Defending against Membership Inference Attacks with High Utility by GAN

Li Hu, Jin Li*, *Senior Member, IEEE*, Guanbiao Lin, Shiyu Peng, Zhenxin Zhang, Yingying Zhang, and Changyu Dong, *Member, IEEE.*

Abstract—The success of machine learning (ML) depends on the availability of large-scale datasets. However, recent studies have shown that models trained on such datasets are vulnerable to privacy attacks, among which membership inference attack (MIA) brings serious privacy risk. MIA allows an adversary to infer whether a sample belongs to the training dataset of the target model or not. Though a variety of defenses against MIA have been proposed such as differential privacy and adversarial regularization, they also result in lower model accuracy and thus make the models less unusable. In this paper, aiming at maintaining the accuracy while protecting the privacy against MIA, we propose a new defense against membership inference attacks by generative adversarial network (GAN). Specifically, sensitive data is used to train a GAN, then the GAN generate the data for training the actual model. To ensure that the model trained with GAN on small datasets can has high utility, two different GAN structures with special training techniques are utilized to deal with the image data and table data, respectively. Experiment results show that the defense is more effective on different data sets against the existing attack schemes, and is more efficient compared with most advanced MIA defenses.

Index Terms—Membership Inference Attack, Generative Adversarial Network, Machine Learning, Privacy.

1 Introduction

Recent advances in complex machine learning (ML) models and computing infrastructure, coupled with the availability of large amounts of data, have facilitated the use of machine learning in the realm of everyday life [1], [2]. For example, in the field of computer vision, machine learning is widely used in face recognition, object detection, image classification and so on. In addition to computer vision, machine learning is also being used in areas as diverse as recommendation systems [3] and critical medical management [4], [5]. ML's success has recently prompted leading Internet companies such as Google and Amazon to take machine learning as a service (MLaaS), where they provide training services for data owners to train ML models for different applications. These models are then either published, or accessed in a black-box fashion as prediction APIs.

Although ML model is very popular, it is vulnerable to various privacy reasoning attacks [6]–[10]. The data used to train the ML model often contains sensitive user information, such as clinical records, location traces, personal photos, etc [11]–[13]. Therefore, the ML model trained with sensitive data may disclose sensitive information and pose a privacy threat to the data owner. This paper mainly studies such an attack, namely the membership inference attack. In this case, the adversary has black box or white box access to the target model, and its goal is to determine whether a given sample belongs to the training set of the target model. Membership inference attack distinguishes members and non-members by learning the different behaviors of the target model on members and non-members. It can learn according to the different characteristics of the target model, including output confidence [9], prediction entropy [14], [15], prediction loss [16], data robustness [17], [18] and gradient of input loss relative to model parameters [19]. Membership inference attacks are particularly effective for large neural networks, because such models can better remember their training samples. Attack performance is also significantly improved by white-box access to trained models.

Recent work has also suggested several defenses against membership inference attacks. Existing defense methods can be summarized as confidence score masking [20], [21], regularization [14], [22], [23], knowledge distillation [24]–[27], and differential privacy [28]–[30]. The realization of these defense methods is based on two principles: (1) reduce the degree of over-fitting of the model; (2) add disturbance to the training process or output results of the model, such that membership information can be hidden. However, the defense capability of these methods is limited, there are trade-off between privacy and utility, and they may not be effective against some attacks.

In this paper, we start from the source of privacy leakage and use the powerful generative ability of GAN [31] to generate identically distributed data with the training data. Then we use the generated data to train a protected model and serve the user with this instead of a model trained directly with the sensitive training data, so that the protected model can provide good service while resisting membership inference attack. As is the case with existing defense efforts, we are primarily concerned with the privacy

Jin Li is the corresponding author.

- *Li Hu, Jin Li, Guanbiao Lin, Shiyu Peng, Zhenxin Zhang and Yingying Zhang are with Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangdong 510006, China. E-mail: hl_27@e.gzhu.edu.cn, lijin@gzhu.edu.cn, linguanbiao@e.gzhu.edu.cn, pengshy@e.gzhu.edu.cn, zhangzhenxin@e.gzhu.edu.cn, zhangyingying@e.gzhu.edu.cn.*
- *Changyu Dong with the School of Computing, Newcastle University, Newcastle upon Tyne NE1 7RU, United Kingdom. E-mail: changyu.dong@ncl.ac.uk.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

and utility guarantees implemented by the defense. In terms of privacy assurance, we replace the original training data with the generated data of GAN, and realize the resistance to membership inference attack by avoiding the attacker from contacting the model trained with the original training data. In terms of utility guarantees, we use a SOTA GAN to generate high quality surrogate data, the existing works mainly focus on the diversity and fidelity of generated data in terms of data quality, but in our defense work, we also focus on the utility of generated data. Therefore, our main contributions can be summarized as follows:

- We propose a new defense mechanism to **d**efend against **m**embership **i**nference attacks by **G**AN (DMIG) and achieve the best trade-off between privacy and utility.
- In order to improve the utility of generated data by GAN, we propose to train GAN with techniques such as truncation to ensure the privacy of members without damaging the utility of the model.
- We conduct experimental analysis using a variety of datasets, such as image dataset (CIFAR10 and MNIST) and table dataset (Location and Purchase). To show the effectiveness of this scheme, we test the defense effect on 10 attack schemes and compare the performance with 4 typical defense schemes.

The paper is organised as follows: Section 2 provides background knowledge about Machine Learning, Membership Inference Attack and Generative Adversarial Network. Section 3 introduces related work in the areas of Membership Inference Attack and Generative Adversarial Network. Section 4 introduces the insight and the design of our proposal. In Section 5, we present the evaluation results. Section 7 summarises the paper and concludes our future work.

2 Preliminaries

2.1 Machine learning

In this work, we focus on supervised learning and classification problems. Let $\mathcal{X}$ be the d-dimensional characteristic space and $\mathcal{Y}$ the c-dimensional output space, where c represents the total number of prediction classes. The goal of machine learning is to learn a parameter vector θ that represents a mapping $f_\theta : \mathcal{X} \mapsto \mathcal{Y}$. f_θ outputs a c-dimensional vector, and each dimension represents the input probability that the input belongs to the corresponding class. This function f_θ accurately predicts the labels of new data that have never been seen before.

The training process of a machine learning model is to find the optimal parameter θ that can accurately reflect the relationship between $\mathcal{X}$ and $\mathcal{Y}$. Let $Pr(\mathcal{X}, \mathcal{Y})$ be the distribution of all data points. We choose a training dataset $D_{tr} = \{x_i, y_i\}_{i=1}^{N}$ with N samples, where $x_i \in \mathcal{X}$ is the feature data, $y_i \in \mathcal{Y}$ is the corresponding ground-truth label and $D_{tr} \sim Pr(\mathcal{X}, \mathcal{Y})$.

Considering that $L(f_\theta)$ is a loss function, measure the deviation of the model's prediction of input x_i , $f_\theta(x_i)$, from the ground-truth label y_i of x_i:

$$L_{D_{tr}}(f_\theta) = \mathop{\mathbb{E}}_{\{x_i,y_i\}_{i=1}^{N} \sim D_{tr}} [\ell(f_\theta(x_i), y_i)] \tag{1}$$

The goal of the machine learning model f_θ is to minimize the expected loss of D_{tr}:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} L_{D_{\text{tr}}}(f_\theta) + \lambda\Omega(\theta) \tag{2}$$

where $\Omega(\theta)$ is a regularization term that penalizes model complexity and help avoid overfitting, and λ is a hyperparameter.

2.2 Membership inference attack (MIA)

Membership inference attack is an attack to judge whether data belongs to a model training set. Through this attack, the attacker can result in serious privacy disclosure. For example, if a machine learning model is trained on data collected from a patient with a disease, the attacker can immediately know the health status of the victim by knowing whether the victim's data belongs to the training data of the model. In most real-world scenarios, the target model is opaque but accessible, and the membership inference attack can infer the member information of the target model based on the information obtained by the access.

MIA against machine learning was first proposed by Shokri et al. [9]. In recent years, more and more work about membership inference attack has been carried out successfully in many fields, such as biomedical data and mobility data. Figure1 briefly describes the workflow of MIA. Given an instance x (i.e. query data), by accessing the target model $f_{target}(\theta)$, the attacker can obtain the output $f_{target}(x;\theta)$, and judge whether the instance x belongs to the training set D_{target}^{train} of the target model. We formulate the membership inference attack as a binary classification task where the attacker aims to classify whether a instance x has been used to train a victim model. Formally, we define:

$$\mathcal{A} : (x, f_{target}(\theta)) \rightarrow \{0, 1\} \tag{3}$$

here, 0 means x is not a member of $f_{target}(\theta)$'s training dataset D_{target}^{train} and 1 otherwise. The attack model $\mathcal{A}$ is essentially a binary classifier. Depending on the assumptions, it can be constructed in different ways, which will be described in later sections.

