Differentially Private Data Sharing: Sharing Models versus Sharing Data

Qingrong Chen[1], Chong Xiang[2], Minhui Xue[3], Bo Li[1], Nikita Borisov[1], Dali Kaafar[4], Haojin Zhu[2]
[1]University of Illinois Urbana-Champaign, USA
[2]Shanghai Jiao Tong University, China
[3]The University of Adelaide, Australia
[4]Macquarie University and CSIRO-Data61, Australia

ABSTRACT

In this paper, we study two different approaches to enable data sharing for learning tasks while preserving data privacy. The first approach is to share representation learning models with multiple parties, for which we choose to use a **d**ifferentially **p**rivate **au**toencoder-based **g**enerative **m**odel (DP-AuGM). The second approach is to share generated data with multiple parties through generative models, for which we choose to use a **d**ifferentially **p**rivate **va**riational auto**e**ncoder-based **g**enerative **m**odel (DP-VaeGM). To achieve differential privacy, we train both models by adding differential privacy noise to the gradient. We evaluate the performance of our two proposed approaches across various differential privacy budgets. We also present the robustness of our two proposed approaches against model inversion attacks [15], membership inference attacks [32], and generative adversarial network (GAN) based attacks against collaborative deep learning [20] only in the extended version of this paper available at https://arxiv.org/pdf/1812.02274.pdf.

1 INTRODUCTION

In this paper, we study two different approaches to enable data sharing for learning tasks while preserving data privacy. We aim to protect the data privacy against the state-of-the-art attacks, namely model inversion attacks [15], membership inference attacks [32], and generative adversarial network (GAN) based attacks against collaborative deep learning [20].

The first approach, termed DP-AuGM, is to share models, where we encode the information of the data into a machine learning model for learning data representations and then share the machine learning model instead. This approach is motivated by representation learning [6], which generally aims to use machine learning models to learn a good representation of the data. Then, the models are used to convert data from its raw format into a better representation, thus helping boost the learning efficiency. An example of this representation learning is word2vec [26]. Analogous to natural language processing, in our paper, we choose to use autoencoders [34] for our representation learning model, as these models are commonly used for extracting key features of data. In order to prevent the attackers from inferring sensitive information from the representation learning model, we add differential privacy noise to the training of the representation learning model [4].

As motivation, consider a hospital not allowed to release its medical data to the public for use, but wants to share the data with universities, for example, data-driven disease diagnosis studies [19, 30]. The universities may only possess a small amount of data, such as public medical datasets [1, 3] which are not adequate for training an effective machine learning model. Under this scenario, instead of publishing the medical data directly, the hospital could locally use the medical data to train a representation learning model and then publish it. Any university interested in researching disease diagnosis independently can use the representation learning model to convert their small amounts of medical data into a better representation, boosting learning efficiency. Another motivating example is two companies that want to collaborate on a data intelligence task. A data-rich company $\mathfrak{A}$ may wish to aid a company $\mathfrak{B}$ in developing a model that helps maximize revenue, but is unwilling or legally unable to share its data with $\mathfrak{B}$ directly due to its sensitive nature. Again, the company $\mathfrak{A}$ can train a representation learning model on its large dataset and share it with the company $\mathfrak{B}$.

The second approach, termed DP-VaeGM, is to share data, where we use the shared data for training a generative model which learns the distribution of the data, and then the generative model is used to generate a new dataset for usage and the new dataset can be shared further. More specifically, we choose to use the variational autoencoder (VAE) [21] as the generative model. Similar to DP-AuGM, we train the VAE by adding differential privacy noise to the gradient [4]. The approach of using differentially private data generative models has several advantages. *First*, privacy can be preserved even if the entire trained model or the generated data is exposed to an adversary. *Second*, it can be easily integrated with other learning tasks without adding much overhead, since only the training data is a variable. *Third*, the data generation can be processed locally on the user side, which eliminates the need for a trusted server that can be attacked and compromised.

We evaluate the performance of our two proposed approaches across the differential privacy budget (cf. Section 4). Due to space limitations, we present the robustness of our two proposed approaches against model inversion attacks [15], membership inference attacks [32], and generative adversarial network (GAN) based attacks against collaborative deep learning [20] only in the extended version of this paper available at https://arxiv.org/pdf/1812.02274.pdf.

