Empirical evaluation of amplifying privacy by subsampling for GANs to create differentially private synthetic tabular data.*

Valtteri A. Nieminen*, Tapio Pahikkala and Antti Airola

University of Turku, Department of Computing, Finland

Abstract

Privacy concerns often limit sharing sensitive data collected from individuals. One proposed solution to make secondary use possible is privacy-preserving synthetic data that attempts to mimic real data. Due to their success on non-private tasks, GAN networks trained with differentially private stochastic gradient descent (DPSGD) have been popular for generating DP synthetic data. In recent years, a prominent approach to achieving better privacy guarantees has been to train ensembles of discriminator networks with DPSDG on mutually exclusive subsets to obtain better differential privacy guarantees by taking advantage of the synergy between GANs and privacy amplification by subsampling. However, this research has been done almost exclusively on images, and empirical evaluations of this strategy on other types of data are lacking. This work focuses on the effects of subsampling in creating DP synthetic tabular data with GANs. We evaluate synthetic data utility by training classification models on synthetic- and testing on real data at varying subsampling rates. Further, we complement the evaluation with a qualitative examination of the generated data. Our findings show that while subsampling does bring benefits with tabular data in terms of the prediction performance for classifiers trained on synthetic data, the resulting samples can be very distorted compared to original real data. The results suggest that the benefits obtainable via this method of training DP GAN can differ significantly based on the type of data used.

Keywords

Machine Learning, Differential Privacy, GAN, Synthetic Data, Privacy Amplification by Subsampling, Tabular Data

1. Introduction

The success of machine learning (ML) has created high demand for data as researchers and businesses alike seek to capitalize on advances in computational methods. One consequence of this has been a push to allow the secondary use of sensitive individual data, which currently can not be utilized due to privacy concerns and laws. The potential value in enabling the use of electronic health records (EHR), for example, has also inspired a wealth of research on privacy-preserving data analysis methods in the past decade.

Synthetic data, which attempts to mimic the statistical properties of some real data, has been proposed as a natural framework to make privacy-preserving data sharing possible. The idea is that synthetic data would be shared in place of real data as a privacy-preserving proxy. Synthetic data has proven to be an appealing proposition that has attracted significant interest both in and outside academia. Many tutorials have been published for the non-technical audience [1, 2], and multiple start-ups creating synthetic data for privacy purposes have sprung up around the subject in the last years.

Unfortunately, creating useful, high-quality synthetic data that would be privacy-preserving has proven to be a complex and difficult problem; it has repeatedly been shown that generative models and synthetic data produced with them are not inherently privacy-preserving but vulnerable against, for example, membership inference attacks [3, 4]. The gold standard to deal with this vulnerability has come to be combining generative models with *differential privacy* [5]. DP is a mathematical framework that can be used to quantify privacy. This quantification is expressed as a worst-case privacy guarantee; the maximum amount of private information leaked about an individual, denoted with ϵ [5].

Combining DP with generative models results in what is called *differentially private synthetic data*. For machine learning models, privacy guarantees can be attained via a DP optimization algorithm, the most popular being differentially private stochastic gradient descent (DPSDG) [6]. In DPSGD, per-example gradients are manipulated prior to updating model parameters. First, these gradients are clipped to some maximum norm, bounding each observation's maximum influence, referred to as sensitivity. Second, statistical noise sampled from a chosen distribution whose parameters depend on the strictness of the imposed guarantee ϵ is added to the gradient. Unfortunately, privacy does not come for free. While perturbation makes it worse, whether or not the calculation is perturbed, there is a tradeoff [5, 7] between privacy

TKTP 2023: Annual Symposium for Computer Science 2023, June 13–14, 2023, Oulu, Finland

*Corresponding author.

vajnie@utu.fi (V. A. Nieminen)

0000-0002-3550-0561 (V. A. Nieminen); 0000-0003-4183-2455 (T. Pahikkala); 0000-0002-1010-4386 (A. Airola)

and utility. How good DP synthetic data is, comes down to optimizing this tradeoff.

Due to their success in non-private settings, Generative adversarial networks (GAN) (see e.g. [8]), trained privately, have been among the most utilized model types to create DP synthetic data. GAN comprises two types of neural networks, a generator G and a discriminator D, both initialized simultaneously and trained in tandem in an adversarial setup [9]. The goal is that G learns to produce samples similar to those of the real data distribution based on the feedback of D. This feedback concerns how well D can discern whether the sample was synthetic or came from the real data distribution based on the approximation D has learned of the real data distribution during training.

DPSGD [6] was groundbreaking because it enabled the training of deep ML models with meaningful privacy guarantees. At the heart of why DPSGD made this possible is that it takes advantage of *privacy amplification by subsampling* (PABS) [6, 10, 5] a well-known result vital for modern privacy-preserving algorithms. PABS describes how privacy is amplified when an algorithm is run only on a subset of the whole data. In DPSGD, this result is applied to mini-batching, seeing each batch as a subset. Intuitively, amplification results from an adversary being unable to know which points were chosen for which iteration of the algorithm. The resulting improvement to privacy is very significant even for datasets of modest size and can, depending on the sampling strategy, be roughly as large as $\frac{m}{n} * \epsilon$, where m stands for the size of the subset and n the size of the whole training data.