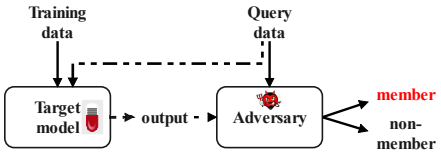


Fig. 1: The illustration of MIA.

2.3 Generative Adversarial Network (GAN)

GAN consists of two neural network modules, a generator G and a discriminator D, which are trained simultaneously in an adversarial manner. The generator takes random noise z (latent code) as input to generate samples with approximate distribution of training data, while the discriminator takes training dataset and samples generated by the generator as input, and optimizes it to distinguish generated data from training data. During training, the two modules compete and evolve so that the generator learns

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

to generate more and more realistic samples to fool the discriminator, which learns to more accurately distinguish between the two data sources. The training objective can be formulated as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))] \quad (4)$$

where P_{data} is the real data distribution, while the P_z is the prior distribution of the latent code. When updating the discriminator, the first item in the optimization objective forces the discriminator to output high scores given real data samples. The second item causes the discriminator to output a lower score on the generated sample. When updating the generator, the goal is to maximize the output score of the discriminator to the generated samples. Usually, once the training of GAN is completed, the discriminator is no longer useful. The generator will receive new potential code samples z taken from a known prior distribution (usually Gaussian distribution) and output synthetic data samples, which will be collected and used for downstream tasks.

3 Related work

3.1 Types of Membership inference attack

In order to verify the effectiveness of the scheme, DMIG, we will test different membership inference attack schemes, analyze the following types of attacks according to the capabilities of the attackers, and prove that our scheme can defend against any attack scheme through experiments.

Attacks with label knowledge in black-box scenarios. Label based attack in black box scenarios is the most likely to happen in actual scenarios, because when the model provides services, such as face recognition, that directly tell you who the person is, an attacker can easily access this information and execute an attack. By obtaining the minimum output knowledge, label-only attack can still achieve strong attack performance on a series of dataset [16]–[18]. This indicates that the ML model is more vulnerable to privacy attacks than we expected.

Attacks with partial output knowledge in black-box scenarios. The attack of partial output knowledge in the black box scenario is the classical MIA proposed by Shokri et al [9], by training an attack model, the attacker can distinguish between member and non-member with the confidence of the predicted output of the target model as the input of the attack model. Subsequently, Salem et al. [14] made improvements on this basis by relaxing the requirements on the structure of the shadow model, the data of the training shadow model and the number of shadow models, and also designed a method to determine whether the access data is a member by using the maximum value of the model output confidence. After this, literature [15] refined this attack by setting different thresholds for different class label.

Attacks with total output knowledge in black-box scenarios. In this attack, the attacker can use some statistical information of the target model to realize MIA, such as the average loss of model training data and prediction entropy. Literature [16] proposed that the attacker could judge whether the access data belonged to member data by calculating the prediction loss value of the access data, and showed that the attack only needed less computing resources and background knowledge to achieve the same performance as the neural network based attack proposed by Shokri et al. [9]. Subsequently, article [14] proved the effectiveness of using prediction entropy to carry out attacks, and literature [15] also proposed another improved attack based on prediction entropy. They believed that prediction entropy does not contain any information about real labels, which may lead to misclassification of members and non-members.

3.2 Defense Mechanisms against Membership inference attack

In order to demonstrate the advantages of this defense scheme in terms of privacy and utility, we compare it with existing defense schemes. The current defenses against MIA can fall into four categories, i.e., noise perturbation, regularization, transfer learning, and differential privacy.

Noise Perturbation Defense Methods. Adversary noise perturbation is a common privacy protection method. It protects sensitive information by adding noise to it. We can subdivide these defense methods into two categories: Differential Privacy and Confidence Score Perturbation. Differential privacy provides theoretical guarantees for protecting membership privacy for individual samples [28]–[30]. Although the differential privacy defence is widely applicable and effective, it has the disadvantage that it is difficult to implement trade-off between the model utility and privacy. In some MIAs, the success of MIA is to exploit the difference in confidence score vectors between members and non-members. Therefore, the defence method of adding noise perturbations to the confidence score vector emerges [20]–[22].

Regularization Defense Methods. Regularization technology is a collective term for a series of technologies. These techniques are proposed to reduce the overfitting of the model to improve the generalization ability of the model.The feature of regularization techniques determines that they can naturally become a method for defending membership inference attacks. From the perspective of defending membership inference attacks, we divide these regularization techniques into the following categories and describe their work in defending against MIA.

Transfer Learning Defense Methods. In the field of machine learning, in order to protect data privacy, relevant works [33]–[36] have combined knowledge transfer and differential privacy. For membership inference attacks, knowledge transfer can be used to protect membership privacy of target data. Moreover, recent studies [24]–[27] have showed that knowledge transfer can be used to train the model with membership privacy. By reducing the access to the target data and hiding the target data with similar but different data, it can prevent the attacker from inferring the privacy of the members of the target data and provide a better trade-off between the privacy of the members and the accuracy of classification.

3.3 GAN

In order to demonstrate the universality of the defense scheme in this paper, most of the experimental data used

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

in membership inference attack are verified experimentally. The experimental data we used were divided into two types: image data and table data. We will experiment with the generation of raw data using different GAN networks.

StyleGAN. Stylegan [37] is a new automatic learning, unsupervised advanced attribute separation generative adversal network introduced by NVIDIA in 2019. Stylegan2-ada [39] is an improved version of Stylegan and Stylegan2 [38]. The ada of Stylegan2-ada refers to adaptive discriminator augmentation, which wishes to avoid manual adjustment of the enhanced strength p and to control it dynamically based on the degree of overfitting. Ada can achieve better quality generation with less sample data and ensure that the generated images will not have corresponding enhancement traces under the premise of data expansion. The adaptive discriminator enhancement ensures that the model is not easily overfitting and the model is more stable.

CTGAN. CTGAN [40] tries to find a flexible and robust model to learn the distribution of columns with complex distributions that can be better distributed than Bayesian network learning. CTGAN's condition generator and sampling training can overcome the problem of unbalanced training data. In addition, conditional generators can help generate data with specific discrete values that can be used for data augmentation.

4 Introducing DMIG for Membership Privacy

We present DMIG for membership privacy, whose goal is to train ML models that are resilient to membership inference attacks. The intuition of the scheme is to publish the model trained by generated data, which can hide the member information of the original training data by means of generating data, so as to disturb the attacker's judgment of the member information of the original training data.

4.1 Notations

We begin by introducing the notation used throughout the paper. We consider the dataset $D_{\text{tr}} = \{(x_{(n)}, y_{(n)})\}_{n=1}^{N}$ is a private training dataset and its true underlying distribution P_{data}. We define a prior on input noise variables distribution p_z and use D_{tr} to train a generative adversarial network D-G. Then, we use $D_{\text{g-tr}} = \{(x_{(m)}, y_{(m)})\}_{m=1}^{M}$ to represent the alternate dataset generated by the GAN network, and use this to train a protected model, denoted by θ_p. We use $D_{\text{te}} = \{(x_{(n)}, y_{(n)})\}_{n=1}^{N}$ to represent the test dataset, which has no intersection with the dataset D_{tr} and follows the same distribution P_{data}.

4.2 Defense framework

Using training data to train a classifier, the classifier will learn the conditional probability distribution of training data. Unfortunately, DNN classifiers tend to over-fitting training data, resulting in inconsistent performance of the classifier on training data and non-training data, which makes it vulnerable to membership inference attacks. Our insight is that if we replace the training dataset with a new dataset that is independent and equally distributed over the training data, then we can mitigate membership inference attacks while preserving the test accuracy of the original classifier.

We propose a new approach to satisfy both properties. Here we present the details of main phases of DMIG. The defense framework is shown in Figure2. We first learn the distribution of the training data by training a GAN, and then generate a new dataset through this GAN. Finally, we use the new dataset to train a classifer.

