SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目： Privacy Preserving Machine Learning

学生姓名: 陈清荣

学生学号: 5140309203

专　　业: 计算机科学与技术

指导教师: 朱浩瑾

学院 (系): 电子信息与电气工程学院

PRIVACY PRESERVING MACHINE LEARNING

ABSTRACT

Deep neural networks (DNNs) have recently been widely adopted in various applications, and such success is largely due to a combination of algorithmic breakthroughs, computation resource improvements, and access to a large amount of data. However, the large-scale data collection required for deep learning also brings great privacy concerns. Prior research has shown several successful attacks in inferring sensitive training data information, such as model inversion, membership inference, and generative adversarial networks (GAN) based leakage attacks. To enable learning efficiency as well as protect data privacy, we propose differentially private generative models that can generate data with privacy guarantees and retain high data utility. In this paper, we mainly propose two such differentially private generative models: a differentially private autoencoder based generative model (DP-AuGM) and a differentially private variational autoencoder based generative model (DP-VaeGM). We provide theoretic analysis of differential privacy guarantees for the generated data. We also empirically evaluate the two models over four datasets, and demonstrate that our methods can protect privacy and maintain high data utility. We compare the proposed models with state-of-the-art private learning approaches, such as Deep Learning with Differential Privacy[1] and Scalable Private Learning with PATE[2], and show that DP-AuGM outperforms both of these methods in terms of utility. In addition, to evaluate the robustness of proposed models, we apply several strong adaptive attacks to the proposed generative models, including the model inversion attack, membership inference attack, and GAN based attack against collaborative deep learning. We show that DP-AuGM can effectively defend against all these attacks and DP-VaeGM is robust to the membership inference attack. Finally, we show that the proposed models—DP-AuGM and DP-VaeGM, can be easily integrated with existing real-world machine learning applications, such as *machine learning as a service* and *federated learning*, which are previously threatened by the membership inference attack and GAN based attack, respectively. We show that the integrated system can both protect data privacy and retain high data utility for real-world applications.

KEY WORDS: Differential Privacy, Generative Model, Autoencoder, VAE

Contents

Chapter 1 Introduction

Machine learning, especially deep neural networks (DNNs), have been applied with great success to a variety of areas, including speech processing[3], medical diagnostics[4], image processing[5], and robotics[6]. Such success largely depends on massive data collections for training machine learning models. However, these data collections often contain sensitive information and therefore raise many privacy concerns. For instance, recently it has been demonstrated that adversaries can infer users' personal identifiable information (PII) attributes (e.g., demographics, interests, behaviors, phone numbers, etc.) as well as online visits by exploiting the rough reach estimates given by Facebook[7]. More privacy violation attacks have also been proposed to show that it is possible to extract PII from different learning systems. Specifically, Fredrikson et al.[8] proposed to infer sensitive features of input data by actively probing the outputs. Later, Fredrikson et al.[9] developed a more robust model inversion attack, where attackers can recover part of the training set, such as human faces. Shokri et al.[10] proposed the membership inference attack, which tries to predict whether a data point belongs to the training set. More recently, a generative adversarial network (GAN) based attack against collaborative deep learning[11] has been proposed against distributed machine learning systems, where users collaboratively train a model by sharing gradients of their locally trained models through a parameter server. Given the fact that Google has proposed federated learning based on distributed machine learning[12] and already deployed it to mobile devices, such a GAN based attack[11] raises severe privacy threats.

Given these privacy concerns, regulations, such as the Privacy Rule of the Health Insurance Portability and Accountability Act (HIPAA) of 1996 (when disclosing medical records)[13], the Federal Rules of Civil Procedure (when disclosing court records)[14], and the European General Data Protection Regulation (GDPR)[15], have recommended the removal of identifiable information. Numerous data protection methods have also been studied in the past several decades[16]. These methods aim at various goals, such as hiding individuals in a crowd (e.g., k-anonymity[17]) or achieving differential privacy (DP) to ensure that little can be inferred about an individual even with arbitrary side information (e.g., -differential privacy[18]). Specifically, for deep learning, the leading approach for privacy to date is the *di erentially private deep learning* algorithm[19, 20].

Although DP shows great potential in practice, only making the training algorithm differentially private may not be sufficient to preserve privacy. For instance, the GAN based attack against collaborative deep learning[11] has shown that even when the training process is differentially private, it is still possible to mount an attack to extract sensitive information from original training data[11]. It is noted that such a vulnerability stems from the computational setting in collaborative deep learning rather than DP itself, since the trusted server may leak information with/without intention[11]. Therefore, we need an alternative method beyond just adopting a differentially private training algorithm to address these attacks and provide stronger privacy guarantees for diverse learning tasks.

In this paper, we propose to develop differentially private data generative models to publish synthetic data that can both protect privacy and retain high data utility. Such data generative models are trained over private/sensitive data (we will denote it as private data to be aligned with the definition in[2]), and able to generate new surrogate data for later learning tasks. These generated data should preserve similar statistical properties with the private data to retain learning efficiency. The approach of using differentially private data generative models has several advantages. *First*, with the generative models, privacy can be preserved even if the entire trained model or the generated data is accessible to an adversary. *Second*, it can be easily integrated with other learning tasks without adding much overhead, since the data only needs to be generated once. *Third*, the data generation process can be done locally on the user side, which eliminates the necessity of a trusted server (which may be attacked) for protecting the private data from users. *Fourth*, as differential privacy is used in our proposed models for protecting the data, we can provide provable privacy guarantees. *Fifth*, we can prove that any machine learning model which is trained over the generated data is also differentially private w.r.t. the private data. In addition, as demonstrated in our empirical studies, our proposed generative models can also mitigate most state-of-the-art privacy attacks[10, 11, 21]. *Finally*, the proposed models can also be easily integrated into a number of popular real-world applications, such as machine learning as a service and federated learning of which the settings are particularly at risk to the aforementioned privacy violation attacks.

We mainly introduce two differentially private generative models to generate publishable data. First, we propose a differentially private autoencoder based data generative model, which we denote as DP-AuGM. For this method, we will use the private data to train the autoencoder in a differentially private way. Then, by gathering some publicly available data (such as from some open datasets[22, 23]) and passing it through the autoencoder, we can generate new data for later learning tasks and provide privacy guarantees for the private data at the same time. Evaluation on four different datasets show that the generated data from DP-AuGM can achieve high data utility even if we use a small privacy bound (i.e., < 1). In addition, we also compare our method with *Deep Learning with Di erential Privacy* (DP-DL)[1]. The result shows that DP-AuGM significantly outperforms DP-DL for any given privacy budget (i.e., and). Furthermore, we compare our method with *Scalable Private Learning with PATE* (sPATE)[2]. Under the same setting, with the same amount of private and public data, our method outperforms sPATE by 0.2% in terms of accuracy. Second, considering if public data is not available, we further develop the differentially private variational autoencoder (VAE) based data generative model, which we denote as DP-VaeGM. Compared with the ordinary autoencoder, VAE has an extra sampling layer which can sample from a Gaussian distribution and generate new data. Thus, we do not need public data for generating new data in DP-VaeGM. As VAE is widely used for generating images, we mainly evaluate DP-VaeGM on the image dataset and the result shows that DP-VaeGM can also maintain high data utility and preserve data privacy at the same time. Under the setting of $= 8$ and $= 10^{3}$, the prediction accuracy of DP-VaeGM is over 97% on MNIST.

To demonstrate the robustness of our proposed models, we evaluate both DP-AuGM and DP-VaeGM on three existing attacks—model inversion attack[8, 9], membership inference attack[10], and GAN based attack against collaborative deep learning[11]. The results show that DP-AuGM can effectively mitigate all of the aforementioned attacks and DP-VaeGM is robust against the membership inference attack.

In addition, we have integrated our proposed generative models with two real-world applications, which are threatened by the aforementioned attacks. The first application is *machine learning as a service*. Traditionally, users need to upload all of their data to the platforms (such as Amazon Machine Learning[24]) to train a model, due to the lack of computational resources on the user side. However, if these platforms are compromised, all of the users' data will be leaked. Thus, we propose to integrate DP-AuGM and DP-VaeGM with this application, so that even if the platforms are compromised, the privacy of users' data can still be protected. We empirically show that after being integrated with DP-AuGM and DP-VaeGM, this application still maintains high utility. The second application is *federated learning*[12], which is recently threatened by the GAN based attack[11]. As DP-AuGM is more effective in defending against this attack, we try to combine DP-AuGM with this application. We successfully show that DP-AuGM can be integrated with federated learning with ease. Even under small privacy budgets ($= 1$, $= 10^{5}$), it only decreases original utility within 5%.

In summary, we make the following contributions:

- We propose two differentially private data generative models DP-AuGM and DP-VaeGM, which can provide differential privacy guarantees for the generated data, and retain high data utility for various machine learning tasks. In addition, we compare the learning efficiency of the generated data with state-of-the-art private training methods. We show that the utility of DP-AuGM outperforms sPATE[2] and DP-DL[1] under any given privacy budget. We also show that DP-VaeGM can achieve comparable learning efficiency with DP-DL.
- We empirically evaluate and demonstrate that the proposed model DP-AuGM is robust against existing privacy attacks—model inversion attack, membership inference attack, and GAN based attack against collaborative deep learning; and DP-VaeGM is robust to the membership inference attack.
- We integrate the proposed generative models with *machine learning as a service* and *federated learning* to protect data privacy. We show that such integration is very convenient, and can retain high utility for these real-world applications, which are currently threatened by privacy attacks.

To the best of our knowledge, we are the first to propose and systematically examine differentially private data generative models which can defend against the contemporary privacy violation attacks.

Chapter 2 Background

In this chapter, we will introduce preliminary knowledge about deep learning, privacy violations, differential privacy, and data generative models.

2.1 Deep learning

Deep learning is the process of learning nonlinear features and functions from complex data. Surveys of deep-learning architectures, algorithms, and applications can be found in[25, 26]. Deep learning has been shown to outperform traditional techniques for speech recognition[27–29], image recognition[30, 31], and face detection[32]. A deep-learning architecture based on a new type of rectifier activation functions is claimed to outperform humans when recognizing images from the ImageNet dataset[33].

Deep learning has shown promise for analyzing complex biomed- ical data related to cancer[34–36] and genetics[37]. The training data used to build these models is especially sensitive from the privacy perspective, underscoring the need for privacy-preserving deep learning methods.

Our work is inspired by recent advances in parallelizing deep learning, in particular parallelizing stochastic gradient descent on GPU/CPU clusters[38], as well as other techniques for distribut- ing computation during neural-network training[39–41]. These techniques, however, are not concerned with privacy of the training data and all assume that a single entity controls the training.

