

# Privacy and Utility Evaluation of Synthetic Tabular Data for Machine Learning

Felix Hermsen<sup>1,2</sup> and Avikarsha Mandal<sup>2</sup>

<sup>1</sup> RWTH Aachen University, Germany

<sup>2</sup> Fraunhofer FIT, Germany

{felix.hermsen, avikarsha.mandal}@fit.fraunhofer.de

**Abstract.** Synthetic data generation approaches have attracted a lot of attention as a potential substitute for classical anonymization methods. However, synthetic data still pose a wide range of privacy risks, for example, dataset containing data points close to real data points, thus, increasing risks of linkage attacks. While differentially private generative models are generally considered immune to privacy attacks, it is not immediately evident how these models maintain privacy with reasonable utility. In this study, we evaluate the privacy and utility trade-offs in synthetic data generated by the state-of-the-art generative model CTGAN and its differentially private variant DPCTGAN for mixed tabular data domain. We conduct experiments using widely recognized benchmark datasets to highlight the importance of selecting optimal hyperparameters such that the model converges during training and produces synthetic data with satisfactory utility. Our experiments show that synthetic data generators, which were trained with differential privacy, may experience collapse during the training phase. While the addition of a smaller noise allows the training to converge, still could limit risks against privacy attacks such as membership inference and linkage.<sup>3</sup>

**Keywords:** Synthetic Data · Membership Inference Attack · Distance to Closest Record · Generative Adversarial Networks · Differential Privacy

## 1 Introduction

An entire industry has been built around collecting, analyzing and sharing personal data for economic and public gain. On the other hand, this is an incentive for malicious actors to misuse this data to make profits as well. Consequently, most governments choose to protect personal data under respective privacy laws. One of those legislation is the General Data Protection Regulation (GDPR)<sup>4</sup>. Nevertheless, personal data breaches still occur repeatably due to accidents, malicious intent, external attacks or because best practices have not been followed.

---

<sup>3</sup> This is the authors' version of the manuscript. Please refer to the final publication for the proper citation of this work. The final publication is available at Springer via <https://link.springer.com/chapter/10.1007/978-3-031-57978-3-17>.

<sup>4</sup> <https://gdpr-info.eu/>

For example, in May 2023, the health provider Brightline experienced an attack that led to the exposure of personal data belonging to 783,000 pediatric mental health patients [9].

In order to protect sensitive information and facilitate data sharing of personal information Privacy Preserving Data Publishing (PPDP) [10] can be utilized. The standard set of methods to achieve PPDP is called syntactic anonymization [12].

One seminal work, which won the test of time award, that showed the limitations of classical anonymization is the de-anonymization attack of Narayanan. et al. from 2008 [18] on the anonymized Netflix price data set. The authors argue that high dimensional data is fundamentally prone to de-anonymization, because only a few bits of information are enough to identify an individual [19].

Thus, researchers sought alternatives such as Synthetic Data Generation (SDG) methods for publishing data, which received considerable attention [2]. The fundamental concept of an SDG approach is to create a statistical model that captures the concealed joint probability distribution of the training dataset. From this statistical model, synthetic data with equivalent statistical properties can be sampled. In addition, when applying SDG for PPDP use cases, the added objective is to produce synthetic data that is adequately distinct from the initial data, preventing an adversary from reconstructing ground truth data.

**Related Work** Evaluation of utility vs. privacy of synthetic data is essential for understanding the effectiveness of synthetic generation methods and their applicability. One very prominent privacy on synthetic tabular data evaluation was performed by Stadler et al. [26]. Their study showed that some state-of-the-art synthetic data generation methods encounter identical challenges as classical anonymization, despite synthetic data being advocated as a "silver bullet" for these issues. They applied a black-box membership inference attack to synthetic data and differentially private synthetic data. The results show that even differentially private synthetic data did not prevent a membership attack on outliers. Towards general evaluation methodology for synthetic data, work from Giomi et al. [11] proposed Anonymeter, a privacy evaluation framework for estimating the privacy risk of synthetic data. The framework provides three attack-based evaluations for singling out, linkage and inference risk, which are the main indicators of anonymization according to an opinion paper of the European Union. On top of that Rosenblatt et al. [24] also saw the value in analysing differentially private synthetic data generators. They came to the conclusion that different data generators are favourable in different situations with different values of  $\epsilon$ . Finally, there is Fang et. al [8], which trained CTGAN in a differentially private manner and used it to generate synthetic data. They also extended this approach with federated learning to improve the privacy properties. However, they did not perform a privacy analysis.

**Contributions** In this study, we focus on generative adversarial network (GAN)-based synthetic data generation methods used in the tabular data domain which

is one of the most popular data formats. The key research questions of this work are threefold:

- Whether the synthetic data generated by CTGAN and DPCTGAN is susceptible to common privacy attacks such as linkage attacks.
- If the approach is susceptible to an attack, at which time interval during the training phase of the generator, it becomes more prone to the performed privacy attack.
- Can we understand the sweet spot between privacy and utility? For example, at which point during the training phase of the generator one should stop to give the synthetic data better utility with minimum privacy risks.

