5.0	149.43	74.13	31.90	386.28	36.78	1.43	37.75	1.0
10	7.0	165.53	78.27	26.00	188.27	35.89	0.95	47.33	1.0

$(0.1, 10^{-4})$ DP-CGAN generated data

	Pregnancies	Glucose	BP	ST	Insulin	BMI	DPF	Age	Outcome
1	12	139.72	72.78	27.17	52.86	35.82	0.73	47.13	1.0
2	4	90.96	62.24	22.44	43.86	28.22	0.43	25.33	1.0
3	6	130.02	87.48	24.19	40.92	33.06	0.50	55.22	1.0
4	−0	116.40	66.70	13.07	187.89	33.98	1.21	15.03	0.0
5	2	184.57	76.85	22.37	50.44	38.32	0.18	58.22	1.0
6	7	109.36	72.10	27.16	46.38	33.36	0.65	38.75	1.0
7	3	112.95	74.64	24.35	53.69	41.44	0.20	30.22	1.0
8	4	141.17	70.80	20.77	45.77	35.44	0.21	56.29	1.0
9	2	154.55	55.37	23.13	194.98	35.37	0.38	24.37	1.0
10	1	126.50	91.49	32.73	92.90	50.86	1.33	28.67	1.0

Figure 5.4: First 10 samples of synthetically generated content from the PID datasets from CGAN and DPCGAN as an exemplary comparison of raw data[2]

The generated results, regardless of whether or not differentially private mechanisms are involved, can on occasion turn up values for certain features which are outside of the meaningful medical/real-world application range. Clearly, negative pregnancies do not make sense. Row 4 in 5.4 of the DP-CGAN generated dataset is mislabeled: With a BMI of well over 30 and in a young age group with no pregnancies, the chances of the patient being diabetic is much more likely than not[3]. The occurrence

[2]The displayed table contains numbers rounded to two decimal places due to space constraints. The raw data contains the full number of decimal places

[3]The common medical definition of obesity is BMI$\geq$ 30 (weight divided by square of the height)[44] and obesity and overweightness has been linked to a higher prevalence of health issues including diabetes. [45][3]

CGAN generated data

	Age	Sex	CP	TrestBps	Chol	Fbs	RestEcg	Thalach	ExAng	Oldpeak	Num
1	49.38	1.0	1.0	132.75	264.42	0.00	1.00	126.76	1.0	3.71	1.0
2	63.06	1.0	1.0	133.41	217.33	0.00	1.00	138.53	1.0	1.36	1.0
3	63.11	0.0	1.0	135.90	409.70	0.00	0.00	150.11	0.0	1.16	0.0
4	48.78	1.0	2.0	149.45	258.97	0.00	0.00	151.83	0.0	0.05	0.0
5	61.26	1.0	2.0	117.93	243.94	0.00	0.00	145.61	0.0	0.60	0.0
6	54.36	1.0	1.0	156.79	176.49	0.00	0.00	178.66	0.0	−0.18	0.0
7	62.57	1.0	1.0	122.66	322.23	0.00	0.00	136.73	1.0	1.83	1.0
8	53.77	1.0	1.0	126.39	239.85	0.00	0.00	130.76	1.0	1.24	1.0
9	46.41	1.0	2.0	138.43	180.91	0.00	0.00	160.99	0.0	−0.01	0.0
10	63.35	1.0	1.0	114.67	235.99	0.00	0.00	132.82	1.0	2.89	1.0

$(0.1, 10^{-4})$ DP-CGAN generated data

	Age	Sex	CP	TrestBps	Chol	Fbs	RestEcg	Thalach	ExAng	Oldpeak	Num
1	53.84	1.0	1.0	141.43	268.11	1.00	0.00	144.02	1.0	0.68	1.0
2	45.39	1.0	1.0	122.82	236.20	0.00	0.00	147.84	0.0	0.53	0.0
3	37.97	1.0	1.0	136.01	171.17	0.00	0.00	136.09	1.0	−0.73	1.0
4	57.64	1.0	1.0	135.61	208.67	1.00	1.00	111.30	0.0	0.05	1.0
5	42.16	1.0	1.0	152.44	268.48	0.00	0.00	116.86	1.0	1.78	1.0
6	40.91	1.0	1.0	135.69	225.72	0.00	0.00	151.16	0.0	0.41	1.0
7	46.42	1.0	1.0	140.32	206.66	0.00	0.00	110.33	1.0	0.01	1.0
8	43.47	1.0	1.0	133.29	223.48	0.00	0.00	143.44	1.0	0.54	1.0
9	46.85	1.0	1.0	130.68	245.24	0.00	0.00	130.52	1.0	0.52	1.0
10	42.39	1.0	1.0	126.28	172.49	0.00	2.00	128.85	0.0	0.19	0.0

Figure 5.5: First 10 samples of synthetically generated content from the HD datasets from CGAN and DPCGAN as an exemplary comparison of raw data[4]

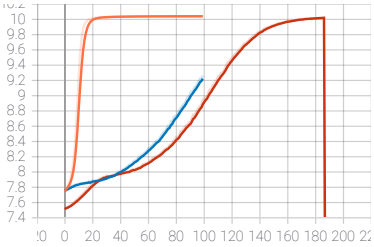
of certain non-usable data points has been observed for both generated datasets. A similar situation can be observed with the data generated from the HD dataset: The Oldpeak attribute occasionally turns negative, which does not have a medical meaning since ST depression measurements are ≥ 0 and in mm [46].

PATE-GAN

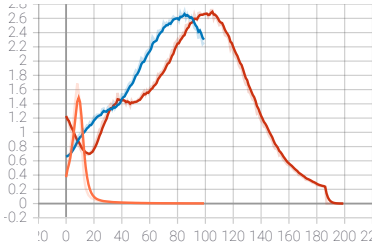
We could not properly evaluate the generated data with respect to DP guarantees due to poor results within this work. Despite various efforts in searching for good hyperparameters, trying to vary training epochs, label smoothing as well as giving a larger ϵ, PATE-GAN did not converge as expected during the training process

[4]The displayed table contains numbers rounded to two decimal places due to space constraints. The raw data contains the full number of decimal places

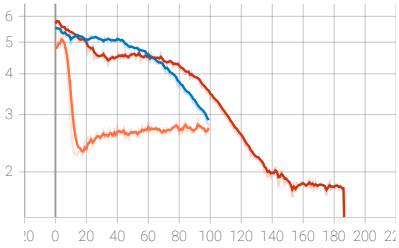
with either dataset and only offered poor quality generated data and bad classifying performance. At larger learning rates, the stabilization in losses of G and S occurs



(a) G loss



(b) norm of gradients of G



(c) S loss of student (log scale)

Figure 5.6: Training metrics for PATE-GAN on PID: orange curves indicate a training run with learning rate $1E^{-4}$, blue and red curves are learning rate $1E^{-5}$ at different training epoch lengths, sudden drop is cut off point when $\epsilon = 100$ is reached

earlier but since gradient norms of G peak and get very small very fast, there are little learning effects after a certain point. Experimenting with smaller learning rates did not significantly change the performance outcome on the generated data.

	Pregnancies	Glucose	BP	ST	Insulin	BMI	DPF	Age	Outcome
1	4.0	124.76	71.19	26.77	89.49	32.01	0.50	33.35	0.0
2	5.0	124.60	73.03	27.01	89.58	31.90	0.44	31.92	1.0
3	4.0	120.99	73.47	27.93	97.90	32.86	0.51	32.75	1.0
4	4.0	118.42	70.01	26.56	90.31	32.57	0.42	33.45	0.0
5	4.0	125.02	72.83	27.09	96.88	32.81	0.46	31.65	0.0
6	4.0	119.57	71.81	27.52	97.35	32.49	0.48	32.82	1.0
7	4.0	121.55	72.76	27.60	96.23	32.71	0.49	33.12	0.0
8	4.0	118.21	71.73	27.27	91.46	33.40	0.47	32.62	1.0
9	4.0	122.98	70.98	27.00	99.00	32.09	0.50	34.11	1.0
10	4.0	118.43	72.29	27.85	93.49	31.91	0.44	33.51	0.0

Figure 5.7: First 10 samples of synthetically generated content from PATE-GAN on PID

The distributions of attributes' values do not mimic the ones in the source datasets