2.2 Privacy Violation in Learning Systems

2.2.1 Model Inversion Attack

This attack was first introduced by Fredrikson et al.[8] and further developed in[9]. The goal of this attack is to recover sensitive attributes within original training data. For example, an attacker can infer the genome type of patients from medical records data or recover distinguishable photos by attacking a facial recognition API. Such a vulnerability mainly lays in the rich information remembered by the machine learning models, which can be leveraged by the attacker to recover original training data by constructing data records with high confidence. In this paper, we mainly focus on a strong adversarial scenario where attackers have white-box access to the model so as to evaluate the robustness of proposed privacy-preserving mechanisms. Within the attack, an attacker aims to reconstruct images used in training phase by minimizing the difference between hypothesized and obtained confidence vectors from the machine learning models.

2.2.2 Membership Inference Attack

Shokri et al.[10] proposed the membership inference attack to determine whether a specific data record is within the training set. This attack also takes advantage of rich information recorded in machine learning models. An attacker first generates data with similar distribution as the original data by querying machine learning models and then uses the generated data to train local models (termed as shadow models in[10]) to mimic the behavior of original models. Finally, the attacker can apply the data provided by the local models to training a classifier and determine whether a given record belongs to the original training dataset.

2.2.3 GAN based Attack against Collaborative Deep Learning

A GAN based attack targeting at privacy-preserving collaborative deep learning[42] has been proposed[11]. An attacker can make use of GANs to generate instances which will approximate data from the other parties. The adversarial generator will be improved based on the information returned from the trusted center, and eventually achieves high attack success rate in the collaborative scenario even when differential privacy is guaranteed for each party.

2.3 Differential Privacy

Differential privacy provides strong privacy guarantees for data privacy analysis[43]. It ensures that attackers cannot infer sensitive information about input datasets merely based on the algorithm outputs. The formal definition is as follows.

Definition 2.1. A randomized algorithm A : D ! R with domain D and range R, is (;)-differentially private if for any two adjacent training datasets $d; d^0$ D, which differ by at most one training point, and any subset of outputs S R, it satisfies that:

$$\text{Pr»A}¹d° \; 2 \; S¼ \quad \text{e} \;\; \text{Pr»A}¹d^{0}° \; 2 \; S¼ + \;\; :$$

Parameter is a privacy budget: smaller budgets yield stronger privacy guarantees. The second parameter is a failure rate for which it is tolerated that the privacy bound defined by does not hold.

Differential privacy has several properties that make it particularly useful in applications such as ours: composability, group privacy, and robustness to auxiliary information. Composability enables modular design of mechanisms: if all the components of a mechanism are differentially private, then so is their composition. Group privacy implies graceful degradation of privacy guarantees if datasets contain correlated inputs, such as the ones contributed by the same individual. Robustness to auxiliary information means that privacy guarantees are not affected by any side information available to the adversary.

A common paradigm for approximating a deterministic real-valued function f : D ! Rwith a differentially private mechanism is via additive noise calibrated to f^0s sensitivity S_f , which is defined as the maximum of the absolute distance kf¹d° f¹d^0°k where d and d^0 are adjacent inputs. (The restriction to a real-valued function is intended to simplify this review, but is not essential.) For instance, the Gaussian noise mechanism is defined by:

$$\text{M}¹d° \;\; , \;\; f¹d° + \text{N}¹0; S_f^2 \quad ^2°$$

where N¹0; S_f^2 ²° is the normal (Gaussian) distribution with mean 0 and standard deviation S_f . A single application of the Gaussian mechanism to function f of sensitivity S_f satisfies ¹ ; °-differential privacy if

$\frac{4}{5}exp$¹ ¹ º²•2º and < 1[43]. Note that this analysis of the mechanism can be applied post hoc, and, in particular, that there are infinitely many ¹ ; º pairs that satisfy this condition.

Differential privacy for repeated applications of additive- noise mechanisms follows from the basic composition theorem[44, 45], or from advanced composition theorems and their refinements[46, 47]. The task of keeping track of the accumulated privacy loss in the course of execution of a composite mechanism, and enforcing the applicable privacy policy, can be performed by the privacy accountant, introduced by McSherry[48].

2.3.1 Deep Learning with Differential Privacy (DP-DL)

DP-DL[1] achieves DP by injecting random noise in 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. DP-DL provides a differentially private training algorithm with tight DP guarantees based on moments accountant analysis[1].

2.3.2 Scalable Private Learning with PATE (sPATE)

To protect the data privacy during learning, private aggregation of teacher ensembles (PATE)[20] is first introduced by training an ensemble of teacher models on the private data. Then these teacher models aggregate their answers on public data to teach student models in a differentially private way. Recently, sPATE improves PATE in terms of scalability by introducing new noisy aggregation mechanism for teacher ensembles, which can provide tighter privacy guarantees[2]. As sPATE is an improved scalable version of PATE, we will focus on comparing our methods with sPATE in the evaluation section.

2.4 Data Generative Models

2.4.1 Autoencoder

An autoencoder is a widely used unsupervised learning model whose goal is to learn a representation of data, typically for the purpose of dimensionality reduction[49–51]. It tries to find the optimal parameters that minimize the norm distance between original and reconstructed data. Through this process, the autoencoder is able to discard those irrelevant features and enhance the performance of machine learning models when facing high-dimension input data. More specifically, an autoencoder comprises two parts. The first part is the encoder which transforms the data from high dimensions into low dimensions. The second part is the decoder which recovers the data from encoded dimension back to the original dimension.

2.4.2 Variational Autoencoder (VAE)

Resembling the autoencoder, an variational autoencoder also comprises two parts: encoder and decoder[52, 53]. 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[52]. Then, a noisy representation of the data will be sampled based on this distribution. Finally, the decoder tries to reconstruct a data point based on sampled noise. To achieve this goal, the loss function of VAE usually comprises two terms. The first term is the reconstruction loss and the second term is the KL-divergence[54] between the output of the encoder and the Gaussian distribution which penalizes the loss when the output of the encoder diverges from the Gaussian distribution.

Chapter 3 Differentially Private Data Generative Models

3.1 Problem Statement

Let $\mathcal{X}$ be the set of training data containing sensitive information, and we will denote it as private data similarly with[20]. We denote $\mathcal{M}$ as a data generative model which is trained on the private data, and is able to generate new data $\mathcal{X}'$ for later training usage, as shown in Figure 3–1. To protect privacy of the private data, the goal of the generative model is to prevent an attacker from recovering $\mathcal{X}$, or inferring sensitive information from $\mathcal{X}$ based on $\mathcal{X}'$. Formally, we give the definition of the differentially private generative model as below.

Definition 3.1. A generative model $\mathcal{M} : \mathcal{D} \to \mathcal{Z}$ with domain $\mathcal{D}$ and range $\mathcal{Z}$, is (ϵ, δ)-differntially private, if for any adjacent private datasets $\mathcal{X}, \hat{\mathcal{X}} \subseteq \mathcal{D}$ which only differ in one entry, and any subset of output space $S \subseteq \mathcal{Z}$, it satisfies that:

$$\Pr[\mathcal{M}(\mathcal{X}) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(\hat{\mathcal{X}}) \in S] + \delta.$$

The goal of the proposed differentially private generative model is to generate data with high utility while protecting sensitive information within the data. Current research shows that even algorithms that have been proved to be differential privacy can also leak private information in the face of certain carefully crafted attacks on different levels. Therefore, in this paper, we will also analyze several existing attacks to show that the proposed differentially private generative models can also defend against the state-of-the-art attacks.

3.2 Approach Overview

To protect private data privacy, we propose to use the private data to train a differentially private generative model and use this generative model to generate new synthetic data for further learning tasks, which can both protect privacy of original data and retain high data utility. As the newly generated data is differentially private w.r.t. the private data, it will be hard for attackers to recover or synthesize the private data, or infer other information about the private data in learning tasks. Specifically, we choose an autoencoder and a variational autoencoder (VAE) as our two generative models. The overview of our proposed differentially private data generative models is shown in Figure 3–1. First, the private data is used to train the generative model with differential privacy, which is either an autoencoder (DP-AuGM) or a variational autoencoder (DP-VaeGM) based model. Then the generated data from the trained differentially private generative model is published and sent to targeted learning tasks. It should be noted that for DP-AuGM, some public data is needed for generating new data while for DP-VaeGM, only sampling from a Gaussian distribution is needed. The goal of our design is that the learning accuracy on the generated data is high for ordinary users (high data utility), while the attackers cannot obtain sensitive information from the private data.

3.3 Privacy and Utility Metrics

Here we will briefly introduce the privacy and data utility metrics used throughout the paper.

3.3.1 Privacy Metric

First, for our differentially private generative models, we theoretically prove differential privacy for the generated data. We refer to the privacy budget (,) as the privacy metric during evaluations. In addition, we evaluate how robust the proposed generative models are against three state-of-the-art attacks—model inversion attack[21], membership inference attack[10], and GAN based attack against collaborative deep learning[11]. Specifically, to quantitatively evaluate how our models deal with the membership inference attack, we propose a new term—privacy loss.

3.3.1.1 **Privacy Loss (PL)**

Within membership inference attack, we measure the privacy loss as the inference precision increment over random guessing baseline (e.g., 0.5), where the adversary's attack precision rate P is defined as the fraction of records that are correctly inferred as members of the training set among all the positive predictions. We define privacy loss PL as follows:

$$PL = \begin{cases} \frac{P-0.5}{0.5}, & \text{if } P > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

3.3.2 Utility Metric

We use the prediction accuracy to measure utility for different models. Considering the goal of machine learning is to build an effective prediction model, it is natural to evaluate how our proposed model performs in terms of prediction accuracy. To be specific, we will evaluate the prediction model which is trained on the generated data from the differentally private generative model.

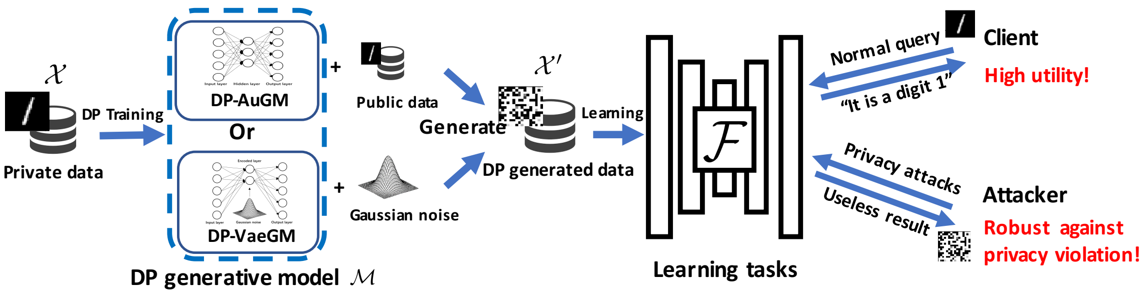


Figure 3–1 Overview of proposed differentially private data generative models. Sensitive private training data X is fed into the generative model M to generate private surrogate dataset X^{0}. After publishing X^{0}, different learning models can be trained on X^{0} to protect privacy of X while achieving high learning accuracy (data utility).