While the first works on DP GANs took DPSGD and more or less simply combined it with a GAN to make information on the sensitive, real data flowing to the discriminator private (see e.g. [11, 12]), soon methods taking advantage of the specifics of GANs were introduced. PATE-GAN [13], took the PATE [14] mechanism, a version of the *subsample-and-aggregate* framework of DP [5, 15], and used it to train a GAN model using aggregated votes of an ensemble of multiple discriminators. Long et al. [16] took this approach further with G-PATE and noted that only the sensitive information flowing to the generator needs to be sanitized as only the generator is released (after the network is trained, discriminators are not needed for generating synthetic data). This allowed for training a large ensemble of discriminators on exclusive subsets of data, taking advantage of the synergy between PABS and the GAN setup, where discriminator and generator networks are separated.

Unlike G-PATE, where information D provides to G is discretized to votes, the work on GS-WGAN by Chen et al. [17] worked the subsample-and-aggregate idea into DPSGD, improving the results of previous works by aggregating the gradient of large discriminator ensembles (> 1000) to train the generator. This application of subsample-and-aggregate [5, 15] to DPSGD results in only the updates to G network cost privacy while D can be trained without privacy costs, while also reaping the benefits of PABS by training multiple discriminators. This opens up many possibilities for further optimization: for example, the D networks can be pre-trained before updates to G and D trained for more iterations G. However, further research on this method has been lacking, with some claiming that the large number of networks trained hinders practical usability due to the time and resources taken to train so many networks [18].

Most DP GAN synthetic data generation research has been conducted on image data. However, the overwhelming majority of, for example, health data is in tabular form. Here, tabular data is referred to as data where observations are on rows and columns represent features of those observations, that may or may not be of mixed type. At time, Chen et al., achieved state-of-the-art results using the handwritten digits MNIST [19] and Fashion-MNIST image datasets [20]. The absence of empirical results on tabular data leaves open interesting questions on data modality and the usability of this method to train DP GAN. As said, the approach requires dividing the data into many mutually exclusive subsets. Unlike tabular data, where features may or may not be correlated or important for some task at hand, the features of images are autocorrelated, as they depict parts of a whole.

This paper presents an empirical investigation into generating DP tabular synthetic data using a GAN trained with the subsampled DPSGD strategy presented by Chen et al. [17]. We conduct experiments with tabular data using the freely available Cardio [21] dataset. The experiments include a standard downstream classification utility task in which classifiers are trained on synthetic and tested on real data. Unlike previous works, we focus on the effect of this training strategy in particular by varying the number of discriminators trained with mutually exclusive subsets across the experiments. The downstream classification utility experiment is augmented with a qualitative examination of the structure of the synthetic data generated. To the authors' knowledge, this work is the first to present an evaluation that focuses on subsampled DPSGD training of DP-GAN with tabular data rather than images.

2. Preliminaries

2.1. Differential privacy

A randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private if for adjacent datasets D_1 and D_2, meaning they differ by at most one record, and for all measurable sets S of outputs, the following inequality holds:

$$Pr[\mathcal{M}(D_1) \in S] \leq e^{\epsilon} Pr[\mathcal{M}(D_2) \in S] + \delta \quad (1)$$

Where $\mathcal{M}$ is, for example, one iteration of DPSGD training, ϵ is the upper bound for privacy loss, and δ is a small probability of a catastrophic breach of the DP-guarantee [5]. A smaller ϵ stands for a stronger guarantee. Although a single acceptable value for the privacy budget ϵ can not be given as it depends on the context, in the literature, values of $\epsilon \leq 1$ have been seen as very strong protection [22] and depending on the type of data and task, values of $\epsilon \leq 10$ have been seen to still result in meaningful guarantees [6]. Informally ϵ can be said to depict the worst-case of how much information, that can not be learned from other individuals data, can be learned from the output concerning a specific individual.

The model in this work uses Rényi differential privacy (RDP) [23], another formulation of DP often used with DP deep learning models to get tighter bounds of composing DP guarantees over iterations. In this paper, due to interpretability, Rényi DP bounds are converted to (ϵ, δ). The privacy loss of training is tracked via the subsampled Rényi moments accountant [24]. There exist many ways to compose privacy costs of sequential runs of a DP algorithm. Naively, this is a summation, but by using advanced techniques, a more efficient composition can be achieved.

This work uses differentially private stochastic gradient descent (DPSGD) [6] to optimize the DP GAN. DPSGD differs from its non-private counterpart in that prior to updating model parameters, the maximum influence of an individual data point can have on the output, called the *sensitivity* [5] of the function is bounded by clipping gradients. Clipping is followed by adding noise from a *noise mechanism*. Noise mechanisms like the *gaussian mechanism* used in this work are functions from which noise calibrated to a specific sensitivity can be sampled [25]. The choice of noise mechanism largely depends on the type of information sanitized.

A DPSDG training step, that is, one run of the DP optimization algorithm, can be summarized as follows:

1. Gradients before sanitation are calculated with backpropagation as in non-DP SGD. At training step t these are: $\nabla^t_{\mathcal{L}}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G\right)$, where θ_D and θ_G are the discriminator and generator network's weights.
2. Gradient information is sanitized by bounding the sensitivity, clipping the gradient vectors to a maximum of C, and adding noise:

$$\hat{\boldsymbol{\nabla}}^t = \mathcal{M}_{\sigma,C}\left(\boldsymbol{\nabla}^{(t)}\right) = \text{clip}\left(\boldsymbol{\nabla}^{(t)}, C\right) + \mathcal{N}\left(0, \sigma^2 C^2 \boldsymbol{I}\right)$$

3. The parameters of the model are updated using the sanitized gradients as in normal gradient descent: $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} - \eta \cdot \hat{\boldsymbol{\nabla}}^{(t)}$

2.2. DP synthetic data

DP synthetic data generation is possible due to the post-processing property [5] of differential privacy, which guarantees that outputs, in this case, synthetic data, of any process that is DP are also DP. Importantly, DP guarantees are not actually over the synthetic data but the algorithm that generated it.