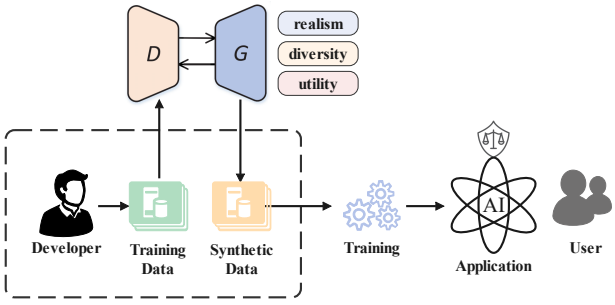


Fig. 2: The illustration of DMIG.

1)Train-GAN phase: In this phase, we need to get a high quality generator to generate labeled data to train a high quality classifier. A high quality conditional-GAN model $D - G$ will be trained on the sensitive labeled dataset D_{tr} with the objective function of a two-player minimax game:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim D_{\text{tr}}}[\log D(\boldsymbol{x} \mid \boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z} \mid \boldsymbol{y})))] \tag{5}$$

When the discriminator and the generator reach nash equilibrium, the GAN model converges. At this point, the discriminator cannot distinguish the authenticity of the data generated by the generator from the sensitive training data D_{tr}, which indicates that the distribution of the data generated by the generator is consistent with that of the original data.

In our scheme, we want the data generated by the trained generator to be able to train models with high classification accuracy. To achieve this goal, we need to choose a GAN that generates high quality data. For example, we choose Stylegan2-ada network structure for image data, which can train on small sample data and generate high-quality images, and which can provide us with the possibility to train models with high classification accuracy. In addition, the quantity of GAN training data will also affect the quality of generated data. Therefore, we need to consider the experimental setting of membership inference attack to optimize the training process of GAN. Specific optimization techniques will be described in detail in the next section. Therefore, in the GAN network training stage, we need to meet the original requirements of high GAN network generation quality, and at the same time to ensure that the model generating data training can achieve high classification accuracy.

2)Generation phase: In this phase, we need to get a new dataset $D_{\text{g-tr}}$ of the independent and identically distributed as the training dataset D_{tr} by generator G:

$$D_{\text{g-tr}} = G(\boldsymbol{z} \mid \boldsymbol{y}), \boldsymbol{z} \sim p_z \tag{6}$$

In order to verify the effectiveness of the defense scheme, we will generate different amounts of data to train the target model, and show that the model trained on the generated data can achieve the same classification accuracy as the original data training model. In this stage, different data generation model generated by means of data processing is consistent, and our goal is still to generate data can train a high classification accuracy of model. Therefore, for different data, the generation process we also have different optimization scheme, the concrete will be described in detail in the next section.

3)Training phase: In this phase, we will train a protected classification model using the generated data $D_{\text{g-tr}}$ obtained by the GAN. We can simply train the protected model on the $D_{\text{g-tr}}$ using the stochastic gradient descent (SGD) algorithm:

$$L_{D_{g-tr}}\left(f_{\theta_p}\right) = \underset{\{x_i,y_i\}_{i=1}^{N}\sim D_{g-tr}}{\mathbb{E}}\left[\ell\left(f_{\theta_p}(x_i), y_i\right)\right] \tag{7}$$

The goal of the machine learning model f_{θ_p} is to minimize the expected loss of D_{g-tr} :

$$\theta_p^* = \underset{\theta_p}{\operatorname{argmin}} L_{D_{\text{g-tr}}}\left(f_{\theta_p}\right) + \lambda\Omega(\theta_p) \tag{8}$$

where $\Omega(\theta_p)$ is a regularization term that penalizes model complexity and help avoid overfitting, and λ is a hyperparameter.

4.3 Details of the Generate technique

To verify the effectiveness of the scheme, image data and table data will be analyzed and compared respectively. For image data, we choose Stylegan2-ada to generate data, and for table data, we choose CTGAN to generate data. In order to improve the quality of the generated data, different optimization techniques are adopted in the generation process for different types of data.

For image data, in order to ensure the quality and diversity of the generated images and restore the dataset as much as possible, truncation technique is adopted in the training process. The prior distribution z is generally the standard normal distribution N(0, 1). We truncate the sampling of z by setting the threshold value, and the values beyond the range are resampled to fall into the range, $z \sim Z$.

$$Z = \{z|z \sim N(0,1) \cap -2 \leqslant z \leqslant 2\} \tag{9}$$

The decrease of the threshold value will improve the quality of the image, but also reduce the diversity of the image. In general, the decrease of the threshold value will lead to the continuous rise of IS value, but the FID value will decrease first and then rise, which means that the diversity of the generated image can be improved, but the fidelity will not always be improved. After repeated attempts, we finally determine a reasonable threshold value to ensure the utility of generated images while selecting appropriate diversity and fidelity.

The decrease of threshold will improve the quality of image and reduce the diversity of image. In general, a decrease in the threshold will lead to an increasing IS value, while the FID value will decrease and then increase, which means that the diversity of the generated images can be improved, but fidelity IS not always improved. After repeated attempts, we finally determined a reasonable threshold value, while choosing the right diversity and fidelity, to ensure the practicality of the generated image.

For table data, we process each table data into a high dimensional pair feature vector with only 0, 1 (this is a common practice). For example, for the Purchase dataset, we preprocess each data into a 600 dimensional feature vector with only 0, 1 (such as (0, 0, 1, 1, ..., 1, 0)). It is worth noting that after we have preprocessed the dataset, every column type is the same, so we do not need to specify that a column (or columns) is a discrete column when using CTGAN. This also improves the quality of CTGAN generation to some extent (as mentioned above, the difficulty of table generation is the existence of discrete columns and discrete columns are often unbalanced).

For the generation of Location dataset, label is used as a condition and CTGAN is input to generate data. This process is not different from most conditional GAN training. However, there is a problem with the generation of the Purchase data: the original Purchase data categories are obtained by kmeans clustering, and the generated data cannot be guaranteed to be in the same cluster center as each class of the original data. This causes the generated data to have a different label than the original data. To solve this problem, we used the following strategy to generate usable data:

(1) Firstly, we use k-means clustering to obtain the clustering center of each class for the original Purchase data. For example, on Purchase2 data, we use k-means clustering algorithm to obtain two clustering centers $C1$ and $C2$:

$$\{C1, C2\} = k - means(D_{tr}) \tag{10}$$

(2) Then, we train CTGAN with raw data D_{tr} and generate data D_{g-tr} with CTGAN. At this time, the dataset D_{g-tr} is unlabeled data. In order to ensure the availability of D_{g-tr}, we use the cluster center $\{C1, C2\}$ generated by the previous cluster to label the generated data. If it is closer to which cluster center, it will be marked as the corresponding label.

$$y_i = \underset{Ci}{min}(D(x_i, C1), D(x_i, C2))(x_i \sim D_{g-tr}) \tag{11}$$

In this way, we can ensure that the classification accuracy of the generated data training model after re-clustering is the same as that of the original data training model.

5 Evaluation

We present the evaluation of DMIG in this section. We first summarise the experimental settings for the evaluation. Subsequently, we present the evaluation results of DMIG with different attack methods. In the evaluation, we compare DMIG with another four state-of-the-art defences, namaly DP(differential privacy), MemGuard, Regularization, KD(knowledge distillation).

5.1 Experimental Setup

We use four datasets and corresponding model architectures that are consistent with the previous work [14]. For

image data, we use Staygan-ada to generate image data, and for table data, we use CTGAN to generate table data.

A. Data

MNIST. This is a dataset of 70, 000 handwritten digits formatted as 32 32 images and normalized so that the digits are located at the center of the image. We use 10,520 randomly selected images to train the target model.

CIFAR10. CIFAR10 is composed of 3232 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. CIFAR10 is a benchmark dataset used to evaluate image recognition algorithms [41]. We use the same amount of data, 10,520, as the work [9] to train the target model.

Purchase. The Purchase dataset is based on Kaggle's "acquire valued shoppers" dataset, which contains the shopping histories of thousands of people. The goal is to devise an accurate coupon promotion strategy. Each user record contains more than one year of transactions. These transactions include many fields, such as product name, chain, quantity, and date of Purchase. In our experiment, we derived a simplified Purchase dataset (197,324 records) as we did in [9], where each record consists of 600 binary features. Each feature corresponds to a product and indicates whether the user has purchased it. In the sorting task, records are first clustered into multiple classes, each representing a different buying style. In our experiment, we use five different classification tasks with a different number of categories 2, 20, 100. The task of classification is to predict the buying style of users given 600 feature vectors. We train the target model with 10,000 randomly selected records from the Purchase dataset.

