Privacy Preserving Synthetic Data Release Using Deep Learning

Conference Paper · December 2018

DOI: 10.1007/978-3-030-10925-7_31

5 authors, including:

Nazmiye ceren Abay
University of Texas at Dallas
6 PUBLICATIONS **57** CITATIONS
SEE PROFILE

Yan Zhou
University of Texas at Dallas
19 PUBLICATIONS **189** CITATIONS
SEE PROFILE

Bhavani M. Thuraisingham
University of Texas at Dallas
220 PUBLICATIONS **4,893** CITATIONS
SEE PROFILE

Privacy Preserving Synthetic Data Release Using Deep Learning

Nazmiye Ceren Abay[1], Yan Zhou[1], Murat Kantarcioglu[1,2], Bhavani Thuraisingham[1], and Latanya Sweeney[2]

[1] University of Texas at Dallas, Richardson, Texas
{nca150130,yan.zhou2,muratk,bxt043000}@utdallas.edu
[2] Harvard University, Cambridge, Massachusetts
latanya@fas.harvard.edu

Abstract. For many critical applications ranging from health care to social sciences, releasing personal data while protecting individual privacy is paramount. Over the years, data anonymization and synthetic data generation techniques have been proposed to address this challenge. Unfortunately, data anonymization approaches do not provide rigorous privacy guarantees. Although, there are existing synthetic data generation techniques that use rigorous definitions of differential privacy, to our knowledge, these techniques have not been compared extensively using different utility metrics.
In this work, we provide two novel contributions. First, we compare existing techniques on different datasets using different utility metrics. Second, we present a novel approach that utilizes deep learning techniques coupled with an efficient analysis of privacy costs to generate differentially private synthetic datasets with higher data utility. We show that we can learn deep learning models that can capture relationship among multiple features, and then use these models to generate differentially private synthetic datasets. Our extensive experimental evaluation conducted on multiple datasets indicates that our proposed approach is more robust (i.e., one of the top performing technique in almost all type of data we have experimented) compared to the state-of-the art methods in terms of various data utility measures.

Keywords: Differential Privacy · Deep learning · Data Generation.

1 Introduction

Increasingly more data is collected about almost every aspect of human life ranging from health care delivery to social media. As the amount of collected data increases, more opportunities have emerged for leveraging this collected data for important societal purposes.

Since the collected data can be used to offer important services and facilitate much needed research, many organizations are striving to share the data that they collect. For example, many novel applications, such as smart cities [34] and personalized medicine [27], require the collection and sharing of privacy sensitive micro-data, i.e., information at the level of individual respondents.

Due to the importance of these goals, many organizations advocate for openly sharing data to serve important social, economic and democratic goals. For example, in 2016, the City of Seattle announced an open data policy where the city's data would be "open by preference" except when such data sharing may affect individual privacy. Similarly, National Institute of Health (NIH) requires the sharing of genomic data created as a part of NIH funded research with other researchers.

At the same time, sharing micro data carries inherent risks to individual privacy. For example, a municipal dataset that contains information about bike sharing has been used to identify individuals and their transit patterns [32]. Similarly, a taxi ride data set from New York have been used to identify certain individuals' addresses and their trips to certain night clubs [33]. These examples show that there is an important societal need in sharing micro data while protecting individual privacy.

To address this privacy challenge, solutions have been proposed in two broad categories. In the first category, the data anonymization based approaches (e.g., [30]) try to use various definitions to sanitize data so that it cannot be easily re-identified. Although these approaches have some important use cases, they are not usually based on rigorous privacy definitions that can withstand various types of re-identification attacks. In the second category, synthetic data generation approaches have been proposed to generate realistic synthetic data using rigorous differential privacy definition [12]. Although these approaches have been shown to work in some limited cases, they have not been extensively tested on different types of use cases with different requirements (e.g., high dimensionality, correlation among features). Therefore, it was not clear which technique works well under what conditions for what type of data sets. We answer these questions by conducting extensive experimentation. Furthermore, we provide a new differentially private deep learning based synthetic data generation technique to address the limitations of the existing techniques.

In this paper, we propose an auto-encoder technique ($\mathrm{DP\text{-}SYN}$), a generative deep learning technique that generates privacy preserving synthetic data. We test our approach on benchmark datasets and compare the results with other state-of-the-art techniques. We show that our proposed technique outperforms them in terms of three evaluation metrics.

Our contributions can be summarized as follows:

- We test existing techniques using different datasets with different properties using three utility metrics. We show that none of the existing techniques consistently outperforms others on all types of data sharing tasks and datasets.
- We propose a novel differentially private deep learning based synthetic data generation technique that is shown to be robust under different utility metrics with respect to different synthetic data generation tasks.
- We show that our approach does not deteriorate when faced with imbalanced or high dimensional datasets. Due to an inner partitioning of the latent structure, our approach gives more robust results in noise addition and works with both relational and image data.

This work is organized as follows. In Section 2, we discuss the related work. In Section 3, we provide the preliminaries for our work. In Section 4, we discuss our novel differentially private auto-encoder based technique for synthetic data generation. In Section 5, we run extensive empirical analyses to understand the relative strengths of the existing techniques and show that our technique works well in almost all of the given datasets. Finally, we conclude in Section 6.