3.4 DP Autoencoder based Generative Model (DP-AuGM)

Autoencoders have been widely applied in multiple real-world applications to capture the underlying representations of data[55]. Here we introduce how to apply the differentially private autoencoder based generative model (DP-AuGM) for protecting privacy of the private data while retaining high utility for the generated data.

An autoencoder consists of two parts, the encoder and decoder. The encoder compresses the input data while the decoder recovers the data from its compressed form. The training goal of an autoencoder is to minimize the mean square error (i.e., L^2) between the output and input of the autoencoder. In this way, an autoencoder can learn an inner representation of the input data in its encoded layer. This model is well studied and deployed in many scenarios, such as natural language processing[56] and image recognition[57].

For DP-AuGM, we first train an autoencoder with our private data using a differentially private training algorithm. Then, we publish the encoder and drop the decoder. New data will be generated (encoded) by feeding the public data into the encoder. These newly generated data can be used to train the targeted learning systems in the future with privacy guarantees. In this way, statistical information of the private data can be preserved in the generative model and help to equip the public data with similar properties. Thus, as we will show in the later experiments, even if only a small amount of public data is available, the learning accuracy of the machine learning model can be very high. During inference time, the encoder will also be used to encode the test data for model predictions. Considering the encoder is trained with a differentially private algorithm, it does not compromise the privacy when publishing the encoder.

The DP-AuGM proceeds as below:

First, trains DP-AuGM with private data using a differentially private algorithm.

Second, generates new differentially private data by feeding the public data to the encoder.

Third, uses the generated data to train any machine learning model.

3.4.1 DP Analysis for DP-AuGM

In this paper, we adopt the training algorithm developed by Abadi et al.[1] to achieve differential privacy. Based on the moments accountant technique applied in[1], we obtain that the training algorithm is $(O(q\sqrt{T}),)$-differentially private. Here T is the number of training steps, q is the sampling probability, and $(,)$ denotes the privacy budget[1]. Further, by applying the post-processing property of differential privacy[43], we can guarantee that the generated data is also differentially private w.r.t. the private data and shares the same privacy bound with the training algorithm. In addition, we will also prove that any machine learning model which is trained on the generated data from DP-AuGM, is also differentially private w.r.t. the private data and shares the same privacy bound. This also shows the benefit of training a differentially private generative model: we only need to train one DP generative model and all the machine learning models which are trained over the generated data will be differentially private w.r.t. the private data. Let $\mathcal{M}$ denote the differentially private generative model and $\mathcal{X}$ be the private data. Any machine learning model trained over the generated data $\mathcal{M}(\mathcal{X})$, is also differentially private w.r.t. the private data $\mathcal{X}$.

Proof. We denote the machine learning model trained on X as $f(X)$, and the learning model trained over the generated data as $f(M(X))$. By directly applying the post-processing property of differential privacy[43], it is shown that the learning model is also differentially private w.r.t. the private data and shares the same privacy bound with the differentially private generative model.

3.5 DP Variational Autoencoder based Generative Model (DP-VaeGM)

For DP-AuGM, we have to note that to generate data for later training tasks, public data is needed. So, a natural question is—what if public data is not available? To address this problem, we further develope DP-VaeGM, which can also achieve a differentially private generative model while not requiring the existence of public data.

The entire DP-VaeGM proceeds as below:

First, initializes with n variational autoencoders (VAE), where n is the number of the classes for the specific data. Each model M_i is responsible for generating the data of a specific class i.

Second, uses a differentially private training algorithm (such as DP-DL) to train each generative model M_i. Note we empirically observe if we train n generative models, the data utility will be higher than training a single model for all the data. This can also help to solve data imbalance problem.

Third, samples Gaussian noise $z \sim N(0, 1)$ for the sampling layer of each variational autoencoder. Returns the entire generated data X' by taking the union of generated data from each generative model M_i.

We prove in Theorem 3.5.1 that each generative model is differentially private w.r.t. the private data, which maintains the same privacy bound as the differentially private training algorithm. We prove in Theorem 3.5.1 that the entire DP-VaeGM is differentially private w.r.t. the private data and shares the same privacy bound.

3.5.1 DP Analysis for DP-VaeGM

We have adopted the algrorithm developed by Abadi et al.[1] to train each VAE. Thus each training algorithm is $(O(q\sqrt{T}), \delta)$-differentially private. Next we prove that each variational autoencoder (VAE) is a differentially private generate model and the entire DP-VaeGM is also $(O(q\sqrt{T}), \delta)$-differentially private. Formally, to introduce notations, we let X be the private data, θ be model parameters, and X' be the generated data (the output of a single VAE).

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

Proof. This theorem directly follows from the post processing property[43] of differential privacy.

Let a generative model (VAE) of class i $M_i : X_i \to X'_i$ be (ϵ, δ)-differentially private. Then if $G_n : X \to \bigcup_{i=1}^{n} X'_i$ is defined to be $G_n = \bigcup_{i=1}^{n} M_i$, G_n is (ϵ, δ)-differentially private, for any integer n. See proof in Appendix A.

3.6 Conclusion of Two Methods

Both DP-VaeGM and DP-AuGM can realize a differentially private generative model w.r.t. the private data. The main difference is that DP-AuGM needs public data while DP-VaeGM does not. Besides, for DP-AuGM, we use the output of the encoder as the generated data, while for DP-VaeGM we just use the output of the VAE as the generated data. As we will show in the following evaluations, both methods can retain high data utility for the generated data and defending against existing privacy attacks.

Chapter 4 Experimental Evaluation

In this section, we will first describe datasets that we use for evaluation, followed by the empirical results for the proposed data generative models, DP-AuGM and DP-VaeGM. We then take a deep dive into how robust our differentially private generative models are against three existing privacy attacks—model inversion attack, membership inference attack, and GAN based attack against collaborative deep learning. All the generative model and machine learning model structures involved in the experiments will be illustrated in Appendix B.

4.1 Datasets

4.1.1 MNIST

MNIST[58] 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 image. To be consistent with[11], we choose to use the 32 32 version of MNIST dataset when evaluating our generative models against the GAN based attack.

4.1.2 Adult Census Data

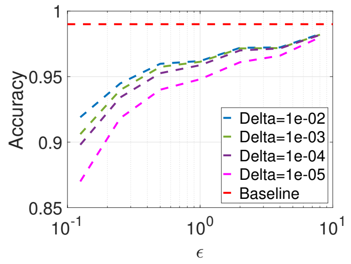
The Adult Census Dataset[59] includes 48,843 records with 14 sensitive attributes, including gender, education level, marital status, and occupation. This dataset is commonly used to predict whether an individual makes over 50K dollars in a year. 32,561 records serve as a training set and 16,282 records are used for testing.

4.1.3 Hospital Data

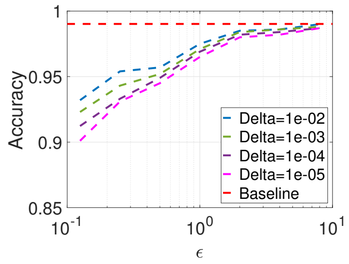
This dataset is based on the Public Use Data File released by the Texas Department of State Health Services in 2010Q1[60]. Within the data, there are personal sensitive information, such as gender, age, race, length of stay, and surgery procedure. We use part of categorical attributes to infer the main procedures of patients. 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.

4.1.4 Malware Data

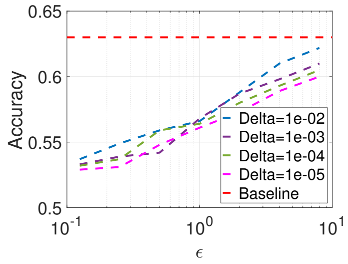
To demonstrate the generality of the proposed models, we also include the Android mobile malware dataset[61] for diversity purposes. This dataset is previously used to determine whether an Android application is benign or malicious based on 142 binary features, such as user permission request. We randomly choose 3,240 instances as training data and 2,000 as testing data.



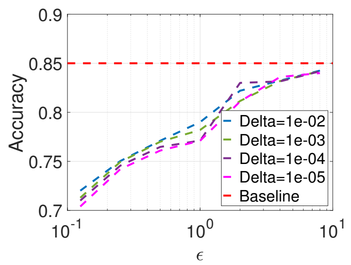
(a) Accuracy of machine learning models trained on generated data by DP-AuGM and pristine data (Baseline) under different levels privacy on MNIST



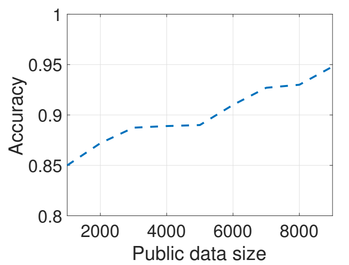
(b) Accuracy of machine learning models trained on generated data by DP-AuGM and pristine data (Baseline) under different levels privacy on Malware Data



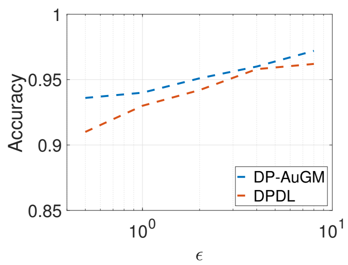
(c) Accuracy of machine learning models trained on generated data by DP-AuGM and pristine data (Baseline) under different levels privacy on Hospital Data



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



(e) Prediction accuracy on MNIST dataset based on different sizes of public data



(f) Comparison between DP-DL and DP-AuGM on MNIST with $= 10^{5}$

Figure 4–1 Evaluation of DP-AuGM

4.2 Evaluation of DP-AuGM

In this section, we first show how DP-AuGM performs in terms of utility under different privacy budgets on four datasets. To evaluate performance, for each dataset, we split the test data into two parts: one serves as public data while the rest serves as test data. For MNIST, we split the test data into two parts: 90% is used as public data and the rest 10% is used as a hold out to evaluate test performance, in the same way as sPATE[2] does. For Hospital Data, Malware Data and Adult Census Data, the test data is evenly split into two halves: the first serves as public data and the second is used for evaluating test performance. All the training data is regarded as private data of which the privacy we want to protect. Then we analyze how public data size influences DP-AuGM in detail on MNIST dataset. Finally, we compare our method with some state-of-the-art differentially private algorithms developed for deep learning, such as DP-DL[1] and sPATE[2]. For the differentially private training algorithm required by DP-AuGM, we choose to use DP-DL[1].

4.2.1 Effect of Different Privacy Budgets

To evaluate the effects of privacy budgets (i.e., and) on prediction accuracy for machine learning models, we vary (,) to test learning efficiency (i.e., the utility metric) on different datasets. The results are shown in Figure 4–1(a)-(d). In these figures, each curve corresponds to the best accuracy achieved for a fixed , as varies between 0:2 and 8. In addition, we also show the baseline accuracy (i.e., without DP-AuGM) on each dataset for comparison. From Figure 4–1, we can see that the prediction accuracy decreases as the noise level increases (decreases), while we see DP-AuGM can still achieve comparable utility with the baseline even when is tight (i.e., around 1). When = 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 further learning tasks.