Related work. Most privacy-preserving or secure data-sharing systems use cryptographic or statistical techniques to enable sensitive data protection and sharing [9, 16, 17]. These systems are generally designed as either centralized (e.g., CryptDB [28] and Mona [24]) or decentralized [27]. Unlike previously proposed techniques, the proposed approaches achieve the following three goals: protect the privacy of training data; enable users to locally customize the privacy preference by configuring the generative models; retain the high utility for generated data. The proposed approaches achieve these goals at a lower computational cost than the aforementioned

differentially private paradigms [5, 12, 14, 31] and cryptographic techniques such as homomorphic encryption [17].

2 BACKGROUND

2.1 Differential Privacy

Definition 1 ((ϵ, δ)-differential privacy [13]). *A randomized algorithm $\mathcal{A} : \mathcal{D} \rightarrow \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$, is (ϵ, δ)-differentially private if for any two adjacent training datasets $d, d' \subseteq \mathcal{D}$, which differ by at most one training point, and any subset of outputs $S \subseteq \mathcal{R}$, it satisfies that:*

$$\Pr[\mathcal{A}(d) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(d') \in S] + \delta.$$

The parameter ϵ is often called a privacy budget: a smaller budget yields stronger privacy guarantees. The second parameter δ is a failure rate that is not tolerated by the privacy bound defined by ϵ.

2.2 Representation Learning

Representation learning aims to automatically extract the key features from the input data and a good representation of the data usually leads to the success of further classification tasks [6].

Autoencoder. An autoencoder is a widely used unsupervised learning model for representation learning in many scenarios, such as natural language processing [11] and image recognition [25]. Its goal is to learn a representation of data, typically for the purpose of dimensionality reduction [18, 33, 34]. It simultaneously trains an encoder, which transforms a high-dimenstional data point to a low-dimensional representation, and a decoder, which reconstructs a high-dimensional data point from the representation, while trying to minimize the 2-norm distance l_2 between the original and reconstructed data. Through this process, the autoencoder is able to discard irrelevant features and enhance the performance of machine learning models when facing high-dimensional input data.

2.3 Variational Autoencoder

Resembling the autoencoder, a variational autoencoder (VAE) also comprises two parts: the encoder and the decoder [21, 29] with a latent variable z sampled from a prior distribution $p(z) = p_{noise}$. Different from the autoencoder of which the encoder only tries to reduce the data into lower dimensions, the encoder inside VAE tries to encode the input data into a Gaussian probability density domain [21]. Mathematically, the encoder approximates $q(z|x)$, which is also a neural network (encoder), with input z conditioned on the data x. Then, a representation of the data will be sampled based on the output from the encoder. Finally, the decoder tries to reconstruct a data point based on sampled noise, which approximates the posterior $p(x|z)$. The two neural networks, encoder and decoder, are trained to maximize a lower bound of the log-likelihood of the data $\log p(x)$:

$$\mathbb{E}_{q(z|x)}[\log p(x|z)] - \mathrm{KL}(q(z|x)||p(z)),$$

where KL is the Kullback-Leibler divergence [10].

Sampling from the VAE is achieved by sampling from the (typically Gaussian) prior $p(z)$ and passing the samples through the decoder network.

3 DIFFERENTIALLY PRIVATE DATA SHARING

3.1 Data Sharing through Sharing Models (The Case of DP-AuGM)

We propose the first approach that shares data through sharing of the representation learning model, autoencoder, to protect privacy of the shared data while retaining high utility for machine learning usage (see overview in Figure 1).

For DP-AuGM, we first train an autoencoder with the shared data using a differentially private training algorithm. We then publish the encoder and drop the decoder. Next, users feed their raw data into this encoder to obtain better data representations which help boost their learning efficiency. Later, these data with new representations could be used to train the targeted learning systems in the future with privacy guarantees for the shared data. We adopt the deep learning with differential privacy (DP-DL) algorithm [4] to train the representation learning model autoencoder. DP-DL [4] achieves differential privacy by injecting random noise in a stochastic gradient descent (SGD) algorithm. At each step of SGD, DP-DL computes the gradient for a random subset of training points, followed by clipping, averaging out each gradient, and adding noise in order to protect privacy. The algorithm of DP-AuGM is outlined in Algorithm 1.