2 Related Work

Extensive research has been conducted on publishing private data for preserving privacy. Despite their success in data utility measures, most of the proposed methods in the literature are impractical to be implemented for high dimensional data. In this section, we discuss the related techniques with their strengths and limitations.

In statistical analysis, publishing a marginal table while preserving the privacy has been a fundamental research goal. One of the initial efforts in addressing this problem is proposed by Barak et al. [3]. In this method, a full contingency table constructed on the original data is represented by the Fourier coefficients. The noise is then added to these coefficients in order to construct the desired k-way marginal tables, instead of perturbing the original data. Despite its feasibility and widespread use in low dimensional data, the number of Fourier coefficients, 2^d, grows exponentially with increased dimensionality. This results in intractable computational cost when working with high dimensional data. Another method is designed by Ding et al. [9] to work with high dimensional data such as online analytical processing (OLAP). In this framework, strategic cuboids that are useful to generate other cuboids are chosen first, and a private version of these cuboids is constructed by using differential privacy. The main limitation of this study arises while constructing the strategic cuboids. As all possible cuboids are iteratively traversed and selected, the number of the cuboids grows with the dimensions of the data, resulting in an increased complexity. A more practical and efficient approach, known as PriView, addresses the high dimensionality problem [24]. PriView also constructs the private k-way marginal tables for $k \geq 3$. While constructing private marginal tables, PriView first extracts low-dimensional marginal views from the flat data and adds noise to the views. Next, PriView applies a reprocessing technique to ensure the consistency of the noisy views. Afterwards, PriView applies maximum entropy optimization on the views to obtain the k-way marginal tables. PriView is reported as a more efficient technique in terms of time and space complexity; however, it can be employed on binary data only.

There are other frameworks, designed particularly for differential optimization problems. First, Dwork et al. [11] propose an output perturbation technique that directly add noise to the regularized objective function after optimization. This technique is outperformed by the objective perturbation technique proposed by Chaudhuri et al. [8] which adds noise to the objective function before optimization. We denote this work as PrivateSVM and compare its results to those of DP-SYN in the experiments section.

Differential privacy has been implemented in a number of data analysis tasks, including regression models [7, 37], classification models [17, 26, 31] and association rule mining [20, 35].

Generating artificial data from the original one is another privacy preserving technique for data publication. Here, instead of using the sanitization models discussed previously, Rubin [25] introduces repetitive perturbation of the original data as a substitute to the original data. To execute this technique, Zhang et al. [36] present a synthetic data generation technique, PrivBayes. PrivBayes is defined as a differential generative model that decomposes high dimensional data into low dimensional marginals by constructing a Bayesian network. Afterwards, noise is injected into these learned low dimensional marginals to ensure differential privacy and the synthetic data is inferred from these noised marginals. Although PrivBayes is credited as an effective technique, as we will show in our experiments, our proposed technique has a significant improvement over PrivBayes.

Acs et al. [2] model generative neural networks to produce synthetic samples. The authors first cluster the original datasets into k clusters with private kernel k-means. Afterwards, they use generative neural networks for each cluster to create synthetic data. In our experiments, we denote this work with DP-VAE and compare its results to our method.

Bindschaedler et al. [5] present another differential generative framework. The authors introduce an idea of *plausible deniability*, rather than adding noise to the generative model directly. Plausible deniability is ensured by a *privacy threshold* in releasing synthetic data. Here, an adversary cannot tell whether a particular input belongs to the original data by observing synthetic records.

Park et al. [22] propose a private version of the iterative expectation maximization algorithm. They effectively combine differential privacy and expectation maximization algorithm to cluster datasets. Here, we use this approach to discover patterns in latent space. We observed an improvement in the performance of this technique when used with partitioning the original dataset into unique data label groups. We use this modified version in our experiments [22] as DP-EM(SYN) and compare its results in the experiments section.

In some cases, combining differentially private algorithms has been proven to be useful in formulating more complex privacy solutions. However, such combinations may result in degradation of the privacy protection as more information is leaked by multiple usage of the private techniques. To track the total privacy loss while executing such mechanisms, Dwork et al. [12] propose basic and advanced composition theorems. Abadi et al. [1] propose another advanced composition theorem known as the *moments accountant* and verify that it has the best overall privacy bound in the literature. Abadi et al. also utilize moment accountant while constructing a deep learning technique to classify images.

3 Preliminaries

This section briefly recalls the definitions and standards of differential privacy and the principles of deep learning.

3.1 Differential Privacy

Differential privacy is the formal mathematical model that ensures privacy protection, and it is primarily used to analyze and release sensitive data statistics [11]. Differential privacy utilizes randomized algorithms to sanitize sensitive information while bounding the privacy risk of revealing sensitive information.

Theorem 1 ((ε, δ)-Differential Privacy [11]). *For two non-negative numbers ε, δ, a randomized algorithm, $\mathcal{F}$, satisfies (ε, δ)-differential privacy iff for any neighboring pair d, d' and $S \subseteq$ Range($\mathcal{F}$), the following formula holds:*

$$Pr[\mathcal{F}(d) \in S] \leq e^{\varepsilon} Pr[\mathcal{F}(d') \in S] + \delta. \tag{1}$$

Here, the neighboring pair differ from each other with only one entry while the remaining entries are identical. In Theorem 1 [11], δ is a relaxation to ε*-differential privacy* that formulates the probability of privacy leakage. However, to avoid such leakage, Dwork et al. [12] shows that δ must be chosen smaller than $1/n$ for a data of n samples.