4.2.2 Effect of Public Data Size

We further examine how utility is affected when we vary the size of the public data on the dataset MNIST. The public data size varies from 1,000 to 9,000 by a step of 1,000. The privacy budget and is set as 1 and 10 ^{5}, respectively. The result is shown in Figure 4–1(e). As we can see, even if the public data size drops nearly 90%, the influence over accuracy is still limited within 10%. This demonstrates that public data size does not have a big impact on the final result. This also shows although the private data is only used to train the differentially private generative model, the generated data still contains enough statistical information from the dataset. Considering the baseline accuracy (without using DP-AuGM) 99% is achieved when 50,000 data samples are used to train the machine learning model, our method shows a great potential to protect privacy of the private data while achieving high data utility.

4.2.3 Comparison with the Differentially Private Training Algorithm (DP-DL)

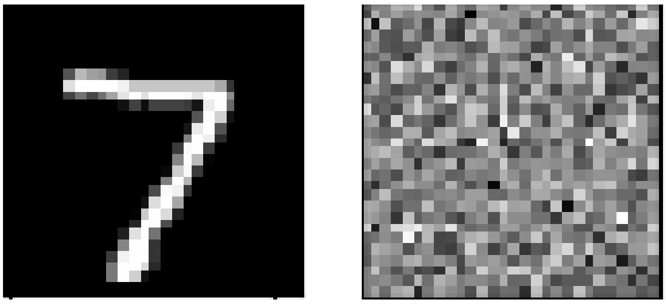
Although our method leverages DP-DL as the differentially private training algorithm, we can show that our method performs better on training the machine learning model under the same privacy budget. For comparison, we choose the feed-forward neural network model of which the architecture is specified in[1] on MNIST dataset. In addition, we use 90% of the test data as public data and the rest acts as the test data for both methods. For DP-DL, the public data simply serves as its training data. As for the privacy budget, we fix as 10 ^{5} and vary from 0:5 to 8. The result is shown in Figure 4–1f. As we can see, under different , our method outperforms DP-DL consistently.

4.2.4 Comparison with Scalable Private Learning with PATE

Scalable Private Learning with PATE (sPATE)[2] is recently proposed by Papernot et al., which can also realize a differentially private training algorithm w.r.t. the private data. We have compared this method with our proposed DP-AuGM on MNIST over the utility metric (i.e., prediction accuracy). The machine learning model uses the CNN model as specified in[2]. We use the same way as[2] does to split the test data into two parts. One part serves as public data while the second serves as test data. The result is shown in Table 4–1. As we can see, the proposed method has outperformed sPATE by 0.2% in terms of prediction accuracy.

Table 4–1 Comparisons between DP-AuGM and sPATE on MNIST

Methods	Privacy budget	Privacy budget	Accuracy
sPATE[2]	1.97	10 ^{5}	0.985
DP-AuGM	1.97	10 ^{5}	0.987



(a) (b)

Figure 4–2 Visualization of (a) Private data (b) Decoded data on dataset MNIST

4.2.5 DP-AuGM against Decoder Exposure Attack

As DP-AuGM publishes only the encoder part to be the generative model, here we will evaluate what if an adversary gets access to the whole model (both encoder and decoder parts). Here we set the differential privacy budget as = 10 ^{5} and = 1. We show the original private data and decoded data on MNIST based on the white-box access in Figure 4–2. It is shown that the decoded data does not tell useful information about the original private data. Thus, even if an adversary has access to the entire autoencoder model, it will still be hard for the attacker to recover sensitive information. This shows DP-AuGM cannot only protect privacy of the private data, but also is robust against such decoder exposure attack.

4.3 Evaluation of DP-VaeGM

In this subsection, we empirically evaluate how our proposed data generative model DP-VaeGM performs in terms of utility. For this method, we do not need the availability of public data, so all the training data will be regarded as private data and all the test data will be used for testing. As DP-VaeGM is usually used to generate high quality images, currently we will evaluate this method on the image dataset, MNIST.

4.3.1 Effect of Different Privacy Budgets

We vary the privacy budget to test DP-VaeGM on MINST dataset. The result is shown in Figure 4–3, where each curve corresponds to the best accuracy given fixed , and varies between 0:2 and 8. We show 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 with the baseline. For instance, when is greater than 1, the accuracy is always higher than 92%. When is 8 and 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.

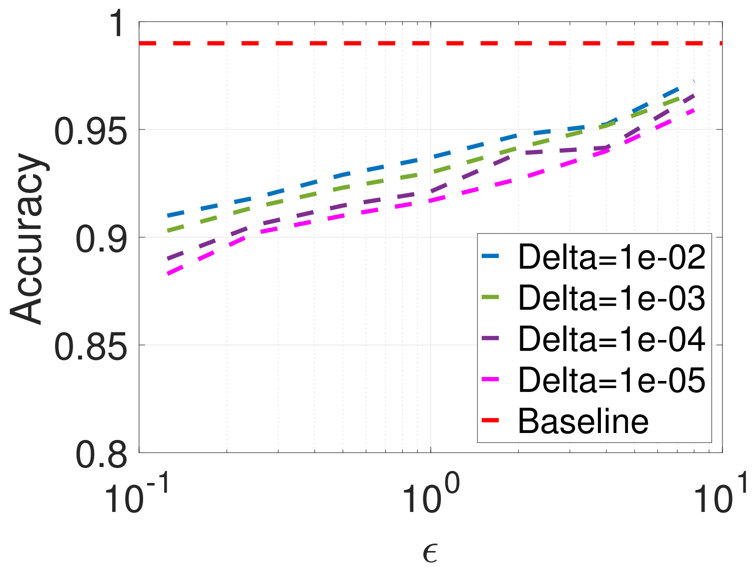


Figure 4–3 Accuracy of DP-VaeGM under various privacy budgets on MNIST dataset

4.3.2 Quality of Generated Data Samples

As VAEs are good at generating high quality images from noise, we want to show that even after imposing differetial privacy, this property still holds. Part of the results on dataset MNISTare shown in Figure 4–4. From the result, we can see that for each class of MNIST, the image is generated with high quality.

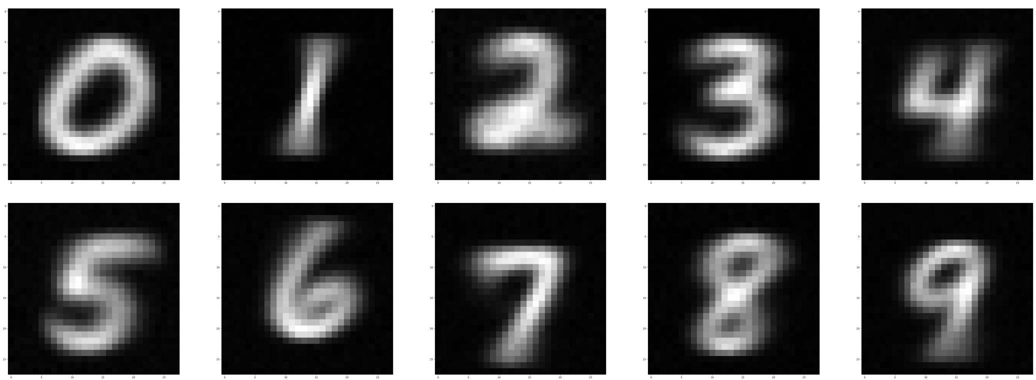


Figure 4–4 Quality of generated data from DP-VaeGMon MNIST

4.3.3 Comparison with the Differentially Private Training Algorithm (DP-DL)

We have also compared DP-VaeGM with DP-DL on MNIST. As for the privacy budget, we fix as 10^{5} and vary from 0:5 to 8. The result is shown in Figure 4–5. From Figure 4–5, we can see that DP-VaeGM achieves comparable utility with DP-DL. Moreover, we want to emphasize that for DP-VaeGM, we only need to generate data once and any model trained over the generated data from DP-VaeGM will always be differentially private w.r.t the private data. However, for DP-DL, we need to perform the training algorithm on every new model. From this perspective, we can see that DP-VaeGM saves a lot of overheads in comparison to DP-DL.

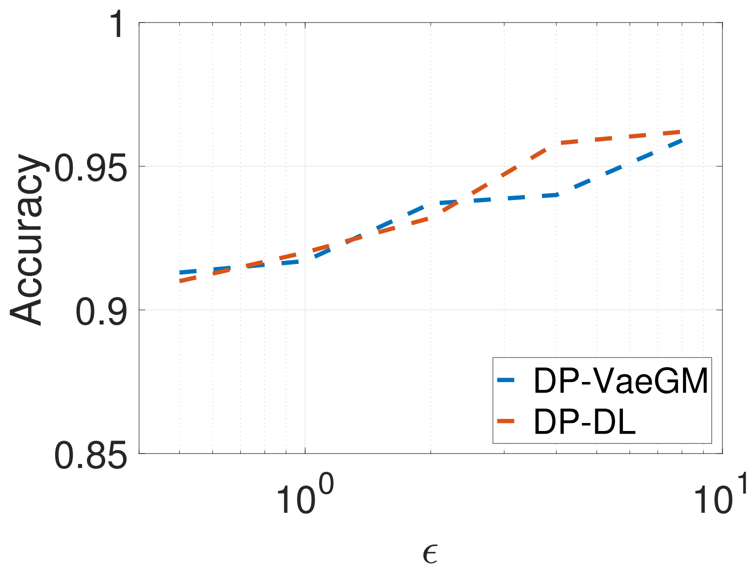


Figure 4–5 Comparison between DP-VaeGM and DP-DL on MNIST with = 10^{5}

4.3.4 Comparison with Scalable Private Learning with PATE

We also compare Scalable Private Learning with PATE (sPATE)[2] with DP-VaeGM on MNIST over the utility metric (i.e., prediction accuracy). The learning model applies the CNN structure as specified in[2]. As sPATE requires the presence of public data, we split the test data into two parts, in the same way specified by[2]. Considering DP-VaeGM does not need public data, this part of data is used for training DP-VaeGM. In addition, the privacy budget and is set to be 1:97 and 10^{5}, respectively. The result is shown in Table 4–2. From the result, we can see that DP-VaeGM falls behind sPATE by approximately 2%, but we have to notice that sPATE only works when public data is available. Instead, DP-VaeGM can be applied regardless of the availability of public data. So DP-VaeGM performs as a competitive option for realizing differentially private algorithms in machine learning systems.

