UNECE

CONFERENCE OF EUROPEAN STATISTICIANS

Expert Meeting on Statistical Data Confidentiality

1-3 December 2021, Poland

Generating tabular data using generative adversarial networks with differential privacy.

Giacomo Astolfi (European Central Bank)
giacomo.astolfi@ecb.europa.eu

Abstract

Generation of synthetic data has shown many advantages over masking for data privacy. Depending on the application, data generation faces the challenge of faithfully reproducing the statistical properties of the original dataset. In this area, artificial neural nets are a powerful tool, because they can learn and reproduce even highly non-linear relations within the original dataset. Generative Adversarial Neural Nets (GANs) is a state of the art approach in machine learning, capable of generating data with high resemblance. GANs have been explored heavily with image data, but are rather unexplored in the field of tabular data sets.

Here we build upon previous work by Xu et Al. (2019) employing GANs to learn to generate tabular data with numerical and categorical attributes. We show how this approach can be joined with differential privacy to provide the data holder with a privacy control mechanism.

Generating Tabular Data using Generative Adversarial Networks with Differential Privacy

Astolfi Giacomo* and Köpfer David**

* European Central Bank, giacomo.astolfi@ecb.europa.eu
** European Central Bank, david.kopfer@ecb.europa.eu

Abstract: Generation of synthetic data has shown many advantages over masking for data privacy. Depending on the application, data generation faces the challenge of faithfully reproducing the statistical properties of the original dataset. In this area, artificial neural nets are a powerful tool, because they can learn and reproduce even highly non-linear relations within the original dataset. Generative Adversarial Neural Nets (GANs) is a state-of-the-art approach in machine learning, capable of generating highly accurate tabular data. GANs have been explored heavily with image inputs but are less developed in the field of tabular data sets. Here we build upon previous work by Xu et Al. employing GANs to learn to generate tabular data with numerical and categorical attributes. We show how this approach can be joined with differential privacy to provide the data holder with an incremental privacy control mechanism.

Keywords: Machine Learning, Differential Privacy, Microdata Protection, Neural Networks, CTGAN

1 Introduction

The quote "Data is the new Oil" by Clive Humby is a widely accepted paradigm in the Information age. Just like the extraction of the natural resource, the collection of data comes with risks: leaking confidential data can potentially be very harmful for individuals, companies, and societies at large. Data Maintainers do not only need to make sure to protect sensitive information but also need to be able to bring data on different systems to make it available to researchers or the public. To make this possible, we propose a machine learning model that can make use of Deep Learning capabilities to produce useful data while keeping the individuals data private using Differential Privacy. We know from Ullmann et Al. [1] that generating private synthetic data can be very challenging, therefore it is appropriate to use techniques like Generative Adversarial Neural Networks.

1.1 Generative Neural Network

Generative Neural Networks (GAN) is a class of machine learning frameworks originally designed by Ian Goodfellow and his colleagues in 2014 [3]. These models aim at reproducing as accurately as possible the distribution of the data given as input. GANs are based composed of two neural networks that train in turn against each other, the Generator and the Discriminator. At each iteration, the discriminator is trained to differentiate between the real data and fake data, produced by the generator. In a second step, generator is trained to produce data passing as real data in the discriminator. Thus,

the discriminator successively infers how the real data looks like based on the performance of the discriminator. After several iterations of successive training rounds of discriminator and generator, the generator will reproduce the structure of the data very consistently, despite never having been in direct contact with it.
GAN Models have been largely employed in a wide variety of computer vision applications and are the state of the art for accomplishing data generation tasks. These models have been so successful, because once trained, they can generate many images without needing to store the original data. Thus, they can be trained in one place and easily used in another context. This greatly simplifies the usage of such model in production environments, as data can just be generated on demand. However, these models have rarely been used with tabular data and, until now, did not consider anonymity of data. In this work we want to show, how these issues can be addressed, and we will propose a methodology capable of generating both private and accurate representations of tabular datasets.

