A survey on generating private synthetic data

Mahdi Alikhasi
Department of Computing Science
University of Alberta
Edmonton, AB T6G 2R3
alikhasi@ualberta.ca

Abstract

The goal of synthetic data generation is to create a database that matches the original ones in the sense of statistics. Nowadays, using machine learning tools became a trend to this end. One concern of these methods is that machine learning tools can compromise the privacy of individuals. In this project, we first survey existing techniques and then propose a new method.

1 Introduction

The wide range of datasets' usage in modern applications is undeniable. From a simple data analysis task to more complicated machine learning problems like speech recognition and image processing, we heavily rely on datasets more than ever. With the increasing amount of data and the trend of releasing datasets by companies, the concept of gathering and using this information brings a lot of concern about privacy because these datasets usually contain individuals' sensitive data, especially in the fields like medical diagnoses or genomics.

The problem of users' privacy has been studied from those early works on anonymization and differential privacy (Dwork , 2011). More formally, having a differentially private algorithm gives individuals the ability to denial. The idea here is that removing one row of data would not change the output of a query much, and the effect of each individual is neglectable, which gives them the ability to deny their presence in the database (Ullman and Vadhan , 2011).

However, providing the privacy of a query on a dataset has its limitations, and it is not enough. To name a few, we have to change our software and analytical tools in order to work with differential privacy methods like the Laplacian mechanism (Dwork et al. , 2006), which is an interactive algorithm. Moreover, sometimes it is necessary to have access to the whole database. One such scenario is when two companies want to collaborate and share data to improve accuracy. In these situations, it is more suitable to have access to the dataset instead of querying it. The problem of having access to the dataset while preserving the differential privacy can be addressed by generating a synthesized dataset from the original one in a private manner (Ullman and Vadhan , 2011).

In this project, first, we compare the existing state-of-the-art methods with each other. This comparison is necessary since it gives us an insight into different approaches, their empirical quality in various tasks, and their limitation. In the next step, we propose a new framework for generating synthetic datasets in a differentially private approach that can both address the issue of privacy of individuals and have a good quality, which makes it suitable to be used as a training set in other machine learning tasks. The main goal here is that the proposed model should at first be private, and secondly, it should be able to create a synthesized dataset with similar statistical properties to the original dataset.

Submitted to Nidhi Hedge for Project Report in CMPUT 622, Fall 2021 .

2 Motivation

Machine learning approaches, especially deep learning, have shown good results in recent years. However, these methods usually suffer from the need for large datasets to train. The first impact of being able to generate a synthetic dataset privately is that we could use this unlimited trope of synthetic data to train our model without being worried about problems like privacy. Besides, it enables us to share our datasets without privacy concerts.

The advantages and the perks of having a private synthetic dataset are even more. It will create an extra layer of privacy-preserving against the adversaries. It means that, if we use the private synthetic data to train a model in order to solve some problem; even if the adversary has access to the model or training dataset (which is the synthetic dataset here), the adversary can not infer any sensitive information about individuals from it (Chen et al. , 2018). More generally, we can show that any learning model trained on the private synthetic dataset is also differentially private (Chen et al. , 2018).

3 Related Works

One of the first works on the generation of synthetic data was the work of Barak et al. , 2007. In this work, the main focus was that the synthesized data should have the same statistical properties as the original. To satisfy it, the authors presented a non-polynomial time algorithm for creating a new database with the same two-way margin property while preserving differential privacy. Moreover, the work of Ullman and Vadhan , 2011 has shown that there is no polynomial-time algorithm to create a two-way margin synthesized private data. The limitation here is that the two-way margin property is not sufficient in complicated datasets like images.