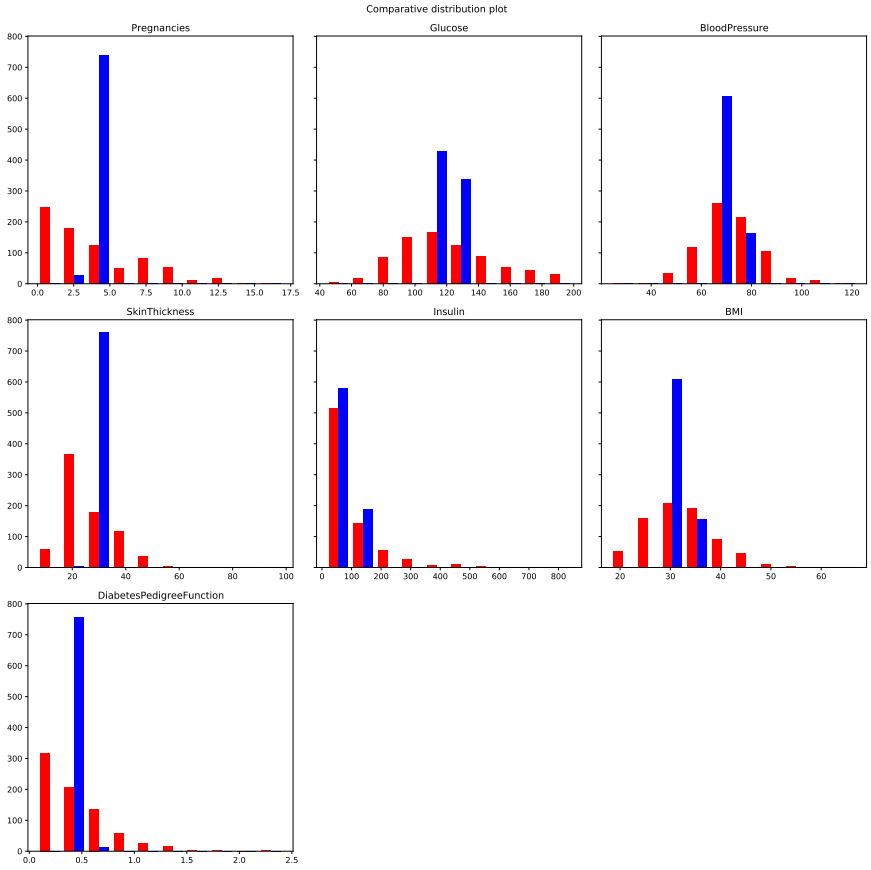


Figure 5.8: Histogram of the first 768 data points of the generated data from PATE-GAN run on PID with the original source data as comparison *(red: source data, blue: generated)*

as seen in 5.8, indicating the training process has not converged and has lead to a collapse of certain attributes into few values (e.g. Pregnancies are all either 4 or 5). The classifying performance is thus very bad, giving us AUROC values around 0.5 and an accuracy of 0.5 as well across all models, showing no ability to successfully classify records into diabetic/non-diabetic patients (7.2 and 7.3). A similar picture emerges from training on the HD set; for brevity reasons, we have put the corresponding figures into the appendix (page 49 ff).

5.2 Variation of parameters ϵ and C

Looking into the influence of privacy parameters on the classification performance metrics, there were mixed results. Different series of runs were made using different ϵ privacy budgets and clipping norm values C for the calculation of the gradient descents. The PID data suggests a light trade-off correlation between ϵ values and

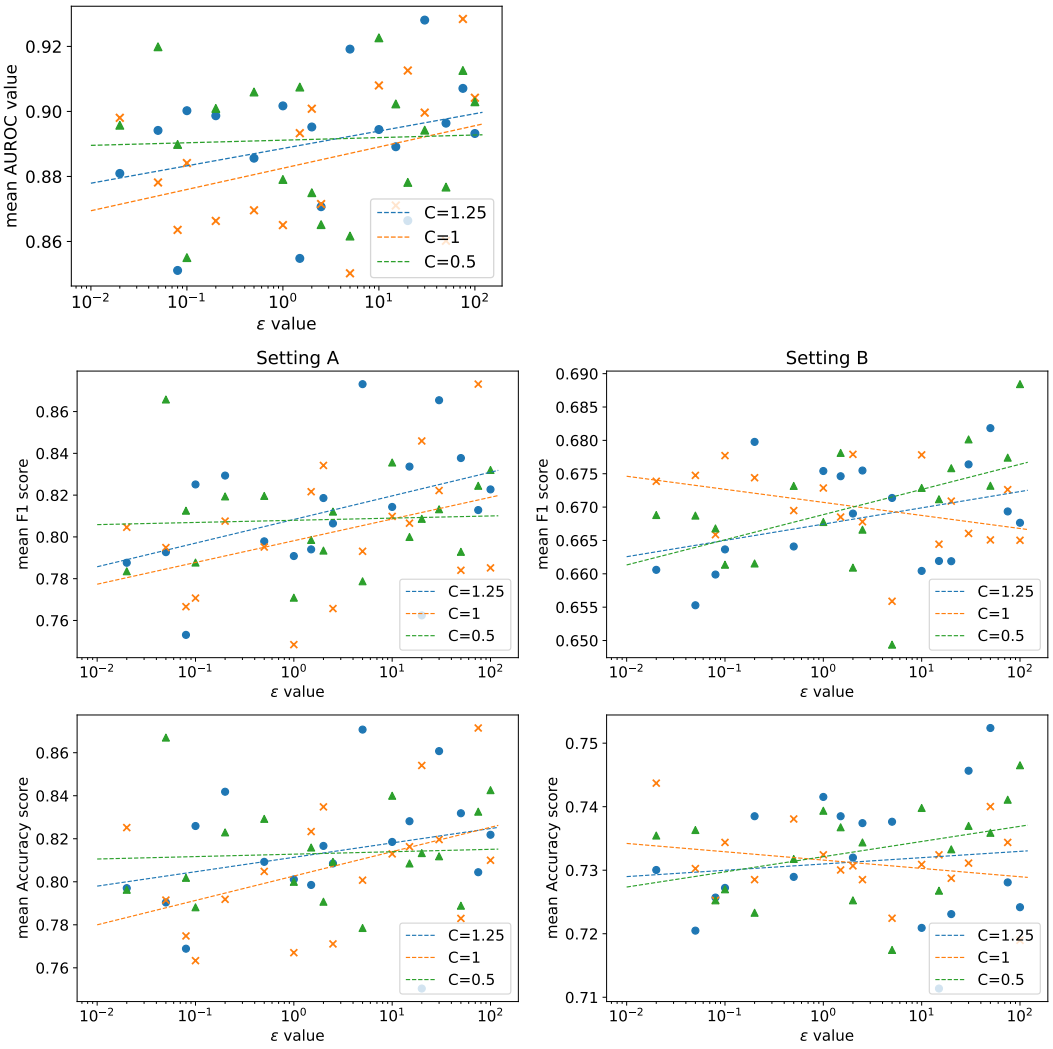


Figure 5.9: Data collection runs showing the relationship between different C, ϵ on the performance metrics values of classifiers on the PID dataset models

AUROC, F1 and to a lesser degree accuracy scores for Setting A on the PID dataset, fitting the findings in [37] and [30]. Higher privacy budgets in the form of higher

values of ϵ correlate to rising scores in AUROC. The variation across different clipping norm factors C seems to point to higher C having a bigger impact on the score variability, showing stronger slopes in trending curves.

However, when we look at setting B, the findings are not clear. For both the F1 and accuracy scores, the data points for the $C = 1$ are an anomaly: The data counter-intuitively points to a slight fall in scores the larger the privacy budget gets.

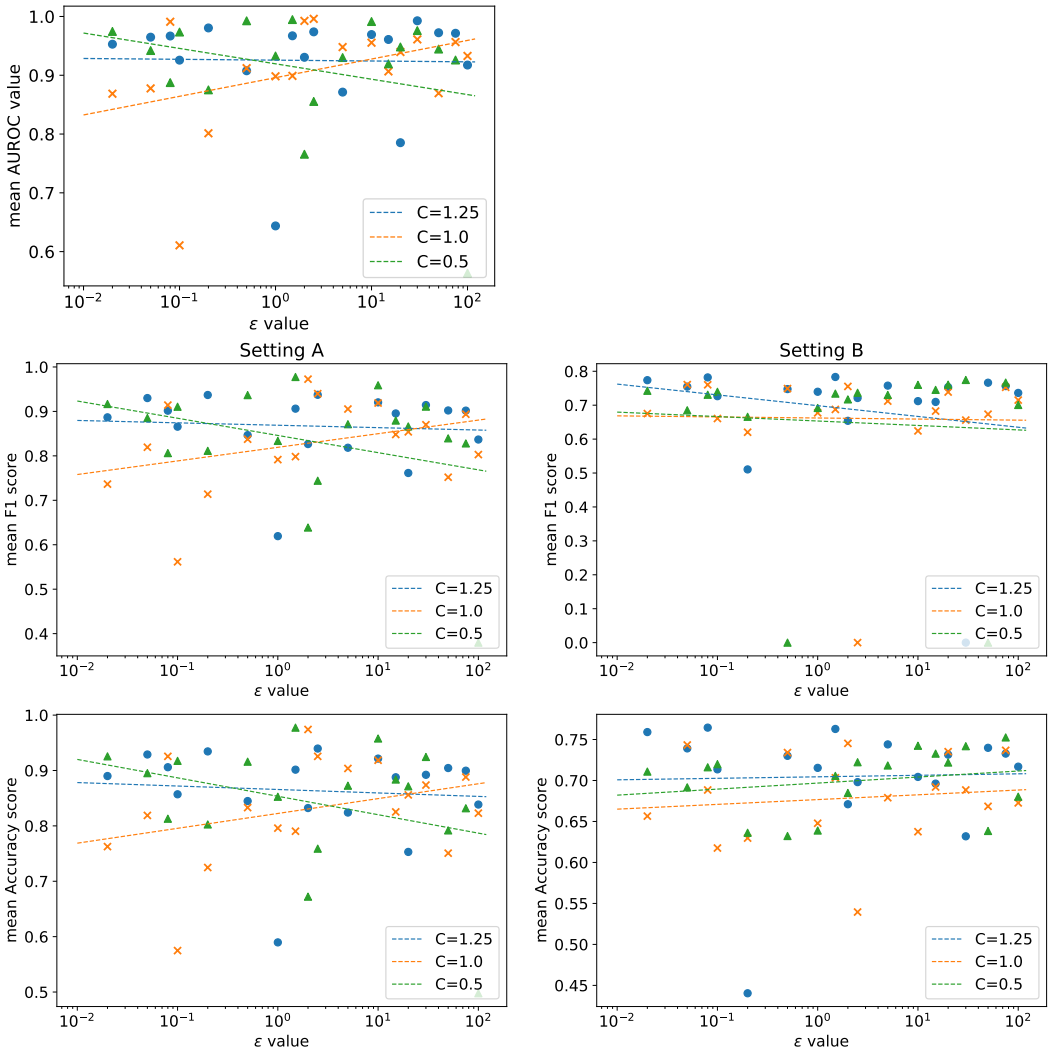


Figure 5.10: Data collection runs showing the relationship between different C, ϵ on the performance metrics values of classifiers on the HD dataset models. Some scores calculated on the runs were recorded as non-finite, which we have plotted as 0 and should be disregarded.