DP Analysis for DP-AuGM. In this paper, we adopt the training algorithm by Abadi et al. [4] to achieve differential privacy. Based on the moments accountant technique applied in [4], we obtain that the training algorithm is $(O(q\epsilon\sqrt{T}), \delta)$-differentially private, where T is the number of training steps, q is the sampling probability, and (ϵ, δ) denotes the privacy budget [4]. In addition, we will prove that any machine learning model which is trained on the data fed into DP-AuGM, is also differentially private w.r.t. the shared data and shares the same privacy bound. This also shows the benefit of sharing a representation learning model: we only need to train one representation learning model and all the machine learning models trained over the data from the representation learning model are differentially private w.r.t. the shared data.

3.2 Data Sharing through Sharing Generated Data (The Case of DP-VaeGM)

We propose the second approach that shares data via building a generative model and sharing a new dataset from the generative model (see overview in Figure 1).

As the main challenge for leveraging the generative model is to generate a new dataset, with both new training vectors and their labels. Otherwise, without their class labels, the new dataset may only be applied in unsupervised learning tasks. To address this challenge, we propose to build a multi-modal variational autoencoder motivated by Guassian Mixture Models [7]. Conceptually, each mode of VAE is used to capture the distribution of the data for each class. Thus, the entire dataset is modeled by the mixture of these modes. Traditionally, Linear Discriminant Analysis (LDA), Bayes Net, and mixture of Gaussians also utilize this type of generative models; henceforth this multi-modal model is shown to be effective for classification.

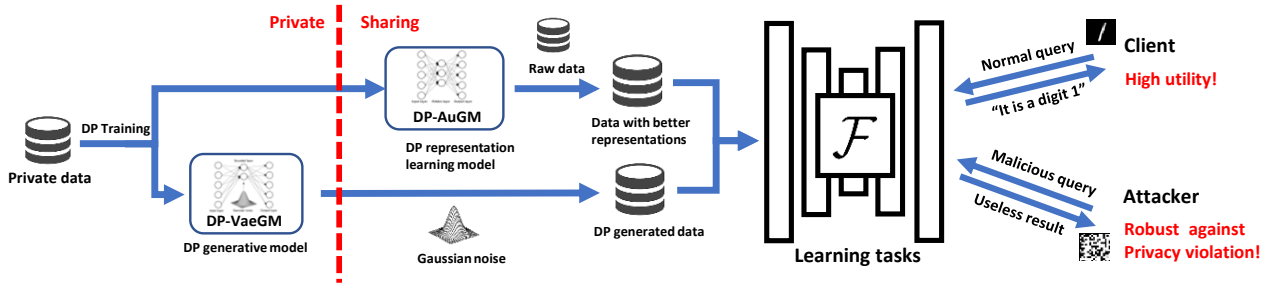


Figure 1: Overview of proposed differentially private data sharing approaches. Differentially private data sharing of private data $\mathcal{X}$ is achieved by 1) sharing a representation learning model (DP-AuGM) trained on the private data $\mathcal{X}$, and 2) by generating new surrogate data $\mathcal{X}'$ via a generative model (DP-VaeGM). After publishing $\mathcal{X}'$, different learning models can be trained on $\mathcal{X}'$ to protect privacy of $\mathcal{X}$ while achieving high learning accuracy (data utility).

Input : Private data $\mathcal{X} = \{x_1, x_2, \dots\}$, batch size B, learning rate λ_t, privacy budget (ϵ, δ), noise scale σ, gradient bound C, number of iterations T, encoder structure F_e, decoder structure F_d

Output : Differentially private sharing encoder $F_e(x; \theta_e)$ with parameter θ_e

Randomly initializes θ_e and θ_d for encoder $F_e(x; \theta_e)$ and decoder $F_d(x'; \theta_d)$, respectively. Let $\Theta = \{\theta_e, \theta_d\}$;
for $t \leftarrow 1$ **to** T **do**
  Randomly Samples a batch of data $\mathcal{X}_t$;
  foreach $x_i \in \mathcal{X}_t$ **do**
    Computes Loss $\mathcal{L}(x_i) = \|x_i - F_d(F_e(x_i))\|_2^2$;
    Computes gradient: $g(x_i) \leftarrow \nabla_\Theta \mathcal{L}(x_i)$;
    $g(x_i) \leftarrow \max(1, \frac{\|g(x_i)\|_2}{C})$;
  end
  $\Theta \leftarrow \Theta - \lambda_t \frac{1}{B}(\sum_i g(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbb{I}))$;
end
return $(F_e(x; \theta_e))$