With the emergence of machine learning, another trend has been created. Instead of preserving the statistical properties of the dataset using a holistic algorithm, we assume that there is an underlying distribution from which the original dataset was sampled. Then using machine learning, we aim to learn and approximate this underlying distribution. Finally, sampling from this learned underlying distribution will create new samples with the same statistical properties as the original one for us, which we can use as a synthetic dataset. This approach of learning the probabilistic model and statistical properties of the original dataset under privacy has been studied in the last decade (Zhang et al. , 2016, and Zhang et al. , 2017). Moreover, we can use deep learning to learn a more accurate presentation of the underlying distribution. This approach and its methodology have been shown in the works of Xie et al. , 2018, and Jordon et al. , 2018. Both these works used generative adversarial networks, also known as GAN, as their generative models. The difference here is that in the former one, they used the DP-SGD method (Abadi et al. , 2016), and in the latter, they used the PATE framework (Papernot et al. , 2016) to preserve privacy. Furthermore, there are more generative models besides GANs. There is also the work of Chen et al. , 2018 which uses two private deep networks side by side. One autoencoder and one variational autoencoder as a generator. Additionally, there is the work of Takahashi et al. , 2020 which again uses the DP-SGD (Abadi et al. , 2016) to train an autoencoder as a data generator.

One gap here among all the mentioned methods is that there is no work comparing these methods. Moreover, most of these methods show their utility and quality through visualization and the quality of synthesized data on image datasets like MNIST. In this project, we compare these methods both visually and theoretically. To compare methods visually, we consider a case of an image dataset when the goal is that the synthesized dataset has visual similarities. On the other hand, for the case of theoretical comparison, we analyze statistical similarities of the synthesized and original datasets with each other.

Furthermore, methods that use DP-SGD, including DP-GAN, require a large privacy budget to preserve privacy while their final results have poor quality. It motivates us in this project to use a private prediction or a noisy private sampling instead of making the learning algorithm private. To address it, we propose a new framework using variational autoencoder and noisy sampling in the following sections.

4 Proposed Solution and Methodology

Since the project aims to provide a solution for the private synthesize data generation, it is logical to start by comparing different existing state-of-the-art methods with each other, which is somehow absent in the previous studies. To compare these methods, we first try to see the results visually. The thing here is that although privacy is crucial, the main goal here is that the generated synthesized dataset has good quality, and it is easier to see the quality of different methods visually. For this part, we use the MNIST dataset as our original dataset and learn models on it. Moreover, to make the conditions fair, we choose an upper bound for the privacy budget. Then, after training the models on the MNIST dataset, we demonstrate the quality of models' output through random sampling from the generated synthesized dataset.

After visualizing the output of various frameworks, we can get a hunch on which method could be more accurate in real data. The next step is to understand whether the synthesized dataset is similar to the original one or not. For doing so, we need a measure that takes the whole dataset, and not limited to the individuals' information, into consideration and computes similarity. Thus, we use the method provided by the work of Snoke and Slavković , 2018. In the mentioned paper, the authors claim that a synthesized dataset has good quality if an arbitrary classification algorithm can not distinguish between the samples from the original one and the synthesized one. According to their work, we first generate a synthesized dataset. Then we train a classifier on half of the synthesized dataset and half of the original dataset. The labels for samples in the synthesized dataset will be one, and the label for the original samples will be zero. Using the pMSE metric (Snoke and Slavković , 2018), we will have:

$$pMSE = \frac{1}{N}\sum_{i=1}^{N}(p_i - 0.5)^2$$

Where p_i is the classification's prediction for data i in the test dataset. Here, the test dataset will be the other half of the synthesized dataset and the other half of the original dataset that we did not use during training.

The idea here is that if the classifier can not distinguish between the synthesized dataset and the original dataset on the test set, then p_i would be near to 0.5, which is a random choice. Thus, by calculating a mean square error on p_i we can see that on average, how far is our prediction from random choice, 0.5. The lesser pMSE is, the more similar the synthesized dataset and original dataset are, and that means the classifier can not distinguish between them. By this means, we can compare different methods of generating synthesized data and their quality.

