Rishav Chourasia, Batnyam Enkhtaivan, Kunihiro Ito, Junki Mori, Isamu Teranishi, and Hikaru Tsuchida

Knowledge Cross-Distillation for Membership Privacy

Abstract: A *membership inference attack (MIA)* poses privacy risks for the training data of a machine learning model. With an MIA, an attacker guesses if the target data are a member of the training dataset. The state-of-the-art defense against MIAs, distillation for membership privacy (DMP), requires not only private data for protection but a large amount of unlabeled public data. However, in certain privacy-sensitive domains, such as medicine and finance, the availability of public data is not guaranteed. Moreover, a trivial method for generating public data by using generative adversarial networks significantly decreases the model accuracy, as reported by the authors of DMP. To overcome this problem, we propose a novel defense against MIAs that uses knowledge distillation without requiring public data. Our experiments show that the privacy protection and accuracy of our defense are comparable to those of DMP for the benchmark tabular datasets used in MIA research, Purchase100 and Texas100, and our defense has a much better privacy-utility trade-off than those of the existing defenses that also do not use public data for the image dataset CIFAR10.

Keywords: privacy-preserving machine learning, membership inference attacks, knowledge distillation

DOI 10.2478/popets-2022-0050
Received 2021-08-31; revised 2021-12-15; accepted 2021-12-16.

Rishav Chourasia: National University of Singapore, E-mail: e0427764@u.nus.edu
Batnyam Enkhtaivan: NEC Corporation, E-mail: b-enkhtaivan@nec.com
Kunihiro Ito: NEC Corporation, E-mail: kunihiro.ito.220@nec.com
Junki Mori: NEC Corporation, E-mail: junki.mori@nec.com
Isamu Teranishi: NEC Corporation, E-mail: teranisi@nec.com
Hikaru Tsuchida: NEC Corporation, E-mail: h_tsuchida@nec.com

1 Introduction

1.1 Background

Machine learning (ML) has been extensively used in various aspects of society [6]. We have seen great improvements in areas such as image recognition and natural language processing.

However, in the recent years, it has been reported that the privacy of the training data can be significantly undermined by analyzing ML models. Since, in most applications, privacy-sensitive data are used as the training data for the models, protecting the privacy of the training data is crucial for getting approval from data providers or essentially society.

Following the growing concern for privacy in society worldwide, many countries and regions are introducing regulations for data protection, e.g., the General Data Protection Regulation (GDPR) [10], California Consumer Privacy Act (CCPA) [2], and Health Insurance Portability and Accountability Act (HIPAA) [13]. Moreover, guidelines and regulations designed specifically for trustworthiness in artificial intelligence (AI) and ML are under discussion [9].

Membership Inference Attacks: One of the most fundamental attacks against the privacy of a ML model is the *membership inference attack (MIA)* [5, 15, 23, 24, 30, 31, 34–36, 43, 44], where an attacker guesses whether the given target data is in the training data of a ML model.

MIAs are dangerous because they reveal the information of individual pieces of data rather than the trend of the whole population of training data. For instance, consider an ML model for inferring a reaction to some drug from a cancer patient's morphological data. An MIA attacker who knows the victim's data and has access rights to the ML model can know whether the victim has cancer or not, although the victim's data itself do not directly contain this information.

The authors are alphabetically ordered.

Another reason that MIAs are dangerous is that they can be executed through legitimate access to ML models only, meaning that they cannot be prevented by the conventional security methods such as data encryption and access control [34].

Defense against MIAs: The current state-of-the-art defense against MIAs is *Distillation for Membership Privacy (DMP)* [33]. It can protect even against various state-of-the-art MIA attacks [5, 24, 35], which the previous defenses [17, 23] cannot protect against very well, and its success comes from the "semi-supervised assumption" that a defender can obtain public unlabeled data. Specifically, DMP exploits a knowledge transfer technique [12]; a defender trains an ML model using their own private data, feeds public data to the ML model to obtain the outputs of them, and trains another ML model using the public data and the corresponding outputs. Such indirect usage of private data makes knowledge distillation-based methods highly effective in protecting the privacy of private data.

However, in many domains of ML applications, public data are scarce due to the sensitive nature of the data, e.g., financial and medical data. To overcome this, utilization of synthetic data is proposed [33] as well. However, this method decreases accuracy [33] due to the decrease in data quality.