To investigate the aforementioned questions, we conducted several experiments and used state-of-the-art synthetic data generators (e.g., CTGAN [29]) for tabular data generation. We analysed the intermediate results during the training of the SDG for tabular data. The reason behind this approach is not only to get a better understating of the generator as well to investigate at which training point the generated data is not susceptible to well-known privacy attacks and could also give good utility. Our finding reveals an inherent correlation between data utility and privacy. This means that it is hard to produce synthetic data that has good utility and can simultaneously protect against privacy attacks. However, at certain acceptable utility, the synthetic data may serve as useful proxy data to aid in the initial stages of project development and testing.

Furthermore, our experiments have revealed that differential privacy can have a profound impact on the training stability of a GAN. For differentially private GANs, when the noise parameter is set too high, the training process fails to make any progress, resulting in the impression that differential privacy consistently produces synthetic data of low quality. However, in reality, the noise introduction was too high initially, leading to the collapse of the training procedure. If the noise parameter is set accurately, differential privacy may decrease the speed of the training procedure, resulting in a decline in utility whilst keeping the training time (epochs) constant, but does not collapse it.

## 2 Synthetic Data Generation

### 2.1 Overview

The main idea of SDG methods is to create new fake data from some original (raw) data which has similar stochastic properties. Therefore, any knowledge derived from the synthetic data should be similar to knowledge derived from the raw data. However, if any data point in the synthetic data is too close to a raw data point, it could leak private information.

To create a SDG model, the assumption is that all data points have been generated from an unknown probability density function (PDF). The objective is to create a stochastic model, which approximates this PDF.

In addition, the stochastic model must support randomized sampling to create new data from the learned distribution.

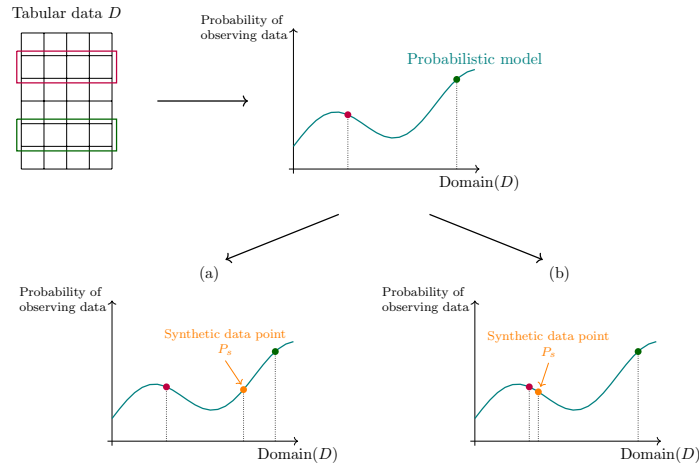


Fig. 1: This Figure depicts an example of generating two synthetic data points from a tabular data set using a probabilistic model. In scenario b), there is a privacy problem, whereas in scenario a), there is no problem

Figure 1 shows an example of the tabular data generation process. Suppose the red row has a rare combination of attributes, whereas the green row has a common combination. As a consequence, the stochastic model learns to assign the green row a higher probability to be generated at random than the red row. In scenario a) we generated a new synthetic data point that is sufficiently far away from any real data row. Thus, it is not critical. However, in scenario b) the synthetically generated record at random is very close to a real record, which could make it feasible to link it to the red row and leak sensitive information about the person belonging to the red record.

## 2.2 Generative models in this study

For this study we have reviewed different deep learning-based SDG methods [13] [20]. Among them are Variational Autoencoders [29], GANs [5] [29] [21] [31] [30] and Transformer based models [4] Our study concluded that the GAN is by far the most popular one with Transformer-based approaches utilizing Language Models starting to achieve even better results [25].

Thus, we choose to use CTGAN [29] as our baseline approach, since it serves as a baseline in many studies. In addition, CTGAN outperforms two other very popular synthetic data generators Med-GAN [5] and Table-GAN [21]. Having said this, CTGAN gets outperformed by CTAB-GAN+ [32] which combines the advantages of two other popular GAN-based approaches, namely Med-GAN [5] and [21] with the advantages of CTGAN. In addition, CTGAN is part of the synthetic data vault<sup>5</sup>, which is a fast growing community that concerns itself

<sup>5</sup> <https://sdv.dev/>

with synthetic data generation. Thus, CTGAN is highly relevant and useful to be analyzed. Thus, we make the assumption that knowledge gained from analysing CTGAN translates to more advanced approaches as-well.

**CTGAN** CTGAN [29] is an abbreviation for Conditional Tabular Generative Adversarial Networks and was developed to generate mixed structured tabular data. One assumption is that rows are independent and identically distributed random variables (i.i.d.). Therefore it is not suited to generate time series-based data. In more detail, the problem with generating mixed tabular data is that most categorical columns are highly imbalanced and do not follow simple distributions. Thus, although in theory, a GAN can approximate any distribution, it is harder to achieve it in practice. Thus, to prevent mode collapse Xu et al. [29] use an efficient re-sampling technique during training, which is enforced using a conditional GAN. In addition, they added mode-specific normalisation, to improve the data utility of numerical columns that follow a complicated multivariate probability distribution. Moreover, to increase training stability they use the W-GAN [28] objective and PAC-GAN discriminator.