Although pMSE will provide us with a metric for comparison, it will not provide us with an explanation of the results. It can be done by constructing a distance metric and studying the parameters and factors that affect this distance separately. For finding the distance between the synthesized dataset and the original one, first, we calculate the correlation matrix of the features in the original dataset and the synthesized one alongside the momentum of each feature. Then, by comparing the correlation matrix of the synthesized dataset and the original one, we can specify whether the relation between features is preserved or not. What is more, by comparing the momentums it is possible to highlight the similarities in each feature between the synthesized dataset and the original one in a stand-alone manner. We introduce a distance measurement to compare different methods, which is the weighted sum of distances of different momentums and correlation matrix. The less this distance measure is, the more realistic and similar the synthetic dataset is. For categorical features, comparison of the plot and the relative proportion of features' categories are sufficient. To this end, the Census Adult dataset will be used, and we analyze models and synthesized datasets on this dataset. The dataset is selected from the OpenML library [1].

All these metrics studied so far are associated with the quality of the synthesized data. However, another crucial part of our approach is privacy. To have an insight into the privacy of these models, we take into consideration the nearest neighbor of each synthesized sample in the original dataset. Our methodology here is that for each sample in the synthesized dataset first, we find the nearest individual in the original dataset. Then we calculate the euclidean distance between these two. By calculating the distance for all samples in the synthesized dataset, it ables us to study the variance and the mean of this distance. The intuition here is that for being private, all the samples should

[1]The dataset that we used can be found at https://www.openml.org/d/1119

have a similar effect on the data generation of the synthesized dataset. This idea is very close to the motivation behind the clipping part of DP-SGD (Abadi et al. , 2016). It can be interpreted that the variance of the distance between each sample in the synthesized dataset and its nearest match in the original one should be as low as possible. In addition, we prefer that the generative model does not overfit the samples in the original dataset. It can be seen as the mean distance between samples in the synthesized dataset and their corresponding match in the original one. Thus, we prefer to have a high mean distance. Another study here is to estimate the gap between the nearest match of samples and the second nearest match.

Finally, we propose a new framework for private synthetic data generation. Here, our approach is very similar to the PATE (Papernot et al. , 2016). In this paper, to address the limitations of methods like DP-SGD (Abadi et al. , 2016), the authors came up with a teacher-student model which uses an ensemble model as a teacher. Moreover, they use noisy voting to provide privacy. Since the teacher models are not private themself, privacy will be compromised if an adversary has access to the models. They used a student model which will train on the output of the teacher part. Their work is based on having a private way of prediction, contracting with DP-SGD that makes the learning algorithm private. Similarly, our proposed method is based on making the sampling of data generation private. The intuition here is similar to the one in PATE.

Our strategy here is to divide the dataset into several sub-datasets and train a generative model on each of these sub-datasets separately. After the training phase, we can use these models to generate new samples, which can be done using a sample selection block. Here, at each timestep, all the generative models will generate new samples, and the sample selection block will choose one of them. Through repeating this procedure n times, we can obtain an intermediate synthesized dataset. The details of the sample selection block will be described further. This structure is much similar to the method known as ensemble learning (Dietterich , 2000). The difference is that in ensemble learning, a weighted voting approach is used to aggregate the results of each model, while our structure has a sample selection block.

The only structure here left to discuss is the sample selection block. This block inputs a series of samples alongside the training dataset, and selects one of the input samples, and returns it as an output. To do so, for each sample it finds the closest match, using Euclidean distance, in the training dataset. After that, it will add a Laplacian noise to this calculated distance to compute a noisy distance. It chooses the one with the largest noisy distance value among all samples. Our intuition here is that if there is an outlier in the training dataset, the generator can get overfit on it to increase the accuracy, while this can compromise privacy. In this case, the distance between that individual and a synthesized sample is low, and the sample selection block will not choose it with a high probability because of its structure. Generally speaking, if a synthesized sample gets very close to a match in the original one, its noisy distance is low, and the sample selection block will not choose it. On the other hand, because of the noise added here, there is a kind of uncertainty about selecting a sample that would guarantee privacy.