Evaluation of DP synthetic data can be said [1] to lie on three axes: *privacy*, *utility*, and *fidelity* (also sometimes called sample quality). ***Utility*** is simply the usefulness of the synthetic data for a given task. In this work, the downstream classification task, where a *downstream model*, a model trained on synthetic data, is evaluated on real data, is concerned with utility. ***Fidelity***, refers to, how closely the statistical properties of the synthetic data are preserved. What exactly this means differs on the measure used, but often, for example, correlational structure and distributions are compared between synthetic and real data.

2.3. GAN

Generative Adversarial Networks [9] are a type of generative model where training is formulated as a competitive game between two networks: a generator G and a discriminator D. The goal is that G learns a mapping from some bounded domain, usually a noise vector denoted with $\mathbf{z}$, to an approximation of the distribution of real data based on the dis-network's feedback.

The G can be used to generate samples from the distribution it has learned, that is, p_{model}, by feeding $\mathbf{z}$ to the network as input: $G(\mathbf{z})$. D discriminates between this generated output $G(\mathbf{z})$ and real data. GAN have been thought to have some inherent privacy attributes, such as resistance to overfitting [8], because G only interacts with the real data indirectly by receiving information from D, which does not define an explicit density function but learns an approximation during training. Few works on these inherent properties exist, but even non-DP GAN have been shown to provide some weak protection against membership inference attacks [26].

The model used in this work is based on a Wasserstein GAN with gradient penalty (WGAN-GP) [27]. In the context of Wasserstein GAN, [27], the discriminator is called a critic, but in this work, it is referred to as a discriminator as well to avoid extra terminology.

The choice of the Wasserstein loss and use of gradient penalty [27] is non-trivial as it has privacy-synergies with DPSGD clipping [17]. The Wasserstein loss is based on the Earth Mover's distance (EMD). For EMD to be

valid, the 1-Lipschitz continuity condition must hold (see Definition 1).

Definition 1 (Lipschitz-continuity). *A function* $f : \mathbb{R}^n \to \mathbb{R}^m$ *is globally L-Lipschitz continuous if there exists an* $L \geq 0$ *such that* $\|f(x) - f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^n$

If the continuity holds, gradient magnitudes during training are approximately between $[-1, 1]$ [17]. The gradient penalty regularization term [27] is used as a soft restraint to make the condition hold. This is beneficial for DPSGD training, as then setting the clipping bound $C = 1$ should be a close to optimal choice and a costly search for the hyperparameter value is avoided [17].

Definition 2 (Wasserstein-1 loss of D and G [27]).

$$\mathcal{L}_D = -\mathbb{E}_{\boldsymbol{x}\sim\hat{p}_{data}}[D(\boldsymbol{x})] + \mathbb{E}_{\tilde{\boldsymbol{x}}\sim p_{model}}[D(\tilde{\boldsymbol{x}})] + \mathbf{GP}$$
$$\mathcal{L}_G = -\mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))]$$

Where $P_{\boldsymbol{z}}$ *is the noise sampled from a normal distribution given as input to G to generate samples,* λ *is the regularization strength hyperparameter of the gradient-penalty term,* $\boldsymbol{x}$ *is the real data and* $\tilde{\boldsymbol{x}}$ *is the data generated by G. GP is the gradient penalty term* $\lambda\mathbb{E}\left[(\|\nabla_D(\alpha\boldsymbol{x} + (1-\alpha)\tilde{\boldsymbol{x}})\|_2 - 1)^2\right]$ *and* $\alpha \sim \mathcal{U}[0, 1]$ *is the interpolation coefficient and* ∇_D *[27].*

2.4. Subsample-and-aggregate and privacy amplification

The work of Chen et al., [17] can be seen to be a successor to a line of works, especially the G-PATE [16], which adapts the subsample-and-aggregate [5] framework of DP, first formalized by Nissim et al. [15], to DPSGD and training of multiple discriminators on a GAN setup to reap privacy amplification by subsampling (PABS) benefits.

Privacy amplification by subsampling is a well-studied subject with bounds for different sampling strategies, such as without replacement or with replacement having been worked out extensively, especially in the works of Kasiviswanathan et al., [28] and Balle et al., [10]. PABS is induced in the model of this work by training a large number of D networks on mutually exclusive subsets and randomly querying them at each G update step. This corresponds to PABS for sampling without replacement [10], with an amplification effect roughly proportional to $\frac{n}{m}$, where m is the number of mutually exclusive subsets the data is split into and n the size of the whole training data.

Figure 1 depicts the sanitation of gradients [17] during the update steps of the generator. As seen from the figure, *where* the sanitation bound, or "privacy barrier" as called by [17] is placed "between" the two networks. This is an important emphasis, because, it is what allows training the D networks without incurring privacy costs. If the sanitation would be between, for example, the real data and the D networks, every time they see real data would result in a privacy cost.