Algorithm 1: DP-AuGM

More specifically, DP-VaeGM proceeds as below and the algorithm is outlined in Algorithm 2:

- Firstly, it initializes with n variational autoencoders (VAEs), where n is the number of the classes for the specific data. Each model $\mathcal{M}_i$ is responsible for generating the data of a specific class $1 \le i \le n$. We empirically observe that training n generative models results in higher utility than training a single model; this is because a single model would need to capture the class label latent variables following a Gaussian distribution. Using n separate models can also generate a balanced dataset even if the original data are imbalanced.
- Secondly, it uses a differentially private training algorithm (such as DP-DL) to train each generative model $\mathcal{M}_i$.
- Finally, it samples data from Gaussian distribution $\mathcal{N}(0, 1)$ for the sampling layer of each variational autoencoder. It returns the entire generated data $\mathcal{X}'$ by taking the union of generated data from each generative model $\mathcal{M}_i$.

DP Analysis for DP-VaeGM. We have adopted the algorithm developed by Abadi et al. [4] to train each VAE. Thus each training algorithm is $(O(q\epsilon\sqrt{T}), \delta)$-differentially private. Next we prove that each variational autoencoder (VAE) is a differentially private generate model (see Theorem 1) and the entire DP-VaeGM is also $(O(q\epsilon\sqrt{T}), \delta)$-differentially private (see Theorem 2). Formally, to show proofs, we let $\mathcal{X}$ be the shared data, Θ be model parameters, and $\mathcal{X}'$ be the generated data (the output of a single VAE).

Input : Number of label classes n, private data $\mathcal{X}_m$ with label m $(m \in \{1, 2, \dots, n\})$, batch size B, learning rate λ_t, privacy budget (ϵ, δ), noise scale σ, gradient bound C, number of iterations T, size of each generated dataset K

Output : Differentially private sharing data $\hat{\mathcal{X}}_m$ with label m, $m \in \{1, 2, \dots, n\}$

for $m \leftarrow 1$ **to** n **do**
  Randomly initializes the weights θ_m for variational autoencoder $F_m(x; \theta_m)$;
  for $t \leftarrow 1$ **to** T **do**
    Randomly samples a batch of data $\mathcal{X}_m^t$ from $\mathcal{X}_m$;
    foreach $x_i \in \mathcal{X}_m^t$ **do**
      Computes Loss $\mathcal{L}(x_i) =$;
      Computes gradient: $g(x_i) \leftarrow \nabla_\Theta \mathcal{L}(x_i)$;
      $g(x_i) \leftarrow \max(1, \frac{\|g(x_i)\|_2}{C})$;
    end
    $\theta_m \leftarrow \theta_m - \lambda_t \frac{1}{B}(\sum_i g(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbb{I}))$;
  end
  $\hat{\mathcal{X}}_m = \{\}$;
  for $k \leftarrow 1$ **to** K **do**
    samples z from $\mathcal{N}(0, 1)$;
    $\hat{x}_i \leftarrow F_m(z; \theta_m)$;
    $\hat{\mathcal{X}}_m \leftarrow \hat{\mathcal{X}}_m \cup \{\hat{x}_i\}$
  end
end
return $\hat{\mathcal{X}}_m$, $m \in \{1, 2, \dots, n\}$

Algorithm 2: DP-VaeGM

THEOREM 1. *Let $\mathcal{T} : \mathcal{X} \to \Theta$ be a VAE training algorithm that is (ϵ, δ)-differentially private based on [4]. Let $f : \Theta \to \mathcal{X}'$ be a mapping that maps model parameters to output, with Gaussian noise generated from a sampling layer of VAE as input. Then $f \circ \mathcal{T} : \mathcal{X} \to \mathcal{X}'$ is (ϵ, δ)-differentially private.*

PROOF. The proof is immediate by applying the post processing property of differential privacy [13]. □