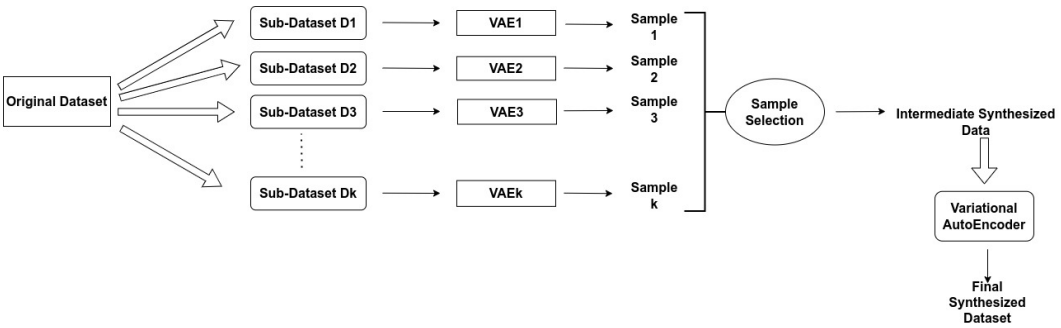
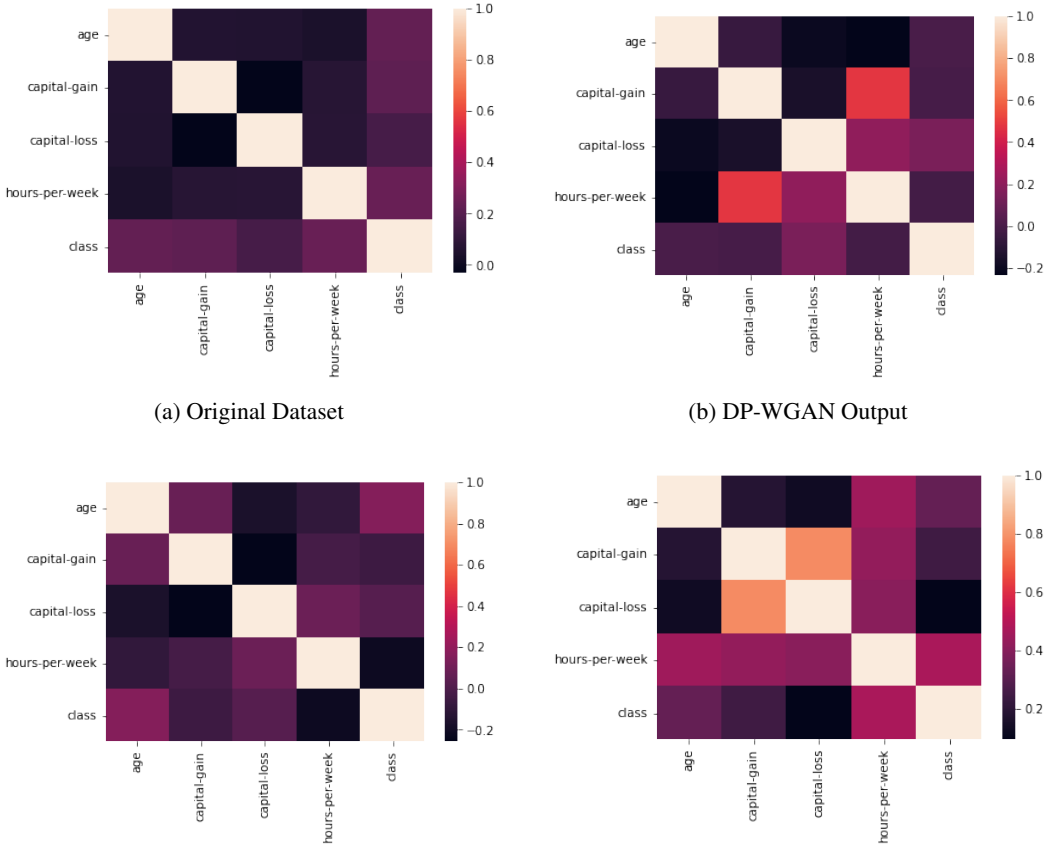