**DPCTGAN** In addition to the vanilla CTGAN, we also choose to evaluate CTGAN trained with differential privacy [7] (DP). The reason behind this is that synthetic data has no inherent privacy property.

In a nutshell, differential privacy is a theoretical privacy property of a randomized algorithm, which is defined over a data set  $D$ . The core idea of DP is to bind the influence a single data record  $d \in D$  can have on the computation  $M(D)$  by adding carefully calibrated noise to it. Most well-known is  $\epsilon$ -differential privacy [7], which has only one hyperparameter  $\epsilon$ . This hyperparameter is also known as the privacy parameter. The lower the value of  $\epsilon$ , the greater the amount of noise that is added to the randomized algorithm, thereby resulting in a higher level of privacy.

Translated to deep learning, this means that the training process is enhanced with differential privacy to create a neural network that is differentially private. [22] Applied to a GAN this boils down to only training the discriminator network with it since only the discriminator depends on the training data. Hence, the post-processing theorem of differential privacy guarantees a differentially private generator that allows to generate differentially private synthetic data.

For this study we made CTGAN differentially private by training the discriminator network with Opacus<sup>6</sup>, which is an established differential privacy library for deep learning. Note, other researchers have also adapted CTGAN to be differentially private [24].

### 2.3 Threat Model for Synthetic Data Generation

In this section, we introduce our threat model against Synthetic Data Generation (SDG). As a reminder, in this study, we aim to analyse the privacy and utility

<sup>6</sup> <https://opacus.ai/>

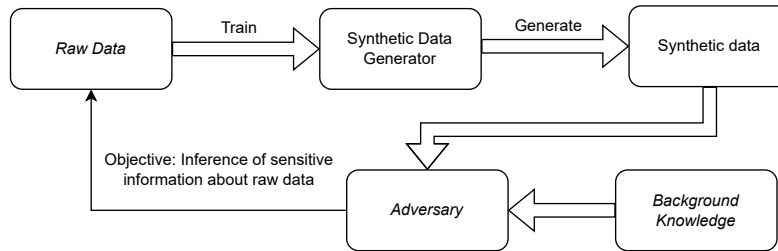


Fig. 2: Threat model for the creation of synthetic data, assuming an honest but curious adversary who uses the generated data to infer unknown knowledge about the raw (sensitive) data based on some background knowledge.

of synthetic data that is used for downstream tasks. The SDG process consists of three main steps and is depicted with blue arrows in Figure 2. The first step is the training of the synthetic data generator model on the original data (raw data)  $R$ . After this, the trained model is used to generate synthetic data  $S$ .

We define a threat model for a semi-honest adversary. A semi-honest or honest-but-curious adversary follows the defined protocols but tries to extract as much knowledge from the output of computations as possible [23].

In this threat model, the adversary has access to one or more generated synthetic data sets. This scenario corresponds to the threat model of anonymization and is the most realistic one, which is why we chose it as our focus. Here the adversary tries to re-identify individuals or infer knowledge about individuals by linking background knowledge to the released synthetic data.

### 3 Evaluation Framework

The conceptual approach for this study is depicted in Figure 3 and consists of three stages. We call the first stage the data generation stage. It is used to generate a sequence of  $N$  synthetic data sets  $S_{seq} = S_1, \dots, S_N$ , where each element corresponds to an intermediate step in the total training procedure of a generative network. More precisely, we train CTGAN for 300 epochs and generate a synthetic dataset every 10 epochs based on the current model. Thus we end up with 30 synthetic dataset to analyse for each GAN model that we train. In addition, we generate data not only from CTGAN [29] but also from our differentially private version, The output of the second stage is an evaluation tuple  $ET_i$  and currently consists of four evaluations. Two utility evaluations and two privacy evaluations.

For a broad picture of the data utility, it is advisable to evaluate the data utility with general and specific utility evaluation approaches [17]. Specific utility evaluations consist of training a specific machine learning model and checking the model accuracy on a separate test set. The test set is a portion of the raw data  $R$ . Thus  $R = R_{train} \cup R_{test}$ , where  $R_{train} \cap R_{test} = \emptyset$ . In contrasts, a general utility measure aims to check the stochastic similarity between synthetic

data  $S_i$  and raw data  $R_{train}$ . The privacy evaluation is comprised of a linkability evaluation and membership inference evaluation.

After this, we obtain a sequence of evaluation tuples which we use in our visualisation stage to create a visualized representation. We chose to make this a separate stage since it averages over multiple cross-validations. Additionally, for comparison, we also apply the evaluation stage to the training data  $R$  of the generator model to establish a baseline.