Our proposed technique sanitizes sensitive data based on a widely used differentially private technique, the Gaussian mechanism [6]. The deterministic function f takes d as input. $f(d)$ perturbs the input with noise sampled from the normal distribution $\mathcal{N}$, based on ϵ, δ, and s_f which is the sensitivity of f defined as follows:

Definition 1 (Sensitivity [11]). *For a given function f, the sensitivity of f is defined as a maximum absolute distance between two neighboring pairs (d,d')*

$$s_f = \max_{(d,d')} \|f(d) - f(d')\|, \tag{2}$$

where $\|.\|$ is L_1 norm.

The (ε, δ)*-differential privacy* of function f over data d is guaranteed by $\mathcal{F}(d)$ with the Gaussian mechanism:

$$\mathcal{F}(d) = f(d) + z, \tag{3}$$

where z is a random variable from distribution $\mathcal{N}(0, \sigma^2 {s_f}^2)$. Here, when $\varepsilon \in [0, 1]$, the relation among the parameters of Gaussian mechanism [12] is such that

$$\sigma^2 \varepsilon^2 \geqslant 2 \ln\left(1.25/\delta\right) s_f^2.$$

3.2 Deep Learning

Deep learning is a subfield of machine learning that can be either supervised or unsupervised [16]. The power of deep learning comes from discovering essential concepts of data as nested hierarchy concepts where simpler concepts are refined to obtain complex concepts. Deep learning has been applied to many different research areas including computer vision [16], speech recognition [18] and bio-informatics [19].

We focus on the auto-encoder, an unsupervised deep learning technique that outputs a reconstruction of its input.

Similar to an artificial neural network, an auto-encoder is trained by optimizing an objective function. Stochastic gradient descent (SGD) is used as a scalable technique [29] to solve this optimization problem. Rather than iterating over every training instance, SGD iterates over a *mini-batch* of the instances. For a given training set of m samples, $D{=}\{x_i\}_{i=1}^{m}$ and $x_i \in \mathbb{R}^d$, the objective function is given as:

$$\min_{w} \mathcal{L}(w) = \frac{1}{|B|} \sum_{x_i \in B} \ell(w; x_i), \tag{4}$$

where B is the *mini-batch*, w is auto-encoder model parameter and ℓ is the discrepancy cost of example x_i and its reconstruction $\tilde{x}_i$.

At each step t, model gradient is computed for a given batch B_t and learning parameter η. Then, the model parameter is updated for the next step as follows:

$$w_{t+1} = w_t - \eta \left(\frac{1}{|B_t|} \sum_{x_i \in B_t} \nabla_w \, \ell(w; x_i) \right). \tag{5}$$

Figure 1 presents two main phases of an auto-encoder: the **encoder** and the **decoder**.

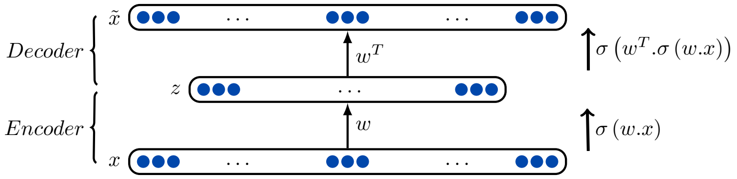


Fig. 1: One hidden layer auto-encoder that encodes the input to the latent space, and decode the latent space to the reconstruction of input.

The **encoder** maps its input to a hidden intermediate layer that usually has less neurons than the input size to get a latent representation of the input. Here, the element-wise activation function σ maps $x \in \mathbb{R}^d$ into $z \in \mathbb{R}^{d'}$ where $d' < d$. On the other hand, the **decoder** takes the latent representation z, and reconstructs $\tilde{x} \in \mathbb{R}^d$.

In Section 4, we describe how the SGD model defined in Equation 5 is used in the inference of model parameter w.

4 Methodology

This section describes the main components of our differentially private synthetic data generation approach. We first introduce our private auto-encoder and explain the private expectation maximization algorithm. Next, we present the privacy analysis of the proposed technique.

4.1 Differentially Private Synthetic Data Generation Algorithm

Our main framework aims to generate synthetic data without sacrificing the utility. A similar approach is proposed in [1] which designs a private convolutional neural network on supervised learning. However, this method can only be used in classification tasks. We combine this method with DP-EM and to create a generative deep learning model.

Assume that we have the sensitive dataset $D = \{(x_1, y_1), \dots, (x_m, y_m)\}$, where every instance $x \in \mathbb{R}^d$ has a label $y \in \{1, \dots, k\}$. We partition the sensitive dataset D into k groups denoted as $\hat{D}_1 \dots \hat{D}_k$ such that every instance x in a group $\hat{D}_i \in D$ has the same label y. The value of k is limited by the number of unique labels in dataset D.

Figure 2 shows the two main steps of our approach. For each data group we build a private generative auto-encoder which are denoted with DP-SYN. The lower pane of the figure shows the inner working of a DP-SYN.