1.2 Our Contributions

In this paper, we propose a novel knowledge distillation-based defense that uses only private data for model training.

Our contributions are as follows.

- We propose a novel MIA defense called *knowledge cross-distillation (KCD)*[1]. Unlike the state-of-the-art defense, DMP, it does not require any public or synthetic reference data to protect ML models. Hence, KCD allows us to protect the privacy of ML models in areas where public reference data are scarce.
- For the benchmark tabular datasets used in MIA research, Purchase100 and Texas100, we empirically show that the privacy protection and accuracy of KCD are comparable to those of DMP even though KCD does not require public or synthetic data, unlike DMP.
- For the image dataset CIFAR10, we empirically show that the accuracy of KCD is comparable to that of DMP, and KCD provides a much better privacy-utility trade-off than those of other defenses that do not require public or synthetic reference data.

1.3 Other Related Works

We focus only on related works that are directly related to our contributions. See Hu et al. [15] for a comprehensive survey of MIAs.

Membership Inference Attacks: One of the earliest works considering MIAs is by Homer et al. [14], and MIAs were introduced in the ML setting in a seminal work by Shokri et al. [34]. A series of MIA attacks, which is now called the *neural network-based attack*, was proposed by Shokri et al. [34] and was studied in detail by Salem et al. [31] and Truex et al. [42]. Later, a new type of MIA attack, the *metric-based attack*, was proposed by Yeom et al. [44] and studied by Song et al. [36], Salem et al. [31], and Leino et al. [19]. Then, Song et al. [35] summarized and improved upon them and proposed the state-of-the art metric-based attack as well.

Choo et al. [5] and Li et al. [20] independently and concurrently succeeded in attacking neural networks in a *label-only setting*, where an attacker can get only labels as outputs of a target neural network, while the attackers of other known papers require confidence scores as the outputs of it. Nasr et al. [24] proposed an MIA attack in a *white-box setting*, where an attacker can obtain the structure and parameters of the target neural network.

Known Defenses: MIAs can be mitigated using one known method, *differential privacy* [7, 8], which is a technique for guaranteeing worst-case privacy by adding noise to the learning objective or model outputs. However, defenses designed to protect against MIAs specifically have better privacy-utility trade-offs. Three MIA-specific defenses were proposed: AdvReg by Nasr et al. [23], MemGuard by Jia et al. [17], and DMP [33].

An important technique for protecting MLs against MIAs is knowledge transfer [12]. Using this technique, PATE by Papernot et al. [26, 27] achieved DP, Cronos [3] by Chang et al. protected ML from an MIA in a federated learning setting, and DMP [33] achieved a higher

[1] After we submitted our work to PETS 2022 Issue 2, Tang et al. [39] published a concurrent and independent work similar to ours in arXiv.

privacy-utility trade-off by removing public data with low entropy.

Currently, DMP is the best defense in the sense of the privacy-utility trade-off. However, it requires public data. Other known defenses, AdvReg and MemGuard, have an advantage in that they do not require public reference data.

2 Preliminaries

2.1 Machine Learning

An ML model for a classification task is a function F parameterized by internal *model parameters*. It takes a d-dimensional real-valued vector $x \in \mathbb{R}^d$ as input and outputs a c-dimensional real-valued vector $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_c)$. The output $\hat{y}$ has to satisfy $\hat{y}_i \in [0, 1]$ and $\sum_i \hat{y}_i = 1$. Each $\hat{y}_i$ is called a *confidence score*. Its intuitive meaning is the likelihood of x belonging to class i. The $\underset{i}{\operatorname{argmax}}\, \hat{y}_i$ is called a *predicted label* (or *predicted class*).

An ML model F is trained using a *training dataset* $D \subset \{(x, y) \mid x \in \mathbb{R}^d,\ y \in \{0, 1\}^c\}$, where x is a data point, and y is a one-hot vector reflecting the true class label of x. In the training procedure, the model parameters of F are iteratively updated to reduce the predetermined *loss* $\sum_{(x,y)\in D} L(F(x), y)$, which is the sum of errors between the prediction $F(x)$ and true label y. For inference, F takes input x and outputs $\hat{y} = F(x)$ as a *prediction*.