Location. This dataset is based on Foursquare dataset, which contains locationcheck-inrecords of several thousand individuals. We obtain a Location dataset by processing the dataset in the same way as [9]. The dataset contains 5,010 data samples with with 446 binary features. Each feature corresponds to a certain region or location type and represents whether the individual has visited the region/location or not. All data samples are clustered into 30 classes representing different geosocial types. The classification task is to predict the geosocial type based on the 466 binary features. In our paper, we use 1,250 data samples to train a model.

Sizes of dataset splits. Table1 shows the division of these different datasets. For the image classification task, we use a raw training dataset D_{tr} of size 10520, a D_{tr} of size 10000 for the Purchase data, and a D_{tr} of size 1250 for the Location data. When training the protected model, for CIFAR10 data, we choose 50k generated data to construct $D_{\text{g-tr}}$; for MNIST data, we select 60k generated data to construct $D_{\text{g-tr}}$; for Purchase2 data, we chose 15k generated data to construct $D_{\text{g-tr}}$; for Purchase10,20,50,100 data, we selected 30k generated data to construct $D_{\text{g-tr}}$; for Location data, we chose 25k generated data to construct $D_{\text{g-tr}}$.

B. Model Architectures

Target Model. For the CIFAR10 and MNIST datasets, following previous works [14], we use two convolution layers plus one hidden layer as the target classifier. The convolution kernel is 32*32, and the number of neurons for the hidden layer is 64. For convolution layer, we use the popular activation function ReLU, and for hidden layer, we use activation function Tanh. The activation function in the output layer is softmax. All the training parameters are consistent with [14].

TABLE 1: For different data, the division of training set and test set, there is no intersection between them, and the amount of generated data used in the training of protected model.

Dataset	Original Data		Synthetic Data
	—D_{tr}—	—D_{te}—	—$D_{\text{g-tr}}$—
CIFAR10	10520	10520	50000
MNIST	10520	10520	60000
Purchase2	10000	10000	15000
Purchase20	10000	10000	30000
Purchase100	10000	10000	30000
Location	1250	1250	25000

For the Location and Purchase datasets, we use a fully-connected neural network with one hidden layer as the target classifier. The number of neurons for the layers is 64. We use activation function Tanh for the neurons in the hidden layers. The activation function in the output layer is softmax. All the training parameters are consistent with the previous works.

Generate Model. For image data, we use Stylegan2-ada model structure to generate data, we follow the original configuration of the authors'code[1]. For table data, we use CTGAN model structure to generate data, we follow the original configuration of the authors'code[2].

C. Evaluation Metrics

We use the following metrics to measure the utility, defense performance and efficiency of a defense method.

Classification Accuracy. It is measured on the training dataset D_{tr} or $D_{\text{g-tr}}$ and the test dataset D_{te} of the target classifier. It reflects how good the target classifier is on the classification task. To ensure the stability of the experimental results, we chose the average of the three tests as the final result.

Inference Accuracy. This is the classification accuracy of the attackers attack model in predicting the membership of input samples. It is measured on member dataset and non-member dataset.

5.2 Evaluate the quality of the generated data

In this section, we evaluate the quality of generated data by DMIG scheme on different datasets. Note that besides privacy-preserving, it is also crucial to ensure that the generated data is highly utility, so we evaluate the prediction accuracy of models trained on the DMIG's generated data and test their accuracy on real test dataset. Unlike existing data generation models, the quality of the "visualization" of the data generated by DMIG is not the primary goal of this article, and we will simply evaluate the visualization quality of the generated data for understanding. In addition, we evaluate the influence of the amount of generated data

1. https://github.com/NVlabs/stylegan2-ada
2. https://github.com/DAI-Lab/CTGAN

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

on the prediction accuracy of the training model, and select the appropriate amount of data for subsequent experimental comparison.

In order to compare with relevant articles, we choose the same data amount of target model training data to train GAN. For CIFAR10 and MNIST data, we use 10,520 data to train Stylegan2-ada; for Purchase and Location data, we use 10,000 data to train CTGAN. As shown in Figure3, you can see that the GAN we trained can produce data with the same utility as the original data, or even better than the original data. Of course, this requires us to use more multiples of data than the original data to achieve this effect. This phenomenon is also understandable, because there is more data, and the generated data features can better cover the features of the original data. For comparison, as shown in the data list in Table1, for CIFAR10, we choose to generate 50,000 data to train the protected model, whose test accuracy is 0.599, and the test accuracy of original data is 0.596. For MNIST, we choose to generate 60,000 data to train the protected model, and its test accuracy is 0.96, while the original data test accuracy is 0.985. For Purchase2, we choose to generate 15,000 data to train the protected model with test accuracy of 0.962 and original data with test accuracy of 0.958. For Purchase20, we choose to generate 30,000 data to train the protected model, and its test accuracy was 0.781, while the original data test accuracy was 0.761. For Purchase100, we selected to generate 30,000 data to train the protected model, and its test accuracy was 0.578, while the original data test accuracy was 0.592. With respect to Location data, we choose to generate 25,000 data to train the protected model, and the test accuracy was 0.651 and the test accuracy of original data is 0.638.

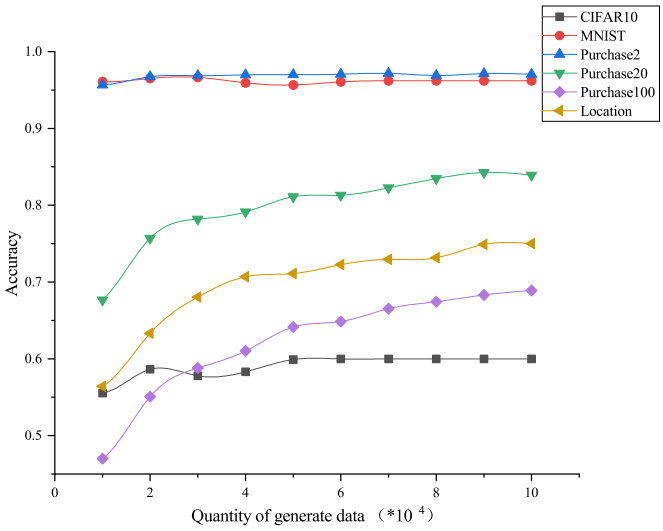


Fig. 3: The impact of the quantity of generated data on model utility.

5.3 Evaluation with Different Attacks

In this section, we use our defense scheme, DMIG, to defend different membership inference attacks, and compare the attack accuracy before and after defense to show the effectiveness of our scheme. We begin with a brief review of the ten MIA strategies we evaluated. Based on previous work, we use the accuracy of attack on the target model as a measure of their membership privacy.

Attack1 - A1 [14], the attacker chooses a shadow training dataset D_{s-tr} and a shadow test dataset D_{s-te} drawn from the same distribution as D_{tr}, where $|D_{s-tr}| = |D_{s-te}| = |D_{tr}|$ and there is no intersection between the three dataset, then trains a shadow model θ_s by using a shadow training dataset D_{s-tr}. Here, the shadow model is used to mimic the behavior of the target model. After this, the attacker computes predictions of θ_s on D_{s-tr} and D_{s-te}, labels the predictions of D_{s-tr} as members and that of D_{s-te} as non-members. Then, the adversary picks the 3 maximal posteriors of these predictions, again ordered from high to low, and trains binary attack model.

Attack2 - A2 [14], the strategy of the adversary is very similar to Attack1. The only difference is that the attacker chooses a shadow training dataset D_{s-tr} and a shadow test dataset D_{s-te} drawn from the different distribution as D_{tr}. For example, D_{tr} is CIFAR10 dataset, then D_{s-tr} and D_{s-te} are MNIST dataset. The shadow model here is not to mimic the target model's behavior, but only to summarize the member status of a data point in the training dataset of a machine learning model.

Attack3 - A3 [14], the membership inference attack uses a threshold on the prediction confidence, if the prediction confidence is greater than the threshold, it is considered as a member. The intuition is that the target model is trained by minimizing prediction loss over training data, which means the prediction confidence of a training sample should be close to 1. On the other hand, the model is usually less confident in predictions on a test sample. Thus, we can rely on the metric of prediction confidence for membership inference.

Attack4 - A4 [17], the strategy of the adversary is also similar to Attack1. The difference is that the attacker uses the disturbed version of the data to extract more subtle membership information, in which case the attacker only needs to obtain the label results output by the target model. The main idea of the attack is to extract fine-grained information about the classifier's decision boundary by combining multiple queries to the disturbed version data. By evaluating the robustness of the target model to the disturbed data with different inputs, the data with high robustness can be inferred as a member.