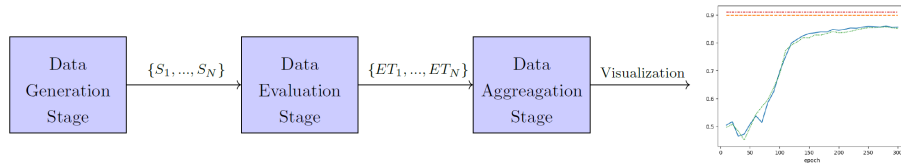


Fig. 3: Overview of the conceptual approach. The first stage generates a sequence of synthetic data sets, which is fed into the data evaluation stage. The data evaluation stage creates an evaluation tuple for each element in the input sequence. Finally, the sequence of evaluation tuples is used as an input for the data aggregation stage, which visualizes the results.

## 4 Data Utility Evaluation

We begin by explaining our data utility evaluation stage, which is composed of two parts.

**The specific utility evaluation** is our first utility evaluation. For it we choose to implement one decision tree-based classifier (XGBoost) and one neural network-based classifier (tabular learner of fastAI<sup>7</sup>). The tabular learner comprises a multi-layered feed-forward neural network featuring non-linear activation functions between hidden layers and an embedding layer after the input layer. Our implementation adopts the standard configuration of two hidden layers. In more detail, we use  $R_{train}$  to train two models which we denote as  $M_{xgb}^{train}$  and  $M_{fastai}^{train}$  to establish a baseline. After this we use  $S_{seq}$  to train  $M_{seq} = \{\{M_{xgb}^{S_1}, \dots, M_{xgb}^{S_N}\}, \{M_{fastai}^{S_1}, \dots, M_{fastai}^{S_N}\}\}$ . Then, all of these models are evaluated against  $R_{test}$ , where the *AUC* is the evaluation score.

**The general utility evaluation** is the Propensity Mean Squared Error (pMSE) The pMSE is referred to as one of the most useful measures for assessing the

<sup>7</sup> <https://docs.fast.ai/tabular.learner.html>

general effectiveness of artificial data [6]. The aim is to train a classifier that differentiates real records from synthetic ones. To train the classifier we join  $S_i$  and  $R_{train}$  into one data set and add labels to distinguish both. As classifier, we choose the XGBoost. The quality of the generated data is determined by how indistinguishable it is from the real data, with detectable synthetic data indicating unsatisfactory results. The pMSE score is calculated as follows:

$$pmse(R) = \sum_{i=0}^n (p_i - c)^2, \quad (1)$$

where  $c = \frac{S_i}{S_i + R_{train}}$  and  $n = |S_i| + |R_{train}|$ . In our case we set  $|S_i| = |R_{train}|$ . Therefore, the score will be 0, if  $S_i$  and  $R_{train}$  are indistinguishable wrt. the trained classifier, which is the case for the baseline. On the other hand, when the classifier is able to distinguish  $S$  from  $R$ , then the score will be 0.25.

## 5 Data Privacy Evaluation

As outlined in section 3, the evaluation of privacy encompasses two additional components.

**The distance to closest record**, which is a popular way of measuring the identifiability risk [31] is the first part. It estimates how many records of the generated synthetic data set  $S_i$  are too close to the training data  $R_{train}$ . The assumption is that the closer the distance to real data, the higher the risk for re-identification.

Most approaches use distance measures as the Euclidean distance and encode categorical values with one hot vector encoding. However, this method loses semantic information between similar attribute values and gives a false impression of the true risk, since an adversary could make use of semantic similarities. Vengurlekar et al. [27] also identified this and suggested other distance measures like the Gower Distance, instead of the Euclidean distance on one hot encoded categorical attribute. We create embeddings that are generated implicitly by training the TabularLearner from fastai on our baseline  $R_{train}$  in the data utility evaluation step. After this we use the same approach as proposed by Vengurlekar et al. [27] to decide if a record is too close. The calculation is performed as follows: Given a synthetic data point  $s_i \in S$  and the training data  $R_{train}$ , a synthetic data point  $s_i$  is said to be too close to  $R_{train}$  if the external similarity smaller than the internal similarity, where the internal similarity is defined as:

$$\min_{r_k \neq r'_k} d(r_k, r'_k), \text{ where } r_k, r'_k \in R_{train} \quad (\text{Internal Similarity})$$

and the external similarity is defined as

$$\min d(r_k, s_i), \text{ where } r_k \in R_{train} \quad (\text{External Similarity})$$

**A membership inference attack** is the second part. Our membership inference attack is an adaption of the MIA that has been proposed by Stadler et al [26]. In a nutshell, a MIA tries to estimate the probability, of whether a target record is part of the training set  $R_{train}$  that was used to train an SDG, which generated  $S_i$ . In their experiments, the authors used handpicked outliers from  $R_{train}$  as target records. The rationale behind it is that outliers have a stronger impact on the distribution. Thus, the presence of outliers in  $R_{train}$  is easier to detect. Their results show that some outliers are classified to be in the train set with very high confidence. In contrast, their method did not detect a privacy problem for random records drawn from  $R_{train}$ .