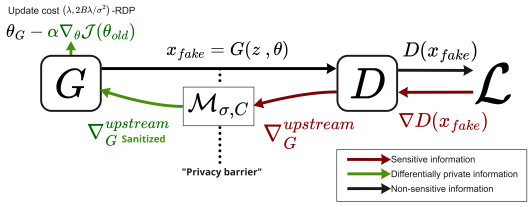

Figure 1: Flow of information for one DPSGD update step of the Generator modified from the work of [17]. The information that flows from G to D is not sanitized, but the information from D, that sees the private data to G is. The notation $D(\boldsymbol{x}_{fake})$ refers in this picture to the "fake" examples produced by G being given to D to get the value for the loss. Sensitive information is marked with a red color, whereas non-sensitive (not depending on real data), information is marked with a black color. Green denotes sanitized information, that is information under a DP guarantee.

3. Materials and methods

3.1. Model specifications

The freely available code [29] of GS-WGAN by Chen et al., [17] was used as a basis of the implementation, but the architecture and gradient clipping procedure code was re-implemented to fit the tabular data case, replacing the convolutional architecture with a fully-connected one using Pytorch (v. 1.4.0) [30]. Choices for the new architecture specifications were made based on a limited number of tests with less than five training runs with different seeds per choice, such as the width or depth of the network and the number of D training repetitions per generator iteration, denoted n_{dis}. Unless mentioned here, hyperparameter choices were those recommended by [27].

The G network used was a fully connected network with two hidden layers, the largest being of size 256 with 16 outputs. The size of the noise vector $\mathbf{z}$ was set to 32, based on experimenting on a few usual settings. The activation function used was ReLU [31], except in the last layer of where a hyperbolic tangent (TanH) was used, due to the range of the Wasserstein loss function (both [-1, 1]). The D classifier network was a typical multi-layer-perceptron architecture with one hidden layer, size 128. Instead of a ReLU, as in the G network, a LeakyReLU [32] with α (negative slope) value set to 0.2 was used in the hidden layers as in [27].

Table 1
Features of the Cardio dataset

Feature	Type
Age	numeric (days)
Height	numeric (cm)
Weight	numeric (kg)
Gender	binary
Systolic blood pressure (ap_lo)	numeric
Diastolic blood pressure (ap_hi)	numeric
Cholesterol	categorical 1 = normal 2 = above normal 3 = well above normal
Glucose	categorical 1 = normal 2 = above normal 3 = well above normal
Smoking	binary
Alcohol intake	binary
Physical activity	binary
Cardiovascular disease (cardio)	binary

3.2. Data

The publicly available Cardiovascular Disease dataset [21] consists of 70 000 observations and 12 features, five binary, two categorical, and five numeric. The classification target is to predict the presence of cardiovascular disease (the feature 'Cardio'). Table 1 lists all features and their types. This dataset was chosen for reproducibility and to work as a feasible proxy for common tabular EHR data; the condition to predict is common, and the features are routinely collected during doctor's examinations. The number of patients with cardiovascular disease in the dataset is balanced. Blood pressure values were limited to a range of +-20 from values indicating a hypertensive crisis according to Finnish national standards [33], affecting 1064 values of ap_hi and 312 values of ap_lo. KNN-imputation (see [34]) with $k = 3$ was carried out to replace these, using the implementation from the scikit-learn package [35] (version 1.0.2). The two categorical variables 'cholesterol' and 'glucose' were one-hot encoded, after which all features were min-maxed to [-1,1] to match the feature values range of the G network output layer.

4. Experiments and evaluation

The quality of synthetic data is evaluated from two viewpoints; downstream classification utility and sample fidelity. This section gives an overview of the downstream utility experiment conducted and the DP synthetic data generation process. The downstream classification utility experiment is depicted in Algorithm 1.

4.1. Downstream utility experiment

Downstream classification utility is a standard way of evaluating DP synthetic data and the method used to generate it (see e.g. [36, 17, 37]). In this experiment, synthetic data is used to train a *downstream model*, which is tested against real data. In this work's binary classification task, a logistic regression (LR) classifier from the [35] (version 1.0.2) package was used as the model of choice for classification and accuracy was measured with the AUC metric [38].

Five private Generators were trained for the downstream classification task each up to a maximum of 40 000 iterations, using an amount of pre-trained D networks corresponding to the subsampling rates, $\gamma = 1/250, 1/500, 1/750, 1/1000, 1/1500$, that is fraction of real data in each mutually exclusive subset. In addition, a non-private G network was trained to compare the effects of the generating process only. In all cases, the discriminators D were pre-trained for 2000 iterations.

Every model was saved once per 1000 iterations, in this work referred to as *checkpoints*. Each of these saved states of the model were evaluated separately. Hyperparameter optimization over the choice of regularization term and regularization strength was conducted for the logistic regression classifier separately for each checkpoint at different iterations. The real and synthetic data used for this experiment were split to training, validation, and test sets with size corresponding to fractions (0.8/0.1/0.1) of the real dataset. The resulting set sizes were 56000 for the synthetic training, for the synthetic validation and 7000 for the real test set.

A total of 287 full model selection runs (40 checkpoints at each of the 6 different γ settings and the additional real versus real baseline case) consisting of hyperparameter optimization and evaluation with the best hyperparameters settings were conducted.