With the experimental runs on the HD dataset, we can see some inconsistent results as well. All scores show a positive correlation with rising privacy budgets, meaning larger ϵ values lead to equal or even slightly better scores for a clipping norm value of $C = 1$. Scores in experimental setting B show less deviation than in setting A, where a flat or slightly positive trend curve is emerging from the data collected. For $C = 0.5$ however, the trend curve is showing a fall with larger ϵ values, especially in setting A. The influence of the clipping norm value on the results here requires further studies.

5.3 Memorization and disclosure risks

Since we train on the source datasets in CGAN/DP-CGAN, accidental privacy leaks by unintentional memorization of some original data points during training, while being seemingly improbable, still remain a point of concern. A crude but simple method of investigating any memorization issues is checking the generated datasets for identical records. In this work, none of the generated datasets have any identical records to the original set the generative networks have been trained on.

Looking at the average euclidean distance between all records of the generated sets and the closest records in the original set allows us to gauge how far away a particular generated point actually is and whether or not this might lead to some issues of being able to retrace the original data points.

Datasets	Source	$\bar{d}$
PID	CGAN	1.035±0.477
	$(50, 10^{-4})$DP-CGAN	1.014±0.479
	$(10, 10^{-4})$DP-CGAN	1.039±0.488
	$(1, 10^{-4})$-DP-CGAN	0.994±0.451
	$(0.1, 10^{-4})$-DP-CGAN	0.996±0.462
HD	CGAN	1.828±0.743
	$(50, 10^{-4})$DP-CGAN	2.224±0.602
	$(10, 10^{-4})$DP-CGAN	2.045±0.592
	$(1, 10^{-4})$-DP-CGAN	1.948±0.519
	$(0.1, 10^{-4})$-DP-CGAN	2.016±0.665

Figure 5.11: Average pairwise euclidean distance $\bar{d}$ between synthetic data points and their closest relative data point in the original datasets

As seen in 5.11, the distance of generated data points for the HD runs are on average slightly larger for the DP-CGAN runs than for the non-DP CGAN run. However, we cannot clearly say the same about the PID runs, where the average distance of

DP generated points is even slightly below the non-DP ones. We could not see a clear correlation between privacy costs and distance metric.

This gives us a mixed picture: The differences of distances are all well within standard deviation. Thus using DP generation can potentially increase the distance of data points to the closest points in the original, thus decreasing the chance of a potential privacy leak concern, but it depends on the dataset that is considered as well as the model parameters.

As an example, we have tried to compare a single data point on a generated dataset with its closest data point in the original dataset. In the case shown in 5.12, there is a privacy concern as the age and sex attribute of the generated point is fairly close to the original. However, it is arguable that the medical data values associated with the generated point is different to the original. Age and sex alone do not contribute to a major violation of privacy if the medical data points associated with them do not make them recognizable or identifiable.

Selected data point in the generated dataset $(0.1, 10^{-4})$-DPCGAN

	Age	Sex	CP	TrestBps	Chol	Fbs	RestEcg	Thalach	ExAng	Oldpeak	Num
1	53.84	1.0	1.0	141.43	268.11	1.00	0.00	144.02	1.0	0.68	1.0

Closest data point in the original set, $\bar{d} = 2.006$

	Age	Sex	CP	TrestBps	Chol	Fbs	RestEcg	Thalach	ExAng	Oldpeak	Num
529	54.0	1.0	4.0	122.0	286.0	0.0	2.0	116.0	1.0	3.2	1.0

Figure 5.12: Comparison between a single data point of the generated dataset by $(0.1, 10^{-4})$-DPCGAN and its closest point in the original HD source dataset. Both data points were normalized and standardized to measure their euclidean distance and then transformed back

To further study the severity of privacy leaks, we conducted a small evaluation to judge presence disclosure risk in case of a breach (potential adversary can guess if a compromised data point is actually part of the training dataset). To that extend, we have sampled a random number r of patient records each from training and test section of the source dataset to be the collection of compromised data records that an adversary might have. Assuming the potential attacker has complete information over these records, we then calculate the k-nearest neighbors within the generated datasets for every compromised record. If at least one such neighboring synthetic data point provides an euclidean distance $d < d_{\text{threshold}}$, we count this as a match, meaning the adversary would claim that the point he has is part of the GAN training data set.

Since we have two different sets that we sampled from, there are now four distinct possibilities: the adversary was indeed correct and the target data point is present

in the original source training set (T_p), the adversary was mistaken and the point is from the test set (F_p), the adversary was correct in that the target point was never part of the training set (T_n) or the adversary was wrong and has missed that the point was actually in the original training set (F_n) [40].

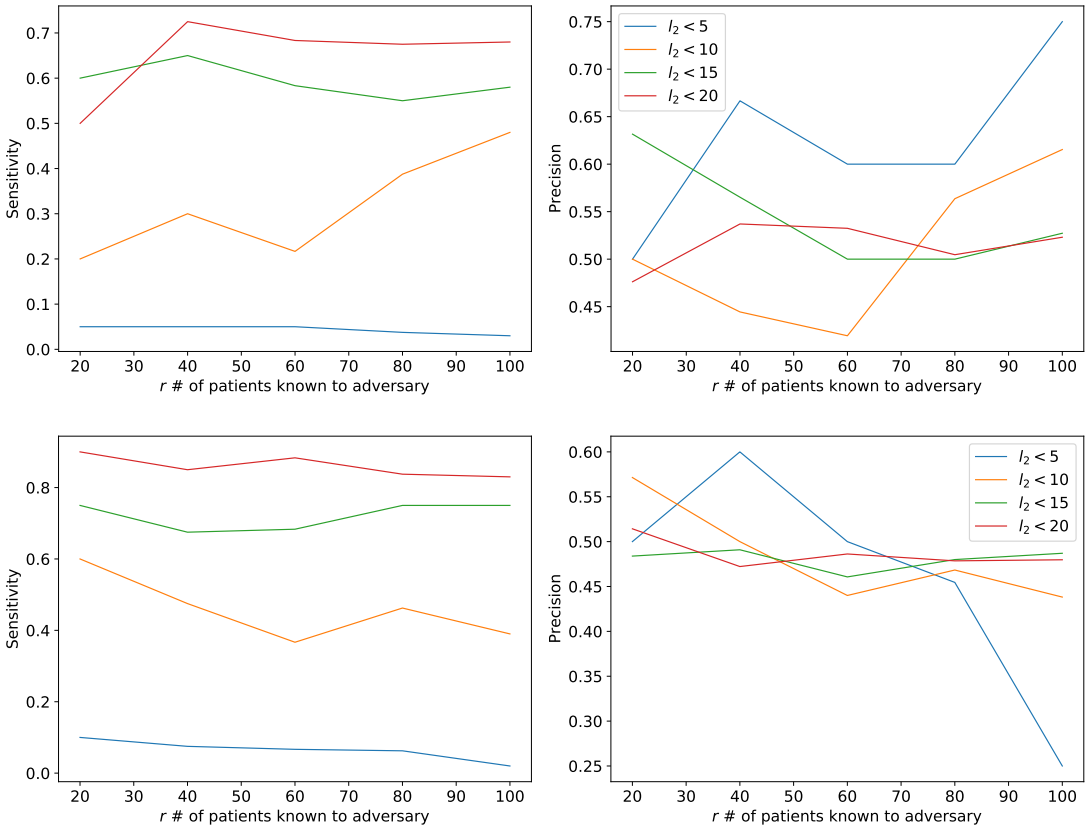


Figure 5.13: Sensitivity and precision of the presence disclosure test for the PID dataset (*top:* CGAN generated synthetic data, *bottom:* $(0.1, 10^{-4})$-DPCGAN generated data), for varying $d_{\text{threshold}} = l_2$ and r

In 5.13, we can see the results for changing the amount of compromised patient data. The percentage sensitivity indicates that the adversary has successfully discovered that $x\%$ of his already known compromised data was used for GAN training purposes while precision here shows the actual proportion of compromised data points that were used to train compared to the number the adversary claimed.

The graphs show that if the distance threshold is kept low enough, the adversary never discovers more than 15% of known patient points that were used in training. The higher the threshold, the more likely it is to discover a large majority. Using the synthetic data from non-DP CGAN generation, it seems that for a middle distance

threshold of $l_2 < 15$, the sensitivity actually goes up with the number of compromised patients, against the trend of other runs.