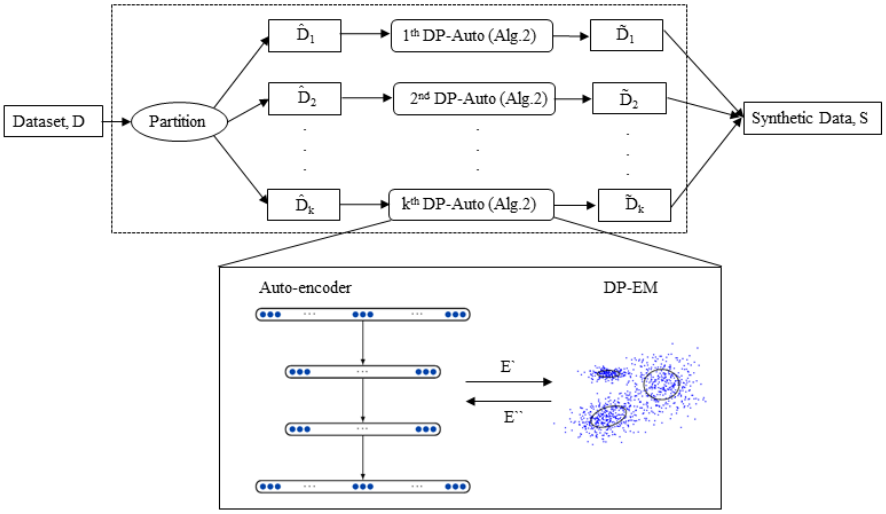


Fig. 2: Differentially Private Synthetic Data Generation DP-SYN

The details of our technique are shown in Algorithm 1. After partitioning the dataset D into k groups (Line 1 in Alg. 1), the noise injected to each group is also partitioned (Line 4 in Alg. 1), as specified in the sequential composition theorem [10]. For each previously obtained group we build one private auto-encoder (Line 5 in Alg. 1), which is detailed in Algorithm 2. Next, we obtain the private latent representation of the group (Line 6 in Alg. 1), and inject it into a differentially private expectation maximization (DP-EM) function. The DP-EM function is detailed in [22]. The main task of DP-EM is to detect different latent patterns in the encoded data and to generate output data with similar patterns. Here, DP-EM

Algorithm 1 DP-SYN: Differentially Private Synthetic Data Generation

```
Input: $D$: $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ where $x \in \mathbb{R}^d$, $y \in \{1, . . . , k\}$; $\eta$: learning rate; $T$:
    iteration number; $\varepsilon$: privacy budget; $\delta$: gaussian delta; $\sigma$: standard deviation; $C$:
    clipping constant.
Output: $S$: Synthetic data.
 1: $\{\hat{D}_1 \ldots \hat{D}_k\} \leftarrow$ Partition data records in $D$ based on associated labels
 2: $S \leftarrow \{\}$
 3: for $i = 1$ to $k$ do
 4:     Partition $\varepsilon$ into $\varepsilon_A = \varepsilon/2$, $\varepsilon_H = \varepsilon/2$ and $\delta$ into $\delta_A = \delta/2$, $\delta_H = \delta/2$
 5:     $W \leftarrow$ DP-Auto $\left(\hat{D}_i, \eta, \mathrm{T}, \varepsilon_A, \delta_A, \sigma, \mathrm{C}\right)$ // see Alg. 2
 6:     $E' \leftarrow encode\left(W, \hat{D}_i\right)$
 7:     $E'' \leftarrow$ DP-EM $\left(E', \varepsilon_H, \delta_H\right)$ // see [22]
 8:     $\tilde{D}_i \leftarrow decode\left(W, E''\right)$
 9:     $S \leftarrow S \cup D'_i$
10: return $S$
```

is used to sample encoded data (Line 7 in Alg. 1) and newly sampled encoded data is decoded with using the model parameter W (Line 8 in Alg. 1) . $\tilde{D}_i$ is the synthetic data associated with an inputted group $\hat{D}_i$ and appended to the S to be output (Line 9 in Alg. 1).

4.2 Building a private auto-encoder

In this section, we discuss the private auto-encoder given in Algorithm 2.

Our private auto-encoder employs steps to improve the optimization process with gradient computation and clipping. While a gradient is computed for a batch in the standard stochastic training techniques, we compute the gradient for each training instance instead. This approach improves the optimization process since it reduces the sensitivity of the gradient present at each instance [14]. Norms of the gradients define the direction that optimizes the network parameters. However, in some deep networks, the gradients can be unstable and fluctuate in a large range. Such fluctuations can inhibit the learning process due to the increased vulnerability of the networks. To avoid this undesired situation, we bound norms of the previously computed gradients by a clipping constant C [23].

After clipping the gradients, noise is sampled from the Gaussian distribution with zero mean and standard deviation of $\sigma\, C$ and added to the previously clipped gradients (Line 9—10 in Alg. 2). At the end of each batch, model parameters of the network are updated with the negative direction of the learning rate η multiplied by the averaged noisy gradients. At the end of this step, the private auto-encoder outputs the model parameter w (Line 11 in Alg. 2).