1.2 Differential Privacy

Differential privacy is a definition of privacy tailored to the problem of privacy-preserving data analysis (Dwork et Al. [5]). It formulates a mathematical guarantee that the output of a privacy preserving system won't change if a single individual is added or removed from a dataset. This would effectively render the output of the algorithm resistant to attackers that want to infer attributes of individuals based on the response of an algorithm. We report the definition of differential privacy with some useful properties to later formulate our proposed method.

Definition 1 (Neighbouring Datasets) *Datasets D and D' are said to be neighbouring if* $\exists x \in D$ *s.t.* $D \setminus \{x\} = D'$

Definition 2 (Differential Privacy) *A randomized algorithm, M, is* (ε, δ)*-differentially private if for all* $S \subset O$ *and for all neighbouring datasets D, D':*

$$P(M(D) \in S) \leq e^{\varepsilon} P(M(D') \in S) + \delta$$

with O being the output space and P is taken with respect to the randomness of M.

Differential privacy provides an intuitively understandable notion of privacy - a particular sample's inclusion or exclusion in the dataset does not change the probability of the seen outcome: it does so by a multiplicative factor e^{ε} and an additive amount δ.

Theorem (Post-processing) *Let M be an* (ε, δ)*-differentially private algorithm and let* $f : O \rightarrow O'$ *where O' is any arbitrary space. Then* $f \circ M$ *is* (ε, δ)*-differentially private.*

The above theorem states that no matter the amount of post-processing that is applied to the output resulting from a differentially private algorithm, the information will remain (ε, δ)-differentially private. This is useful for GANs as simply proving the

discriminator to be differentially private will imply the generator to be differentially private as well. We need the discriminator and not the generator to be differentially private because the training of the Discriminator in GANs also updates the Generator.

2 Background

Our goal is to use deep learning models to generate a synthetic version of a private table T to be released to researchers in a controlled environment, a privacy constraint that can be decided by the data holder. The table dataset T contains N_c continuous columns and N_d discrete columns where each column is a random variable. Existing research only suggests a handful of models that can be employed in this setting.

2.1 Tabular GANs

The most notable deep learning models that can achieve this are GANs and Autoencoders (AE), but as argued in Xu et Al. [2] GANs are generally easier to integrate with differential privacy and show better performance over real datasets. In case of tabular dataset, Côté et Al. [6] suggests MC-WGAN-GP (Camino et Al. [7]) and CTGAN (Xu et Al. [2]) as the best models to synthesize real data. The MC-WGAN-GP model is an adaptation of the more common WGAN-GP model (Gulrajani et Al. [11]) made to handle datasets with multiple categories ($N_d>1$). CTGAN instead makes use of a conditional vector to handle multiple categories and has a particular pre-process that aims at making the numerical variables easier to learn. Côté et Al. [6] also show that MC-WGAN-GP shows slightly better performance, while CTGAN is easier to use in practice. The CTGAN model also provides the benefit of being able to impose a categorical condition on the samples to be generated.

2.2 Differentially Private GANs

Some effort has been put into developing differentially private GAN models, although not specifically for tabular datasets. The CTGAN authors point out that their model would be easier than an autoencoder to integrate with differential privacy by using PATE-GAN (Jordon et Al. [4]). PATE-GAN is a model that adapts the Private Aggregation of Teacher Ensemble method to implement differential privacy into a GAN. It is remarkable as the generator of PATE-GAN never sees any data; it only manages to learn based on the indication given to the student discriminator by the teachers. The model that the authors of PATE-GAN compare against is the DPGAN model (Xie et Al [8]), which uses a noisy back-propagation system to achieve differential privacy called DP-SGD (Abadi et Al. [9]). Like PATE-GAN, G-PATE (Yunhui et Al. [10]) uses the PATE system but removes the need for the student discriminator and instead applies PATE directly on the gradient passed to the Generator.