4.2. Sample fidelity

A comparative assessment of the effects of different privacy- and subsampling levels on the method's ability to retain sample fidelity in this work consists of three comparisons; correlational structure using Spearman's rank correlation coefficient, a visual examination of the change of the continuous feature distributions, and a visual examination of the change in binary and categorical variable distributions.

5. Results

5.1. Downstream classification utility

Figure 2 compares results of the downstream classification utility experiment (train on synthetic, test on

Algorithm 1: Downstream Classification Quality Experiment

Create sets $h \in \mathbf{H}$, where $\mathbf{H}$ denotes combinations between 20 values of c_{reg} randomly sampled from a logarithmic space between (0, 1) and regularization term choice of either $l1$ or $l2$
for *subsampling rate* γ *in {1, 1/250, 1/500, 1/750, 1/1000, 1/1500}* **do**
  pretrain K (denominator of γ), networks D_k^γ
  train (M_γ using D^γ), save "checkpoint models" $M_\gamma^{1k}, M_\gamma^{2k} \ldots M_\gamma^{40k}$ every 1000 iterations
for γ *in {1, 1/250, 1/500, 1/750, 1/1000, 1/1500}* **do**
  for *i in {1k, 2k, …, 40k}* **do**
    split dataset $\mathbb{X}$ of size n with stratification by the Y variable 'cardio' to $\mathbb{X}_{train}$, $\mathbb{X}_{val}$, and $\mathbb{X}_{test}$ with proportions 0.8/0.1/0.1 * n.
    sample s $= 0.9 * n$ synthetic data points from M_γ^i and split with stratification to $synth_{train}$ and $synth_{val}$
    for $h \in \mathbf{H}$ **do**
      train classifier $LR_{validation}$ with hyperparameters h and data $synth_{train}$
      evaluate $LR_{validation}$ against $synth_{val}$
      save best h in h_{best}
    train classifier LR_{test} with h_{best} and data $synth_{train}$ combined with $synth_{val}$
    evaluate LR_{test} using $\mathbb{X}_{test}$
    save the result of the downstream classification utility test for M_γ^i
    empty the set h_{best}

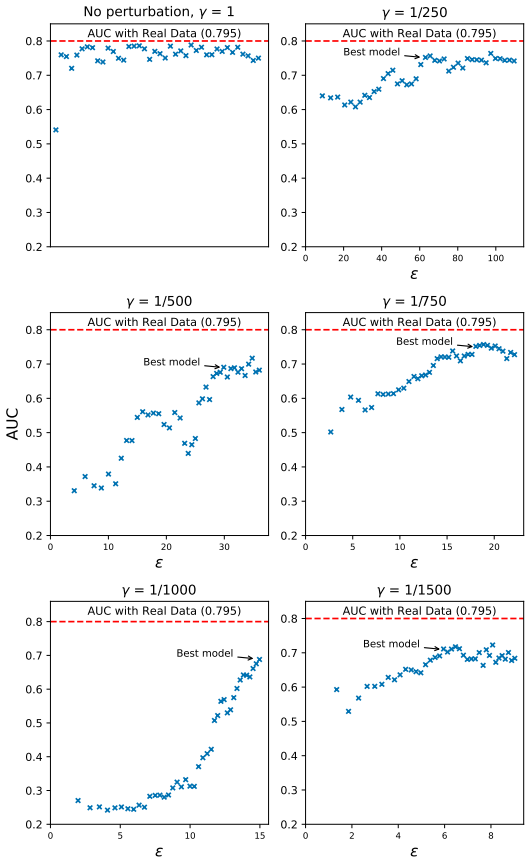

Figure 2: Downstream classification task results training a logistic regression model on synthetic and testing on real data, comparing AUC and ϵ with a non-private model $M_{baseline}$ and models with differential privacy using a noise scale of σ = 1.07 and a clipping bound C = 1. Each marker represents a 'checkpoint' at every 1000 iterations, where a synthetic dataset was sampled. The model at a checkpoint with the lowest ϵ to AUC ratio close to the highest AUC of the checkpoint evaluations for a specific model is chosen as the "best model". The visualizations of distributions in Figure 3 are from data generated by these "best" models.

real data) with multiple synthetic datasets sampled at different checkpoints (every 1000 iterations) of five differentially private models with different subsampling rates γ. The models are denoted M_{250} for $\gamma = 1/250$, M_{500} for $\gamma = 1/500$, M_{750} for $\gamma = 1/750$, M_{1000} for $\gamma = 1/1000$ and M_{1500} for $\gamma = 1/1500$ and are compared to a non-private Generator denoted $M_{baseline}$.

The best overall results in terms of the ϵ - AUC tradeoff were achieved with synthetic data sampled from the model with the highest subsampling rate, M_{1500} with $\epsilon = 6.45$ and AUC $= 0.717$. Compared to the real data logistic regression (LR) model AUC = 0.795, the difference was 0.078. Accounting for the loss of information caused by generating data with any model, that is, when compared to the results obtained with synthetic data sampled from the non-private $M_{baseline}$ (AUC = 0.788) this difference drops to 0.071.

Other private models required more than double the privacy budget to reach classification accuracy similar to M_{1500} with the closest model, M_{750} achieving AUC = 0.715 at $\epsilon = 13.9$. The next best tradeoff was obtained with data sampled from M_{1000}, which ultimately failed to reach the same AUC, with its highest score being 0.687 at ϵ = 14.7.