Attack5 - A5 [17], the attack described here aim to predict member using a point's distance to the model's decision boundary. Suppose $disth(x, y)$ represents the $L2$ distance from the point to the decision boundary, set a threshold τ, if $disth(x, y) < \tau$, we can infer (x, y) is a member of the training set.

Attack6 - A6 [17], this attack is a combination of A4 and A5, that will combine augmentations attack with decision boundary attack to evaluate robustness of many patterns.

Attack7 - A7 [42], the membership inference attack base on whether the input is classified correctly or not. The intuition is that the target model is trained to predict correctly on training data (members), which may not generalize well on test data (non-members). Thus, we can rely on the prediction correctness metric for membership inference.

Attack8 - A8 [15], the strategy of the adversary is very similar to Attack3. The only difference is that attacker sets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

different threshold values for different class labels. The reason is that the dataset may be unbalanced so that the target model indeed has different confidence levels for different class labels.

Attack9 - A9 [15], the strategy of the adversary uses prediction entropy for membership inference attack. The intuition is that the target model is trained by minimizing the prediction loss over training data, which means the prediction output of a training sample should be close to a one-hot encoded vector and its prediction entropy should be close to 0. On the other hand, the target model usually has a larger prediction entropy on an unseen test sample.

Attack10 - A10 [15], the strategy of the adversary is very similar to Attack9. Attack9 has one serious issue: it does not contain any information about the ground truth label. In fact, both a correct classification with probability of 1 and a totally wrong classification with probability of 1 lead to zero prediction entropy values. Therefore, the attack introduces ground truth label to modify calculation of prediction Entropy.

Now we use our defense scheme against different attacks to test the effectiveness of the scheme. During the experiment, A1, A3 and A7-A10 attacks use the same training data, test data, training data generation and test data generation, while A2 and A4-A6 attacks have slightly different, so we will describe the corresponding defense effects respectively.

As shown in the Table2, we test different data for A1, A3 and A6-A9 attacks respectively, and count the gap between the training accuracy and the test accuracy of target model, the test accuracy of target model and the accuracy of attack. First, we can see that when defense is not used, the gap between training accuracy and test accuracy is relatively large, so it will lead to higher attack accuracy. However, after using our defense scheme, the gap between the training accuracy and the test accuracy is almost zero, which indicates that the attacker cannot distinguish the training dataset from the test dataset at this time, and thus cannot judge the member and non-member information. In addition, we can see that the test accuracy of the target model basically does not change after using defense. On the contrary, the test accuracy is better after defense is used in some data. For example, for Purchase100, when the defense scheme is not used, the corresponding test accuracy is 0.581, and when defense is used, the test accuracy is 0.594, which is improved by 0.013.

For A2 attack, we choose table data to train target model and image data to train shadow model for analysis. As shown in the Table3, the rows of the table represent these attack datasets, and the columns of the table represent these datasets being attacked. As described above, the shadow model in A2 attack learns the member and non-member states of the data rather than imitating the behavior of the target model, so this attack strategy still has strong attack capability. However, our defense plan is also better able to resist this attack. As shown in the Table3, under our defense, the attack accuracy of image data to table data is about 0.5, and the attack strategy will also be invalid.

For A4-A6 attack, we reproduce the experimental results of reference [17] and verify whether the attack will be invalid with our defense strategy under the same settings. We selected 2,500 CIFAR10, 1,000 MNIST, 10,000 Purchase100 and 1,600 Location data respectively to train the target model. For the A4 attack, we analysis CIFAR10 and MNIST data, and we implement with two data enhancements, one is translation, with the parameter d=1 selected for comparison, and the other is rotation, whose corresponding parameter is set to r=4. For attack 5, we also implement two methods to find the decision boundary of the model: HopSkipJump and Random Noise. For the HopSkipJump method, we analyze it on CIFAR10 and MNIST data, and for the Random Noise method, we also compare it on Purchase-100 and Location. For A6, we combine the methods in A4 with the Random Noise method in A5. As [17] shows, A5 attack would be better than A4 attack. Then, we add our defense on this basis. As can be seen from the statistical data in the Table4, MDMG can well resist these three kinds of attacks.

TABLE 3: For different data, compare the performance of A2 attack without using DMIG and using DMIG. The columns represents the dataset being attacked, the rows represents the dataset used for training the shadow model.

W/O Defense				
Dataset	Purchase2	Purchase20	Purcahse100	Location
CIFAR10	0.5425	0.7403	0.8870	0.7716
MNIST	0.541	0.5549	0.5222	0.5592
W/ Defense				
Dataset	Purchase2	Purchase20	Purcahse100	Location
CIFAR10	**0.4994**	**0.4987**	**0.4952**	**0.4984**
MNIST	**0.5005**	**0.4929**	**0.4984**	**0.5024**

5.4 Comparison with Different Defences

In this part, we use different membership inference attack defense schemes to compare with our defense scheme, DMIG, and show that DMIG can achieve better privacy protection while ensuring the accuracy of the target model is not affected. In order to intuitively compare the defense effect of our defense scheme with other defense schemes, we compare A3, A6-A10 attacks in various data, because these attacks are executed under the same setting.

Comparison with DP-SGD. As has been reported in many papers [28]–[30], differential privacy is an effective method to defend against membership inference attack, which provides theoretical guarantee for protecting member privacy of individual samples. For each dataset, we will choose two different privacy budgets for experiments and compare with our scheme, one has a large privacy budget, the defense effect is general, but the utility damage is small, the other has a smaller loss budget, the defense effect is good, but the utility has obvious and serious damage.

For CIFAR10 data, we choose privacy budget $\varepsilon = 2.89$ and $\varepsilon = 25.41$ respectively for experiment. It can be seen that when $\varepsilon = 2.89$, the model's test accuracy is 0.439, while when $\varepsilon = 25.41$, the model's test accuracy is 0.503, that is, when we add more noise, the damage to the test accuracy of the target model is greater. The defense effect achieved by DP is relatively good for different attacks. When the privacy budget is smaller, the defense effect is stronger, and the attack effect is basically equal to random guess. Through experiments, we can see that DP does have a good

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

TABLE 2: For different data, compare the performance of A1, A3 and A7-A10 attacks without using DMIG defense scheme and using DMIG defense scheme.

Dataset	W/O Defense								W/ Defense							
	gap	test-acc	A1-acc	A3-acc	A7-acc	A8-acc	A9-acc	A10-acc	gap	test-acc	A1-acc	A3-acc	A7-acc	A8-acc	A9-acc	A10-acc
CIFAR10	0.394	0.606	0.8252	0.844	0.697	0.862	0.841	0.864	0.008	0.6	0.499	0.496	0.504	0.502	0.498	0.505
MNIST	0.015	0.985	0.5373	0.533	0.507	0.531	0.528	0.532	-0.007	0.965	0.5	0.495	0.496	0.499	0.5	0.502
Purchase2	0.043	0.957	0.5407	0.56	0.522	0.566	0.562	0.566	-0.002	0.961	0.5001	0.495	0.499	0.497	0.493	0.497
Purchase20	0.24	0.76	0.7534	0.775	0.62	0.791	0.779	0.792	0.003	0.783	0.5022	0.505	0.504	0.507	0.507	0.507
Purchase100	0.419	0.581	0.9098	0.931	0.709	0.932	0.927	0.933	-0.016	0.594	0.5	0.495	0.492	0.492	0.504	0.489
Location	0.362	0.638	0.896	0.907	0.678	0.906	0.894	0.907	-0.016	0.661	0.4984	0.502	0.492	0.496	0.484	0.49

TABLE 4: For different data, compare the performance of A4-A6 attacks without using DMIG defense scheme and using DMIG defense scheme.

W/O Defense						
Attack / Dataset		A4-acc		A5-acc		A6-acc
	gap	translation d=1	rotation r=4	HopSkipJump	RanDom	translation d=1&RanDom
CIFAR10	0.5024	0.75	0.7706	0.839	0.777	0.8032
MNIST	0.044	0.5195	0.509	0.554	0.523	0.5505
Purchase-100	0.353	N/A	N/A	N/A	0.8966	N/A
Location	0.38	N/A	N/A	N/A	0.9066	N/A

W/ Defense						
Attack / Dataset		A4-acc		A5-acc		A6-acc
	gap	translation d=1	rotation r=4	HopSkipJump	RanDom	translation d=1&RanDom
CIFAR10	0.003	0.5024	0.505	0.5088	0.51	0.503
MNIST	0.007	0.4955	0.505	0.4885	0.4985	0.497
Purchase-100	0.003	N/A	N/A	N/A	0.5368	N/A
Location	0.004	N/A	N/A	N/A	0.5	N/A

defense effect, but DP has a fatal defect. If we want to achieve a better defense effect, it will cause serious damage to the test accuracy of the model. As shown in the Table5, for CIFAR10 data, compared with DP, our defense effect is basically the same as that when $\varepsilon = 2.89$, but at this time, the test accuracy corresponding to DP is only 0.439, while our test accuracy is 0.6. Basically the same as the target model without defense. The same is true for other data. When the privacy budget is smaller, the test accuracy of the model will be lower and the defense ability will be stronger. However, when the privacy budget is too small, the accuracy of the model will be seriously damaged, and the model will no longer be available, so there will be no privacy at all.