THEOREM 2. *Let a generative model (VAE) of class i $\mathcal{M}_i : \mathcal{X}_i \to \mathcal{X}_i'$ be (ϵ, δ)-differentially private. Then $\mathcal{G}_n : \mathcal{X} \to \Pi_{i=1}^n \mathcal{X}_i'$ is defined to be $\mathcal{G}_n = \bigcup_{i=1}^n \mathcal{M}_i$, $\mathcal{G}_n$ is (ϵ, δ)-differentially private, for any integer n.*

PROOF. Given two adjacent datasets $\mathcal{X}_1$ and $\mathcal{X}_2 = \mathcal{X}_1 \bigcup \{b\}$, without loss of generalization, we assume b belongs to class k $(1 \le k \le n)$. Fix any subset of events $S \subseteq \Pi_{i=1}^n \mathcal{X}_i'$. Since the n generative models are pairwise independent, we obtain $\Pr[\mathcal{G}_n(\mathcal{X}_1) \in S] = \Pi_{i=1}^n \Pr[\mathcal{M}_i(x_i^1) \in S]$, where $x_i^1 \subseteq \mathcal{X}_1 = \bigcup_{i=1}^n x_i^1$ denotes

the training data of $\mathcal{X}_i$ for the ith generative model. Similarly, $\Pr[\mathcal{G}_n(\mathcal{X}_2) \in S] = \Pi_{i=1}^{n} \Pr[\mathcal{M}_i(x_i^2) \in S]$. Since $\mathcal{X}_1$ and $\mathcal{X}_2$ only differ in b, we have $x_i^1 = x_i^2$ and $\Pr[\mathcal{M}_i(x_i^1) \in S] = \Pr[\mathcal{M}_i(x_i^2) \in S]$, for any $i \neq k$. Since $\mathcal{M}_k$ is (ϵ, δ)-differentially private, then we have $\Pr[\mathcal{M}_k(x_k^1) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}_k(x_k^2) \in S] + \delta$. Therefore, we obtain $\Pr[\mathcal{G}_n(\mathcal{X}_1) \in S] = \Pi_{i=1}^{n} \Pr[\mathcal{M}_i(x_i^1) \in S] = \Pr[\mathcal{M}_1(x_1^2) \in S] \times \cdots \times \Pr[\mathcal{M}_k(x_k^1) \in S] \times \cdots \times \Pr[\mathcal{M}_n(x_n^2) \in S] \leq e^{\epsilon} \Pi_{i=1}^{n} \Pr[\mathcal{M}_i(x_i^2) \in S] + \delta = e^{\epsilon} \Pr[\mathcal{G}_n(\mathcal{X}_2) \in S] + \delta$. The inequality derives from the fact that any probability is no greater than 1. Hence, $\mathcal{G}_n$ is (ϵ, δ)-differentially private, for any n. □

4 EVALUATION

4.1 Datasets

MNIST. MNIST [22] is the benchmark dataset containing handwritten digits from 0 to 9, comprised of 60,000 training and 10,000 test examples. Each handwritten grayscale image of digits is centered in a 28×28 or 32×32 image. To be consistent with [20], we choose to use the 32×32 version of MNIST dataset when evaluating our generative models against the GAN-based attack.

Adult Census Data. The Adult Census Dataset [23] includes 48,843 records with 14 sensitive attributes, including gender, education level, marital status, and occupation. This dataset is commonly used to predict whether an individual makes over 50K dollars in a year. 32,561 records serve as a training set and 16,282 records are used for testing.

Hospital Data. This dataset is based on the Public Use Data File released by the Texas Department of State Health Services in 2010Q1 [2]. Within the data, there are personal sensitive information, such as gender, age, race, length of stay, and surgery procedure. We focus on the 10 most frequent main surgery procedures, and exploit part of categorical features to make inference for each patient. The resulting dataset has 186,976 records with 776 binary features. We randomly choose 36,000 instances as testing data and the rest serves as the training data.

Malware Data. To demonstrate the generality of the proposed models, we also include the Android mobile malware dataset [8] for diversity purposes. This dataset is previously used to determine whether an Android application is benign or malicious based on 142 binary features, such as user permission request. We randomly choose 3,240 instances as training data and 2,000 as testing data.

4.2 Evaluation of DP-AuGM

In this subsection, we show how DP-AuGM performs in terms of utility under the various differential privacy budget on four datasets. To evaluate performance, for all the four datasets, we assume 90% of the training data is used as shared data while the remaining 10% still serves as the training data. To demonstrate how DP-AuGM helps boost the learning efficiency, we compare the learning efficacy between: when only using 10% training data and when combining it with DP-AuGM for better data representations.