Looking at the precision graphs, we can also see that the graphs tend to hover around the 50% mark, meaning in a lot of cases, the compromised information the adversary has on hand is less useful than initially thought, lessening the severity of leakage. The development for increasing number of compromised patients seem to lead to better results for the adversary in the case of CGAN synthetic data while in the DP-CGAN case, the precision seems to be stable at around 50% or even decreasing with rising numbers of patients for $l_2 < 15$. A more balanced picture emerges from

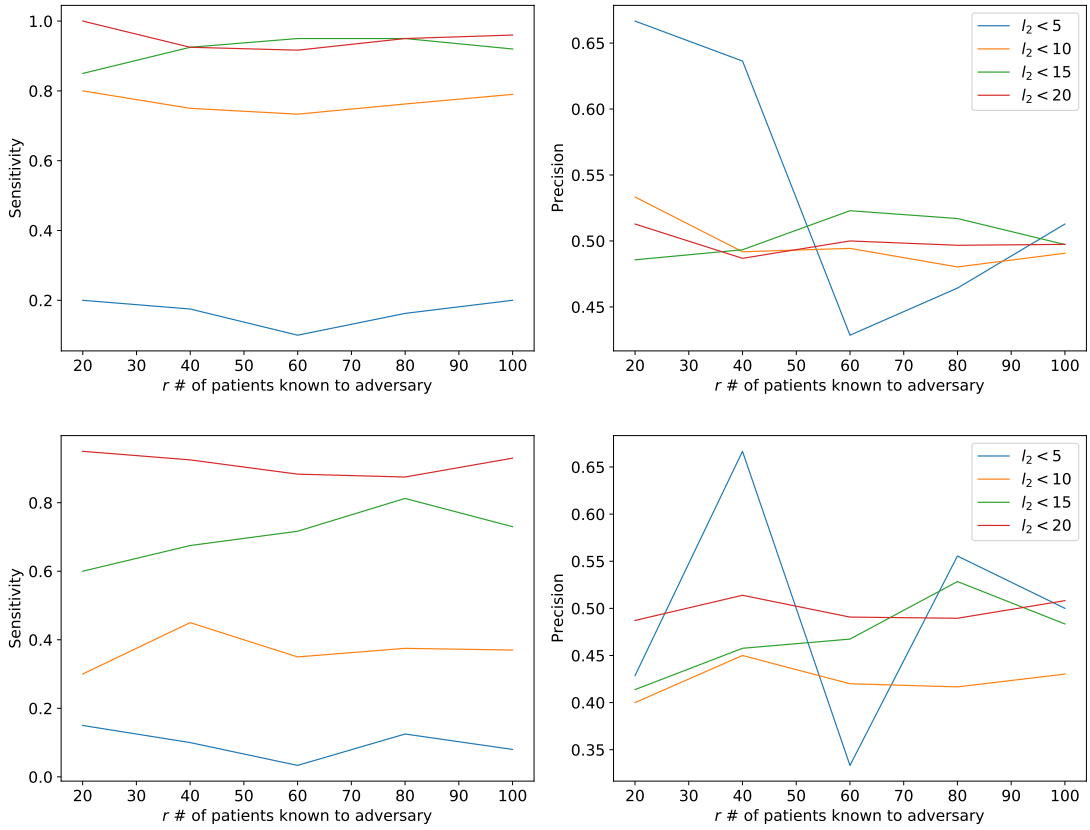


Figure 5.14: Sensitivity and precision of the presence disclosure test for HD dataset (*top:* CGAN generated synthetic data, *bottom:* $(0.1, 10^{-4})$-DPCGAN generated data), for varying $d_{\text{threshold}} = l_2$ and r

the evaluations on the HD dataset, where precision curves for both the non-DP and DP generated datasets are mostly around the 50% mark. The influence of varying the threshold seems to be larger with the non-DP synthetic data, with sensitivity settling at a much higher level compared to the DP-CGAN generated sets.

6 Discussion

6.1 Applicability of DP in generative models

As the previous chapter has demonstrated, DP mechanisms are possible to integrate into generative algorithms to produce data for other ML tasks. In certain scenarios with quite low privacy cost, the generated data was still useful enough to use for standard classifying tasks with a reasonable amount of accuracy compared to the performance on the original dataset. For the PID dataset, even very good privacy guarantees with $\epsilon = 0.1$ enable utility metrics on-par with or just below non-DP CGAN as well as directly training classifiers on the source data. For the HD dataset, the accuracy for $\epsilon = 0.1$ in setting B was 5 percentage points off compared to training on the original dataset, a bigger accuracy loss than in the PID dataset but it is still well within usable range on all other metrics and not too far off the non-DP CGAN run. However, this work points to some non-trivial unexplained dynamics of clipping norm values on the behavior of our models which need further investigation.

Generally, there are a certain number of limitations which hinders the ability to deploy DP-mechanisms for generative models. While useful in a general sense, the stability of training with DP-inclusive generative methods like DP-CGAN and PATE-GAN is very variable and has an impact on the practical usability of such methods.

For comparatively small source datasets with some skewed features, we have found that GANs with DP mechanisms are bound to not always converge and frequently face collapse issues in certain feature distributions while the non-DP GANs were able to generate decent synthetic sets with more consistency. Generated datasets also have had traces of either inconsistent singular records or of nonsensical medical data, regardless of involvement of differential privacy.

Choosing the right parameters with respect to training (clipping norm C, ϵ privacy guarantee) might be hard for external users to fully understand without deeper knowledge of the intricacies of behavior of the DP GANs.

The discussion of DP in GANs should also not overshadow the general discussion on what differential privacy can or cannot achieve. While quantifying privacy in a more concrete way is certainly an improvement, differential privacy in of itself does

not mean freedom from all potential harm.

If trained models generate data points that lets you infer conclusions that hold true even for certain individuals, this inference of private attributes from publicly observable attributes would not be considered evidence of DP violation. Individuals might not even have added their data into the input dataset at all.

Another fact to consider is the question of how catastrophic the real-world consequences of a violation of privacy guarantee as defined by ϵ-DP really is. For small ϵ, failure to meet ϵ-DP (for example, instead being 2ϵ-DP) can be considered acceptable as the nature of privacy guarantees in these ranges are similar. To assess the severity of a privacy breach for mechanisms with large ϵ, one still needs to consider other factors as well; a large ϵ merely means that there are neighboring datasets and an output o from a dataset for which probabilities of observing such o is large. If an adversary does not have the necessary information to realize that such a reveal has happened or does not know enough details about the dataset itself, then a breach in this case might be meaningless [13]. As we can see from the presence disclosure tests, the risk of mass disclosure even in the case of a breach is relative to the size of compromised data that an adversary has. However, further studies on the robustness of GANs with DP with regards to adversarial attacks are needed to give a more detailed picture on the findings in this work.

A possible contention point might be the usage of ϵ-DP in larger scale commercial applications in and outside of the machine learning community. Advertising privacy using the notion of ϵ-DP in the process might be misleading to users as its specific implementation in an application is not always well documented, poorly communicated to the public or with too large ϵ that are not configurable by users to have a good guarantee [47]. Complicating the deployment of DP furthermore is the difficulty of translating the abstractness of factors like ϵ or δ into user-understandable language so they can make a reasonable decision as to which privacy loss is acceptable in their specific case [19]. Unclear or imprecise communication ultimately leads to suggestions of total privacy guarantees to users where there might not actually be one.

6.2 GAN training issues

Common recurring problems with GANs reported within the ML community were also affecting the results in this work. GAN training tends to be highly sensitive to hyperparameters where even slight changes in certain parameters like the learning rates of either G or D or the ratio between the two can lead to a significantly worse generator, leading the performance on the synthetic datasets to degrade drastically.

In the worse case, convergence of D and G does not happen and the losses either explode/vanish or continue to erratically bounce within a certain range with no improvement.

To combat some of these problem, there have been multiple points of potential improvements pointed out by Salimans et al. [48]. One-sided label smoothing, where instead of a hard 1/0 label for the discriminator D, slightly noised values off from 0 or 1 has been applied to the CGANs in this work with some success. It is shown to prevent the network from being overconfident and more robust towards adversarial example attacks.

We have also found that one-hot encoding can problematic for general usage; for the HD dataset, it has lead to temporarily inflated feature numbers for the GAN to learn; this works for datasets with few features but can blow up to very large input dimensions for the GAN if used on datasets with high feature numbers, making it computationally expensive to learn these.

Another problem that seems to occur more often with encoded categorical data is the tendency of generators of GANs to collapse their results into very few or even singular values of attribute distributions, which also negatively affect classifier performance.

PATE-GAN training has shown some specific problems of the generator being too weak, making the generated data useless. With G being weak, the teacher and student discriminator models have an easy job distinguishing between real and fake examples, making the student improvement process, which relies on data provided by G as well as labels by the teacher ensemble come to a standstill. The rapid decline of the gradient norms that was observed for the generator indicate the learning process is ineffective. Suggestions in [38] to use Xavier initialization (equivalent to TF `GlorotNormal` initialization) in model layers to ensure some realistic generated samples of G in the beginning have not worked. As a result, we could not replicate the results by Jordon et al. or conclusively evaluate the classification performance of PATE-GAN generated data or the influence of larger or smaller privacy guarantees on it.