3 Our Approach

Since not a lot of research has been put into developing and testing differentially private models for tabular data, we intend to expand on the claim of the CTGAN authors that GANs can be made differentially private. We do not explore MC-WGAN-GP further as CTGAN has been more thoroughly tested in several settings[1] and naturally provides the capability for conditioning that may be of use to researchers that want to generate specific (or balanced) datasets.

To make CTGAN differentially private we must choose one Differential Privacy system between the ones that we highlighted earlier. In Rosenblatt et Al. [12], the authors used a CTGAN with two different Differentially Privacy architectures, one achieved through the PATE system and the other through DPGAN. Unfortunately, they do not address in detail how their models work. Therefore, we will now explore the different possibilities and the required steps to transform CTGAN into a differentially private model.

3.1 Pre-Processing

The pre-processing step in CTGAN creates problems in differential privacy, as it embeds additional information into the model itself that must be privatized. Therefore, a small part of our privacy budget will be spent in ensuring that the pre-processing step is also differentially private. This portion of the budget can be 0, depending on the assumptions that we make about our private datasets.

3.1.1 Categorical Attributes and Conditional Loss

In the pre-processing phase CTGAN transforms the categorical attributes in their respective 1-Hot-Encoded versions. The network then computes the univariate frequency of these categorical attributes and uses this input to guide the generator towards which categorical attributes are to be used for generation. During training, the generator uses the chosen conditional vector samples to create a new tuple according to the categorical attributes decided by the given condition. The Conditional Loss, an additional loss based on the conditioned portion, is computed to force the generator to adhere to the scheme. From an information perspective, this loss propagates the information of the frequency of the categorical attributes into the generator. In our interest of guaranteeing differential privacy, this may cause a problem. We can:

1. assume that the univariate categorical frequencies are publicly known
2. enforce the model to use privatized frequencies
3. disable the conditional loss altogether

[1] https://sdv.dev/SDV/

The first option is viable, but it should be kept in mind that if the categorical attributes are not privatized their frequency will be considered public by the model. Instead, choosing to follow either point two or three from our options achieves privacy: having the categorical frequencies be already private before entering the generator will ensure privacy on the loss thanks to the post-processing theorem, while disabling this loss will make sure that the Generator cannot learn these frequencies. The third case is simpler to achieve, but it leads to a much worse utility. As the ablation study from the CTGAN authors highlight, removing the conditioning results in a 36% performance decrease. Therefore, it is the second point achieves the highest utility while maintaining privacy. We can do so by adding to the pre-processing phase a (private) noise to the computed frequencies by using Differentially Private algorithms to sample histograms or more flexible schemes such as Private-PGM (McKenna et Al. [13]).

3.1.2 Numerical Attributes and Gaussian Mixture Models

The pre-processing of the Numerical Attributes is also troublesome, as it relies on a Variational Gaussian Mixture Model (VGMM). Each numerical attribute is encoded into a pair, one representing the position of the value within a mode and the other the mode to which it belongs. This means that CTGAN also accounts for the presence of a Decoder which will know the univariate distribution of the categorical attributes. If the CTGAN model is shared with a third party, the decoder will directly disclose this information. To address this issue, we can:

1. Hand out generated data instead of the model itself
2. Ensure that we only move the model in trusted environments
3. Make the VGMM have differentially private outputs
4. Assume that the univariate distributions are already known

As in the previous section, we can either protect this information with noise by using a private version of VGMM, like the one presented in Gautam et Al. [14]. We also want to note that CTGAN uses VGMM for only 1 iteration and not until convergence. Therefore, the privacy budget to be spent on privatizing this algorithm will be small.

3.2 CTGAN Learning

CTGAN's architecture will learn the individuals if kept non-private. To avoid this, we need to embed one of the GAN solutions illustrated in section 2 for differential privacy. It is important to note that CTGAN makes use of the PAC-GAN (Lin et Al. [15]) framework which would make the privacy guarantee of any system refer to the size of the PAC (10 by default) instead of the single sample. To avoid this issue, we will always consider a PAC of 1.