Algorithm 2 DP-Auto: Differentially Private Auto-encoder

```
Input: $\eta$: Learning rate; $T$: iteration number; $\varepsilon_A$: privacy budget; $\delta$: gaussian delta;
    $\sigma$: standard deviation; $C$: clipping constant.
Output: $w$: Model parameter.
 1: $\ell$ is the objective function
 2: $\nabla\ell$ is the gradient of objective function
 3: initialize $w_0$ randomly
 4: $\varepsilon'_A = 0$
 5: for $t = 1$ to $T$ do
 6:     if $\varepsilon'_A < \varepsilon_A$ then
 7:         $B_t \leftarrow$ random batch
 8:         $i_t \sim b$ where $x_{i_t} \in B_t$
 9:         $z_{i_t} \sim \mathcal{N}(0, \sigma^2 C^2)$
10:         $w_{t+1} \leftarrow w_t - \eta \cdot \left( \frac{1}{|B_t|} \sum_{i_t} \left( \nabla\ell(w_t; x_{i_t}) + z_{i_t} \right) \right)$
11:         $\varepsilon'_A \leftarrow$ calculate privacy loss with moments accountant
12: return $w$
```

4.3 Privacy Analysis

The privacy analysis of our proposed technique employs the *moments accountant* approach developed by Abadi et. al. [1] to keep track of the privacy cost in multiple iterations. Moments accountant is a combination of both the strong composition theorem [13] and the privacy amplification theorem [4]. Moments accountant has an improvement on estimating of the privacy loss for composing differentially private Gaussian mechanisms, and it is the best for overall estimation of privacy budget in literature [1].

In our proposed work, while training the auto-encoder, we track the privacy loss at the end of each batch iteration. As given in Lines 5—11 of Alg. 2, we compute the value of current privacy loss ε' that has been spent on private auto-encoder in a given iteration $t \in T$. Training ends when ε' reaches the final privacy budget ε.

According to moments accountant, Algorithm 2 is (ε, δ)-differentially private if the privacy loss for any $\varepsilon' < k_1(|B|/n)^2 T$ is such that for some constants k_1, k_2:

$$\varepsilon' \geq k_2 \frac{|B|/n\sqrt{T \log(1/\delta)}}{\sigma},$$

where T is the number of training steps and $|B|$ is the mini-batch for a data of n samples with a given privacy budget ε, gaussian delta δ and standard deviation σ of the Gaussian distribution.

5 Experiments

In this section, we present the experimental results to demonstrate the efficiency of our proposed approach. We compare our results with other state-of-the-art techniques. To ensure fairness, we also employ the *Gaussian mechanism* in these techniques.

We start the evaluation by explaining the experimental settings. We evaluate the performance with statistical measures, accuracy in machine learning models and agreement rate. For each task, 70% of the data is used as a training set, while the rest is used for testing.

5.1 Experimental Settings

Datasets. We test the proposed approach on nine real datasets. The following is a brief description of each dataset:

(i) The **Adult** [21] dataset contains the information of **45222 individuals**, extracted from the 1994 US census. The dataset shows whether the income of the individuals exceeds fifty thousand US dollars. The dataset contains **15 features**.

(ii) The **Lifesci** [21] dataset contains **26733 records** and **10 principal components** from chemistry and biology experiments.

(iii) The **Optical Digit Recognition** (ODR) [21] dataset contains **5620 handwritten digits of 10**. Each instance is represented by **64 numeric features**.

(iv) The **Spambase** [21] dataset contains **4601 emails**, each of which is labeled as spam or non-spam. Each instance has **58 attributes**.

(v) The **Contraceptive Method Choice** (CMC) [21] dataset contains **9 features** of **1473 married women** to predict their current contraceptive method choice.

(vi) The **German Credit** [21] dataset contains the anonymized information of **1000 customers** with **20 features**. Each customer is classified as good or bad credit risk.

(vii) The **Mammographic Mass** [21] dataset contains the information of **961 patients'** mammographic masses of with **5 attributes**. The class value shows that patient has breast cancer or not based on mammographic mass.

(viii) The **Diabetes** [21] dataset contains the information of female patients who are at least 21 years old. Each patient is classified as diabetic or not diabetic. The dataset has **768 records** with **8 features**.

(ix) The **BreastCancer** [21] dataset contains the information about whether a patient has breast cancer or not. It has **699 patient records** with **10 features**.

In all experiments, we compare our results with four state-of-the-art techniques: PrivateSVM [8], PrivBayes [36], DP-EM(SYN) [22] and DP-VAE [2].

We repeat each experiment 10 times for each task and report the average measurements in our experimental results. In total, our experiments consist of 7840 runs of the mentioned techniques. Here, we only report the best results from each algorithm.

5.2 Accuracy in Machine Learning Models

In this set of experiments, we evaluate the accuracy in a Support Vector Machine (SVM) [15] task. More specifically, we report the percentage of incorrectly classified tuples as the *misclassification rate*. For the PrivateSVM, out of two proposed approaches we only report results from the objective perturbation approach because it outperforms the output perturbation approach.

For each training set, we generate synthetic data by using each method and construct SVM models on the synthetic data. Performance of these models is evaluated on the test set. PrivateSVM has a regularization parameter λ for the objective function. We run PrivateSVM with $\lambda \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-8}\}$ and pick the model that reports the lowest misclassification rate.

Figure 3 presents the misclassification rate of the techniques for a given (ε, δ) pair. In the figure, the black straight line shows the misclassification rate on the original dataset, i.e., without privacy. Here, it presents the best case to aim for. We now compare the performance of DP-SYN with respect to each state-of-the-art method.