6.3 Conclusions

All in all, this work has outlined the applicability of DP in generative methods on medically sensitive data, the results show that for certain generative methods with DP, the performance results are on-par and sometimes better than their non-private counterparts while simultaneously giving a reasonably good privacy guarantee with ϵ.

However, the various hurdles and problems encountered show that implementing DP into GANs is not always a trivial task and the models perform differently not only on different data types but also on different datasets of the same type, making it hard to easily deploy it for various different medical datasets.

The choice and meaning of an appropriate ϵ at generation time remains hard to convey to an outside user of such synthetically generated data and it needs additional study on how to appropriately explain the privacy guarantee of such synthetic data for mass usage.

The work has also shown that the success of DP-inclusive GANs fundamentally depend on the convergence of the underlying GAN on the dataset. Stability and convergence issues with GANs itself are known in the community and finding solutions can be difficult. We suggest further investigations into the sensitivity of parameters within PATE-GAN and studying the convergence behavior of PATE-GAN as well as for GAN training with DP in general.

7 Appendix

Mathematical addendum

(Expectation) For the discrete case: Let X be a random variable with outcomes $x_1, x_2, \ldots x_k$ with probabilities $p_1, p_2, \ldots p_k$. The expected value $\mathbb{E}(X)$ is

$$\mathbb{E}(X) = \sum_{i=1}^{k} x_i p_i \tag{7.1}$$

For the continuous case: Let X be a random variable with a probability density function $f(x)$. The expected value $\mathbb{E}(X)$ is

$$\mathbb{E}(X) = \int x f(x) \mathrm{d}x \tag{7.2}$$

The expectation value is also called the first moment of a random variable.

(Lipschitz continuity) Let $S \in \mathbb{R}$ be a subset of $\mathbb{R}$, $L \geq 0$ and $f : D \to \mathbb{R}$ a function.

$$f \text{ is L-Lipschitz continuous} \quad \Leftrightarrow \quad \exists L \forall x, x' \in D : |f(x) - f(x')| \leq L|x - x'| \tag{7.3}$$

(Tail bounds) In statistical analysis, it is often useful to be able to bound the probability that a random variable X deviates far from its mean. The most lose bound is the Markov bound: Let X be a non-negative random variable, $k > 0$

$$P(X \geq k) \leq \frac{\mathbb{E}(X)}{k} \tag{7.4}$$

Dataset feature descriptions

Attribute feature	Type	Description
PID dataset		
Pregnancies	numeric int	Number of pregnancies a patient has experienced
Glucose	numeric	Plasma glucose concentration 2h after glucose tolerance test in mg/dl
Blood pressure	numeric	Diastolic blood pressure in mmHg
Skin thickness	numeric	Triceps skin fold thickness in mm
Insulin	numeric	Insulin level 2h after glucose administration in μU/ml
BMI	numeric	Body Mass Index (weight in kg/(height in m)^2)
DPF	numeric	Diabetes pedigree function
Age	numeric int	Age of patient in years
Outcome	binary	Prediction label diabetic or healthy
HD dataset		
age	numeric	Age of patient in years
sex	binary	Male = 0, Female=1
cp	categorical int	Chest pain type 1 (typical angina), 2 (atypical angina), 3 (non-anginal pain), 4 (asymptomatic)
trestbps	numeric	Resting blood pressure in mmHg
chol	numeric	Serum cholesterol level in mg/dl
fbs	binary	Fasting blood sugar > 120 mg/dl
restecg	categorical int	Resting electrocardiographic results: 0=normal, 1=ST-T wave abnormality, 2=probable or definite left ventricular hypertrophy
thalach	numeric	Maximum heart rate achieved in exercise test
exang	binary	Exercise induced angina
oldpeak	numeric	ST depression induced by exercise relative to rest
num	binary	Prediction label healthy or heart disease

Figure 7.1: List of attribute descriptions for PID and HD datasets; for the HD dataset, the number of attributes was reduced to 10+outcome label due to large numbers of values missing

Additional diagrams

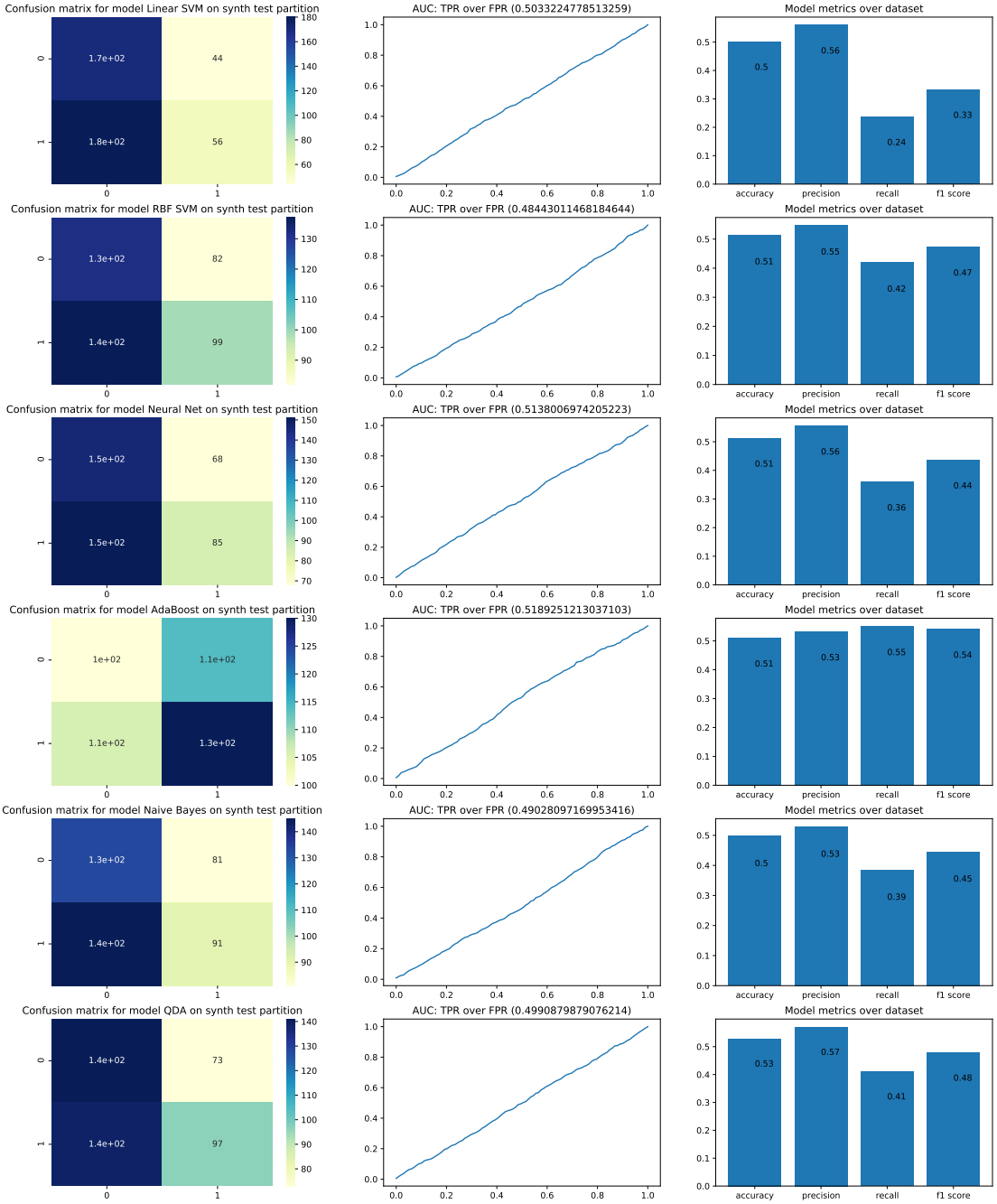


Figure 7.2: Performance metrics for each classifier in setting A for PATE-GAN generated data on PID