Table 4–2 Comparisons between DP-VaeGM and sPATE on MNIST

Methods	Privacy budget	Privacy budget	Accuracy
sPATE[2]	1.97	10^{5}	0.985
DP-VaeGM	1.97	10^{5}	0.968

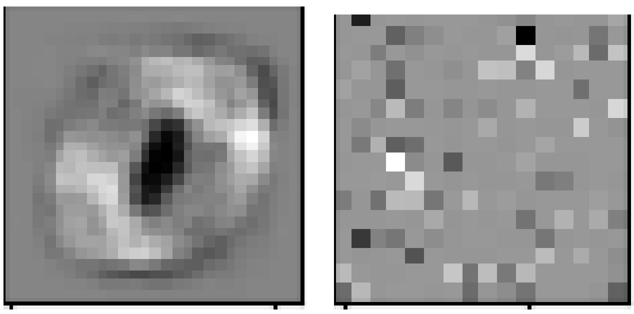
In summary, we have empirically shown that DP-AuGM and DP-VaeGM can achieve high data utility and protect privacy of private data at the same time. Although DP-AuGM performs better in terms of the utility metric compared with DP-VaeGM, DP-VaeGM still gives a good option when all the data requires protection (i.e., no public data).

Chapter 5 Defending against Existing Attacks

To demonstrate the robustness of proposed generative models, here we evaluate these models against three state-of-the-art privacy violation attacks—model inversion attack, membership inference attack, and the GAN based attack against collaborative deep learning.

5.1 Model Inversion Attack

We choose to use the one-layer neural network to mount the model inversion attack[9] over MNIST dataset setting because the simplicity of the network and data structure would increase the attack success rate, considering[11] has claimed that the model inversion attack might not work on deep neural networks. For the original attack, we use all the training data to train the one-layer neural network and then try to recover digit 0 by exploiting the confidence values[9]. The result is presented in Figure 5–1a. As we can see from Figure 5–1a, the digit 0 is almost recovered. Then, we try to evaluate how DP-AuGM performs in defending against the attack. We use the generated data from DP-AuGM to train the one-layer neural network. The privacy budget and for DP-AuGM is set to be 1 and 10 ^{5}, respectively. We then mount the same model inversion attack on the one-layer neural network. Figure 5–1b shows the result after deploying DP-AuGM. We can clearly see that after deploying DP-AuGM, nothing can be learned from the attack result as shown in Figure 5–1b. So we can see DP-AuGM can mitigate the model inversion attack effectively. However, we find that DP-VaeGM is not robust enough in mitigating the model inversion attack. We will discuss this in Section 5.4.



(a) Original attack (b) Attack with DP-AuGM

Figure 5–1 The efficiency of the model inversion attack on MNIST dataset before and after deploying DP-AuGM

5.2 Membership Inference Attack

We evaluate how DP-AuGM and DP-VaeGM perform in mitigating this attack on MNIST using one-layer neural networks. The training set size is set to be 1,000 and the number of shadow models[10] is set to be 50. We have set the privacy budget and to be 1 and 10 ^{5}, respectively. For this attack, we mainly consider

Table 5–1 Privacy loss for the membership inference attack

Original attack (MNIST)	0.2	0.6	0.2	0.2	0.1	0.2	0.1	0.1	0.2	0.0
With DP-AuGM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
With DP-VaeGM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

whether this attack can predict the existence of private data in the training set. To evaluate the attack, we use the standard metric—precision, as specified in[10] that the fraction of the records inferred as members of the private training dataset that are indeed members. The result is shown in Figure 5–2. As we can see from Figure 5–2, after deploying DP-AuGM, the attack precision for all the classes drops at least 10% and for some classes, the attack precision is approaching zero, such as classes 2 and 5. Similarly for DP-VaeGM, the attack precision drops over 20% for all the classes. Thus, we conclude that, with DP-AuGM and DP-VaeGM, the membership inference attack can be effectively defended against. The privacy loss on MNIST is also tabulated in Table 5–1. As we can see, with our proposed generative models, the privacy loss for each class can be reduced to zero.

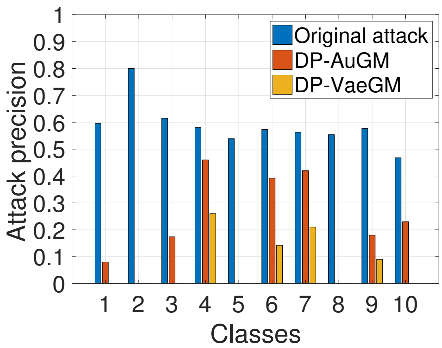


Figure 5–2 Evaluation of DP-AuGM, and DP-VaeGM against the membership inference attack on MNIST

5.3 GAN based Attack against Collaborative Deep Learning

We analyze GAN based attack on MNIST in order to analyze the strongest attacker due to the simplicity of the dataset. We create two participants in this setting, where one serves as an adversary and the other serves as an honest user, as suggested in[11]. We follow the same model structure as specified in[11], where the CNN is used as a discriminator and the DCGAN[62] is used as a generator. Users can apply the proposed differentially private generated data or original data to training their local models. We show defense results for DP-AuGM in Figure 5–3, where the first row represents the images obtained by adversaries without deploying generative models, while the second row shows the obtained images which have been protected by DP-AuGM. As we can see from Figure 5–3, the proposed model DP-AuGM significantly thwarts the attacker's attempt to recover anything from the private data. However, similar with the results from model inversion attack, DP-VaeGM is not robust enough to defend against this attack. We will also discuss in detail in Section 5.4.

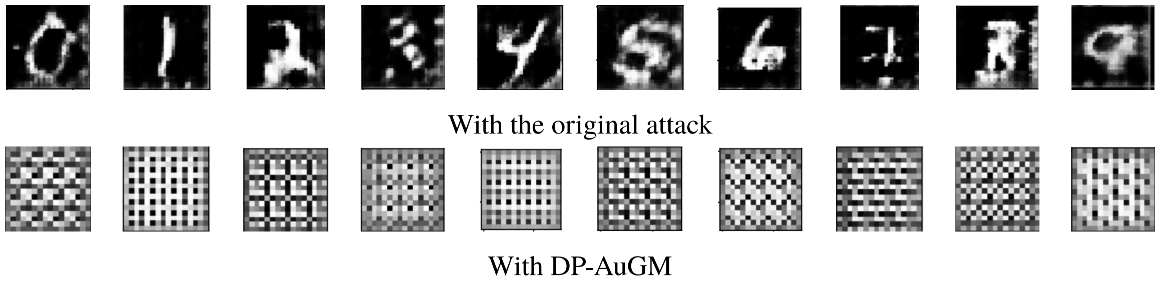


Figure 5–3 Images generated by the GAN based attack against collaborative deep learning on the MNIST dataset

5.4 Discussion

Although both DP-VaeGM and DP-AuGM are differentially private generative models, the results show that DP-AuGM is robust against all the attacks while DP-VaeGM can only defend against the membership inference attack. The main difference between these two models is that DP-AuGM uses the output of the encoder (a part of the autoencoder) as the generated data while DP-VaeGM uses the output of the VAE. As the encoder functions can reduce the dimensions of the input data, we can envision that this operation will incur a big norm distance between the input data and the generated data in DP-AuGM. However, for DP-VaeGM, we can see from Figure 4–4, the generated data still maintains good quality. Considering the model inversion attack and GAN attack both target at recovering part of the training data of a model, the best result on DP-AuGM will be successfully recovering those encoded data while for DP-VaeGM, the result will be recovering the figures as shown in Figure 4–4. Therefore, it seems that the key to defend against these two attacks is not only differential privacy, but also the appearance of the generated data. This is also mentioned by Hitaj et al.[11], as they asserted that differential privacy is not effective in mitigating the developed GAN attack because differential privacy is not designed to solve such a problem. Differential privacy in deep learning targets at protecting the specific elements of training data, while the goal of these two attacks is to construct a data point which is similar to the training data. Even if the attacks are successful, differential privacy is not violated since the specific data points are not recovered. So we think this can explain why DP-AuGM and DP-VaeGM behave differently. We also provide a general principle for defending against the attacks which try to recover private data—construct new training data that is dissimilar to the private data.

In summary, experiments show that DP-AuGM can mitigate all the three attacks. DP-VaeGM is only robust to the membership inference attack.

Chapter 6 Deploying Data Generative Models on Real-World Applications

To demonstrate the applicability of the proposed generative models DP-AuGM and DP-VaeGM, here we will show how they can be easily integrated with two real-world applications: Machine Learning as a Service (MLaaS) that is commonly supported by major Internet companies and federated learning supported by Google.

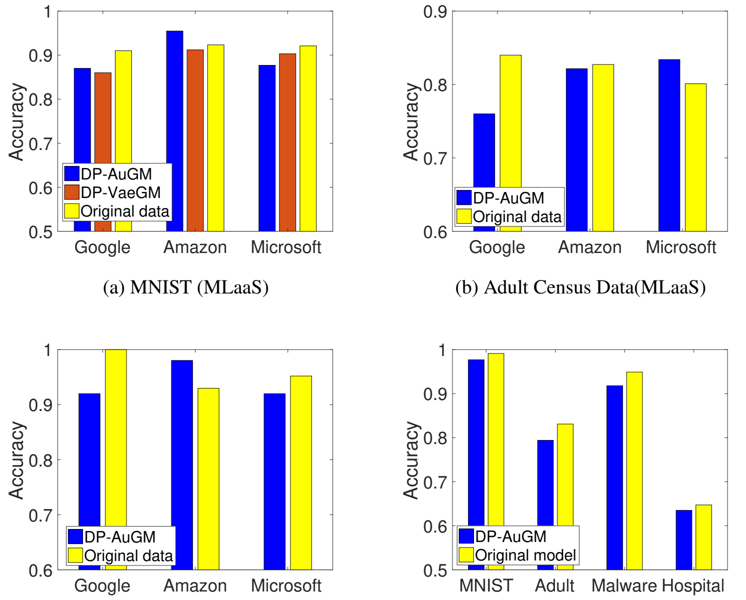
For DP-AuGM, we integrate it with MLaaS and federated learning over all the datasets. For DP-VaeGM, we integrate it with the real-world application MLaaS and evaluate it on image dataset MNIST, as currently VAEs are widely used for generating images. Considering federated learning is mainly threatened by the GAN based attack[11] but can be defended by DP-AuGM, we mainly focus on studying its utility when being integrated with DP-AuGM. We will make all of our source codes publicly available upon acceptance.

6.1 Machine Learning as a Service

MLaaS platforms are cloud-based systems that provide simple APIs as a web service for users who are interested in training and querying machine learning models. For a given task, a user first submits the private data through a web page interface or an mobile application created by developers, and selects the features for the task. Next, the user chooses a machine learning model from the platform, tunes the parameters of the model, and finally obtains the trained model. All these processes can be completed inside the mobile application. However, the private data submitted by innocent users can be maliciously exploited if the platform is compromised, which raises serious privacy concern. In this paper, our DP-AuGM and DP-VaeGM can serve as a data privacy protection module to protect privacy of the private data. To this end, users can first build DP-AuGM or DP-VaeGM locally, train the generative models with the private data, and then upload the generated data for later training. As we will show in the experiment, this will incur negligible utility loss for training, while significantly protecting data privacy. With DP-AuGM and DP-VaeGM, even if these platforms are compromised, the privacy of sensitive data can still be preserved.