The *accuracy* of F for dataset D is the ratio between the number of elements $(x, y) \in D$ satisfying $\underset{i}{\operatorname{argmax}}\, F(x)_i = \underset{i}{\operatorname{argmax}}\, y_i$. Here, $F(x)_i$ and y_i are the i-th component of $\hat{y} = F(x)$ and y, respectively. The *training accuracy* and *testing accuracy* of F are for the training and testing datasets, respectively. Here, *testing dataset* is a dataset that does not overlap with the training dataset. The *generalization gap* of F is the difference between training and testing accuracies.

2.2 Membership Inference Attack (MIA)

MIA is an attack in which an attacker attempts to determine whether given data (called *target data*) are used for training a given ML model (called a *target model*). In the discussion of MIAs, the training data of the target model are called *member data*, and non-training data are called *non-member data.*

There are two types of MIAs, *white-box* and *black-box* [24, 34]. Attackers of the former can take as input the model structure and model parameters of the target model. Attackers of the latter do not take them as input but are allowed to make queries to the target model and obtain answers any number of times. A black-box MIA can be divided into the two sub-types, *MIA with confidence scores* and *label-only MIA* [5]. Attackers of the former can obtain confidence scores as answers from the target model but attackers of the latter can obtain only predicted labels as answers.

In all types of MIAs, the attackers can take the target data and *prior knowledge* as inputs. Intuitively, the prior knowledge is what attackers know in advance. What type of prior knowledge an adversary can obtain depends on the assumed threat model. An example of prior knowledge is a dataset sampled from the same distribution as the training data of the target model, not overlapping with the training data. Another example is a portion of the training data. The prior knowledge we focused in this study is described in Section 4.3. The *attack accuracy* of an attacker for an MIA is the probability that they will succeed in inferring whether target data are member data. As in the all previous papers, the target data are taken from member data with a probability of 50%.

One of the main factor causing MIA risks is *overffitting* of an ML model on the training (=member) data. The member data can be distinguished from non-member data [31, 44] depending on whether it is overfitted to the target model, e.g. by checking whether the highest confidence score of output of the target model is more than a given threshold.

2.3 Distillation for Membership Privacy (DMP)

DMP [33] is a state-of-the-art defense method against MIAs that leverages knowledge distillation [12]. Distillation was originally introduced as a model compression technique that transfers the knowledge of a large *teacher model* to a small *student model* by using the output of a teacher model obtained on unlabeled *reference dataset.* DMP needs public reference dataset R disjoint from the training dataset D to train ML models with membership privacy.

The training algorithm of DMP is given in Algorithm 1. Here, L is the loss function. First, DMP trains a teacher model F using a private training dataset D (Step 1). F is overfitted to D and therefore vulnerable to MIA. Next, DMP computes the *soft labels* $F(x)$ of each peice of data x of public reference dataset R and lets $\bar{R}$ be the set of $(x, F(x))$ (Step 2). Finally, to obtain a protected model, DMP trains a student model H using the dataset $\bar{R}$ (Step 3). H has MIA resistance because it is trained without direct access to the private D. Note that DMP uses H with the same architecture as F. The authors of DMP [33] proposed three different ways of achieving the desired privacy-utility tradeoffs:

- increasing the temperature of the softmax layer of F,
- removing reference data with high entropy predictions from F,
- decreasing the size of the reference dataset.

All of the above changes reduce MIA risks but also the accuracy of H and vice versa. When we use the second or third way to tune DMP, we select samples from the reference dataset and use them as R in Step 2.

Algorithm 1 Training algorithm of DMP

```
Input: training dataset $D \subset \{(x, y) \mid x \in \mathbb{R}^d, y \in \{0,1\}^c\}$, reference dataset $R \subset \{x \mid x \in \mathbb{R}^d\}$, and initialized parameters of $F, H$.
Output: Distilled student model $H$.
1: Train $F$ by using $D$ as a training dataset until the training converges to minimize the loss
       $\sum_{(x,y)\in D} L(F(x), y)$.
2: Let $\bar{R}$ be a dataset $R$ with soft labels,
       $\bar{R} = \{(x, F(x)) \mid x \in R\}$.
3: Train $H$ by using a dataset $\bar{R}$ until the training converges to minimize the loss
       $\sum_{(x,y')\in \bar{R}} L(H(x), y')$.
4: Return $H$.
```

3 Our Proposed Defense

In this section, we propose a new defense that can protect ML from MIAs without using a reference dataset.