The possible choices to make CTGAN be differentially private are:

PATE-GAN teacher and student in CTGAN: The PATE system implemented in PATE-GAN does not consider Wasserstein loss and can't easily be extended for a WGAN-GP such as CTGAN. To eliminate this issue, we must convert CTGAN to use the standard Binary Cross Entropy Loss (BCE). Moreover, it is theoretically possible to divide the dataset into shard and assign each one to a teacher, but this can lead to unstable behaviour. It can happen that a certain conditional vector is invalid - as a certain category might not be present - in a dataset shard assigned to a teacher. Solving this would require stripping CTGAN of the conditioning or build the data shards in a non-random way. The latter would break the PATE privacy guarantee. Instead, stripping CTGAN of the conditioning portion would incur in a significant utility loss. Therefore, PATE-GAN is not well suited for our setting. We also know that it has not been tested in highly dimensional settings.

G-PATE in CTGAN: This framework, differently from PATE-GAN, allows us to use the WGAN-GP loss as the teacher vote does not decide the category of a sample. It is instead the gradient that should be passed back to the generator that is discretized with PATE. However, using this approach with the conditional generator of CTGAN is not possible as we would have to sample from the conditional generator and use the same samples for each Teacher, thus creating an instability problem in case one Teacher does not see a particular category of a sample. Just like PATE-GAN, this approach might not be well suited for our setting where we employ multi-categorical attributes and imbalanced datasets.

Using DP-SGD in CTGAN: this is the simplest approach, which does not require any modification to CTGAN as it only involves the substitution of the Discriminator optimizer with one that injects noise directly into its gradients. This approach unfortunately provides the worst privacy bound among the presented models. Therefore, it might be inefficient especially for particularly low privacy budgets.

We propose to use DP-SGD to make CTGAN be differentially private without stripping the model of any features. This will allow us to test the capabilities of CTGAN when embedding differential privacy. We will refer to this approach as **DP-CTGAN.**

4 Experimental Results

To test the performance and utility loss of the CTGAN model against its differentially private version, we will use the benchmarking suite with which CTGAN was evaluated in the first place. Developed by MIT, SDGym[2] is a framework to benchmark the performance of synthetic data generators. It contains metrics to test single and multiple

[2] https://github.com/sdv-dev/SDGym

table generation for machine learning and publication purposes. Metrics include statistical and machine learning utility measures as well as privacy metrics.

4.1 Experimental Setup

We run CTGAN, our proposed DP-CTGAN and a baseline model on various metrics using multiple datasets.

Datasets: We selected 4 commonly used machine learning datasets (*Adult*, *Census*, *News and Credit*) from the SDGym repository[3], with features and label columns in a tabular form. General characteristics datasets that we are going to use are highlighted in **Table 1.1**.

Name	**Continuous Columns**	**Binary Columns**	**Multi-Class Columns**	**# Records**	**Task**
Adult	6	2	7	33K	C
Census	7	3	31	300K	C
Credit	29	1	0	284K	C
News	45	14	0	40K	R

Table 1.1 Dataset in our benchmark

Models: We compare the utility and privacy loss of the proposed DP-CTGAN model against a standard CTGAN and we use an Identity Generator (a generator that outputs the original dataset) as a baseline model. To better compare the performance of the privacy scheme, we will consider the univariate attributes in our dataset be publicly know. The GANs will be trained with the same hyper parameters, using a batch size of 500 and each model will be trained for 100 epochs. For DP-CTGAN, we will train using different levels of noise multipliers (nm = 10^{-5}, 0.01, 0.1, 1) to achieve different privacy levels and we will report the mean epsilon of the models over the different datasets in the results section. The objective of this evaluation is to show that we start from the same performance of CTGAN when injecting a very low noise (10^{-5}) and we end up with a strong privacy requirement ($\varepsilon \sim 1$) when injecting a high noise. For all models and evaluations, δ is always set to 10^{-3} (indicating a very high estimated resistance to adversarial attacks in our privacy guarantee).