When applying the proposed DP-AuGM and DP-VaeGM to MLaaS, we choose to examine three mainstream MLaaS platforms, which are Google Prediction API[63], Amazon Machine Learning[24], and Microsoft Azure Machine Learning[64]. The transparency of each step along the training pipeline exposed to users varies from each platform. Note that Google exposes none of the steps in the pipeline to the user but provides a "1-click" mode that simply trains a model using an uploaded dataset. Amazon does not expose the selection of learning models but allows the users to control a few meta-parameters. Microsoft provides full control of almost every step along the pipeline.

We set the differential privacy budget and to be 1 and 10 ^{5}, respectively, for DP-VaeGM and DP-AuGM. Similar with the evaluation section, we regard all the training data as private data and for DP-AuGM,



(a) MNIST (MLaaS)

(b) Adult Census Data(MLaaS)

(c) Hospital Data(MLaaS)

(d) Malware Data(MLaaS)

(e) Accuracy on four datasets (federated learning)

Figure 6–1 Accuracy of trained models when integrating proposed generative models with MLaaS and federated learning platforms

we split the test data the same way as we do in Section 4.2. As we can see from Figure 6–1, using the generated data by DP-AuGM for training, we can achieve comparatively high accuracy (accuracy deteriorating within 8%) on all three platforms for all datasets. Strikingly, we find that the model trained with generated data sometimes even outperforms the one trained with original data (see trained models on Amazon over MNIST). This observation provides great advantage for deploying DP-AuGM on current MLaaS platforms to both protect privacy and preserve high data utility. For DP-VaeGM, the result is shown in Figure 6–1a. We can see that DP-VaeGM can achieve comparable utility (accuracy deteriorating within 3%) on all the three platforms on MNIST. This clearly shows that DP-VaeGM and DP-AuGM have the potential to be well integrated into Machine Learning as a Service platforms and provide privacy guarantees for users' private data and retain high data utility at the same time.

6.2 Federated Learning

Federated learning[12], which is proposed by Google, enables mobile users to collaboratively train a shared prediction model and keep all their distributed training data local. Users typically train the model locally on their own device, upload the summarized parameters as a small focused update, and download the parameters averaged with other users' updates collaboratively using secure multiparty computation (MPC), without needing to share their personal training data in the cloud.

Federated learning is demonstrated to be private since the individual users' data is stored locally and the updates are securely aggregated by leveraging MPC to compute model parameters. However, the recent paper[11] declares that federated learning is secure only if we consider the attacker is the cloud provider who scrutinizes individual updates. If the attackers are the casual colluding participants, private data of one participant can still be recovered by other users who aim to attack. Hitaj et al. have shown that only applying differential privacy in federated learning is not enough to mitigate the GAN based attack, and a malicious user is able to successfully recover private data of others.

From Section 5, we show that DP-AuGM is robust enough to mitigate the GAN attack. Thus, in this part, we will mainly consider whether it can be well integrated into the federated learning to protect privacy and retain high data utility. We show the concrete steps toward integrating DP-AuGM as below. Note that the first two steps are added to the original federated learning platform.

1. *Users first train DP-AuGM locally with the private data.*
2. *After training DP-AuGM, users use DP-AuGM and some public data to generate new training data.*
3. Users train the local model with *generated data* locally and upload the summarized parameters to the server.

Next we will empirically show that DP-AuGM can be well integrated into federated learning over four datasets. We will then study in detail the model sensitivity over MNIST dataset.

6.2.1 Settings

The structure of autoencoder and differential privacy parameters can be specified by a central server such as Google, and will be publicly available to any user. As proof-of-concept, we hereby set the differential privacy parameters and to be 1 and 10 ^{5}, respectively. For each user in the federated learning, we evenly split the private data and public data for usage.

6.2.2 Hyper-parameters

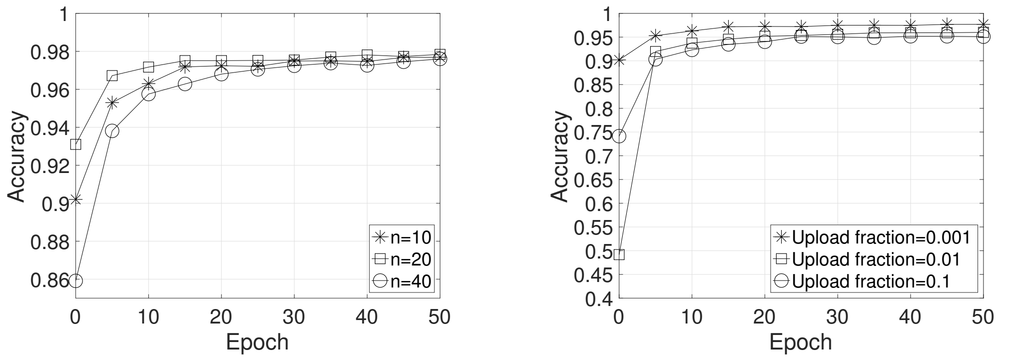
We set the default learning rate to be 0.001, the batch size to be 100, the number of users to be 10, and the uploading fraction to be 0.1. We will also test how DP-AuGM performs across different parameters later.

6.2.3 Comparison with the Original Federated Learning

We apply DP-AuGM to federated learning and compare it with the original setting without DP-AuGM. As we can see from Figure 6–1e, after we add DP-AuGM model to the pipeline, the accuracy drops only within 5% for all datasets. Hence, it shows the proposed DP-AuGM can be well integrated into federated learning without affecting its utility too much. In the following part, we study in detail about the model sensitivity on the MNIST dataset.

6.2.4 Effect of Other Parameters

We further examine the effect of the number of users and the upload fraction over the privacy-preserving federated learning model.



(a) Accuracy on MNIST with different number of users

(b) Accuracy on MNIST across different upload fraction

Figure 6–2 The performance of federated learning integrated DP-AuGMunder different hyper-parameters

6.2.4.1 **Number of Users**

We choose the number of users to be 10, 20, and 40. From Figure 6–2a, we can see the difference in number of users will only affect the speed of convergence a bit without affecting the final data utility. We find that although more users will take slightly more time for the model to converge, the accuracy of the privacy-preserving model actually converges to the same result within 50 epochs.

6.2.4.2 **Upload Fraction**

We choose the upload fraction as 0.001, 0.01, and 0.1 to analyze the proposed method. As we can see from Figure 6–2b, different learning rates only have negligible impact on the trained model.

In summary, we have shown that DP-AuGM can be well integrated with the commonly used two real-world applications and DP-VaeGM can be well integrated with the Machine Learning as a Service. The integrated models can protect privacy and preserve high data utility at the same time.

Chapter 7 Related Work

7.1 Privacy Attacks on Machine Learning Models

As machine learning models have become ubiquitous, multiple privacy attacks against learning models have been proposed. The goal of such attacks is to recover sensitive information from the inputs via various approaches.

Specifically, Homer et al.[65] show that it is possible to learn whether a target individual was related to certain disease by comparing the target's profile against the aggregated information obtained from public sources. This attack was then extended by Wang et al.[66] by performing correlation attacks, without prior knowledge about the target. Backes et al.[67] propose to conduct the membership inference attack against individuals contributing their microRNA expressions to scientific studies. If an attacker can learn information about individual's genome expression, he can potentially infer/profile the victim's future/historical health records, which can lead to severe consequences. Shokri et al.[10] later show that machine learning models can leak information about medical data records by performing membership attack against well trained models. Recently, Hitaj et al.[11] show that a GAN based attack can compromise user privacy in the collaborative learning setting[42], where each participant collaboratively trains his or her own model with private data locally. Hitaj et al.[11] also warn that simply adding differentially private noise is not robust enough to mitigate the attack. Besides federated learning, Hayes et al.[68] recently study privacy leakage for generative models in MLaaS.

Given these existing privacy attacks, learning with generated data from DP generative models can potentially defend against them, such as the representative model inversion attack, membership inference attack, and GAN based attack against collaborative deep learning. To the best of our knowledge, the learning method that can defend against all these attacks has not been proposed or systematically examined before.

7.2 Privacy-Preserving Learning Methods

The goal of privacy-preserving learning models is to protect sensitive information of individuals within the training set. Differential privacy is a strong and common notion to protect the data privacy[43]. Differential privacy can also be used to mitigate membership inference attacks, as its indistinguishability-based definition formally proves that the presence or absence of an instance does not affect the output of the learned model significantly[10]. A common approach to achieving differential privacy is to add noise from Laplacian[69] or Gaussian distribution[70] whose variance is determined by the privacy budget. In practice, differentially private schemes are often tailored to the spatio-temporal location privacy analysis[71–75].

To protect the privacy of machine learning models, random noise can be injected to input, output, and objectives of the models. Erlingsson et al.[76] propose to randomize the input and show that the randomized input still allows data collectors to gather meaningful statistics for training. Chaudhuri et al.[77] show that by adding noise to the cost function minimized during learning, -differential privacy can be achieved. In terms of perturbing objectives, Shokri et al.[42] show that deep neural networks can be trained with multi-party computations from perturbed model parameters to achieve differential privacy guarantees. Deep learning with differential privacy is proposed[1] by adding noise to the gradient during each iteration. They further use moment accountant to keep track of the spent privacy budget during the training phase. However, the prediction accuracy of the deep learning system will degrade more than 13% over the CIFAR-10 dataset when large differential privacy noise is added[1], which is unacceptable in many real-world applications where high prediction accuracy is pursued, such as autonomous driving[78] and face recognition[79]. This is also aligned with the warning proposed by Hitaj et al.[11] that using differential privacy to provide strong privacy guarantees cannot be applied to all scenarios, especially where the GAN based attack can be applied. Later, private aggregation of teacher ensembles (PATE) has been proposed, which first learns an ensemble of teacher models on a disjoint subset of training data, and aggregates the output of these teacher models to train a privacy-preserving student model for prediction[20]. The queries performed on the teacher models are designed to minimize the privacy cost of these queries. Once the student models are trained, the teacher models can be discarded. PATE is within the scope of knowledge aggregation and transfer for privacy[80, 81]. An improved version of PATE, scalable PATE is proposed by introducing new aggregation algorithm to achieve better data utility[2].

At inference, random noise can also be introduced to the output to protect privacy. However, this severely decays the test accuracy, because the amount of noise introduced increases with the number of inference queries answered by the machine learning model. Note that homomorphic encryption[82] can also be applied to protect the confidentiality of each individual input. The main limitations are the performance overhead and the restricted set of arithmetic operations supported by homomorphic encryption.