We propose the following changes to their evaluation and design: *First* they applied their attack only to generators trained on data sets of size 1000. One of the datasets they ran their evaluations on is the *adult census* data set, which is one of our evaluation data sets as well. Based on our utility evaluation, GANs trained on such small data sets have a lower performance. Therefore, we argue the attack should be run on synthetic data that has been generated from models that were trained on larger train sets.

Hilprecht et al. [14] introduced the concept of a set membership attack. We adopted their idea to test the membership of a randomly selected set to determine if we can identify it. Therefore, we implemented a shadow model attack to support that. In addition, we also check the membership for an outlier set.

Finally, we choose to find outliers with the Copula-Based Outlier Detection (COPOD) [16] method.

## 6 Experiments

We evaluated our implementation on three mixed tabular data sets. For a health-care use case we selected a modified version of MIMIC 3 [15], for a government sector use case we selected the adult census income data set [3] and for the financial domain we chose the bank marketing data set [3]. In this paper, we present our findings for the adult data set, which are representative of our findings in the other two. All our experiments have been performed on the RWTH-Aachen GPU cluster<sup>8</sup>.

### 6.1 CTGAN-Evaluation

**Data Utility Evaluation** We begin this evaluation with our specific data evaluation. Fig. 4 suggests that the generator needs more than just 1000 randomly sampled data points to converge. Hence, running experiments on CTGAN with a data set size of 1000 is not reliable. This result is an important observation as previously, researchers did not use sufficient data points while evaluating privacy properties of synthetic data that trained the generator using the same data [26]. In this case, the generator does not converge, which is depicted on the left in

<sup>8</sup> <https://help.itc.rwth-aachen.de/service/rhr4fjjuttff/>

Fig. 4. On the other hand, if the generator gets around 10000 data points it converges, which is depicted on the right. Furthermore, we can observe that the convergence speed increases, if we increase the training data size.

In addition, the  $pMSE$  evaluation in Fig.5 (left) shows that the value is close to 0.21 after 300 trained epochs. As a reminder, a  $pMSE$  value close to 0 is desirable. Investigating the synthetic data set, we find that two attributes "capital-gain" and "capital-loss" could not be generated with high accuracy, since they follow a very unique long tail distribution. Note, the adult data set consists of 15 attributes. Removing these two attributes from the data set and repeating the experiments, we notice a steep convergence to 0.04 (See Fig. 5 right). Thus, the  $pSME$  classifier is not very good at distinguishing between raw and fake data anymore. From this, we can infer that the data produced without the two challenging attributes achieves sufficient utility. Nevertheless, this raises concerns about the general applicability of SDG. Note, this is not a new finding and approaches like CTAB-GAN+ [32] claim to have improved the generation of long-tail distributions.

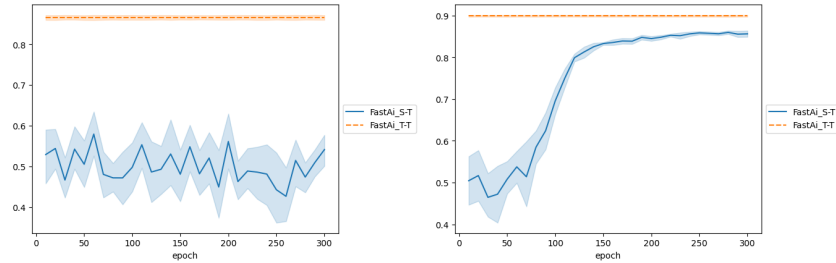


Fig. 4: Machine learning efficiency evaluation of CTGAN with FastAI. In the left CTGAN is trained with  $|R_{train}| \approx 1000$ , in the right on  $|R_{train}| \approx 10000$ . Y-axis represents the accuracy score, while the X-axis represents the training time in epochs.

**Data Privacy Evaluation** We begin the data privacy evaluation of CTGAN by analyzing the distance to the closest record (DCR) first. For this, we picked the SDG that has been trained on 10000 data records on default hyper-parameters, which corresponds to the graph of the right picture in Figure 6. As we can see in the left picture of Figure 6, almost a quarter of records are at risk of being re-identified. This corresponds to the default scenario in which the external similarity is larger than the internal similarity. In addition, we can also multiply the internal similarity with a weight. This allows us to define a less or a more critical scenario. For instance, in the right picture of Figure 6, we have multiplied the internal similarity with 0.8 (for more privacy-critical scenarios). The result is that this doubles the synthetic records at risk. In addition, we can observe

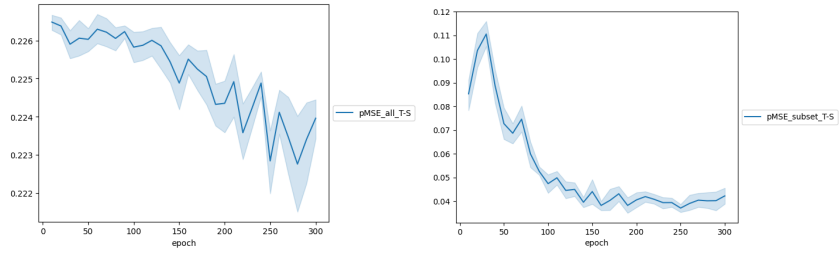


Fig. 5: pMSE evaluation of CTGAN with FastAI. In the left CTGAN is trained with  $|R_{train}| \approx 10000$  with all categorical attributes, whereas on the right-hand side, two attributes have been removed from the adult census dataset ("capital gain" and "capital loss"). Y-axis represents the pMSE score, while the X-axis represents the training time in epochs.

the expected strong correlation with the data utility curves and the DCR curve at around 100 to 150 epochs, which gives the impression that this is enough to train CTGAN on the adult dataset for 150 epochs.