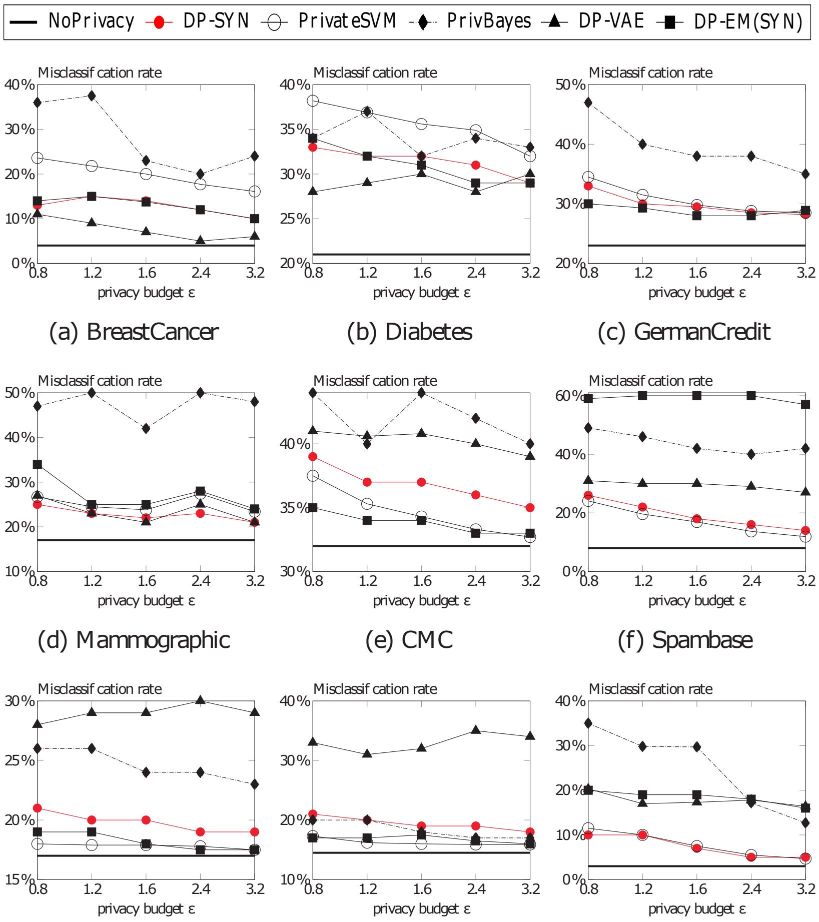
Figure 3 shows that DP-SYN has better performance than PrivBayes for eight out of the nine datasets. Only for the Adult dataset, PrivBayes performs slightly better than our DP-SYN approach. DP-SYN outperforms DP-VAE for seven datasets. For BreastCancer and Diabetes, DP-VAE has better performance; however, it fails to classify any instance in the GermanCredit dataset. For high dimensional datasets such as Spambase and ODR, the misclassification rate of DP-EM(SYN) is two times bigger than that of DP-SYN. A reason for this high misclassification rate shows that DP-EM(SYN) fails in generating synthetic data task when the dimension of data is more than two dozens. DP-SYN also outperforms PrivateSVM in five datasets. PrivateSVM is specifically designed for SVM, and it is expected to have lower misclassification rates in SVM tasks. However, PrivateSVM cannot be employed in other machine learning tasks easily.

Consequently, we point out that on datasets of different types, no single method gives the best misclassification rate consistently. As shown in Figure 3-c with the absence of DP-VAE, some algorithms cannot even classify a dataset if the dataset is highly imbalanced. Considering these issues, Figure 3 shows that DP-SYN can be employed on all datasets and reports competitive results.

5.3 Statistical Measures

We evaluate the quality of synthetic data in terms of statistical utility. We generate k-way marginals of the dataset and compare the probability distribution of the noisy and original marginals. *Total variation distance* [28] is used to report the statistical difference between the noisy and original marginals. The datasets used in the experiments are large, leading to prohibitively large queries. Hence, considering this problem, we generate only 2-way and 3-way marginals as used in [36].

Figure 4 shows that DP-SYN performs better than PrivBayes, DP-VAE and DP-EM(SYN) for the 3-way marginals of BreastCancer and Diabetes datasets. In 2-way marginals of BreastCancer and Diabetes datasets, our method performs