Figure 1: Proposed framework: (1) an ensemble of generative models trained on subsets of the original dataset. (2) each model outputs a sample after training. (3) sample selection block selects one of the samples to create the intermediate dataset. (4) Final student model trains on the intermediate dataset. Note that we use variational autoencoder models as our generator.

Figure 1 shows the structure of our proposed framework.



(a) Original Dataset

(b) DP-WGAN Output

(c) PATE-GAN Output

(d) Proposed Model Output

Figure 2: Heatmap of Correlation matrix

5 Results

Our strategy toward studying the statistical similarity of the original dataset and the synthesized one starts by computing the correlation matrix of features. As discussed earlier, this will provide us with the relation between features in the original dataset and the synthesized one. For this section, we run our experiments on the Census Adult dataset. We used the implementation provided here[2], and here[3] to generate synthesized data on the Census Adult dataset. Then, after generating it, we compared the results[4]. To make the learning faster, we normalized the Census Adult dataset features to be in the range of [0, 1], and after generating synthesized data we reversed the normalization. What is more, for the implementation of our proposed method we used 10 variational autoencoders. The details of this implementation are provided in appendix A.

Figure 2 shows the heatmap of feature correlation in the original dataset. This figure gives us intuition that to some extent all of the provided algorithms preserve the relations between features. To experiment in more details, we calculate the sum square error of the correlation matrix as follow:

$$SSE = \sum (MC - OC)^2$$

Where MC is the correlation matrix of the model under study, and OC is the correlation matrix of the main dataset. The power operator is element-wise power in a matrix, and the sum is the sum over all elements of each row in a matrix. Results are shown in the table 1.

[2]https://github.com/BorealisAI/private-data-generation

[3]https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/pategan

[4]For the DP-GAN implementation, we used the variant that uses Wasserstein distance. This variant is also known as DP-WGAN.

	DP-WGAN	PATE-GAN	Proposed Model
age	0.206777	**0.071572**	0.199806
capital-gain	0.247967	**0.135406**	0.808232
capital-loss	**0.096976**	0.118604	0.783392
hours-per-week	0.328045	**0.237578**	0.458623
class	0.168672	0.307770	**0.070163**

Table 1: Sum of Square Error of correlation matrix between various approaches and original dataset.

In the table 1, the best method in each row is highlighted in bold. It can be interpreted from this table that PATE-GAN can mimic the relation between features more accurately. The main reason for this trend is that since DP-WGAN uses DP-SGD, the noises added to gradient descent will eventually accumulate on each other and affect the final accuracy. In this experiment, the privacy budget is set to be fixed to $\epsilon = 8$. Note that although Census Adult has only 14 features, most of its features are categorical, which makes the training even harder. Another interesting part here is that the Census Adult dataset is a classification task in which the labels are the 'class' feature. As you can see in the table, our proposed method was able to capture the correlation in this feature more accurately.

As we discussed earlier in the methodology section, the next step is to study the similarity of features in the synthesized dataset and the original dataset in a stand-alone manner. To do so, we compute the mean and variance of each feature.