Effect of the Privacy Budget. To evaluate the effects of the privacy budget (i.e., ϵ and δ) on prediction accuracy for machine learning models, we vary (ϵ, δ) to test learning efficiency (i.e., the utility metric) on different datasets. The results are shown in Figures 2(a)-(d). In these figures, each curve corresponds to the best accuracy achieved given a fixed δ, as ϵ varies between 0.2 and 8. In addition, we also show the baseline accuracy (i.e., without DP-AuGM) on each dataset for the comparison. From Figure 2, we can see that the prediction accuracy decreases as the noise level increases (ϵ decreases), while we see DP-AuGM can still achieve comparable utility with the baseline even when ϵ is tight (i.e., around 1). When $\epsilon = 8$, for all the datasets, the accuracy lags behind the baseline within 3%. This demonstrates that data generated by DP-AuGM can preserve high data utility for subsequent learning tasks.

Efficacy of DP-AuGM. We further examine how DP-AuGM helps boost the learning efficacy. We compare the learning accuracy between using 10% training data and combining it with DP-AuGM for getting better data representations. For DP-AuGM, we set the private budget ϵ and δ to be 1 and 10^{-5}, respectively. We do the comparisons on all the datasets and the result is presented in Table 1. As we can see from Table 1, after using DP-AuGM, the learning accuracy increases by at least 6% on all the datasets and by 34% on Malware Data dataset. This demonstrates the significance of using DP-AuGM prior to releasing information about the shared data. DP-AuGM trained over the shared data can better capture the inner representations of the dataset, which boosts the following learning accuracy of machine learning models.

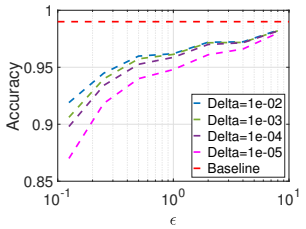
In Comparison with Scalable Private Learning with PATE. Scalable Private Learning with PATE (sPATE) [27], recently proposed by Papernot et al., can also realize a differentially private training algorithm w.r.t. the private data and provide privacy protections for partial data. We try to compare sPATE with DP-AuGM on MNIST in terms of the utility metric. Here, the baseline denotes the scenario where no privacy protection approach is used. We follow [27] to split the test data into two parts. One part serves as public data while the second serves as test data. We also use the same CNN machine-learning model as specified in [27]. As we can see from Table 2, DP-AuGM outperforms sPATE by 0.2% in terms of prediction accuracy and only sits below the baseline by 0.5%. Note that the reason of making a comparison at a specific pair of the privacy budget is that sPATE [27] only presents the result on MNIST for a specific pair of differential privacy parameters. Furthermore, DP-AuGM surpasses sPATE in terms of computational efficiency since 250 teacher models are used in sPATE while DP-AuGM only needs to be trained once.

4.3 Evaluation of DP-VaeGM

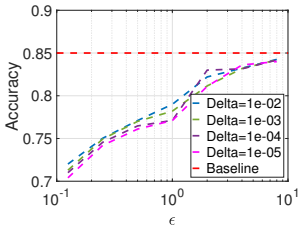
In this subsection, we empirically evaluate utility performance of our proposed data generative model DP-VaeGM. As VAE is typically used to generate high quality images, now we only evaluate DP-VaeGM on the MNIST image dataset.

Table 1: Comparisons of training accuracy between using only public data for training and using both DP-AuGM and public data

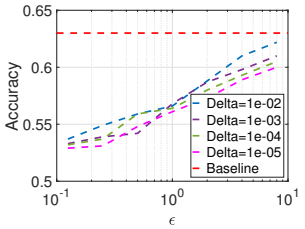
Datasets	With DP-AuGM	Without DP-AuGM
MNIST	0.95	0.89
Adult Census Data	0.78	0.64
Hospital Data	0.56	0.42
Malware Data	0.96	0.62



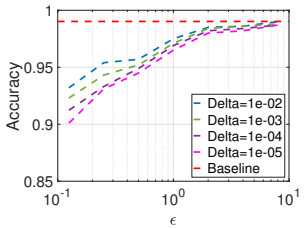
(a) Accuracy of machine learning models trained on generated data by DP-AuGM and pristine data (Baseline) under different levels of privacy on MNIST