3.1 Idea

The starting point with our approach is DMP [33]. That is, we train a *teacher model* F using a training dataset D, compute *soft labels* $F(x)$ to x of public *reference dataset* R, train a *student model* H using $(x, F(x))$, and, finally, use H for inference. DMP can mitigate MIAs as described in Section 2.3.

The problem with DMP is that it requires a public reference dataset, which may be difficult to collect in privacy-sensitive domains [33]. A naïve idea to solve this problem is to use the original D as a reference dataset. However, our experiment shows that this approach does not sufficiently mitigate the MIA risk (see Section 6.1). The main problem of the naïve idea is that data x of the reference dataset $R = D$ is *member data* of F. Therefore, F results in overfitting on x and the confidence score $\hat{y} = F(x)$ is close to the one-hot vector y of the true label. Hence, H trained on $(x, \hat{y})$ results again in overfitting on x, which can be exploited by an MIA.

Our proposed defense, denoted by *knowledge cross-distillation* (KCD) is designed to overcome the above problem. We divide the training dataset into n parts, leave one part as a reference dataset, and train a teacher F_1 using the remaining parts. To increase the accuracy of KCD, we prepare teachers $F_2, \ldots, F_n$ as well and repeat the above procedure for each teacher by changing the reference part. Finally, we use each reference part to distill the knowledge of each corresponding teacher into a single H. Our defense solves the problem of the naïve idea because none of the remaining parts of the training dataset are used to train the teacher model.

3.2 Description

The training algorithm of the our proposed defense, KCD, is given in Algorithm 2 and is overviewed in Figure 1. Here, $F_1, \ldots, F_n$, and H are models with the same structure as that of the model F that we want to protect[2]. L is the loss function.

In Algorithm 2, we divide training dataset D into n disjoint subsets $D_1, \ldots, D_n$ with almost the same size, such that $D = \bigsqcup_{i=1}^{n} D_i$ holds[3] (Step 1). Then, for $i = 1, \ldots, n$, we train the teacher model F_i using the dataset

[2] Although we use the term "distillation," we use teacher and student models with the same structure as in DMP [33]. This is because we are not concerned about the size of the resulting model.

[3] $\bigsqcup_{i=1}^{n} D_i$ denotes a disjoint union of sets

D but exclude D_i (Step 2-4). Let $\bar{D}_i$ be the dataset that is obtained by adding soft labels $F_i(x)$ to $(x, y) \in D_i$ (Step 5). Finally, we train a student model H using the dataset $\bigcup_i \bar{D}_i$ to minimize the combined loss function with hyperparameter α (Step 6).

Our loss function comprises two terms; the first term is the loss for soft labels y', and the second is the loss for the true label y. The hyperparameter α can tune the privacy-utility trade-off of KCD. In fact, if $\alpha = 1$, our defense protects the privacy of the training data due to the reason mentioned in Section 3.1. If $\alpha = 0$, KCD becomes the same as the unprotected ML.

Note that our privacy-utility trade-off based on α cannot be directly applied to the known knowledge distillation-based defenses, DMP [33] and Cronus [3], because the public reference datasets for these defenses do not have the true labels and loss for the predicted scores and true labels cannot be computed.

```
Algorithm 2 Training algorithm of KCD
Input: training dataset $D \subset \{(x, y) \mid x \in \mathbb{R}^d, y \in \{0,1\}^c\}$, hyperparameter $\alpha \in [0,1]$, and initialized parameters of $F_1, \ldots, F_n, H$.
Output: Distilled student model $H$.
1: Divide $D$ into $n$ randomly selected disjoint subsets $\{D_i\}_{i=1}^n$ with almost the same size, such that[3]
       $$D = \bigsqcup_{i=1}^{n} D_i,$$
2: for $i = 1, \ldots, n$ do
3:     Train $F_i$ by using $D \setminus D_i$ as a training dataset until the training converges to minimize the loss
           $$\sum_{(x,y) \in D \setminus D_i} L(F_i(x), y).$$
4: end for
5: Let $\bar{D}_i$ be a dataset $D_i$ with soft label
       $$\bar{D}_i = \{(x, F_i(x)) \mid \exists y \ : \ (x, y) \in D_i\},$$
   and let $\bar{D} = \cup_i \bar{D}_i$.
6: Train $H$ by using a dataset $\bar{D}$ until the training converges to minimize the loss
       $$\alpha \sum_{(x,y') \in \bar{D}} L(H(x), y') + (1 - \alpha) \sum_{(x,y) \in D} L(H(x), y) \quad (1)$$
7: Return $H$.
```