	Original Dataset	DP-WGAN	PATE-GAN	Proposed Model
age	38.5326	1.7792	44.472145	**37.5152**
capital-gain	1033.6284	11579.4356	50058.197550	**8187.5314**
capital-loss	93.6952	-499.8672	2191.371395	**298.2812**
hours-per-week	40.5150	-8.5138	49.421968	**40.5828**
class	0.2442	**0.2514**	0.500198	0.3610

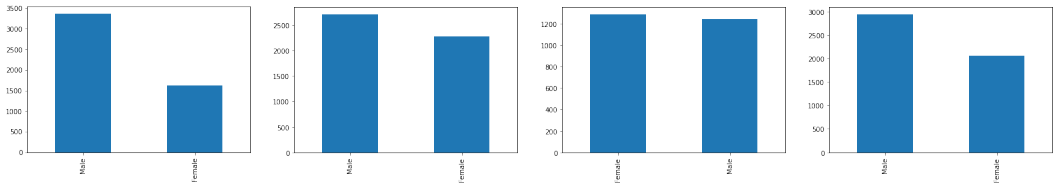
Table 2: Mean value of each feature in original and synthesized datasets.

	Original Dataset	DP-WGAN	PATE-GAN	Proposed Model
age	1.842814e+02	1.062483e+03	**2.101505e+02**	9.927242e+01
capital-gain	4.972772e+07	1.481932e+09	**2.485816e+08**	2.732494e+08
capital-loss	1.687529e+05	1.856199e+06	4.581812e+05	**2.453345e+05**
hours-per-week	1.465983e+02	1.357706e+03	3.388614e+02	**5.432961e+01**
class	1.846033e-01	**1.882357e-01**	2.500988e-01	2.307251e-01

Table 3: Variance value of each feature in original and synthesized datasets.

Table 2 and table 3 show the first two momentum of data. It is more suitable that the momentums be as close as possible to the original ones. The closest values are highlighted in bold. The results show that our proposed method could preserve the statistical properties of the original dataset more accurately in terms of momentum. It is worth mentioning that the results from PATE-GAN are very close to the original dataset too, which shows this method can work on this dataset quite well. Another interesting part is that DP-GAN is not much successful in synthesizing datasets. Our hunch is that it can be mainly because of two reasons. First, the Census Adult dataset is mostly categorical. It has 14 features, but after doing one hot encoding, the number of features will increase to 98. In this situation, our training set is sparse, which can adversely affect the training of this model. Another reason can be that the maximum privacy budget here is not enough for this method to capture the complexity of the training dataset. Both of these intuitions need further experiments.

Moving next to categorical features, we analyze the histogram of them in Figure 3. Results indicate that our model was able to mimic the trend in the categorical data better than the other two methods.



(a) Original Dataset (b) DP-WGAN Output (c) PATE-GAN Output (d) Our Output

Figure 3: Histogram of sex feature

So far, we analyzed datasets using different matrices and various perspectives. The final question here is how to consider all these metrics together. To find the quality of a synthesized dataset here, we introduce a distance metric that uses all the previous analyzes. It is as equation 1.

$$\begin{aligned} Distance = \sum SSE + \sum_{features} (model\ mean - original\ dataset\ mean)^2 \\ + \sum_{features} (model\ variance - original\ dataset\ variance)^2 \\ + \sum_{features} (\sum_{category} (model\ count - original\ dataset\ count)^2) \end{aligned} \quad (1)$$

For more accurate results, we normalized this distance to get equation 2, where w is some weight. The results for equation 2 are shown in table 4.

$$\begin{aligned} Distance = w_1 * \sum SSE + w_2 * \sum_{features} (1 - \frac{model\ mean}{original\ dataset\ mean})^2 \\ + w_3 * \sum_{features} (1 - \frac{model\ variance}{original\ dataset\ variance})^2 \\ + \sum_{features} w_{4i} * (\sum_{category} (1 - \frac{model\ count}{original\ dataset\ count})^2) \end{aligned} \quad (2)$$

DP-WGAN	PATE-GAN	Proposed Model
236.1874	137.8452	**5.9794**

Table 4: Distance of the synthesized dataset from the original one.