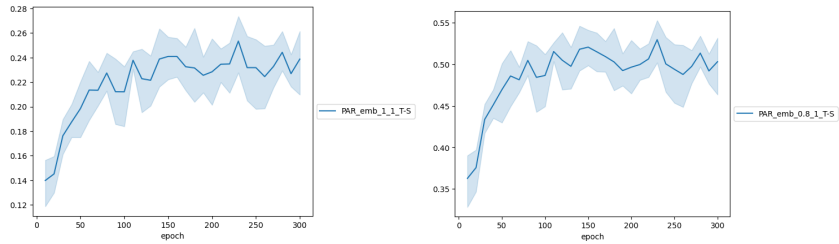


Fig. 6: DCR evaluation of CTGAN. Y-axis represents the distance to closest record score, while the X-axis represents the training time in epochs. Left picture shows default settings for DCR, while the right picture considers a more privacy-critical (weighted) scenarios.

The final privacy evaluation is our set membership inference attack. As a reminder, in a classical membership inference attack, we choose one record and try to detect whether it is a member of  $R_{train}$ . In this attack, we strengthen the attacker and try to infer, if a set of of 50 was present in  $R_{train}$ . Our experimental results show that after 150 epochs the membership attack stabilizes and achieves a high confidence on the presence of the attack set. In the left graph of Figure 7 the attack set is an outlier set, whereas in the right graph, the attack set has been sampled randomly from the training data.

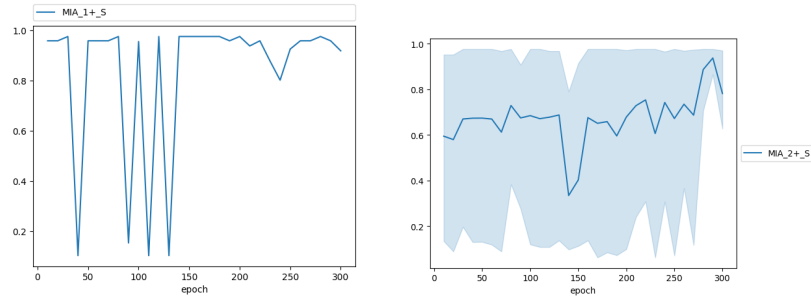


Fig. 7: Membership inference evaluation of CTGAN. Y-axis represents the probability that the attack set was in the training set, while the X-axis represents the training time in epochs.

In addition, the right image includes the variance over multiple performed attacks. In total, the membership attack is a success on outlier sets with low variance. However, the attack is not always successful on randomly sampled target sets. The variance is large, allowing for a high degree of plausible deniability. Nevertheless, depending on the randomly sampled set, the attack can be successful. In particular, if it includes outliers.

## 6.2 DPCTGAN Evaluation

**Data Utility** In this subsection, we discuss the findings of our differentially private implementation of CTGAN (DPCTGAN), which is depicted in Fig. 8. To implement this, we used Opacus, which is governed by a clipping norm and a noise parameter. The default value for the clipping norm is 1, which we also used. The utility evaluation shows that the generator can converge to a reasonable point when added noise is less (top left and top right graph of Fig.8). However, if the noise added is set to high, the GAN convergence collapses (bottom left and bottom right of Fig.8). Experimentally, we observed that when we set the hyperparameter that governs the noise to 3 (bottom left) and to 5 (bottom right), the training collapses from the beginning. Otherwise, convergence can be slowly reached with less added noise during training in comparison to CTGAN. The slowed convergence results in a final value of lower data utility by 5 – 10 percentage points, when we set the noise parameter to 0.5 (top left Fig.8). The total amount of privacy budget  $\epsilon$  is approximately 100 for the top left image, 15 for the top right, 5 for the bottom left, and 2 for the bottom right. It should be noted that a privacy budget ( $\epsilon$ ) greater than 1 is generally considered inadequate for strong privacy. Nonetheless, usage of higher privacy budgets is not unheard of in machine learning applications. For example, Abadi et al. [1] employed an  $\epsilon$  value of 8 in their experiments. As training DPCTGAN with lower epsilon values failed to reach convergence, we conducted privacy attacks in the setting with a higher privacy budget where training at least succeeds.