Utility Metrics: As in Xu et Al. [2], given that evaluation of generative models is not a straightforward process, where different metrics yield substantially diverse results (Theis et Al. [16]), our benchmarking suite evaluates multiple metrics on multiple datasets. With our dataset, we have a machine learning task to use to evaluate synthetic data generation method via *machine learning efficacy* to test for the utility of the Synthetic Dataset. **Figure 1.1** illustrates the evaluation framework. The

[3] http://sdv-datasets.s3.amazonaws.com/index.html

Machine learning models that we will use to perform this evaluation are: Decision Trees, Ada-boost, Logistic Regression, and Multi-Layer Perceptron.

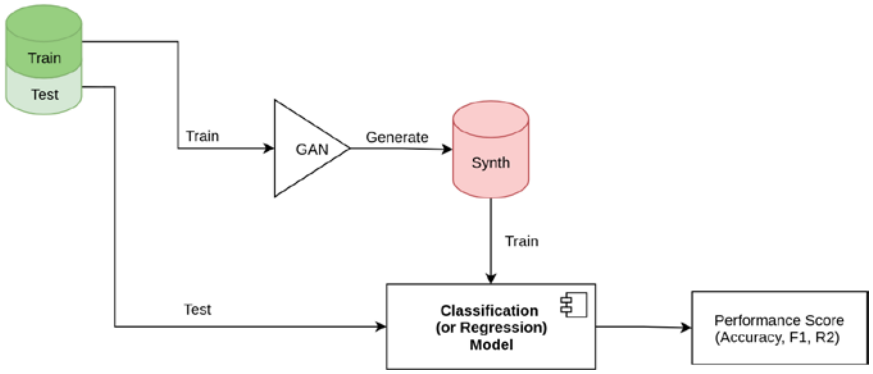

Figure 1.1 Evaluation of Utility of the synthetic dataset against the real dataset

Detection Metrics: We will use a *Detection* approach to evaluate the distinguishability of the Real dataset from the generated one. The two metrics of our choice build a Machine Learning Classifier that learns to tell the synthetic data apart from the real data. The classifier is evaluated using Cross Validation, measuring one minus the average ROC AUC score obtained. The machine learning models of choice for this evaluation are SVC Detection and Logistic Detection, as shown in **Figure 1.2**.

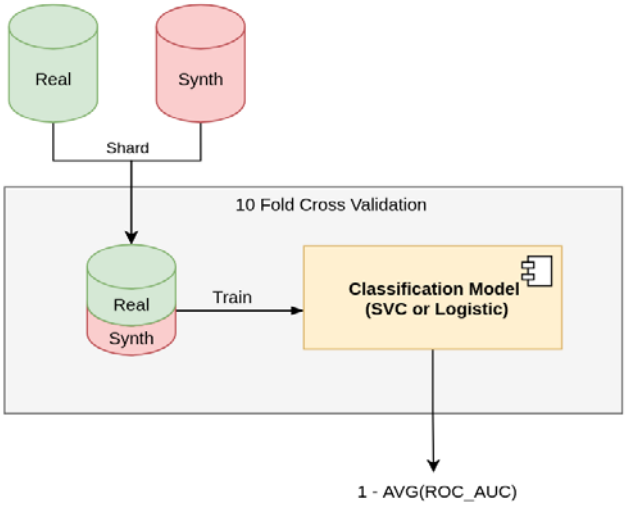

Figure 1.2 Evaluation of Distinguishability of synthetic data from the Real data

Privacy Metrics: This family of metrics measures the privacy of a synthetic dataset by positing the question: given the synthetic data, can an attacker predict sensitive attributes in the real dataset? This is accomplished by fitting an adversarial attacker model on the synthetic data to predict sensitive attributes from “key” (assumed as known) attributes and then evaluating its accuracy on the real data. This metric is defined as one minus the probability of making the correct attack on the real data.