In general, models with weaker privacy guarantees and a smaller subsampling rate were able to reach higher values of AUC eventually, but at high privacy costs. In comparison to M_{1500} that reached the best tradeoff, for example, M_{500} reached the value 0.717, close to that of M_{1500} at $\epsilon = 34.8$, nearly six times more. The best AUC value obtained with privacy-preserving models was reached by M_{250} at an AUC of 0.752 with $\epsilon = 63.0$.

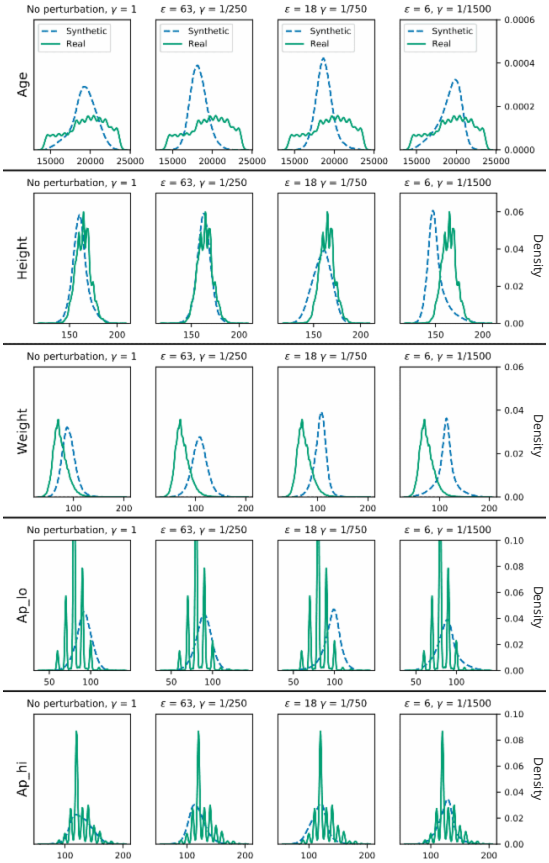

Figure 3: Comparison of distributions of continuous features of synthetic data produced by the best performing models of the downstream classification task based on AUC to ϵ tradeoff. Note that the y-axis varies in range. Real data distributions are included for comparison. Columns are different models at different sampling rates γ and ϵ values whereas rows are the different continuous features.

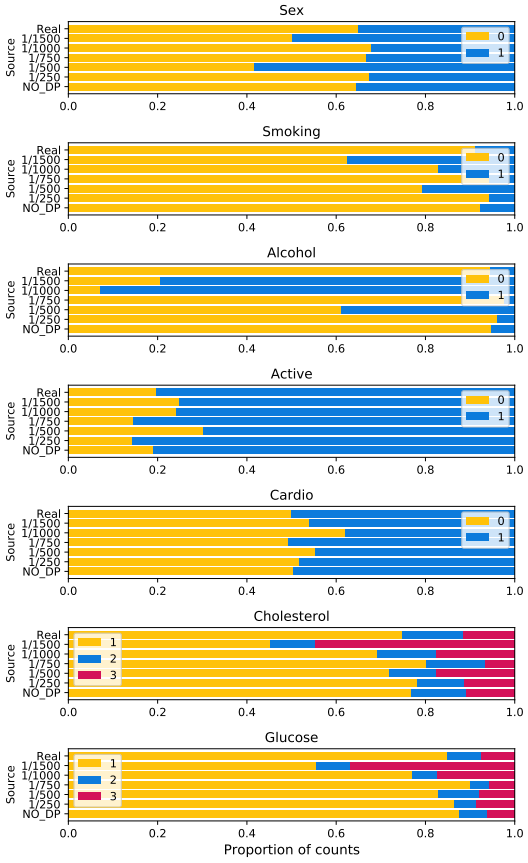

Figure 4: Comparison of frequencies of binary and categorical features between data sampled from the best performing models in the downstream utility experiment. Real data distributions are included for comparison.

5.2. Sample fidelity

Figure 3 shows distributions of continuous features generated by models performing best in the downstream classification task at settings M_{250}, M_{750}, M_{1500} as well as the non-private $M_{baseline}$, all compared with the real data feature distributions. Note that the y-axis density value range varies to provide better resolution for each variable. Interestingly, there is visible x-axis shift in especially the samples from models where γ is larger.

Figure 4 compares binary and categorical feature distributions of data sampled from DP models, the baseline model $M_{baseline}$ and real data. $M_{baseline}$ appears to capture distributions of the real data well, but when DP is applied there are considerable deviations from the real data case, especially with M_{1500} with stricter guarantees $\epsilon = 6$ and a higher γ. In the case of features where the number of positive cases is low to begin with, such as 'alcohol', adding DP seems to often further decrease the amount of positives. For the categorical features 'cholesterol' and 'glucose', stronger privacy guarantees such as in the case of M1500 seem to also balance the size differences between the counts.