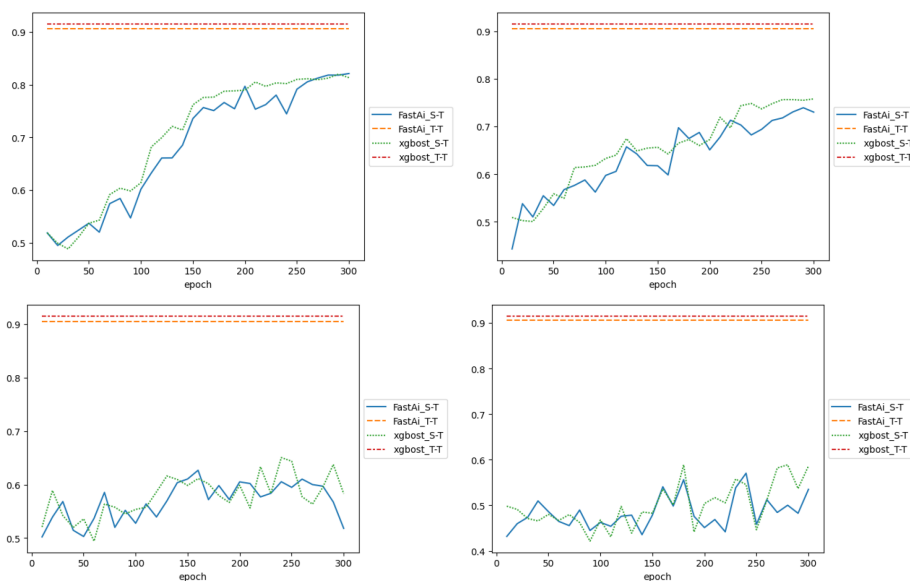


Fig. 8: Utility evaluation of CTGAN trained with Opacus (DPCTGAN): The Y-axis of each graph depicts the accuracy score, whereas the X-axis depicts the training time in epochs. Furthermore, the green line is an XGBoost classifier, while the blue line is a neural network-based classifier. Choice of noise parameters (more noise gives better privacy protection): top left 0.5, top right 1, bottom left 3, bottom right 5.

**Data Privacy** In the two cases where the data utility is still sufficient, we investigated data privacy risks of linkage and membership inference on the Adult dataset. For the DCR, Fig. 9 shows that less than 10% of records are now at risk for re-identification when we analyse both cases. In addition to that, the membership attack which was previously very successful on outlier sets for CTGAN, is now very unreliable and mostly experimentally unsuccessful for DPCTGAN as depicted in Fig. 10.

If the attack set is in the training set (left Fig. 10), we can observe that the classifier believes that the set is not present in the training data although it is. For a different attack set (right Fig. 10), we show the variance and the mean of the performed attacks. This indicates that DPCTGAN with a higher privacy budget could still limit privacy attack success rates compared to CTGAN.

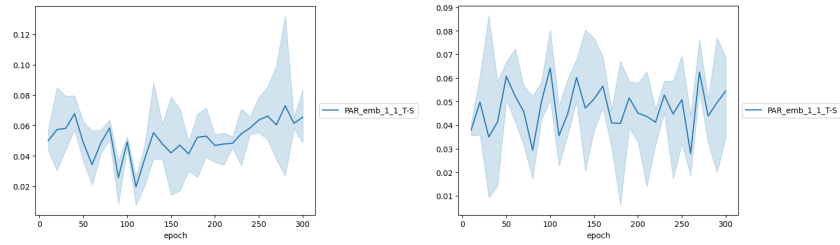


Fig. 9: DCR evaluation of DPCTGAN. Y-axis represents the distance to the closest record score, while the X-axis represents the training time in epochs. Left picture shows the default settings for DCR on the DPCTGAN trained with noise parameter 0.5, while the right picture considers the DPCTGAN trained with noise parameter 1.

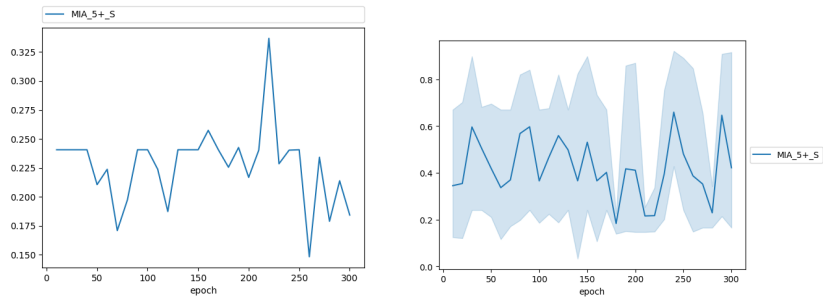


Fig. 10: Membership inference attack evaluation of DPCTGAN. Y-axis represents the probability that the attack set was in the training set, while the X-axis represents the training time in epochs. Left picture shows DPCTGAN trained with noise parameter 0.5 and an attack set consisting of outliers, while the right picture is the same setting on a different attack set.