(a) BreastCancer (b) Diabetes (c) GermanCredit

(d) Mammographic (e) CMC (f) Spambase

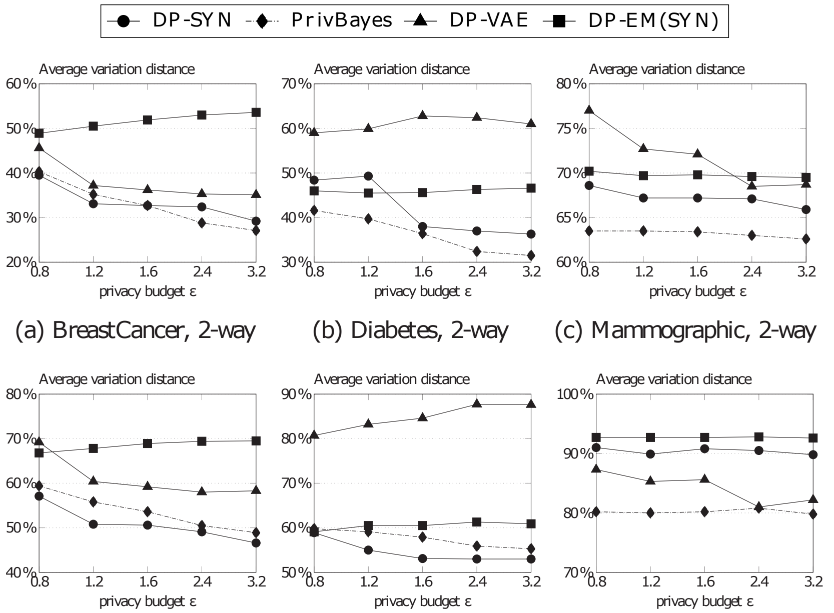
(g) Lifesci (h) Adult (i) ODR

Fig. 3: Misclassification rates for the nine datasets.

better than state-of-the-art with the exception of PrivBayes. However, for 2-way marginals of Mammographic, our results are competitive with those of PrivBayes. Overall, DP-SYN preserves the statistical information better than comparable to the state-of-the-art techniques in all datasets.

5.4 Agreement Rate

In this section, we evaluate the quality of the synthetic data in terms of the *agreement rate* in an SVM label prediction task. Specifically the agreement rate is defined as the percentage of records for which the two classifiers make the same prediction [5].



(a) BreastCancer, 2-way (b) Diabetes, 2-way (c) Mammographic, 2-way

(d) BreastCancer, 3-way (e) Diabetes, 3-way (f) Mammographic, 3-way

Fig. 4: Statistical difference between the noisy and original k-way marginals.

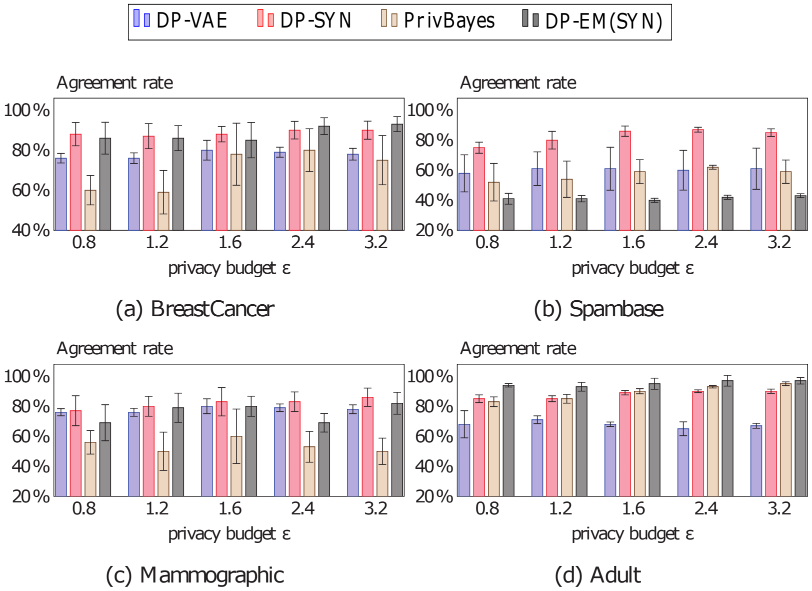
Figure 5 shows the performance of four techniques in terms of SVM agreement rate and its standard deviation which indicates the certainty and consistency in model predictions.

For the BreastCancer dataset, our approach has the highest agreement rate for privacy loss $\varepsilon \in \{0.8, 1.2, 1.6\}$. For the remaining two cases where $\varepsilon \in \{2.4, 3.2\}$, our approach outperforms DP-VAE and PrivBayes and it has slightly lower agreement rate than DP-EM(SYN). PrivBayes has the lowest agreement rate and the highest standard deviation in most cases. This is expected since PrivBayes does not have much improvement on SVM classification of BreastCancer as previously shown in Section 5.2.

For the Spambase and Mammographic datasets, our technique achieves significantly higher agreement rate than that of other state-of-art approaches. For Spambase DP-SYN has lowest standard deviation which indicates high consistency with the SVM classifier that runs on original Spambase training set. We expect such a highest agreement rate because the proposed approach outperforms other techniques in terms of SVM accuracy in Figure 3.

For the Adult dataset, the proposed method outperforms DP-VAE, PrivBayes when $\varepsilon \in \{0.8, 1.2\}$. For the remaining cases, the performance of DP-SYN is better than DP-VAE and comparable to DP-EM(SYN) and PrivBayes.

In conclusion, our approach exhibits a significant improvement in the majority of the test cases as evident from SVM agreement rate.



(a) BreastCancer

(b) Spambase

(c) Mammographic

(d) Adult

Fig. 5: SVM agreement rate of the four methods reported on the four datasets.

6 Conclusion

We propose a new generative deep learning method, DP-SYN, that produces synthetic data from a dataset while preserving the utility of the original dataset. Our generative auto-encoder method partitions the original data into groups, and then employs the private auto-encoder for each group. Auto-encoder learns the latent structure of each group, and uses expectation maximization algorithm to simulate them. This approach eliminates impurity of groups and results in more accurate representations for each latent group.