Various approaches have been proposed for the automatic discovery of sensitive entities, such as identifiers, and redact them to protect privacy. The simplest of these rely on a large collection of rules, dictionaries, and regular expressions (e.g.,[83, 84]). Chakaravarthy et al.[85] proposed an automated data sanitization algorithm aimed at removing sensitive identifiers while inducing the least distortion to the contents of documents. However, this algorithm assumes that sensitive entities, as well as any possible related entities, have already been labeled. Similarly, Jiang et al.[86] have developed the t-plausibility algorithm to replace the known (labeled) sensitive identifiers within the documents and guarantee that the sanitized document is associated with at least t documents. Li et al.[87] have proposed a game theoretic framework for automatic redacting sensitive information. In general, finding and redacting sensitive information with high accuracy is still challenging.

Unlike previously proposed techniques, our proposed privacy-preserving generative models can guarantee differential privacy while maintaining data utility. The proposed models achieve all three goals: protect privacy of training data; enable users to locally customize the privacy preference by configuring the generative models; retain high data utility for generated data. The proposed models achieve these goals at a much lower computation cost than aforementioned differentially private mechanisms and cryptographic techniques, such as secure multi-party computation or homomorphic encryption. These generative models are also easy to be integrated with MLaaS and federated learning[12] in practice to protect data privacy.

7.3 Privacy guarantees

Early works on privacy-preserving learning were done in the framework of secure function evaluation (SFE) and secure multi-party computations (MPC), where the input is split between two or more parties, and the focus is on minimizing information leaked during the joint computation of some agreed-to functionality. In contrast, we assume that data is held centrally, and we are concerned with leakage from the functionality ’s output (i.e., the model).

Another approach, k-anonymity and closely related notions[88], seeks to offer a degree of protection to underlying data by generalizing and suppressing certain identifying attributes. The approach has strong theoretical and empirical limitations[89, 90] that make it all but inapplicable to de-anonymization of high-dimensional, diverse input datasets. Rather than pursue input sanitization, we keep the under- lying raw records intact and perturb derived data instead. The theory of differential privacy, which provides the analytical framework for our work, has been applied to a large collection of machine learning tasks that differed from ours either in the training mechanism or in the target model.

7.4 Conclusion

We have designed, implemented, and evaluated two differentially private data generative models—a differentially private autoencoder based generative model (DP-AuGM) and a differentially private variational autoencoder based generative model (DP-VaeGM). We show that both models can provide strong privacy guarantees and retain high data utility for machine learning tasks. We empirically demonstrate that DP-AuGM is robust against the state-of-the-art privacy violation attacks, such as the model inversion attack, membership inference attack, and GAN based attack against collaborative deep learning, and DP-VaeGM is robust to the membership inference attack. Furthermore, we show that the proposed generative models can be easily integrated with two real-world applications—machine learning as a service and federated learning, which are previously threatened by the membership inference attack and GAN based attack, respectively. We empirically demonstrate that the integrated system can both protect privacy of users' data and retain high data utility.

Through the study of privacy attacks and corresponding defensive methods, we claim it is important to generate differentially private synthetic data for various machine learning systems to secure current learning tasks. We are the first to propose differentially private data generative models that can defend against the contemporary privacy violation attacks. We hope that our work will help pave the way toward designing more effective privacy-preserving learning methods.

Appendix A Proof of Theorem 2

Theorem 2. *Let a generative model (VAE) of class i* $M_i : X_i \to X_i'$ *be* (ϵ, δ)*-differentially private. Then if* $G_n : X \to \prod_{i=1}^{n} X_i'$ *is defined to be* $G_n = \bigcup_{i=1}^{n} M_i$, G_n *is* (ϵ, δ)*-differentially private, for any integer n.*

Proof. Given two adjacent datasets X_1 and $X_2 = X_1 \cup \{b\}$, without loss of generalization, assume b belongs to class k $(1 \le k \le n)$. Fix any subset of events $S \subseteq \prod_{i=1}^{n} X_i'$. Since the n generative models are pairwise independent, we obtain $\Pr[G_n(X_1) \in S] = \prod_{i=1}^{n} \Pr[M_i(x_i^1) \in S]$, where $x_i^1 \subseteq X_1 = \bigcup_{i=1}^{n} x_i^1$ denotes the training data of X_i for the ith generative model. Similarly, $\Pr[G_n(X_2) \in S] = \prod_{i=1}^{n} \Pr[M_i(x_i^2) \in S]$. Since X_1 and X_2 only differ in b, we have $x_i^1 = x_i^2$ and $\Pr[M_i(x_i^1) \in S] = \Pr[M_i(x_i^2) \in S]$, for any $i \neq k$. Since M_k is (ϵ, δ)-differentially private, then we have $\Pr[M_k(x_k^1) \in S] \le e^{\epsilon} \Pr[M_k(x_k^2) \in S] + \delta$. Therefore, we obtain $\Pr[G_n(X_1) \in S] = \prod_{i=1}^{n} \Pr[M_i(x_i^1) \in S] = \Pr[M_1(x_1^2) \in S] \cdots \Pr[M_k(x_k^1) \in S] \cdots \Pr[M_n(x_n^2) \in S] \le e^{\epsilon} \prod_{i=1}^{n} \Pr[M_i(x_i^2) \in S] + \delta = e^{\epsilon} \Pr[G_n(X_2) \in S] + \delta$. The inequality derives from the fact that any probability is no greater than 1. Hence, G_n is (ϵ, δ)-differentially private, for any n.

Appendix B Model Architectures

Table B–1 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 B–2 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 B–3 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 B–4 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)

Bibliography

[1] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy[C]II Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM. [S.l.]: [s.n.], 2016: 308–318.

[2] PAPERNOT N, SONG S, MIRONOV I, et al. Scalable Private Learning with PATE[J]. International Conference on Learning Representations, 2018.

[3] HINTON G, DENG L, YU D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82–97.

[4] CIRESAN D, GIUSTI A, GAMBARDELLA L M, et al. Deep neural networks segment neuronal membranes in electron microscopy images[C]II Advances in neural information processing systems. [S.l.]: [s.n.], 2012: 2843–2851.

[5] CIREGAN D, MEIER U, SCHMIDHUBER J. Multi-column deep neural networks for image classification[C]II Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE. [S.l.]: [s.n.], 2012: 3642–3649.

[6] ZHANG F, LEITNER J, MILFORD M, et al. Towards vision-based deep reinforcement learning for robotic motion control[J]. ArXiv preprint arXiv:1511.03791, 2015.

[7] VENKATADRI G, ANDREOU A, LIU Y, et al. Privacy Risks with Facebook ' s PII-based Targeting: Auditing a Data Broker ' s Advertising Interface[C]II Security and Privacy (SP), 2018 IEEE Symposium on. IEEE. [S.l.]: [s.n.], 2018.

[8] FREDRIKSON M, LANTZ E, JHA S, et al. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing.[C]II USENIX Security Symposium. [S.l.]: [s.n.], 2014.

[9] FREDRIKSON M, JHA S, RISTENPART T. Model inversion attacks that exploit confidence information and basic countermeasures[C]II Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. [S.l.]: [s.n.], 2015.

[10] SHOKRI R, STRONATI M, SONG C, et al. Membership inference attacks against machine learning models[C]II Security and Privacy (SP), 2017 IEEE Symposium on. IEEE. [S.l.]: [s.n.], 2017: 3–18.

[11] HITAJ B, ATENIESE G, PEREZ-CRUZ F. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning[J]. CCS, 2017.

[12] MCMAHAN B, RAMAGE D. Federated learning: Collaborative machine learning without centralized training data[R]. Technical report, Google, 2017.

[13] U.S. Dept. of Health and Human Services. Standards for Privacy and Individually identifiable health information; final rule[J]. Federal Register, 2000, 65(250): 82462–82829.

[14] Committe on the Judiciary House of Representatives. Federal Rules of Civil Procedure. 2014.

[15] PARLIAMENT E, of the EUROPEAN UNION C. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data and repealing Directive 95/46/EC (General Data Protection Regulation)[J]. Official Journal of the European Union, 2016, 119: 1–88.

[16] FUNG B, WANG K, CHEN R, et al. Privacy-preserving data publishing: A survey of recent developments[J]. ACM Computing Surveys, 2010, 42(4): 14.

[17] SWEENEY L. K-anonymity: A model for protecting privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05): 557–570.

[18] DWORK C. Differential Privacy: A Survey of Results[C]II Proc. 5th International Conference on Theory and Applications of Models of Computation. [S.l.]: [s.n.], 2008: 1–19.

[19] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy[C]II Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. [S.l.]: [s.n.], 2016.

[20] PAPERNOT N, ABADI M, ERLINGSSON U, et al. Semi-supervised knowledge transfer for deep learning from private training data[J]. ArXiv preprint arXiv:1610.05755, 2016.

[21] FREDRIKSON M, JHA S, RISTENPART T. Model inversion attacks that exploit confidence information and basic countermeasures[C]II Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM. [S.l.]: [s.n.], 2015: 1322–1333.

[22] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998.

[23] DENG J, DONG W, SOCHER R, et al. ImageNet: A large-scale hierarchical image database[C]II Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. [S.l.]: [s.n.], 2009: 248–255.

[24] Amazon Machine Learning. @LLHKYII9OK69E9RGF6;GEIE9;@AF=ƅD=9JFAF?I.

[25] BENGIO Y. Learning Deep Architectures for AI[J]. Foundations & Trendső in Machine Learning, 2009, 2(1): 1–127.

[26] BOS J W, LAUTER K, NAEHRIG M. Private predictive analysis on encrypted medical data[J]. Journal of Biomedical Informatics, 2014, 50(8): 234–243.

[27] GRAVES A, MOHAMED A R, HINTON G. Speech recognition with deep recurrent neural networks[C]II IEEE International Conference on Acoustics, Speech and Signal Processing. [S.l.]: [s.n.], 2013: 6645–6649.

[28] HANNUN A, CASE C, CASPER J, et al. Deep Speech: Scaling up end-to-end speech recognition[J]. Computer Science, 2014.

[29] HINTON G, DENG L, YU D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82–97.

[30] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]II International Conference on Neural Information Processing Systems. [S.l.]: [s.n.], 2012: 1097–1105.

[31] SIMARD P Y, STEINKRAUS D, PLATT J C. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis[C]II International Conference on Document Analysis & Recognition. [S.l.]: [s.n.], 2003: 958.

[32] TAIGMAN Y, YANG M, Marc, et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification[C]II Computer Vision and Pattern Recognition. [S.l.]: [s.n.], 2014: 1701–1708.

[33] HE K, ZHANG X, REN S, et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification[J]. 2015: 1026–1034.

[34] CRUZROA A A, AREVALO OVALLE J E, MADABHUSHI A, et al. A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection.[C]II. [S.l.]: [s.n.], 2015: 403–10.