## 7 Conclusion and Future Work

In summary, we have demonstrated that CTGAN was not able to capture the distribution of certain attributes as tested in the adult census dataset, and thus may not perform well in other datasets. However, by eliminating two problematic attributes, we were able to create high-quality synthetic data that is equally as useful as the original data for two downstream tasks. Thus CTGAN has shown potential to produce good-quality synthetic data given the subsequent downstream task is known. However, when the downstream task is unknown, the usefulness of synthetic data produced by CTGAN cannot be predicted. Furthermore, we conducted experimental evaluations on CTGAN to understand the

risks of privacy leakage as the produced synthetic data may reveal personal information. Unsurprisingly, the synthetic data generated from CTGAN, if trained on privacy-sensitive data, should not be shared as linkage and membership inference privacy attacks have shown to be successful. Employing differential privacy during training of the synthetic data generator is one existing solution to mitigate this issue. However, our analysis of the synthetic data generated with DPCTGAN shows that differential privacy can slow down or even prevent the convergence of the generator if the noise parameter is set too high (for better privacy), which could render the generated data to be unusable. On the other side if the noise parameter is set to low the subsequent privacy protection will be reduced. However, we observed with correctly set hyperparameters, DPCTGAN, trained with a higher privacy budget, can still give basic privacy protection against our evaluated attacks in comparison to CTGAN. In the future, we aim to extend this evaluation with more recent SDGs, more diverse data sets, and privacy attacks. Some of the candidate models for data generation are [31] and [4]. Furthermore, we have only evaluated privacy for an honest but curious adversary. Extending the threat model to a malicious adversary could be another research direction. Finally, it is crucial to understand the distinctions and similarities between syntactic anonymization and synthetic data generation to evaluate their applicability across different domains.

## 7.1 Acknowledgements

This is the authors' version of the manuscript. The final publication is available at Springer via [https://doi.org/10.1007/978-3-031-57978-3\\_17](https://doi.org/10.1007/978-3-031-57978-3_17). Please refer to the final publication for the proper citation of this work. We would like to thank the anonymous reviewers for their useful feedback and suggestions. This work was supported by the BMBF-ANR-funded project Crypto4Graph-AI (funding number 01IS21100A). The simulations were performed with the computing resources granted by RWTH Aachen University.

## References

1. 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*, pages 308–318, 2016.
2. Mohammad Abufadda and Khalid Mansour. A survey of synthetic data generation for machine learning. *2021 22nd International Arab Conference on Information Technology (ACIT)*, pages 1–7, 2021.
3. Arthur Asuncion and David Newman. Uci machine learning repository, 2007.
4. Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators, 2022.
5. 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 *Machine learning for healthcare conference*, pages 286–305. PMLR, 2017.

6. Fida K Dankar, Mahmoud Ibrahim, and Mauro Castelli. Fake it till you make it: Guidelines for effective synthetic data generation. *Applied Sciences (2076-3417)*, 11(5), 2021.
7. Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.
8. Mei Ling Fang, Devendra Singh Dhimi, and Kristian Kersting. Dp-ctgan: Differentially private medical data generation using ctgans. In *International Conference on Artificial Intelligence in Medicine*, pages 178–188. Springer, 2022.
9. Neil Ford. List of data breaches and cyber attacks in 2023, Nov 2023.
10. Benjamin CM Fung, Ke Wang, Rui Chen, and Philip S Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)*, 42(4):1–53, 2010.
11. Matteo Giomi, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. A unified framework for quantifying privacy risk in synthetic data, 2023.
12. Puneet Goswami and Suman Madan. Privacy preserving data publishing and data anonymization approaches: A review. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 139–142. IEEE, 2017.
13. Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neuro-computing*, 493:28–45, 2022.
14. Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019(4):232–249, 2019.
15. Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.
16. Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123. IEEE, 2020.
17. Ofer Mendelevitch and Michael D Lesh. Fidelity and privacy of synthetic medical data. *arXiv preprint arXiv:2101.08658*, 2021.
18. Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.
19. Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets: a decade later. *May*, 21:2019, 2019.
20. Sergey I Nikolenko. *Synthetic data for deep learning*, volume 174. Springer, 2021.
21. Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.*, 11(10):1071–1083, jun 2018.
22. Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.
23. Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56(4):1–34, 2023.
24. Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: Applied evaluations and enhancements. *arXiv preprint arXiv:2011.05537*, 2020.

25. Aivin V. Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers, 2023.
26. Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data-anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1451–1468, 2022.
27. Pushkar Vengurlekar. Generating tabular synthetic data. <https://github.com/Pushkar-v/Generating-Synthetic-Data-using-GANs>, 2020. Accessed: (02.04.2024).
28. Lilian Weng. From gan to wgan. *arXiv preprint arXiv:1904.08994*, 2019.
29. Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.
30. Jinsung Yoon, Lydia N Drumright, and Mihaela Van Der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24(8):2378–2388, 2020.
31. Zilong Zhao, Aditya Kumar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian Conference on Machine Learning*, pages 97–112. PMLR, 2021.
32. Zilong Zhao, Aditya Kumar, Robert Birke, and Lydia Y. Chen. Ctab-gan: Effective table data synthesizing. In Vineeth N. Balasubramanian and Ivor Tsang, editors, *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 97–112. PMLR, 17–19 Nov 2021.