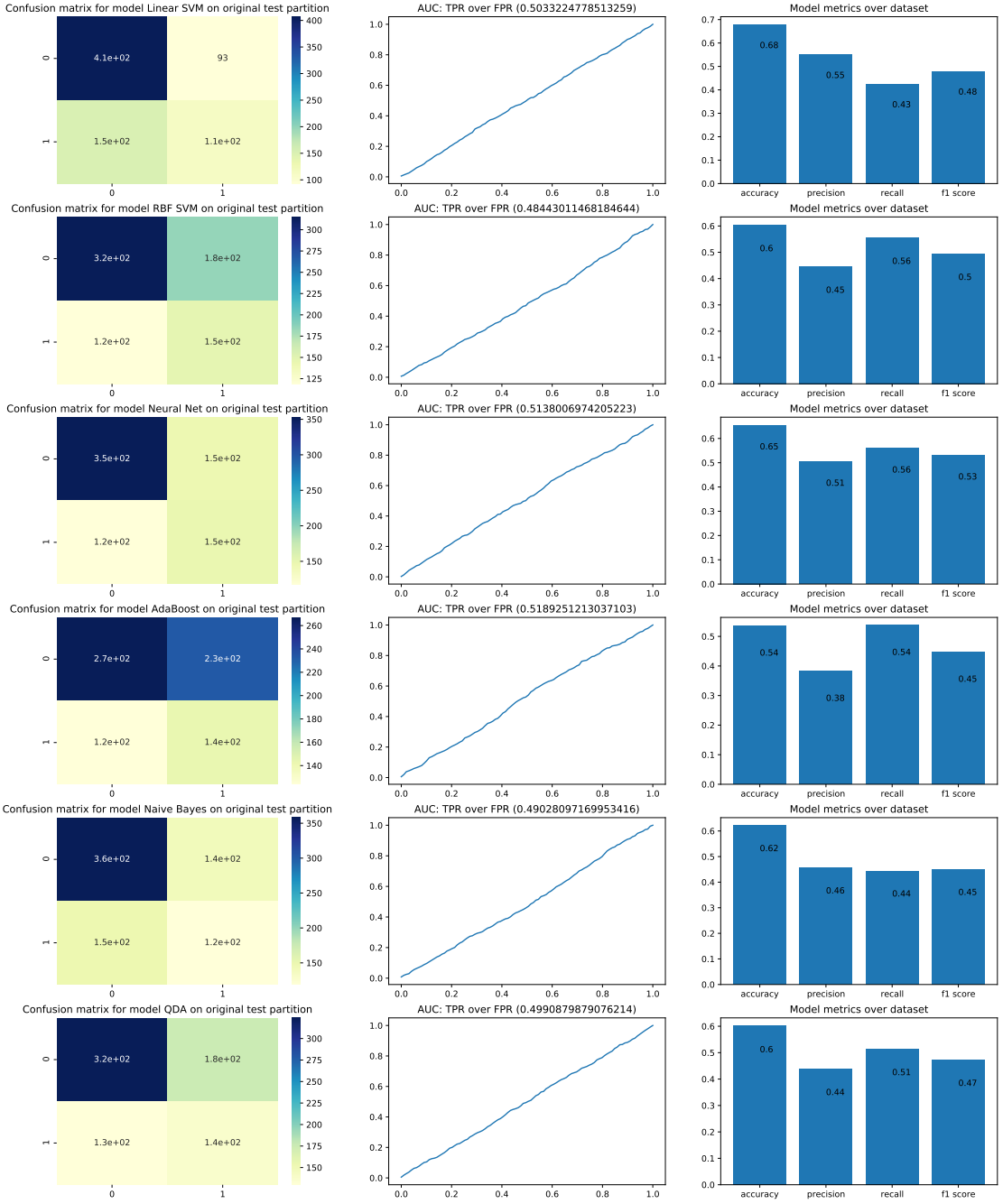


Figure 7.3: Performance metrics for each classifier in setting B for PATE-GAN generated data on PID

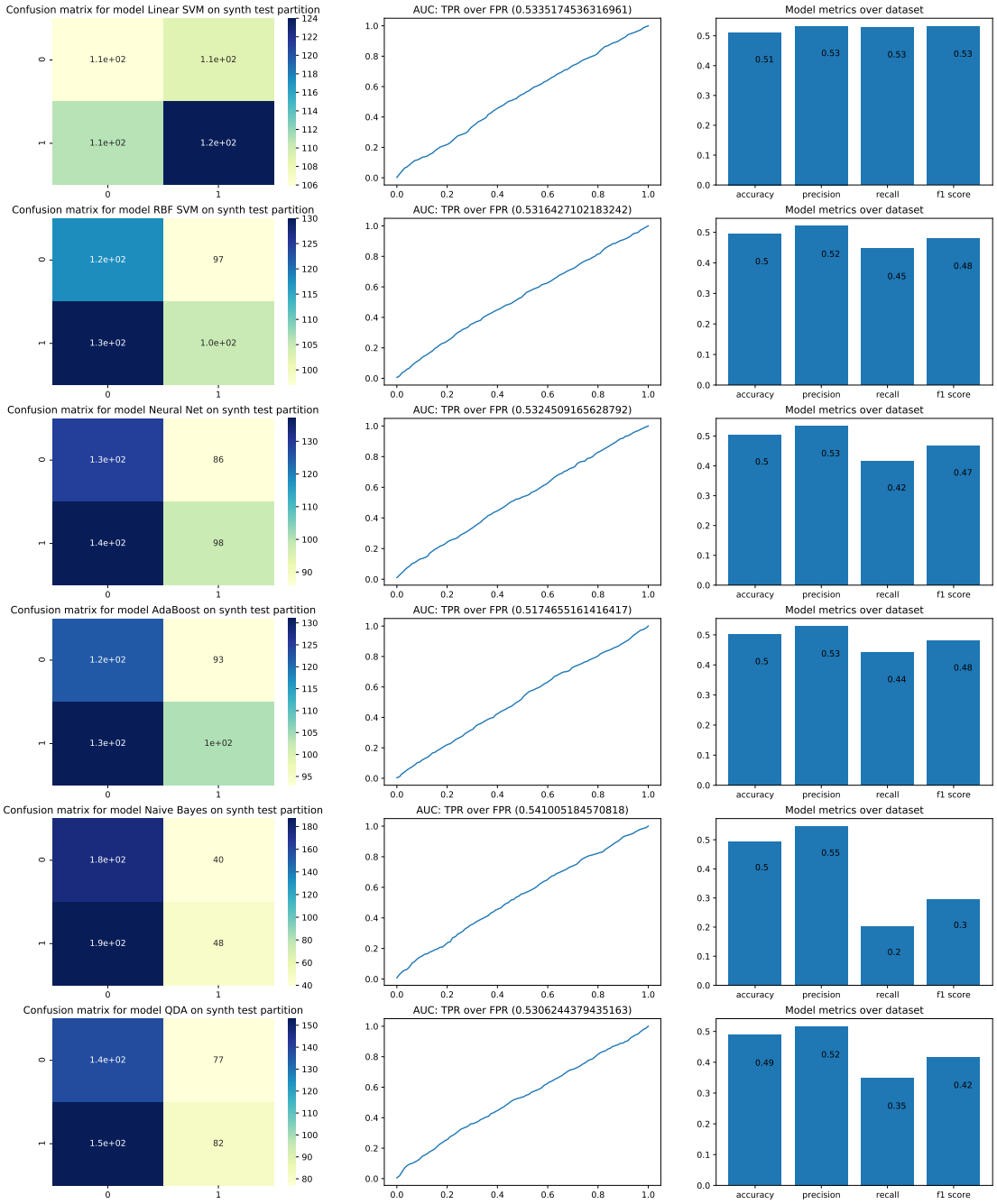


Figure 7.4: Performance metrics for each classifier in setting A for PATE-GAN generated data on HD

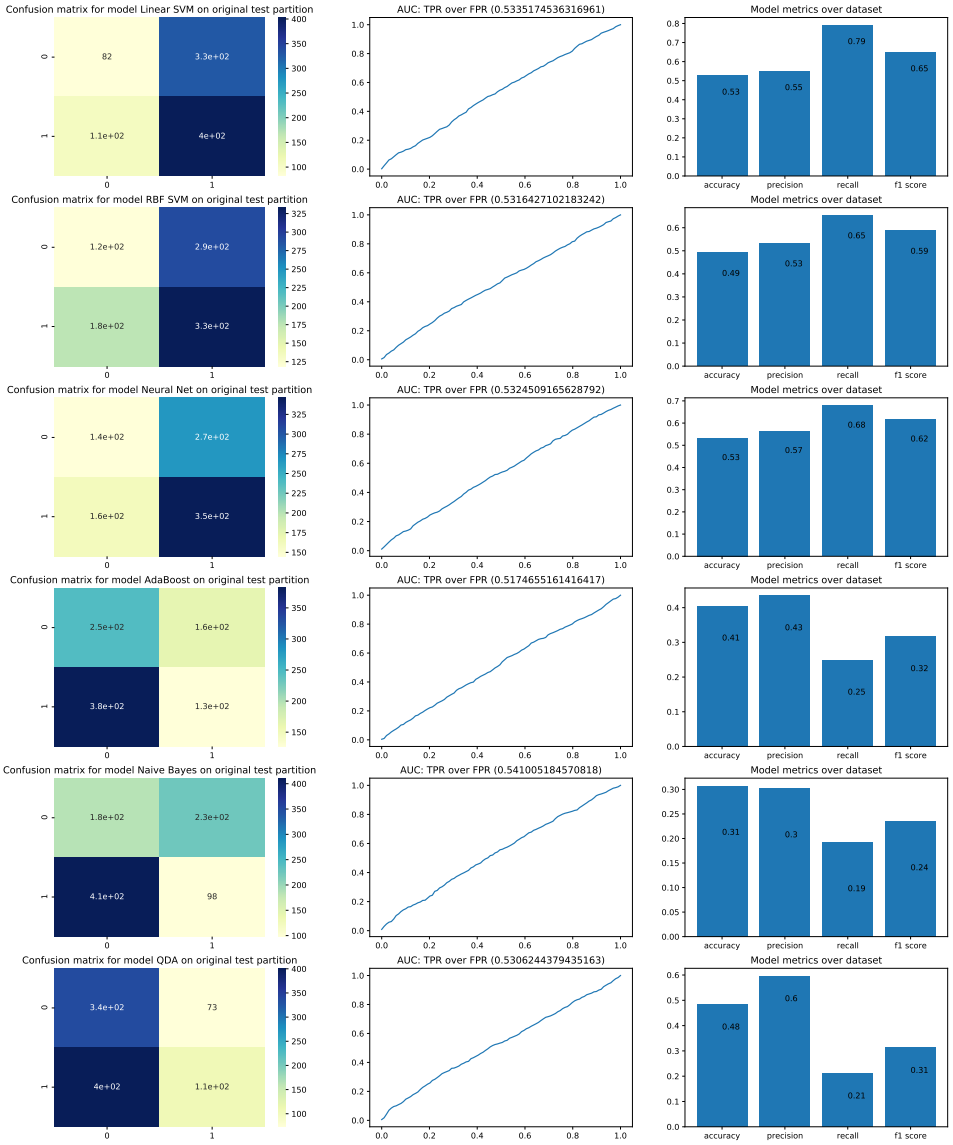


Figure 7.5: Performance metrics for each classifier in setting B for PATE-GAN generated data on HD