Figure 7.3 shows a comparison between Spearman Rank correlation coefficient values calculated between the continuous features across synthetic data sampled from the best performing models shown in 7.2. Significant correlations are marked with (*) for a significance level of p < 0.05 and (**) for p < 0.01. Even in the case of the non-private synthetic data sampled from $M_{baseline}$, many of the dependencies in the real data are lost, as is the case with, for example, the correlation between 'weight' and 'ap_hi'. In addition, the synthetic datasets, especially those sampled from private models show new

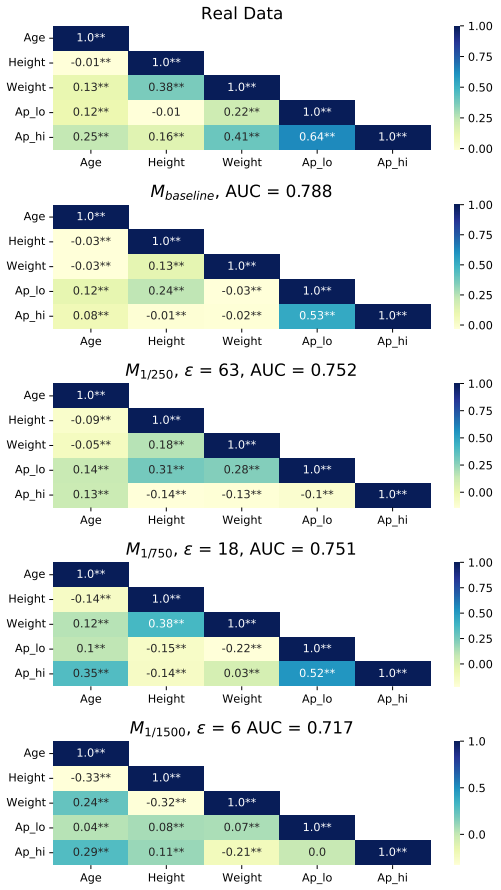

Figure 5: Spearman rank correlation coefficients of continuous variables compared between synthetic and real dataset across three of the best performing private models in the downstream classification utility experiment. The synthetic data generated by the non-private $M_{baseline}$ as well as the real data case are included for comparison. One asterisk (*) denotes a significant at a p-value < 0.05 and ** marks significance at level < 0.01.

correlations that are not present in the real data.

6. Conclusion

In the downstream classification utility task (see Figure 2), the private models, especially those with higher subsampling rates, required more training iterations to obtain high AUC values. However, the benefits to privacy attained with the subsample-and-aggregate DPSGD strategy outweighed these costs. This result suggest that the benefits seen with PABS and image data by Chen et al. [17] can also be reaped in terms of downstream tasks when tabular data with mixed features is used. However as opposed to the case of image data, this seems to happen at the cost of fidelity.

The sample fidelity results of Chen et al. [17] show that images retain fidelity uniformly and, the picture is still very much recognizable. In contrast, the synthetic tabular data produced in this study, especially with DP models at stricter privacy guarantees, have substantial deviations in all the fidelity examinations when compared to real data. This is especially evident in the binary feature distributions when there are few positive observations to begin with, which corresponds with other results showing that DPSGD training affects imbalanced feature distributions disparately [39]. In addition, some correlations turn almost opposite and from non-significant to significant.

The described effect worsens with the subsampling rate rising, but it does not affect the downstream classification metric nearly as much. This could be indicative of the differences of tabular and image data hypothesized earlier: tabular data features may or may not be correlated or important for some task at hand, while the features of images are autocorrelated, as they depict parts of a whole. Tabular data may lose almost all signal in some features, while with images, the perturbation is applied more evenly due to autocorrelation and less imbalance in the distributions.

A clear limitation of this paper is that it only uses one dataset. This is due to the significant computational expenses; the time it took to train a whole model and conduct the experiment at one subsampling rate exceeded 15 hours for the model with the highest subsampling rate $\gamma = 1/1500$, not counting the hyperparameter optimization using three NVIDIA RTX Titan GPU's. Subsample-and-aggregate-based methods have been criticized for being computationally expensive [18]. However, this question is not as black and white because, although expensive, subsampling could incur great privacy benefits for some types of data while only having a small detrimental effect on model performance.

Compared with other works using the same data, Fang et al., [40] reported better results. However, it has been since noted that their approach of adding DP to the conditional GAN of [41] is not DP since it oversamples the data and the DP mechanism is not random. RDP-CGAN of Torfi et al., [36] reported results visually in a figure of approximately $AUC = 0,72$ at $\epsilon = 10$, which fall short of the results of this work.

The results of this study suggest that subsample-and-aggregate DPSDG training also brings benefits with tabular data; however, with a higher cost to fidelity than with images. From a broader perspective, this work adds to the line of thought [1], that useful DP synthetic data can be made specifically for some problems, but making "general" synthetic data, where all features would be preserved well and which could be used like real data is very

difficult if not impossible.

Future work could have a closer look at how benefits from subsampling and data structure relate, using, for example, simulated data to control more parameters. Another direction relates to taking advantage of free training of the discriminators. For example, ways to track when generator training steps would be optimal with this method could result in significant benefits.

7. Acknowledgements

This work has been conducted as part of the PRIVASA project funded by Business Finland (grant number 37428/31/2020).

References

[1] J. Jordon, L. Szpruch, F. Houssiau, M. Bottarelli, G. Cherubin, C. Maple, S. N. Cohen, A. Weller, Synthetic data – what, why and how?, 2022. URL: https://arxiv.org/abs/2205.03257. doi:10.48550/ARXIV.2205.03257.

[2] A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, J. Honaker, K. Nissim, D. R. O'Brien, T. Steinke, S. Vadhan, Differential privacy: A primer for a non-technical audience, Vand. J. Ent. & Tech. L. 21 (2018) 209.