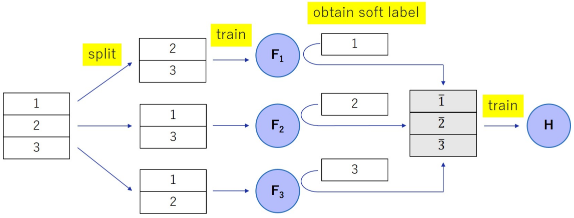

Fig. 1. Outline of KCD when dividing training dataset into three subsets. F_1-F_3: teacher models, H: student model.

4 Experimental Setup

We conducted our experiments using the following datasets and model architectures as in the previous studies [5, 17, 23, 24, 33–35].

4.1 Datasets

CIFAR 10: This is a typical benchmark dataset used for evaluating the performance of image-classification algorithms [18]. It contains $60,000$ RGB images. Each image is composed of 32×32 pixels and labeled in one of 10 classes.

Purchase100: This is a benchmark dataset used for MIAs. It is based on a dataset provided by Kaggle's Acquire Valued Shoppers Challenge [28]. We used a processed and simplified one by Shokri et al. [34]. The dataset has $197,324$ records with 600 binary features, each of which represents whether the corresponding customer purchased an item. The data are clustered into 100 classes representing different purchase styles, and the classification task is to predict which one of the 100 classes an input is in.

Texas100: This is also a benchmark dataset used for MIAs. It is based on the hospital discharge data [40] from several health facilities published by the Texas Department of State Health Services and was processed and simplified by Shokri et al. [34]. It contains the 100 most frequent procedures that patients underwent. The dataset has $67,330$ records with $6,170$ binary features of patients, such as the corresponding patient's symptoms and genetic information. The classification task is to predict which one of the 100 procedures a patient for a piece of an input data underwent.

4.2 Model Architectures

Wide ResNet-28: For CIFAR 10, we used the same model architecture as in a previous study [5], i.e., Wide ResNet-28.

Purchase and Texas Classifiers: For Purchase 100 and Texas 100, we used fully connected NNs with Tanh activation functions. We used the same layer sizes as in a previous study [23], i.e., layer sizes $(1024, 512, 256, 128)$.

4.3 Setting of MIA

As in the previous studies of MIAs [23, 24], we consider a strong setting where the attackers know the non-member dataset and a subset of the member dataset of the target model as prior knowledge. (This subset of the member dataset does not contain the target data, of course). This setting is called *supervised inference* [24].

One may think that the supervised inference setting seems too strong as a real setting. However, the *shadow model* technique [34] allows an attacker to achieve supervised inference virtually [24]. A shadow model is an ML model that is trained by an attacker to mimic a target ML model. The attacker then knows the training data of the shadow model as in the supervised inference setting since the attacker trains it.

4.4 MIAs for Evaluations

We conducted comprehensive experiments for three types of MIA: black-box MIA with confidence score, black-box MIA with only labels, and white-box MIA.

4.4.1 Black-Box MIA with Confidence Score (BB w/Score)

These are attacks such that the attackers know the confidences scores as outputs of the target model. There are two sub-types of these attacks.

NN-based Attack: This is a type of black-box MIA using an NN, called *attack classifier A*. Specifically, the attacker knows a set of non-member data and a subset of member data as their prior knowledge, as mentioned in Section 4.3. They send these data to the target model and obtain their confidence scores as answers. Using these data, the answers, and the knowledge of whether these data are members, they train A. Finally, they infer the membership status of the target data by taking their label and confidence score as input to A. There are two known NN-based attacks [31, 34]. The difference between them is whether the attacker trains an attack classifier for each label class; the original attack by Shokri et al. [34] uses one classifier per each class and a simplified attack by Salem et al. [31], called *ML Leaks Adversary 1*, uses only one common attack classifier for all classes.