[35] FAKOOR R, LADHAK F, NAZI A, et al. Using deep learning to enhance cancer diagnosis and classification[C]II The International Conference on Machine Learning. [S.l.]: [s.n.], 2013.

[36] LIANG M, LI Z, CHEN T, et al. Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach[J]. IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2015, 12(4): 928–937.

[37] XIONG H Y, ALIPANAHI B, LEE L J, et al. RNA splicing. The human splicing code reveals new insights into the genetic determinants of disease.[J]. Science, 2015, 347(6218): 1254806.

[38] DEAN J, CORRADO G S, MONGA R, et al. Large scale distributed deep networks[C]II International Conference on Neural Information Processing Systems. [S.l.]: [s.n.], 2012: 1223–1231.

[39] CHAPELLE O, CHAPELLE O, LANGFORD J. A reliable effective terascale linear learning system[J]. Journal of Machine Learning Research, 2014, 15(1): 1111–1133.

[40] NIU F, RECHT B, RE C, et al. HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent[J]. Advances in Neural Information Processing Systems, 2011, 24: 693–701.

[41] ZINKEVICH M, WEIMER M, SMOLA A J, et al. Parallelized Stochastic Gradient Descent[C]II Advances in Neural Information Processing Systems 23: Conference on Neural Information Processing Systems 2010. Proceedings of A Meeting Held 6-9 December 2010, Vancouver, British Columbia, Canada. [S.l.]: [s.n.], 2010: 2595–2603.

[42] SHOKRI R, SHMATIKOV V. Privacy-preserving deep learning[C]II Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM. [S.l.]: [s.n.], 2015: 1310–1321.

[43] DWORK C, ROTH A, et al. The algorithmic foundations of differential privacy[J]. Foundations and Trendső in Theoretical Computer Science, 2014, 9(3–4): 211–407.

[44] DWORK C, KENTHAPADI K, MCSHERRY F, et al. Our Data, Ourselves: Privacy via Distributed Noise Generation[C]II Advances in Cryptology - EUROCRYPT 2006, International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings. [S.l.]: [s.n.], 2006: 486–503.

[45] DWORK C, LEI J. Differential privacy and robust statistics[C]II ACM Symposium on Theory of Computing. [S.l.]: [s.n.], 2009: 371–380.

[46] DWORK C, ROTHBLUM G N, VADHAN S. Boosting and Differential Privacy[C]II IEEE Symposium on Foundations of Computer Science. [S.l.]: [s.n.], 2010: 51–60.

[47] KAIROUZ P, OH S, VISWANATH P. The Composition Theorem for Differential Privacy[J]. IEEE Transactions on Information Theory, 2015, 63(6): 4037–4049.

[48] MCSHERRY F D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis.[J]. Communications of the Acm, 2010, 53(9): 89–97.

[49] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. [S.l.]: MIT press, 2016.

[50] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders[C]II Proceedings of the 25th international conference on Machine learning. ACM. [S.l.]: [s.n.], 2008: 1096–1103.

[51] VINCENT P, LAROCHELLE H, LAJOIE I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. Journal of Machine Learning Research, 2010, 11.

[52] KINGMA D P, WELLING M. Auto-encoding variational bayes[J]. ICLR, 2014.

[53] REZENDE D J, MOHAMED S, WIERSTRA D. Stochastic backpropagation and approximate inference in deep generative models[J]. ICML, 2014.

[54] COVER T. Information theory and statistics[M]. [S.l.]: Wiley, 1959: 301.

[55] HINTON G E, SALAKHUTDINOV R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006.

[56] DENG L, SELTZER M L, YU D, et al. Binary coding of speech spectrograms using a deep autoencoder[C]II Eleventh Annual Conference of the International Speech Communication Association. [S.l.]: [s.n.], 2010.

[57] MASCI J, MEIER U, CIREAN D, et al. Stacked convolutional auto-encoders for hierarchical feature extraction[J]. Artificial Neural Networks and Machine Learning–ICANN 2011, 2011: 52–59.

[58] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278–2324.

[59] LICHMAN M. UCI Machine Learning Repository. 2013. @LLHYII9J;@AN=6A;K6M;A6=<MIED.

[60] Hospital Discharge Data Public Use Data File. 2018. @LLHKYIIOOO6<K@K6L=P9K6?GNI2&!'!I&GKHAL9DKI"GOFDG9<6K@LE.

[61] CHEN S, XUE M, TANG Z, et al. Stormdroid: A streaminglized machine learning-based system for detecting android malware[C]II Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. ACM. [S.l.]: [s.n.], 2016: 377–388.

[62] RADFORD A, METZ L, CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. ArXiv preprint arXiv:1511.06434, 2015.

[63] Google Prediction API. @LLHKYII;DGM<6?GG?D=6;GEIHJ=<A;LAGFI.

[64] Microsoft Azure Machine Learning. @LLHKYIIKLM<AG69RMJ=ED6F=LI.

[65] HOMER N, SZELINGER S, REDMAN M, et al. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays[J]. PLoS genetics, 2008, 4(8): e1000167.

[66] WANG R, LI Y F, WANG X, et al. Learning your identity and disease from research papers: information leaks in genome wide association study[C]II Proceedings of the 16th ACM conference on Computer and communications security. ACM. [S.l.]: [s.n.], 2009: 534–544.

[67] BACKES M, BERRANG P, HUMBERT M, et al. Membership privacy in MicroRNA-based studies[C]II Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM. [S.l.]: [s.n.], 2016: 319–330.

[68] HAYES J, MELIS L, DANEZIS G, et al. LOGAN: Evaluating Privacy Leakage of Generative Models Using Generative Adversarial Networks[J]. ArXiv preprint arXiv:1705.07663, 2017.

[69] DWORK C. Differential privacy: A survey of results[C]II International Conference on Theory and Applications of Models of Computation. Springer. [S.l.]: [s.n.], 2008: 1–19.

[70] DWORK C, KENTHAPADI K, MCSHERRY F, et al. Our Data, Ourselves: Privacy Via Distributed Noise Generation.[C]II Eurocrypt. Vol. 4004. Springer. [S.l.]: [s.n.], 2006: 486–503.

[71] MACHANAVAJJHALA A, KIFER D, ABOWD J, et al. Privacy: Theory meets practice on the map[C]II Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. IEEE. [S.l.]: [s.n.], 2008: 277–286.

[72] RASTOGI V, NATH S. Differentially private aggregation of distributed time-series with transformation and encryption[C]II Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM. [S.l.]: [s.n.], 2010: 735–746.

[73] SHOKRI R, THEODORAKOPOULOS G, TRONCOSO C, et al. Protecting location privacy: optimal strategy against localization attacks[C]II Proceedings of the 2012 ACM conference on Computer and communications security. ACM. [S.l.]: [s.n.], 2012: 617–627.

[74] ACS G, CASTELLUCCIA C. A case study: privacy preserving release of spatio-temporal density in paris[C]// Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. [S.l.]: [s.n.], 2014: 1679–1688.

[75] TO H, NGUYEN K, SHAHABI C. Differentially private publication of location entropy[C]// Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM. [S.l.]: [s.n.], 2016: 35.

[76] ERLINGSSON Ú, PIHUR V, KOROLOVA A. Rappor: Randomized aggregatable privacy-preserving ordinal response[C]// Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM. [S.l.]: [s.n.], 2014: 1054–1067.

[77] CHAUDHURI K, MONTELEONI C, SARWATE A D. Differentially private empirical risk minimization[J]. Journal of Machine Learning Research, 2011, 12(Mar): 1069–1109.

[78] GEIGER A, LENZ P, URTASUN R. Are we ready for autonomous driving? the kitti vision benchmark suite[C]// Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE. [S.l.]: [s.n.], 2012: 3354–3361.

[79] GRAHAM D B, ALLINSON N M. Characterising virtual eigensignatures for general purpose face recognition[G]// Face Recognition. [S.l.]: Springer, 1998: 446–456.

[80] PATHAK M, RANE S, RAJ B. Multiparty differential privacy via aggregation of locally trained classifiers[C]// Advances in Neural Information Processing Systems. [S.l.]: [s.n.], 2010: 1876–1884.

[81] HAMM J, CAO Y, BELKIN M. Learning privately from multiparty data[C]// International Conference on Machine Learning. [S.l.]: [s.n.], 2016: 555–563.

[82] GILAD-BACHRACH R, DOWLIN N, LAINE K, et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy[C]// International Conference on Machine Learning. [S.l.]: [s.n.], 2016: 201–210.

[83] BECKWITH B A, MAHAADEVAN R, BALIS U J, et al. Development and evaluation of an open source software tool for deidentification of pathology reports[J]. BMC Medical Informatics and Decision Making, 2006, 6: 12.

[84] SWEENEY L. Replacing personally-identifying information in medical records, the Scrub system.[C]// AMIA Fall Symposium. [S.l.]: [s.n.], 1996: 333.

[85] CHAKARAVARTHY V T, GUPTA H, ROY P, et al. Efficient techniques for document sanitization[C]// Proceedings of the 17th ACM conference on Information and knowledge management. ACM. [S.l.]: [s.n.], 2008: 843–852.

[86] JIANG W, MURUGESAN M, CLIFTON C, et al. T-Plausibility: semantic preserving text sanitization[C]// International Conference on Computational Science and Engineering. Vol. 3. [S.l.]: [s.n.], 2009: 68–75.

[87] LI B, VOROBEYCHIK Y, LI M, et al. Scalable Iterative Classification for Sanitizing Large-Scale Datasets[J]. IEEE transactions on knowledge and data engineering, 2017, 29(3): 698–711.

[88] SWEENEY L. K-ANONYMITY: A MODEL FOR PROTECTING PRIVACY[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05): 557–570.

[89] AGGARWAL C C. On k -anonymity and the curse of dimensionality[C]II International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September. [S.l.]: [s.n.], 2005: 901–909.

[90] BRICKELL J, SHMATIKOV V. The cost of privacy: destruction of data-mining utility in anonymized data publishing[C]II ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.]: [s.n.], 2008: 70–78.

Acknowledgements

Upon finishing this paper, I have to thank a lot of people who help me and guide me to work on this project. Firstly, I have to thank Professor Zhu. He is the person who leads me into the academic world and gives me the chance to get in touch with different research topics in computer security. Secondly, I also have to thank my senior classmates in the NSEC lab, like Minhui Xue, Lu Zhou, Huaxin Li, and so on. I have to thank them for sharing some research opportunities with me. Thirdly, I want to thank Xingyun for accompanying me whenever I feel depressed and she always gives me the power to bravely face the numerous difficulties in life. At last, I just want to thank everyone who helps me and cares about me in my four years at SJTU. Without you, I could not make to this stage and I could not gain enough strength to keep me moving. Anyway, upon this opportunity, wishing all of us a bright future!