For each of our datasets, we attack the attributes separately by fitting a SVM and Random Forest models for categorical attributes and a Support Vector Regressor (SVR) and Linear Regression for numerical attributes. The evaluation scheme is depicted in **Figure 1.1**.

An important remark is that these metrics intend to measure the machine learning performance of synthetic data, but in principle any metric can be used to compare against other methods of generating private tabular datasets.

4.2 Results

We evaluated our models using the above-mentioned framework. We summarize the benchmark results in **Table 2**. We split the results in the table based on the tasks which the metrics aim to achieve. We further split in 'Classification' and 'Regression' tasks for ML Utility. We also split 'Numerical' and 'Categorical' metrics in the privacy case.

Method	**Noise**	**Utility**		**Detection**	**Privacy**		**Epsilon *(ε)***
		Classification	*Regression*		*Numerical*	*Categorical*	
Identity	N/A	0.78	0.014	1	0.26	0.62	Inf
CTGAN	N/A	0.59	-0.08	0.7	0.32	0.66	Inf
DPCTGAN	0.00001	0.575	-0.11	0.65	0.51	0.72	>100000
DPCTGAN	0.001	0.565	-0.23	0.61	0.56	0.74	7453
DPCTGAN	0.1	0.6	-2.85	0.34	0.59	0.84	78
DPCTGAN	1	0.58	-12.5	0.15	0.78	0.97	0.94

Table 2 The average results of our tests for the different model initializations

Based on the noise that we inject into the network, we achieve different levels of privacy. Table 2 depicts that the metrics for utility, detection have the highest values on low noise cases. This is to be expected, because we expect the models without noise to reproduce the data faithfully. In turn, the Privacy metrics have the lowest values in those cases as one can deduce individual's information from the dataset. As the noise levels are increased, we see that privacy rises as the information in the synthetic data is decreasingly useful for predictions about individuals. The same holds true for detection; as the noise level increases the synthetic data becomes increasingly different from the real data. For utility, we see that the regressions values (R2) are quite sensitive to the noise level. Especially in the case where privacy level is very high ($\varepsilon < 1$), we observe highly negative R2 values, meaning that the models built on synthetic data do not fit the original data anymore. This is expected, since injecting noise directly into the network weights like in DPGAN provides the worst privacy bound estimation among the DP modalities that we presented earlier, therefore leading to a low utility. We see instead that in the classification case performance is not impacted as much, but it is not much better than random classification (0.5) even in CTGAN without noise.

5 Conclusions

In this work we designed and tested a differentially private way of generating accurate synthetic tabular datasets. We built from CTGAN and analysed different schemes to enforce privacy in the model directly and evaluated our proposal against the identity function and CTGAN itself for different choices of noise. We show that our proposal for integrating differential privacy progressively reduces utility to achieve better privacy constraints. We therefore show that the CTGAN framework can be associated with differential privacy.

As a future work, we want to research the possibility of extending the PATE framework to the WGAN to provide a better privacy/utility trade-off in this setting. It would be also interesting to analyse the possibility of using different base GAN models (such as MC-WGAN-GP) to achieve privacy and compare against the stability of the training procedure.

6 Acknowledgments

This research was conducted as an exploration of the IT Innovation Team (it-innovation@ecb.europa.eu) of the European Central Bank. We want to acknowledge Christoph Schaper (Head of Division) and María Velasco (Team lead) for providing the time and material for this project. We also thank Sebastian Boddenberg for helping us with Infrastructure (especially the GPUs) and the rest of the team for their support. We further want to acknowledge our colleagues from the general directorate of statistics for fruitful discussions and guidance, especially Sebastien Perez Duarte, Thomas Gottron and Nicola Benatti.