[3] J. Hayes, L. Melis, G. Danezis, E. De Cristofaro, Logan: Membership inference attacks against generative models, arXiv preprint arXiv:1705.07663 (2017).

[4] D. Chen, N. Yu, Y. Zhang, M. Fritz, Gan-leaks: A taxonomy of membership inference attacks against generative models, in: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security, 2020, pp. 343–362.

[5] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Foundations and Trends in Theoretical Computer Science 9 (2014) 211–407.

[6] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 308–318.

[7] T. Stadler, B. Oprisanu, C. Troncoso, Synthetic data – anonymisation groundhog day, in: 31st USENIX Security Symposium (USENIX Security 22), USENIX Association, Boston, MA, 2022, pp. 1451–1468. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/stadler.

[8] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, arXiv preprint arXiv:1701.00160 (2016).

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[10] B. Balle, G. Barthe, M. Gaboardi, Privacy amplification by subsampling: Tight analyses via couplings and divergences, Advances in Neural Information Processing Systems 31 (2018).

[11] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, Differentially private generative adversarial network. corr abs/1802.06739 (2018), arXiv preprint arXiv:1802.06739 (2018).

[12] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, R. Lee, S. P. Bhavnani, J. B. Byrd, C. S. Greene, Privacy-preserving generative deep neural networks support clinical data sharing, Circulation: Cardiovascular Quality and Outcomes 12 (2019).

[13] J. Jordon, J. Yoon, M. Van Der Schaar, Pate-gan: Generating synthetic data with differential privacy guarantees, in: International conference on learning representations, 2019.

[14] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, Ú. Erlingsson, Scalable private learning with PATE, arXiv preprint: 1802.08908 (2018).

[15] K. Nissim, S. Raskhodnikova, A. Smith, Smooth sensitivity and sampling in private data analysis, in: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, 2007, pp. 75–84.

[16] Y. Long, S. Lin, Z. Yang, C. A. Gunter, B. Li, Scalable differentially private generative student model via PATE, arXiv preprint : 1906.09338 (2019).

[17] D. Chen, T. Orekondy, M. Fritz, GS-WGAN: A gradient-sanitized approach for learning differentially private generators, Advances in Neural Information Processing Systems 33 (2020) 12673–12684.

[18] T. Cao, A. Bie, A. Vahdat, S. Fidler, K. Kreis, Don't generate me: Training differentially private generative models with sinkhorn divergence, Advances in Neural Information Processing Systems 34 (2021) 12480–12492.

[19] L. Deng, The MNIST database of handwritten digit images for machine learning research, IEEE Signal Processing Magazine 29 (2012) 141–142.

[20] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).

[21] S. Ulianova, Cardiovascular disease dataset, 2019. URL: https://www.kaggle.com/sulianova/.

[22] C. Arnold, M. Neunhoeffer, Really useful synthetic data–A framework to evaluate the quality of differentially private synthetic data, arXiv preprint arXiv:2004.07740 (2020).

[23] I. Mironov, Rényi differential privacy, in: 2017

IEEE 30th computer security foundations symposium (CSF), IEEE, 2017, pp. 263–275.

[24] Y.-X. Wang, B. Balle, S. P. Kasiviswanathan, Subsampled Rényi differential privacy and analytical moments accountant, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1226–1235.

[25] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of cryptography conference, Springer, 2006, pp. 265–284.

[26] Z. Lin, V. Sekar, G. Fanti, On the privacy properties of gan-generated samples, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 1522–1530.

[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of Wasserstein GANs, Advances in neural information processing systems 30 (2017).

[28] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, A. Smith, What can we learn privately?, SIAM Journal on Computing 40 (2011) 793–826.

[29] D. Chen, GS-WGAN github-repository, https://github.com/DingfanChen/GS-WGAN, 2020.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[31] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition?, in: 2009 IEEE 12th international conference on computer vision, IEEE, 2009, pp. 2146–2153.

[32] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., Rectifier nonlinearities improve neural network acoustic models, in: Proc. icml, volume 30, Citeseer, 2013, p. 3.

[33] The Finnish Medical Society Duodecim, Current Care Guidelines: Treatment of hypertensive crisis, 2020. URL: https://www.kaypahoito.fi/hoi04010.

[34] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R. B. Altman, Missing value estimation methods for DNA microarrays, Bioinformatics 17 (2001) 520–525.

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

[36] A. Torfi, E. A. Fox, C. K. Reddy, Differentially private synthetic medical data generation using convolutional GANs, Information Sciences 586 (2022) 485–500.

[37] Y. Tao, R. McKenna, M. Hay, A. Machanavajjhala, G. Miklau, Benchmarking differentially private synthetic data generation algorithms, CoRR abs/2112.09238 (2021). URL: https://arxiv.org/abs/2112.09238. arXiv:2112.09238.

[38] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern recognition 30 (1997) 1145–1159.

[39] V. M. Suriyakumar, N. Papernot, A. Goldenberg, M. Ghassemi, Chasing your long tails: Differentially private prediction in health care settings, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 723–734.

[40] M. L. Fang, D. S. Dhami, K. Kersting, Dp-ctgan: Differentially private medical data generation using ctgans, in: Artificial Intelligence in Medicine: 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Halifax, NS, Canada, June 14–17, 2022, Proceedings, Springer, 2022, pp. 178–188.

[41] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, Advances in Neural Information Processing Systems 32 (2019).