Comparison with Memguard. The strategy of Memguard [20] is to make the distribution of entropy normalised to the confidence vector of members and non-members almost indistinguishable by adding noise to each confidence score vector predicted by the target model. This defense does not affect the accuracy of the model and can effectively reduce the success rate of membership inference attack to the level of random guesses.

As shown in the Tabel6, we compare the different attacks and data with Memguard defense method. The experimental results are consistent with the defense principle. Memguard defense does not affect the test accuracy of the model, so it does not affect the utility of the model. At the same time, compared with not using defense, Memguard can achieve a certain defense effect. For example, for A3 and A8-A10 attacks, the attack accuracy of CIFAR10 data decreased from 0.844, 0.862, 0.841 and 0.864 to 0.5, 0.697, 0.5 and 0.697, respectively. However, Memguard is invalid for attacks that rely on model output labels for judgment, because Memguard itself does not affect the prediction labels of the model. As shown in the experimental data of A7 attack, the defense effect of Memguard is invalid at this time. However, our defense is able to defend against all attacks. Fig4 describes Memguard defense performance compared to our scheme. It can be seen that DMIG can reduce the attack accuracy to about 0.5 for all attacks and related data, while the defense effect of Memguard against some attacks and data is not obvious. Especially for attacks A7, A8, A9, and the data CIFAR10, Purchase100, Location.

Comparison with regularization techniques. Regularization improves the generalization of ML models, and hence, reduce the MIA risk [9]. So, we compare DMIG with two regularization defeses, including dropout and spatial dropout. Dropout [43] is a regularization method that can be used to mitigate the degree of overfitting of models and thus protect against membership inference attacks. While, spatial dropout is a variant of dropout method by Tompson et al. [44] in the field of images. The normal dropout randomly sets some elements to zero, while the spatial dropout randomly sets some regions to zero. In our experiments, We set the random inactivation parameter of the two regularization methods to 0.5.

The results are shown in Table7. By observing the test

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

TABLE 5: Compare the defense performance with DP-SGD with different privacy budgets for different data under different attacks.

dataset	Defense	Privacy budget (ϵ)	training acc	test acc	attack acc				
					A3	A7	A8	A9	A10
CIFAR10	W/O	-	1	0.606	0.844	0.697	0.862	0.841	0.864
	Ours	-	0.608	0.6	0.496	0.504	0.502	0.498	0.505
	DP-SGD	2.89	0.477	0.439	0.505	0.519	0.52	0.501	0.519
		25.41	0.546	0.503	0.496	0.522	0.524	0.51	0.526
MNIST	W/O	-	1	0.985	0.533	0.507	0.531	0.528	0.532
	Ours	-	0.958	0.965	0.495	0.496	0.499	0.5	0.502
	DP-SGD	1.48	0.907	0.901	0.5	0.503	0.506	0.506	0.505
		8.93	0.964	0.951	0.5	0.506	0.503	0.505	0.503
Purchase2	W/O	-	1	0.957	0.56	0.522	0.566	0.562	0.566
	Ours	-	0.959	0.961	0.495	0.499	0.497	0.493	0.497
	DP-SGD	1.51	0.925	0.924	0.494	0.501	0.501	0.489	0.501
		4.27	0.949	0.944	0.499	0.503	0.503	0.5	0.503
Purchase20	W/O	-	1	0.824	0.706	0.588	0.721	0.707	0.721
	Ours	-	0.822	0.829	0.494	0.497	0.502	0.493	0.503
	DP-SGD	1.51	0.719	0.692	0.517	0.514	0.511	0.516	0.512
		17.72	0.8	0.744	0.522	0.528	0.536	0.528	0.538
Purchase100	W/O	-	1	0.581	0.931	0.709	0.932	0.927	0.933
	Ours	-	0.578	0.594	0.495	0.492	0.492	0.504	0.489
	DP-SGD	17.72	0.515	0.395	0.525	0.56	0.563	0.519	0.564
		27.03	0.733	0.567	0.558	0.583	0.581	0.545	0.584
Location	W/O	-	1	0.643	0.907	0.678	0.906	0.894	0.907
	Ours	-	0.645	0.661	0.502	0.492	0.496	0.484	0.49
	DP-SGD	8.27	0.678	0.437	0.534	0.621	0.609	0.533	0.617
		52.92	0.947	0.637	0.668	0.655	0.697	0.659	0.703

accuracy column in the table, it can be seen that for different data, the application of regularization techniques can improve the classification performance of the model. For CIFAR10 data, the test accuracy of the model was improved from 0.606 to 0.669 and 0.658 by using the dropout mechanism and spatial dropout mechanism, respectively. For different attacks, the corresponding attack accuracy decreased from 0.844, 0.697, 0.862, 0.841, 0.864 to 0.563, 0.608, 0.612, 0.559, 0.613 and 0.534, 0.583, 0.58, 0.532, 0.58, respectively. Therefore, for CIFAR10 data, the defense effect of spatial dropout is better than that of dropout. The reason for this is that the model using spatial dropout training has a smaller gap between the training accuracy and the test accuracy, so it is better at defending against MIAs. Compared with our defense effect, we can see that the defense effect of DMIG is far better than that of the two regularization techniques. Especially for Purchase100 and Location data, the dropout and spatial dropout mechanisms achieved significantly less defense than ours. For example, for A10 attack, on the Purchase100 and Location models without defense, the attack effects are high as 0.933 and 0.907, and reduced to 0.832 and 0.886 with dropout mechanism. It can be reduced to 0.832 and 0.874 with spatial dropout mechanism, while our defense scheme can be reduced to 0.489 and 0.49, and the test accuracy of our defense scheme is basically consistent with the results of dropout and spatial dropout mechanism. It can be seen that the defense advantage of DMIG is obviously stronger under the condition of ensuring the same test accuracy.

Comparison with DMP. Knowledge distillation [45] uses the output of large teacher model to train smaller model in order to transfer knowledge from large to small model. It allows smaller student model to have similar accuracy to their teacher model [46]. We can use knowledge distillation to distill a protected alternative model from the unprotected target model, thus resisting membership inference attacks. As described in the paper of DMP [24], the defense effect of DMP defense scheme will be better and better with the increase of T, but it will also cause more and more serious damage to the utility model. In order to compare with the DMP scheme, we just compare the attack performance of all data under A10 attack (for other attack schemes, DMP have the same defense effect), and at the same time, we test the influence of different temperatures on the DMP defense effect and the test accuracy of the reference model.

As shown in the Figure4, the influence of different data on the test accuracy of the reference model and the attack accuracy of A10 under different temperature(T) settings are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

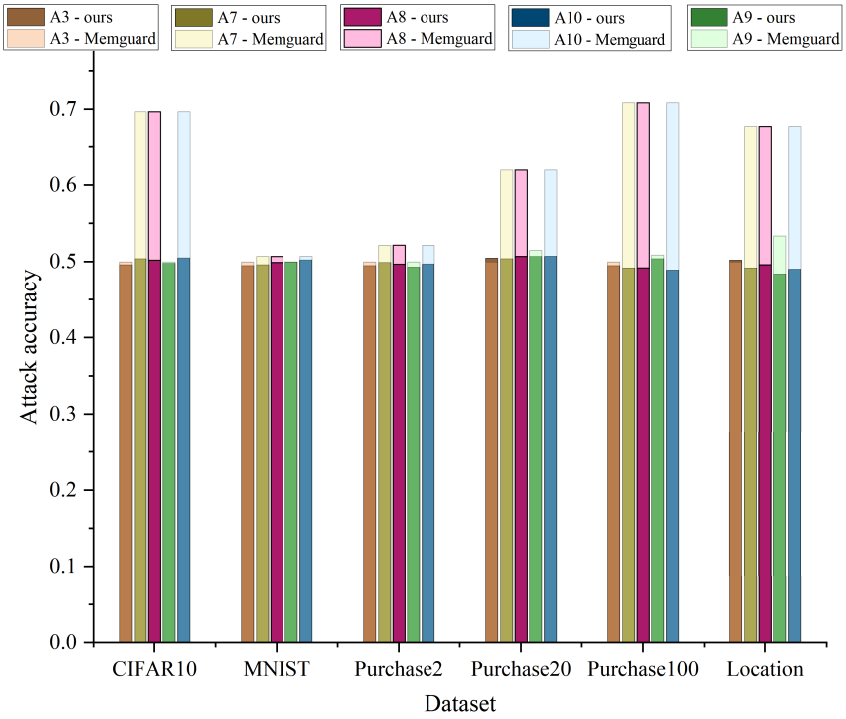


Fig. 4: Compare the defense performance with Memguard for different data and under different attacks. The dark color corresponds to the DMIG defense effect, and the light color corresponds to the Memguard defense effect.