The views expressed in this paper are those of the authors and do not necessarily reflect those of the European Central Bank.

References

[1] Ullman J., Vadhan S. (2011) PCPs and the Hardness of Generating Private Synthetic Data. In: Ishai Y. (eds) Theory of Cryptography. TCC 2011. Lecture Notes in Computer Science, vol 6597. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19571-6_24

[2] Lei Xu and Maria Skoularidou and Alfredo Cuesta-Infante and Kalyan Veeramachaneni.*Modeling Tabular Data using Conditional GAN*. 2019. arxiv: 1907.00503 [cs.LG]

[3] Ian J. Goodfellow and Jean Pouget-Abadie and Mehdi Mirza and Bing Xu and David Warde-Farley and Sherjil Ozair and Aaron Courville and Yoshua Bengio. *Generative Adversarial Networks.* 2014. arXiv: 1406.2661 [stat.ML]

[4] Jordon James and Yoon Jinsung and Van Der Schaar Mihaela., *PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees.* 2019. International Conference on Learning Representations.

[5] Cynthia Dwork and Aaron Roth (2014), "The Algorithmic Foundations of Differential Privacy", Foundations and Trends® in Theoretical Computer Science: Vol. 9: No. 3–4, pp 211-407. http://dx.doi.org/10.1561/0400000042

[6] Marie-Pier Cote and Brian Hartman and Olivier Mercier and Joshua Meyers and Jared Cummings and Elijah Harmon. *Synthesizing Property & Casualty Ratemaking Datasets using Generative Adversarial Networks.* 2020. arXiv: 2008.06110 [stat.ML]

[7] Ramiro Camino and Christian Hammerschmidt and Radu State. *Generating Multi-Categorical Samples with Generative Adversarial Networks.* 2018. arXiv: 1807.01202 [stat.ML]

[8] Liyang Xie and Kaixiang Lin and Shu Wang and Fei Wang and Jiayu Zhou. *Differentially private generative adversarial network.* 2018. arXiv: 1802.06739 [cs.LG]

[9] Abadi, Martin and Chu, Andy and Goodfellow, Ian and McMahan, H. Brendan and Mironov, Ilya and Talwar, Kunal and Zhang, Li. *Deep Learning with Differential Privacy.* 2016. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. http://dx.doi.org/10.1145/2976749.2978318

[10] Yunhui Long and Suxin Lin and Zhuolin Yang and Carl A. Gunter and Bo Li. *Scalable Differentially Private Generative Student Model via PATE.* 2019. arXiv: 1906.09338 [cs.LG]

[11] Ishaan Gulrajani and Faruk Ahmed and Martin Arjovsky and Vincent Dumoulin and Aaron Courville. *Improved Training of Wasserstein GANs.* 2017. arXiv: 1704.00028 [cs.LG]

[12] Lucas Rosenblatt and Xiaoyan Liu and Samira Pouyanfar and Eduardo de Leon and Anuj Desai and Joshua Allen. *Differentially Private Synthetic Data: Applied Evaluations and Enhancements. 2020. arXiv: 2011.05537 [cs.LG]*

[13] Ryan McKenna and Daniel Sheldon and Gerome Miklau. *Graphical-model based estimation and inference for differential privacy. 2019. arXiv: 1901.09136 [cs.LG]*

[14] Gautam Kamath and Or Sheffet and Vikrant Singhal and Jonathan Ullman. *Differentially Private Algorithms for Learning Mixtures of Separated Gaussians. 2019. arXiv: 1909.03951 [cs.DS]*

[15] Zinan Lin and Ashish Khetan and Giulia Fanti and Sewoong Oh. *PacGAN: The power of two samples in generative adversarial networks. 2018. arXiv: 1712.04086 [cs.LG]*

[16] Lucas Theis, Aäron van den Oord, and Matthias Bethge. *A note on the evaluation of generative models. 2016. arXiv: 1511.01844 [stat.ML]*