This result is one of the most crucial parts of our project. The reason that the distance for PATE-GAN and DP-GAN is high is that these methods have trouble synthesizing categorical data. It can be seen in the histograms too. The high difference between the histograms of categorical data leads to this high distance. It also indicates that one aspect of improvement for these methods is to make them more suitable for sparse datasets like categorical ones, which can lead us to some future works.

Our final step toward this project is to study the pMSE. As we mentioned earlier, it shows that to what extent the synthesized data can not be distinguished by a classifier. Thus, it is a good metric to be considered as the quality of the synthesized dataset. Table 5 shows the results.

In this section, an SVM classifier is used for classification. To this end, we used sklearn library[5] with a linear kernel. It indicates that although the data synthesized by PATE-GAN have statistical similarities to the original one, a classifier is still able to distinguish it from the original dataset. Moreover, this data gives us more important outcomes. First, with the same amount of privacy, PATE-GAN leads to

[5]https://scikit-learn.org/

DP-WGAN	PATE-GAN	Proposed Model
0.2447	0.2083	**8.2675e-05**

Table 5: pMSE of the synthesized dataset created by different methods.

better-synthesized data compared to DP-GAN. In other words, DP-GAN requires a higher privacy budget for satisfactory outputs. Secondly, both DP-GAN and PATE-GAN have trouble synthesizing categorical features and sparse datasets. Third, our motivation in the first place was the quality of the synthesized dataset. It can be seen that PATE-GAN is still far from this requirement in our empirical study.

6 Conclusion

The most important result of this survey would be to present a new perspective on studying synthesized datasets. We analyzed a dataset from different aspects and using pMSE we showed that having similar statistics can not be enough, and a classifier is still able to distinguish synthesized data. It is worth mentioning that through our experiments, with the same amount of privacy budget, PATE-GAN showed better results compared to DP-GAN.

Moreover, we showed that the mentioned methods have trouble synthesizing sparse datasets. One future work can be to improve our learning algorithms and frameworks against these kinds of datasets.

We introduced a distance metric, which gives us a foundation for analyzing the quality of a synthesized dataset. Since this metric determines how far two databases are from each other, one way of improving our learning algorithm in future works can be using this metric as some feedback to improve outputs.

Finally, we introduced a new framework that uses the intuition from the PATE framework to do a noisy sampling, and we showed its utility compared to other methods. It is still left to be discussed in further detail the perks and disadvantages of this framework, which can be done in future works.

References

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

B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 273–282, 2007.

Q. Chen, C. Xiang, M. Xue, B. Li, N. Borisov, D. Kaarfar, and H. Zhu. Differentially private data generative models. *arXiv preprint arXiv:1812.02274*, 2018.

T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

C. Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

J. Jordon, J. Yoon, and M. Van Der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

J. Snoke and A. Slavković. pMSE mechanism: differentially private synthetic data with maximal distributional similarity. In *International Conference on Privacy in Statistical Databases*, pages 138–159. Springer, 2018.

T. Takahashi, S. Takagi, H. Ono, and T. Komatsu. Differentially private variational autoencoders with term-wise gradient aggregation. *arXiv preprint arXiv:2006.11204*, 2020.

J. Ullman and S. Vadhan. PCPs and the hardness of generating private synthetic data. In *Theory of Cryptography Conference*, pages 400–416. Springer, 2011.

L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

J. Zhang, X. Xiao, and X. Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proceedings of the 2016 International Conference on Management of Data*, pages 155–170, 2016.

J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

Appendices

A Network Setting

To implement our proposed model, we used the TensorFlow library. Our Model consists of 10 generators and a sample selection block. Each of these generators is a variational autoencoder. The details of each variational autoencoder are as follows: In the encoder part, there is one hidden layer with Relu activation. This layer consists of 64 nodes. The latent space layer, z, consists of 4 nodes. The decoder part consists of one hidden layer with a Relu activation function and 64 nodes. The final layer is an output layer with the shape same as the input layer.