TABLE 6: Compare the defense performance with Memguard for different data and under different attacks.

dataset	Defense	training acc	test acc	attack acc				
				A3	A7	A8	A9	A10
CIFAR10	W/O	1	0.606	0.844	0.697	0.862	0.841	0.864
	Ours	0.608	0.6	0.496	0.504	0.502	0.498	0.505
	Memguard	1	0.606	0.5	0.697	0.697	0.5	0.697
MNIST	W/O	1	0.985	0.533	0.507	0.531	0.528	0.532
	Ours	0.958	0.965	0.495	0.496	0.499	0.5	0.502
	Memguard	1	0.985	0.5	0.507	0.507	0.5	0.507
Purchase2	W/O	1	0.957	0.56	0.522	0.566	0.562	0.566
	Ours	0.959	0.961	0.495	0.499	0.497	0.493	0.497
	Memguard	1	0.957	0.5	0.522	0.522	0.5	0.522
Purchase20	W/O	1	0.824	0.706	0.588	0.721	0.707	0.721
	Ours	0.822	0.829	0.494	0.497	0.502	0.493	0.503
	Memguard	1	0.76	0.5	0.62	0.62	0.515	0.62
Purchase100	W/O	1	0.581	0.931	0.709	0.932	0.927	0.933
	Ours	0.578	0.594	0.495	0.492	0.492	0.504	0.489
	Memguard	1	0.581	0.5	0.709	0.709	0.509	0.709
Location	W/O	1	0.643	0.907	0.678	0.906	0.894	0.907
	Ours	0.645	0.661	0.502	0.492	0.496	0.484	0.49
	Memguard	1	0.643	0.5	0.678	0.678	0.534	0.678

TABLE 7: For different data and under different attacks, compare the defense performance with dropout and spatial dropout regularization technology.

dataset	Defense	training acc	test acc	attack acc				
				A3	A7	A8	A9	A10
CIFAR10	W/O	1	0.606	0.844	0.697	0.862	0.841	0.864
	Ours	0.608	0.6	0.496	0.504	0.502	0.498	0.505
	dropout	0.886	0.669	0.563	0.608	0.612	0.559	0.613
	spatial_dropout	0.824	0.658	0.534	0.583	0.58	0.532	0.58
MNIST	W/O	1	0.985	0.533	0.507	0.531	0.528	0.532
	Ours	0.958	0.965	0.495	0.496	0.499	0.5	0.502
	dropout	1	0.988	0.505	0.506	0.509	0.509	0.503
	spatial_dropout	0.999	0.988	0.503	0.505	0.504	0.499	0.504
Purchase2	W/O	1	0.957	0.56	0.522	0.566	0.562	0.566
	Ours	0.959	0.961	0.495	0.499	0.497	0.493	0.497
	dropout	0.999	0.952	0.523	0.523	0.535	0.526	0.535
	spatial_dropout	1	0.955	0.526	0.522	0.54	0.529	0.54
Purchase20	W/O	1	0.824	0.706	0.588	0.721	0.707	0.721
	Ours	0.822	0.829	0.494	0.497	0.502	0.493	0.503
	dropout	1	0.827	0.626	0.587	0.654	0.629	0.654
	spatial_dropout	1	0.772	0.683	0.614	0.705	0.684	0.705
Purchase100	W/O	1	0.581	0.931	0.709	0.932	0.927	0.933
	Ours	0.578	0.594	0.495	0.492	0.492	0.504	0.489
	dropout	1	0.615	0.808	0.692	0.833	0.809	0.832
	spatial_dropout	1	0.624	0.813	0.688	0.833	0.812	0.832
Location	W/O	1	0.643	0.907	0.678	0.906	0.894	0.907
	Ours	0.645	0.661	0.502	0.492	0.496	0.484	0.49
	dropout	1	0.649	0.883	0.685	0.883	0.884	0.886
	spatial_dropout	1	0.65	0.882	0.683	0.874	0.865	0.874

described. As described in [24], with the increase of temperature, the test accuracy of the reference model of DMP training will decrease and the corresponding defense effect will increase. This effect is more obvious when the over fitting degree of the original target model is more serious. In addition, as shown in the Table8, the defense effect of DMIG is significantly better than that of DMP. For example, according to CIFAR10 data, the attack accuracy of DMP scheme can be close to that of DMIG when the temperature reaches 9, but the test accuracy is 0.535, which is 0.065 less than that of DMIG. Other data are similar. As shown in the Table8, the data in blue represent the temperature and relevant data when the defense effect is similar to that achieved by DMIG. It can be seen that the trade-off between privacy and utility of DMIG is much better than that of DMP.

6 Conclusion

In this paper, we propose a new and effective member privacy protection scheme, DMIG. Compared with the existing methods, DMIG can not only experiment the best privacy protection, but also ensure the accuracy of model classification. In addition, in order to ensure the stability of GAN training process and the availability of generated data, we sample effective generation techniques for different types of data. Specifically, for the generation of image data, we use truncation techniques to ensure the availability of generated data. For table data, we save the clustering center of the original data with the help of clustering algorithm to update the label of the generated data, so as to ensure the quality of generated data. A large number of empirical experiments show that even with a limited privacy budget, the defense performance of DMIG is significantly better than existing DP defense schemes. We hope to explore the applicability of this model to more data types and explore other generative adversarial networks to achieve more efficient defense in the future.

Acknowledgments

This work was supported by National Natural Science Foundation of China for Joint Fund Project (No. U1936218) and National Natural Science Foundation of China (No.61802383, No.62102107, No.62072132, No.62002074).

References

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017. (document)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3174569, IEEE Transactions on Dependable and Secure Computing

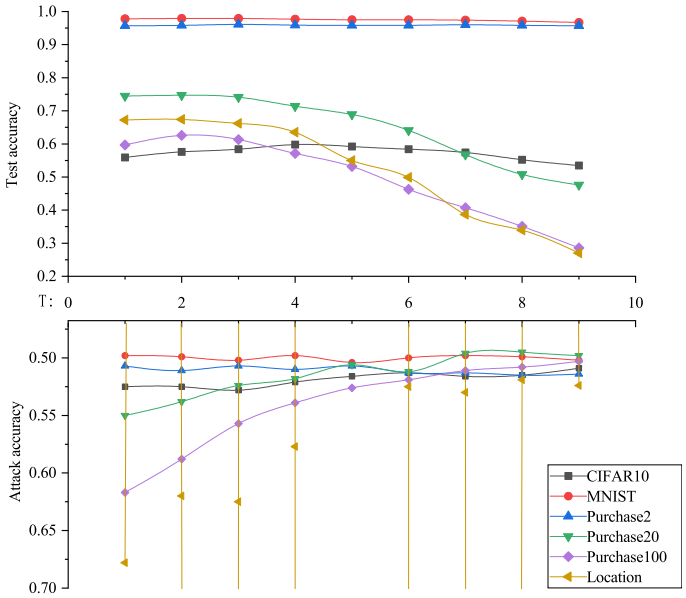


Fig. 5: The influence of different temperatures on model test accuracy and attack accuracy of DMP.