(b) Accuracy of machine learning models trained on generated data by DP-AuGM and pristine data (Baseline) under different levels privacy on Adult Census Data



(c) Accuracy of machine learning models trained on generated data by DP-AuGM and pristine data (Baseline) under different levels privacy on Hospital Data



(d) Accuracy of machine learning models trained on generated data by DP-AuGM and pristine data (Baseline) under different levels privacy on Malware Data

Figure 2: Evaluation of DP-AuGM

Table 2: Comparisons between DP-AuGM and sPATE on MNIST

Models	Privacy budget ϵ	Privacy budget δ	Accuracy	Baseline
sPATE [27]	1.97	10^{-5}	0.985	0.992
DP-AuGM	1.97	10^{-5}	0.987	0.992

Effect of the Privacy Budget. We vary the privacy budget to test DP-VaeGM on MNIST dataset. The result is shown in Figure 3, where each curve corresponds to the best accuracy given δ, and ϵ varies between 0.2 and 8. We plot the baseline accuracy (i.e., without DP-VaeGM) using the red line. From this figure, we can see that DP-VaeGM can achieve comparable utility w.r.t. the baseline. For instance, when ϵ is greater than 1, the accuracy is always higher than 92%. When ϵ is 8 and δ is 10^{-2}, the accuracy is over 97% which is lower than the baseline by 2%. Thus, we can see that DP-VaeGM has the potential to generate data with high training utility while providing privacy guarantees for private data.

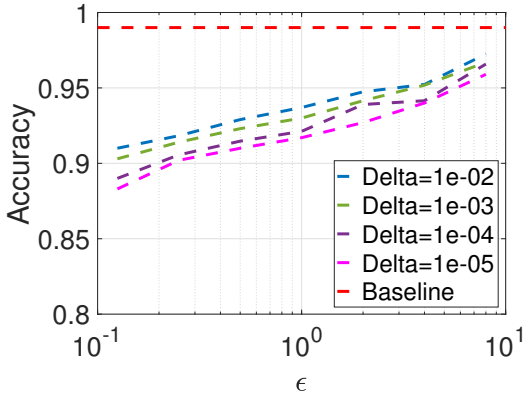


Figure 3: Accuracy of DP-VaeGM across the privacy budget on MNIST dataset

In Comparison with Scalable Private Learning with PATE. We also compare Scalable Private Learning with PATE (sPATE) [27] with DP-VaeGM on MNIST in terms of the utility metric (i.e., prediction accuracy). The learning model applies the CNN structure as specified in [27]. As sPATE requires the presence of public data, we split the test data into two parts in the same way as specified by [27]. Considering DP-VaeGM does not need public data, the private data is discarded for DP-VaeGM. In addition, the privacy budget ϵ and δ is set to be 1.97 and 10^{-5}, respectively. From Table 3, we can see that DP-VaeGM falls behind sPATE by approximately 2%. This is because that sPATE trains the model using both public and private data while DP-VaeGM is only trained with private data.

Table 3: Comparisons between DP-VaeGM and sPATE on MNIST

Models	Privacy budget ϵ	Privacy budget δ	Accuracy
sPATE [27]	1.97	10^{-5}	0.985
DP-VaeGM	1.97	10^{-5}	0.968

5 CONCLUSION

We have designed, implemented, and evaluated two approaches of differentially private data sharing via a differentially private autoencoder-based generative model (DP-AuGM) and a differentially private variational autoencoder-based generative model (DP-VaeGM), respectively. We show that both approaches can provide provable privacy guarantees and retain high data utility for machine learning tasks. We hope that our work will help pave the way toward designing more effective differentially private data sharing approaches in the dynamic digital world.

ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation of China, under Grant No. 61672350.