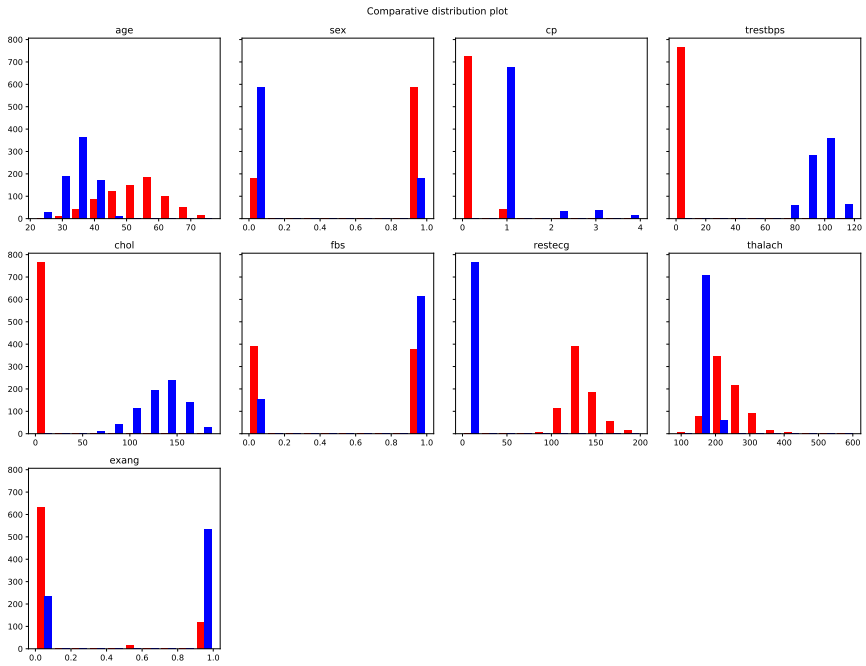


Figure 7.6: Histogram of the first 920 data points of the generated data from PATE-GAN run on HD with the original source data as comparison *(red: source data, blue: generated)*

List of Figures

2.1 The model structure of a vanilla GAN as proposed by Goodfellow et al. with generator G and adversary D as discriminator [24] 11
2.2 Conditional GAN (CGAN) structure proposed in [26] 13

3.1 Ensemble of teacher part of PATE-GAN with a fixed generator G . . 21
3.2 Overall structure of PATE-GAN[38] 21

4.1 Hyperparameter used for training CGAN and DP-CGAN 25
4.2 Schematic model architecture for our CGAN and DP-CGAN after the design in [16] 26
4.3 Parameters for the layers in the CGAN and DP-CGAN models . . . 26
4.4 Hyperparameter used for training PATE-GAN 27

5.1 Comparative performance results: Classification using CGAN, DP-CGAN-generated datasets vs original on $\epsilon = 50, 10, 1, 0.1$ and $\delta = 10^{-41}$ 29
5.2 Correlation between features of the generated datasets in different runs on the PID dataset after [31] 30
5.3 Correlation between features of the generated datasets in different runs on the HD dataset 30
5.4 First 10 samples of synthetically generated content from the PID datasets from CGAN and DPCGAN as an exemplary comparison of raw data[2] 31
5.5 First 10 samples of synthetically generated content from the HD datasets from CGAN and DPCGAN as an exemplary comparison of raw data[3] 32
5.6 Training metrics for PATE-GAN on PID: orange curves indicate a training run with learning rate $1E^{-4}$, blue and red curves are learning rate $1E^{-5}$ at different training epoch lengths, sudden drop is cut off point when $\epsilon = 100$ is reached 33
5.7 First 10 samples of synthetically generated content from PATE-GAN on PID 33
5.8 Histogram of the first 768 data points of the generated data from PATE-GAN run on PID with the original source data as comparison *(red: source data, blue: generated)* 34

5.9 Data collection runs showing the relationship between different C, ϵ on the performance metrics values of classifiers on the PID dataset models 35
5.10 Data collection runs showing the relationship between different C, ϵ on the performance metrics values of classifiers on the HD dataset models. Some scores calculated on the runs were recorded as non-finite, which we have plotted as 0 and should be disregarded. 36
5.11 Average pairwise euclidean distance $\bar{d}$ between synthetic data points and their closest relative data point in the original datasets 37
5.12 Comparison between a single data point of the generated dataset by $(0.1, 10^{-4})$-DPCGAN and its closest point in the original HD source dataset. Both data points were normalized and standardized to measure their euclidean distance and then transformed back 38
5.13 Sensitivity and precision of the presence disclosure test for the PID dataset (*top:* CGAN generated synthetic data, *bottom:* $(0.1, 10^{-4})$-DPCGAN generated data), for varying $d_{\text{threshold}} = l_2$ and r 39
5.14 Sensitivity and precision of the presence disclosure test for HD dataset (*top:* CGAN generated synthetic data, *bottom:* $(0.1, 10^{-4})$-DPCGAN generated data), for varying $d_{\text{threshold}} = l_2$ and r 40

7.1 List of attribute descriptions for PID and HD datasets; for the HD dataset, the number of attributes was reduced to 10+outcome label due to large numbers of values missing 46
7.2 Performance metrics for each classifier in setting A for PATE-GAN generated data on PID 47
7.3 Performance metrics for each classifier in setting B for PATE-GAN generated data on PID 48
7.4 Performance metrics for each classifier in setting A for PATE-GAN generated data on HD 49
7.5 Performance metrics for each classifier in setting B for PATE-GAN generated data on HD 50
7.6 Histogram of the first 920 data points of the generated data from PATE-GAN run on HD with the original source data as comparison *(red: source data, blue: generated)* 51

Bibliography

[1] Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert, and Rickmer F. Braren. "Secure, privacy-preserving and federated machine learning in medical imaging". In: *Nat Mach Intell* 2.6 (June 2020), pp. 305–311. ISSN: 2522-5839. DOI: 10.1038/s42256-020-0186-1. URL: http://www.nature.com/articles/s42256-020-0186-1.

[2] World Health Organization. *Cardiovascular diseases (CVDs)*. URL: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (visited on 06/09/2020).

[3] E. S. Ford, D. F. Williamson, and S. Liu. "Weight Change and Diabetes Incidence: Findings from a National Cohort of US Adults". In: *American Journal of Epidemiology* 146.3 (Aug. 1, 1997), pp. 214–222. ISSN: 0002-9262, 1476-6256. DOI: 10.1093/oxfordjournals.aje.a009256. URL: https://academic.oup.com/aje/article-lookup/doi/10.1093/oxfordjournals.aje.a009256 (visited on 07/21/2021).

[4] The Diabetes Prevention Program Research Group. "The Diabetes Prevention Program (DPP): Description of lifestyle intervention". In: *Diabetes Care* 25.12 (Dec. 1, 2002), pp. 2165–2171. ISSN: 0149-5992, 1935-5548. DOI: 10.2337/diacare.25.12.2165. URL: http://care.diabetesjournals.org/cgi/doi/10.2337/diacare.25.12.2165 (visited on 09/02/2021).

[5] David McClure and J. Reiter. "Differential Privacy and Statistical Disclosure Risk Measures: An Investigation with Binary Synthetic Data". In: *Trans. Data Priv.* 5 (2012), pp. 535–552.

[6] Zhanglong Ji, Zachary C. Lipton, and Charles Elkan. "Differential Privacy and Machine Learning: a Survey and Review". In: *arXiv:1412.7584 [cs]* (Dec. 23, 2014). arXiv: 1412.7584. URL: http://arxiv.org/abs/1412.7584 (visited on 09/01/2021).

[7] David Rubin. "Discussion of statistical disclosure limitation. Journal of Official Statistics". In: vol. 9. 1993, pp. 461–468. URL: https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/discussion-statistical-disclosure-limitation2.pdf.

[8] Ashish Dandekar, Remmy A M Zen, and Stéphane Bressan. "Comparative Evaluation of Synthetic Data Generation Methods". In: (2017), p. 5.

[9] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. "Privacy: Theory meets Practice on the Map". In: *2008 IEEE 24th International Conference on Data Engineering*. 2008 IEEE 24th International Conference on Data Engineering (ICDE 2008). Cancun, Mexico: IEEE, Apr. 2008, pp. 277–286. ISBN: 978-1-4244-1836-7 978-1-4244-1837-4. DOI: 10.1109/ICDE.2008.4497436. URL: http://ieeexplore.ieee.org/document/4497436/ (visited on 09/02/2021).

[10] Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. "Sampling Attacks: Amplification of Membership Inference Attacks by Repeated Queries". In: *arXiv:2009.00395 [cs]* (Sept. 1, 2020). arXiv: 2009.00395. URL: http://arxiv.org/abs/2009.00395.

[11] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. "The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks". In: *arXiv:1802.08232 [cs]* (July 16, 2019). arXiv: 1802.08232. URL: http://arxiv.org/abs/1802.08232.