We test DP-SYN on nine datasets and compare to four state-of-the art methods in synthetic data generation. Our evaluation process uses statistical, machine learning based and agreement rate based metrics. Although not a single method outperforms others consistently in all tasks, our experiments show that DP-SYN gives robust results across all datasets, and performs better than state-of-the-art in multiple settings for both relational and image based datasets. Furthermore, DP-SYN performance does not deteriorate when the original dataset is imbalanced or high dimensional.

Acknowledgement

The research reported herein was supported in part by NIH award 1R01HG006844, NSF awards CNS-1111529, CICI- 1547324, and IIS-1633331 and ARO award W911NF-17-1- 0356.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318. ACM (2016)
2. Ács, G., Melis, L., Castelluccia, C., Cristofaro, E.D.: Differentially private mixture of generative neural networks. CoRR **abs/1709.04514** (2017), http://arxiv.org/abs/1709.04514
3. Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: 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. pp. 273–282. PODS '07, ACM, New York, NY, USA (2007). https://doi.org/10.1145/1265530.1265569, http://doi.acm.org/10.1145/1265530.1265569
4. Beimel, A., Kasiviswanathan, S.P., Nissim, K.: Bounds on the sample complexity for private learning and private data release. In: TCC. vol. 5978, pp. 437–454. Springer (2010)
5. Bindschaedler, V., Shokri, R., Gunter, C.A.: Plausible deniability for privacy-preserving data synthesis. Proceedings of the VLDB Endowment **10**(5), 481–492 (2017)
6. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: the sulq framework. In: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 128–138. ACM (2005)
7. Chaudhuri, K., Monteleoni, C.: Privacy-preserving logistic regression. In: Advances in Neural Information Processing Systems. pp. 289–296 (2009)
8. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. Journal of Machine Learning Research **12**(Mar), 1069–1109 (2011)
9. Ding, B., Winslett, M., Han, J., Li, Z.: Differentially private data cubes: optimizing noise sources and consistency. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. pp. 217–228. ACM (2011)
10. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: Eurocrypt. vol. 4004, pp. 486–503. Springer (2006)
11. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: TCC. vol. 3876, pp. 265–284. Springer (2006)
12. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science **9**(3–4), 211–407 (2014)
13. Dwork, C., Rothblum, G.N., Vadhan, S.: Boosting and differential privacy. In: Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on. pp. 51–60. IEEE (2010)
14. Goodfellow, I.: Efficient per-example gradient computations. arXiv preprint arXiv:1510.01799 (2015)
15. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their applications **13**(4), 18–28 (1998)
16. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine **29**(6), 82–97 (2012)
17. Jagannathan, G., Pillaipakkamnatt, K., Wright, R.N.: A practical differentially private random decision tree classifier. In: Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on. pp. 114–121. IEEE (2009)

18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
19. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R.: The sequence alignment/map format and samtools. Bioinformatics **25**(16), 2078–2079 (2009)
20. Li, N., Qardaji, W., Su, D., Cao, J.: Privbasis: Frequent itemset mining with differential privacy. Proceedings of the VLDB Endowment **5**(11), 1340–1351 (2012)
21. Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
22. Park, M., Foulds, J., Chaudhuri, K., Welling, M.: Dp-em: Differentially private expectation maximization. arXiv preprint arXiv:1605.06995 (2016)
23. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning. pp. 1310–1318 (2013)
24. Qardaji, W., Yang, W., Li, N.: Priview: practical differentially private release of marginal contingency tables. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 1435–1446. ACM (2014)
25. Rubin, D.B.: Discussion statistical disclosure limitation. Journal of official Statistics **9**(2), 461 (1993)
26. Rubinstein, B.I., Bartlett, P.L., Huang, L., Taft, N.: Learning in a large function space: Privacy-preserving mechanisms for svm learning. arXiv preprint arXiv:0911.5708 (2009)
27. Schork, N.J.: Personalized medicine: time for one-person trials. Nature **520**(7549), 609–611 (2015)
28. Shah, I.M.: Introduction to nonparametric estimation. Investigación Operacional **30**(3), 284–285 (2009)
29. Song, S., Chaudhuri, K., Sarwate, A.D.: Stochastic gradient descent with differentially private updates. In: Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE. pp. 245–248. IEEE (2013)
30. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05), 557–570 (2002)
31. Vaidya, J., Shafiq, B., Basu, A., Hong, Y.: Differentially private naive bayes classification. In: Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01. pp. 571–576. IEEE Computer Society (2013)
32. Vogel, P., Greiser, T., Mattfeld, D.C.: Understanding bike-sharing systems using data mining: Exploring activity patterns. Procedia-Social and Behavioral Sciences **20**, 514–523 (2011)
33. Wong, C.: Nyc taxi trips dataset. Online (2017), https://github.com/andresmh/nyctaxitrips
34. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.: Internet of things for smart cities. IEEE Internet of Things journal **1**(1), 22–32 (2014)
35. Zeng, C., Naughton, J.F., Cai, J.Y.: On differentially private frequent itemset mining. Proceedings of the VLDB Endowment **6**(1), 25–36 (2012)
36. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Privbayes: Private data release via bayesian networks. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 1423–1434. ACM (2014)
37. Zhang, J., Zhang, Z., Xiao, X., Yang, Y., Winslett, M.: Functional mechanism: regression analysis under differential privacy. Proceedings of the VLDB Endowment **5**(11), 1364–1375 (2012)