REFERENCES

[1] 2018. Alzheimer's Disease Neuroimaging Initiative. http://adni.loni.usc.edu
[2] 2018. Hospital Discharge Data Public Use Data File. https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm
[3] 2018. Symptom Disease sorting. https://www.kaggle.com/plarmuseau/sdsort
[4] Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 308–318.
[5] Raef Bassily and Adam Smith. 2015. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM, 127–135.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
[7] Christopher M. Bishop. 2007. *Pattern recognition and machine learning, 5th Edition*. Springer. http://www.worldcat.org/oclc/71008143
[8] Sen Chen, Minhui Xue, Zhushou Tang, Lihua Xu, and Haojin Zhu. 2016. Stormdroid: A streaminglized machine learning-based system for detecting android malware. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 377–388.
[9] Henry Corrigan-Gibbs and Dan Boneh. 2017. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*. 259–282.
[10] T Cover. 1959. *Information theory and statistics*. Wiley,. 301 pages.
[11] Li Deng, Michael L Seltzer, Dong Yu, Alex Acero, Abdel-rahman Mohamed, and Geoff Hinton. 2010. Binary coding of speech spectrograms using a deep auto-encoder. In *Eleventh Annual Conference of the International Speech Communication Association*.
[12] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 486–503.
[13] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
[14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1054–1067.
[15] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *USENIX Security Symposium*.
[16] David Froelicher, Patricia Egger, João Sá Sousa, Jean Louis Raisaro, Zhicong Huang, Christian Mouchet, Bryan Ford, and Jean-Pierre Hubaux. 2017. Unlynx: a decentralized system for privacy-conscious data sharing. *Proceedings on Privacy Enhancing Technologies* 2017, 4 (2017), 232–250.
[17] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*. 201–210.
[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.
[19] Kilian Hett, Vinh-Thong Ta, José V Manjón, and Pierrick Coupé. 2018. Graph of hippocampal subfields grading for Alzheimer's disease prediction. In *International Workshop on Machine Learning in Medical Imaging*. Springer, 259–266.
[20] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. *CCS* (2017).
[21] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *ICLR* (2014).
[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[23] M. Lichman. 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[24] Xuefeng Liu, Yuqing Zhang, Boyang Wang, and Jingbo Yan. 2013. Mona: Secure multi-owner data sharing for dynamic groups in the cloud. *IEEE transactions on parallel and distributed systems* 24, 6 (2013), 1182–1191.
[25] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning–ICANN 2011* (2011), 52–59.
[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[27] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. 2018. Scalable Private Learning with PATE. *International Conference on Learning Representations* (2018).
[28] Raluca Ada Popa, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan. 2011. CryptDB: protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. ACM, 85–100.
[29] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *ICML* (2014).
[30] Peter Schulam and Suchi Saria. 2015. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems*. 748–756.
[31] Elaine Shi, HTH Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. 2011. Privacy-preserving aggregation of time-series data. In *Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society.
[32] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.
[33] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine learning*. ACM, 1096–1103.
[34] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11 (2010).

APPENDIX

A MODEL ARCHITECTURES

Table 4: Model structures of DP-AuGM over different datasets

MNIST	Adult Census Data	Texas Hospital Stays Data	Malware Data
FC(400)+Sigmoid	FC(6)+Sigmoid	FC(400)+Sigmoid	FC(50)+Sigmoid
FC(256)+Sigmoid	FC(100)+Sigmoid	FC(776)+Sigmoid	FC(142)+Sigmoid
FC(400)+Sigmoid			
FC(784)+Sigmoid			

Table 5: Model structures of DP-VaeGM over MNIST

MNIST
FC(500)+Sigmoid
FC(500)+Sigmoid
FC(20)+Sigmoid ; FC(20)+Sigmoid
Sampling Vector(20)
FC(500)+Sigmoid
FC(500)+Sigmoid
FC(784)+Sigmoid

Table 6: Structures of machine learning models over different datasets with DP-AuGM

MNIST	Adult Census Data	Texas Hospital Stays Data	Malware Data
Conv(5x5,1,32)+Relu	FC(16)+Relu	FC(200)+Relu	FC(4)+Relu
MaxPooling(2x2,2,2)	FC(16)+Relu	FC(100)+Relu	FC(3)+Relu
Conv(5x5,32,64)+Relu	FC(2)	FC(10)	FC(2)
MaxPooling(2x2,2,2)			
Reshape(4x4x64)			
FC(10)			

Table 7: Structures of machine learning models over different datasets with DP-VaeGM

MNIST
Conv(5x5,1,32)+Relu
MaxPooling(2x2,2,2)
Conv(5x5,32,64)+Relu
MaxPooling(2x2,2,2)
Reshape(7x7x64)
FC(1024)
FC(10)