TABLE 8: Compare the defense performance with DMP under A10 attack for different data, "T" represents temperature, "tr-acc" represents training accuracy, "te-acc" represents test accuracy, and "at-acc" represents attack accuracy.

dataset	Defense	T	tr-acc	te-acc	at-acc A10
CIFAR10	W/O	-	1	0.606	0.844
	Ours	-	0.608	0.6	0.505
	DMP	1	0.608	0.559	0.525
		3	0.639	0.584	0.528
		5	0.629	0.592	0.516
		7	0.601	0.574	0.516
		9	0.557	0.535	0.509
MNIST	W/O	-	1	0.985	0.533
	Ours	-	0.958	0.965	0.502
	DMP	1	0.979	0.978	0.498
		3	0.982	0.979	0.502
		5	0.982	0.975	0.504
		7	0.976	0.974	0.498
		9	0.969	0.967	0.502
Purchase2	W/O	-	1	0.957	0.56
	Ours	-	0.959	0.961	0.497
	DMP	1	0.974	0.957	0.507
		3	0.985	0.961	0.507
		5	0.984	0.958	0.507
		7	0.985	0.96	0.513
		9	0.983	0.957	0.514
Purchase20	W/O	-	1	0.824	0.706
	Ours	-	0.822	0.829	0.503
	DMP	1	0.882	0.776	0.558
		3	0.839	0.756	0.547
		5	0.816	0.754	0.53
		7	0.776	0.739	0.517
		9	0.476	0.476	0.498
Purchase100	W/O	-	1	0.581	0.931
	Ours	-	0.578	0.594	0.489
	DMP	1	0.814	0.597	0.617
		3	0.721	0.613	0.557
		5	0.581	0.532	0.526
		7	0.438	0.407	0.511
		9	0.306	0.286	0.503
Location	W/O		1	0.643	0.907
	Ours	-	0.645	0.661	0.49
	DMP	1	0.941	0.672	0.678
		3	0.866	0.662	0.625
		5	0.667	0.55	568
		7	0.498	0.387	0.53
		9	0.318	0.27	0.524

[2] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE, 2018, pp. 471–478. (document)

[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019. (document)

[4] P. S. Kohli and S. Arora, "Application of machine learning in disease prediction," in *2018 4th International conference on computing communication and automation (ICCCA)*. IEEE, 2018, pp. 1–4. (document)

[5] J. Betancur, F. Commandeur, M. Motlagh, T. Sharir, A. J. Einstein, S. Bokhari, M. B. Fish, T. D. Ruddy, P. Kaufmann, A. J. Sinusas *et al.*, "Deep learning for prediction of obstructive disease from fast myocardial perfusion spect: a multicenter study," *JACC: Cardiovascular Imaging*, vol. 11, no. 11, pp. 1654–1663, 2018. (document)

[6] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333. (document)

[7] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 603–618. (document)

[8] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: Measuring unintended neural network memorization & extracting secrets," *arXiv preprint arXiv:1802.08232*, vol. 5, 2018. (document)

[9] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18. (document), 2.2, 3.1, 5.1, 5.4

[10] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 601–618. (document)

[11] C.-L. Chi, W. N. Street, J. G. Robinson, and M. A. Crawford, "Individualized patient-centered lifestyle recommendations: An expert system for communicating patient specific cardiovascular risk information and prioritizing lifestyle options," *Journal of Biomedical Informatics*, vol. 45, no. 6, pp. 1164–1174, 2012. (document)

[12] I. W. P. Consortium, "Estimation of the warfarin dose with clinical and pharmacogenetic data," *New England Journal of Medicine*, vol. 360, no. 8, pp. 753–764, 2009. (document)

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708. (document)

[14] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018. (document), 3.1, 5.1, 5.1, 5.3

[15] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021. (document), 3.1, 5.3

[16] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018, pp. 268–282. (document), 3.1

[17] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1964–1974. (document), 3.1, 5.3, 5.3

[18] Z. Li and Y. Zhang, "Label-leaks: Membership inference attack with label," *arXiv e-prints*, pp. arXiv–2007, 2020. (document), 3.1

[19] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753. (document)

[20] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 259–274. (document), 3.2, 5.4

[21] M. Xue, C. Yuan, C. He, Z. Wu, Y. Zhang, Z. Liu, and W. Liu, "Use the spear as a shield: A novel adversarial example based privacy-preserving technique against membership inference attacks," *arXiv preprint arXiv:2011.13696*, 2020. (document), 3.2

[22] Y. Kaya, S. Hong, and T. Dumitras, "On the effectiveness of regularization against membership inference attacks," *arXiv preprint arXiv:2006.05336*, 2020. (document), 3.2

[23] H. Liu, J. Jia, W. Qu, and N. Z. Gong, "Encodermi: Membership inference against pre-trained encoders in contrastive learning," *arXiv preprint arXiv:2108.11023*, 2021. (document)

[24] V. Shejwalkar and A. Houmansadr, "Membership privacy for machine learning models through knowledge transfer," *arXiv preprint arXiv:1906.06589*, 2019. (document), 3.2, 5.4

[25] J. Zheng, Y. Cao, and H. Wang, "Resisting membership inference attacks through knowledge distillation," *Neurocomputing*, vol. 452, pp. 114–126, 2021. (document), 3.2

[26] H. Huang, W. Luo, G. Zeng, J. Weng, Y. Zhang, and A. Yang, "Damia: Leveraging domain adaptation as a defense against membership inference attacks," *IEEE Transactions on Dependable and Secure Computing*, 2021. (document), 3.2

[27] H. Huang, "Defense against membership inference attack applying domain adaptation with addictive noise," *Journal of Computer and Communications*, vol. 9, no. 5, pp. 92–108, 2021. (document), 3.2

[28] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318. (document), 3.2, 5.4

[29] S. Rahimian, T. Orekondy, and M. Fritz, "Differential privacy defenses and sampling attacks for membership inference," in *PriML Workshop (PriML)*, vol. 13, 2019. (document), 3.2, 5.4

[30] J. Chen, W. H. Wang, and X. Shi, "Differential privacy protection against membership inference attack on machine learning for genomic data," in *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*. World Scientific, 2020, pp. 26–37. (document), 3.2, 5.4

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. (document)

[32] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, pp. 1–40, 2016.

[33] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016. 3.2

[34] F. Yu-Xiang, Y. B. Qin, and G. W. Shen, "Multi-source data privacy protection based on transfer learning," *Computer Engineering & ence*, 2019. 3.2

[35] Y. Fu, Y.Qin, and G. Shen, "Sensitive data privacy protection method based on transfer learning," *Journal of Data Acquisition and Processing*, 2019. 3.2

[36] J. Wang, W. Bao, L. Sun, X. Zhu, B. Cao, and S. Y. Philip, "Private model compression via knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1190–1197. 3.2

[37] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410. 3.3

[38] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119. 3.3

[39] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *arXiv preprint arXiv:2006.06676*, 2020. 3.3

[40] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *arXiv preprint arXiv:1907.00503*, 2019. 3.3

[41] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009. 5.1

[42] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1605–1622. 5.3

[43] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012. 5.4

[44] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656. 5.4

[45] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. 5.4

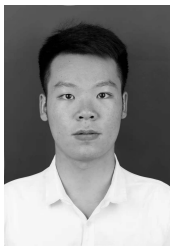
[46] E. J. Crowley, G. Gray, and A. J. Storkey, "Moonshine: Distilling with cheap convolutions." in *NeurIPS*, 2018, pp. 2893–2903. 5.4

Li Hu is a PhD student at Guangzhou University, China. She received her Master degree from Guangzhou University. Her research interests include cryptography and privacy in artificial intelligence.

Jin Li received the BS degree in mathematics from Southwest University, 2002, the MS degree in mathematics from $SunYatsen$ University, 2004, and the PhD degree in information security from Sun Yat-sen University, 2007. He is currently a professor and vice dean of School of Computer Science, Guangzhou University. His research interests include design of secure protocols in cloud computing and cryptographic protocols. He has published more than 100 papers in international conferences and journals, including IEEE INFOCOM, IEEE TIFS, IEEE TPDS, IEEE TOC and ESORICS etc. His work has been cited more than 10000 times at Google Scholar and the H-Index is 34. He also served as program chairs and committee for many international conferences. He received NSFC Outstanding Youth Foundation in 2017.

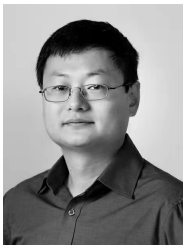
Guanbiao Lin is currently a graduate student at Guangzhou University in China. His research interests include artificial intelligence security and differential privacy.

Shiyu Peng is now a graduate student at Guangzhou University. His research interest is artificial intelligence security.



Zhenxin Zhang is now a graduate student at Guangzhou University. His research interest is artificial intelligence security.

Yingying Zhang is currently a master student at Guangzhou University. Her research area is artificial intelligence security.



Changyu Dong received the PhD degree from Imperial College London. He is currently a senior lecturer with the School of Computing, Newcastle University. He has authored more than 30 publications in international journals and conferences. His research interests include applied cryptography, trust management, data privacy, and security policies. His recent work focuses mostly on designing practical secure computation protocols. The application domains include secure cloud computing and privacy preserving data mining.