[12] Cynthia Dwork. "Differential Privacy". In: *Automata, Languages and Programming*. Vol. 4052. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12. ISBN: 978-3-540-35907-4 978-3-540-35908-1. DOI: 10.1007/11787006_1. URL: http://link.springer.com/10.1007/11787006_1.

[13] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *FNT in Theoretical Computer Science* 9.3 (2013), pp. 211–407. ISSN: 1551-305X, 1551-3068. DOI: 10.1561/0400000042. URL: http://www.nowpublishers.com/articles/foundations-and-trends-in-theoretical-computer-science/TCS-042.

[14] John M. Abowd and Lars Vilhuber. "How Protective Are Synthetic Data?" In: *Privacy in Statistical Databases*. Ed. by Josep Domingo-Ferrer and Yücel Saygın. Vol. 5262. ISSN: 0302-9743, 1611-3349 Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 239–246. ISBN: 978-3-540-87470-6 978-3-540-87471-3. DOI: 10.1007/978-3-540-87471-3_20. URL: http://link.springer.com/10.1007/978-3-540-87471-3_20.

[15] Cynthia Dwork and Moni Naor. "On the Difficulties of Disclosure Prevention in Statistical Databases or The Case for Differential Privacy". In: *Journal of Privacy and Confidentiality* 2.1 (Sept. 1, 2010). ISSN: 2575-8527. DOI: 10.29012/jpc.v2i1.585. URL: http://www.journalprivacyconfidentiality.org/index.php/jpc/article/view/585 (visited on 11/07/2018).

[16] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. "DP-CGAN: Differentially Private Synthetic Data and Label Generation". In: *arXiv:2001.09700 [cs, stat]* (Jan. 27, 2020). arXiv: 2001.09700. URL: http://arxiv.org/abs/2001.09700.

[17] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. "Boosting and Differential Privacy". In: IEEE, Oct. 2010, pp. 51–60. ISBN: 978-1-4244-8525-3. DOI: 10.1109/FOCS.2010.12. URL: http://ieeexplore.ieee.org/document/5670947/.

[18] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. "The Composition Theorem for Differential Privacy". In: *IEEE Transactions on Information Theory* 63.6 (June 2017), pp. 4037–4049. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2017.2685505. URL: http://ieeexplore.ieee.org/document/7883827/ (visited on 08/20/2021).

[19] Franziska Boenisch. "Differential Privacy: General Survey and Analysis of Practicability in the Context of Machine Learning". In: (2019), p. 107.

[20] Ilya Mironov. "Renyi Differential Privacy". In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)* (Aug. 2017), pp. 263–275. DOI: 10.1109/CSF.2017.11. arXiv: 1702.07476. URL: http://arxiv.org/abs/1702.07476.

[21] Ilya Mironov, Kunal Talwar, and Li Zhang. "Rényi Differential Privacy of the Sampled Gaussian Mechanism". In: *arXiv:1908.10530 [cs, stat]* (Aug. 27, 2019). arXiv: 1908.10530. URL: http://arxiv.org/abs/1908.10530 (visited on 10/25/2020).

[22] Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. "Subsampled Rényi Differential Privacy and Analytical Moments Accountant". In: *arXiv:1808.00087 [cs, stat]* (Dec. 4, 2018). arXiv: 1808.00087. URL: http://arxiv.org/abs/1808.00087.

[23] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000056. arXiv: 1906.02691. URL: http://arxiv.org/abs/1906.02691 (visited on 06/28/2021).

[24] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Networks". In: *arXiv:1406.2661 [cs, stat]* (June 10, 2014). arXiv: 1406.2661. URL: http://arxiv.org/abs/1406.2661.

[25] J Schmidhuber. *Making the World Differentiable: On Using Self-Supervised Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environment*. Nov. 1990. URL: https://people.idsia.ch//~juergen/FKI-126-90ocr.pdf.

[26] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *arXiv:1411.1784 [cs, stat]* (Nov. 2014). arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784 (visited on 06/23/2021).

[27] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN". In: *arXiv:1701.07875 [cs, stat]* (Dec. 6, 2017). arXiv: 1701.07875. URL: http://arxiv.org/abs/1701.07875 (visited on 06/23/2021).

[28] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. "The Earth Mover's Distance as a Metric for Image Retrieval". In: (2000), p. 23.

[29] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. "Improved Training of Wasserstein GANs". In: *arXiv:1704.00028 [cs, stat]* (Dec. 25, 2017). arXiv: 1704.00028. URL: http://arxiv.org/abs/1704.00028 (visited on 06/23/2021).

[30] Aleksei Triastcyn and Boi Faltings. "Generating Artificial Data for Private Deep Learning". In: *arXiv:1803.03148 [cs, stat]* (Apr. 28, 2019). arXiv: 1803.03148. URL: http://arxiv.org/abs/1803.03148.

[31] Manhar Singh Walia, Brendan Tierney, and Susan McKeever. "Synthesising Tabular Datasets Using Wasserstein Conditional GANS with Gradient Penalty (WCGAN-GP)". In: *AICS 2020: 28th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin Ireland* (2020), p. 13. DOI: 10.21427/e6wa-sz92. URL: http://arrow.tudublin.ie/scschcomcon/289/.

[32] Roderick J.A. Little. "Statistical Analysis of Masked Data". In: vol. 9. 1993, pp. 407–426. URL: www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/statistical-analysis-of-masked-data.pdf.

[33] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *arXiv:1511.06434 [cs]* (Jan. 7, 2016). arXiv: 1511.06434. URL: http://arxiv.org/abs/1511.06434 (visited on 09/02/2021).

[34] Jost Tobias Springenberg. "Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks". In: *arXiv:1511.06390 [cs, stat]* (Apr. 30, 2016). arXiv: 1511.06390. URL: http://arxiv.org/abs/1511.06390 (visited on 09/02/2021).

[35] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation". In: *arXiv:1703.10847 [cs]* (July 18, 2017). arXiv: 1703.10847. URL: http://arxiv.org/abs/1703.10847 (visited on 09/10/2021).

[36] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks". In: *arXiv:1612.03242 [cs, stat]* (Aug. 4, 2017). arXiv: 1612.03242. URL: http://arxiv.org/abs/1612.03242 (visited on 09/10/2021).

[37] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. "Differentially Private Generative Adversarial Network". In: *arXiv:1802.06739 [cs, stat]* (Feb. 19, 2018). arXiv: 1802.06739. URL: http://arxiv.org/abs/1802.06739.

[38] James Jordon and Jinsung Yoon. "PATE-GAN: GENERATING SYNTHETIC DATA WITH DIFFERENTIAL PRIVACY GUARANTEES". In: (2019), p. 21.

[39] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data". In: *arXiv:1610.05755 [cs, stat]* (Mar. 3, 2017). arXiv: 1610.05755. URL: http://arxiv.org/abs/1610.05755.

[40] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. "Generating Multi-label Discrete Patient Records using Generative Adversarial Networks". In: *arXiv:1703.06490 [cs]* (Jan. 11, 2018). arXiv: 1703.06490. URL: http://arxiv.org/abs/1703.06490.

[41] Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. "Generation and evaluation of synthetic patient data". In: *BMC Med Res Methodol* 20.1 (Dec. 2020), p. 108. ISSN: 1471-2288. DOI: 10.1186/s12874-020-00977-1. URL: https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-020-00977-1.

[42] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS'16: 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna Austria: ACM, Oct. 24, 2016, pp. 308–318. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978318. URL: https://dl.acm.org/doi/10.1145/2976749.2978318 (visited on 09/18/2020).

[43] Andrew P. Bradley. "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern Recognition* 30.7 (July 1997), pp. 1145–1159. ISSN: 00313203. DOI: 10.1016/S0031-3203(96)00142-2. URL: https://linkinghub.elsevier.com/retrieve/pii/S0031320396001422 (visited on 06/24/2021).

[44] Peter G. Kopelman. "Obesity as a medical problem". In: *Nature* 404.6778 (Apr. 2000), pp. 635–643. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/35007508. URL: http://www.nature.com/articles/35007508 (visited on 07/21/2021).

[45] Ali H. Mokdad, Earl S. Ford, Barbara A. Bowman, William H. Dietz, Frank Vinicor, Virginia S. Bales, and James S. Marks. "Prevalence of Obesity, Diabetes, and Obesity-Related Health Risk Factors, 2001". In: *JAMA* 289.1 (Jan. 1, 2003), p. 76. ISSN: 0098-7484. DOI: 10.1001/jama.289.1.76. URL: http://jama.jamanetwork.com/article.aspx?doi=10.1001/jama.289.1.76 (visited on 07/21/2021).

[46] Dr Araz Rawshani. *The ST segment: physiology, normal appearance, ST depression & ST elevation*. URL: https://ecgwaves.com/st-segment-normal-abnormal-depression-elevation-causes/ (visited on 08/30/2021).

[47] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. "Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12". In: *arXiv:1709.02753 [cs]* (Sept. 11, 2017). arXiv: 1709.02753. URL: http://arxiv.org/abs/1709.02753 (visited on 09/08/2021).

[48] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved Techniques for Training GANs". In: *arXiv:1606.03498 [cs]* (June 10, 2016). arXiv: 1606.03498. URL: http://arxiv.org/abs/1606.03498 (visited on 06/23/2021).