Name	Condition
Top 1	$\mathbf{argmax}_i\, \boldsymbol{F(x)_i} \overset{?}{\geq} \boldsymbol{\tau}$
Correctness	$\mathbf{argmax}_i\, \boldsymbol{F(x)_i} \overset{?}{=} \mathbf{argmax}_i\, \boldsymbol{y_i}$
Confidence	$\boldsymbol{F(x)_{\ell[y]}} \overset{?}{\geq} \boldsymbol{\tau_{\ell[y]}}$
Entropy	$\boldsymbol{-\sum_i F(x)_i \log F(x)_i} \overset{?}{\leq} \boldsymbol{\tau_{\ell[y]}}$
Modified Entropy	$\boldsymbol{-(1 - F(x)_{\ell[y]}) \log F(x)_{\ell[y]}}$ $\boldsymbol{-\sum_{i \neq \ell[y]} F(x)_i \log F(x)_i} \overset{?}{\leq} \boldsymbol{\tau_{\ell[y]}}$

Table 1. Known metric-based attacks. Here, $F(x)_i$ and y_i mean i-th component of $F(x)$ and y respectively and $\ell[y]$ is the label corresponding to one-hot vector y, that is, $\mathrm{argmax}_i\, y_i$. τ and $\tau_{\ell[y]}$ are thresholds determined by attackers.

In our experiments, we executed the attack *ML Leaks Adversary 1* [31] since it is simpler and “has very similar membership inference” [31] to that of Shokri et al. [34].

Metric-based Attack: This is a type of black-box MIA that directly uses the fact that the confidence score $F(x)$ of the target data (x, y) differs depending on whether (x, y) is a member. Specifically, an attacker computes a value $m = M(F(x), y)$, called a *metric*, and infers (x, y) as a member if m satisfies a given condition (e.g., greater than a given threshold). There are five known attacks of this type: *Top 1*, *correctness*, *confidence*, *entropy*, and *m-entropy attacks* (Table 1), where Top 1 was proposed in [31], and the other four were proposed in [35] by generalizing or improving known metric-based attacks [19, 31, 31, 35, 36, 44].

In our experiments, we executed all five metric-based attacks [31, 35] mentioned above.

4.4.2 Black-Box MIA Only with Labels (BB Label Only)

These are attacks such that attackers know only the predicted labels as outputs of the target model without knowing the confidence scores. We call such an MIA a *label-only MIA*. There are two known label-only attacks, *boundary distance (BD)* and *data augmentation* [5]. We introduce only the former one because it is stronger than the latter one [5].

A BD attack is an attack that computes the smallest adversarial perturbation Δx satisfying $\underset{i}{\mathrm{argmax}}\, F(x+\Delta x)_i \neq \underset{i}{\mathrm{argmax}}\, y_i$ for the target data (x, y). Here, $F(x+\Delta x)_i$ and y_i are the i-th components of $F(x+\Delta x)$ and y, respectively. The attacker then infers x is a member if the L_2 norm of Δx is larger than a predetermined threshold. A BD attack is a black-box MIA if adversarial perturbation is crafted by HopSkipJump [4]. However, the attack becomes a white-box MIA if we use the Carlini-Wagner method for adversarial perturbation [1].

In our experiments, we executed the BD attack with HopSkipJump. This is because the attack accuracy of the BD attack based on HopSkipJump is asymptotically equal to that of the BD attack with Carlini-Wagner when the number of queries increases [5].

4.4.3 White-Box MIA (WB)

These are attacks such that attackers can take the confidence score of target data besides the model structure and model parameters of the target model as input. Two white-box attacks have been proposed, the *Nasr-Shokri-Houmansadr (NSH) attack* [24] and *Hui's attack* [16].

The NSH attack exploits the fact that the gradient for the model parameter of the target model F on (x, y) becomes smaller if (x, y) is a member of F. Specifically, an attacker computes the gradient of F on the target data and infers the membership of (x, y) by inputting the gradient as well as the confidence score and the class label into an NN trained by the attacker. Hui's attack focuses mainly on reducing the assumption behind the NSH attack. That is, it can be executed without assuming that an attacker has member data as prior knowledge [34].

In our experiments, we executed only the NSH attack, since our assumption was stronger than Hui's; an attacker has member data as prior knowledge (as mentioned in Section 4.3).

4.5 Known Defenses

Known defenses can be categorized into the following three types. We chose the best defense from all three types for comparison with our method.

Regularization-based Methods: These methods use the fact that the regularization techniques of ML models mitigate overfitting, one of the main reasons behind the MIA risk [44]. Regularization techniques, such as L_2-regularization, dropout [37], and early-stopping, also mitigate the MIA risk, as pointed out by Nasr et al. [23], Shokri et al. [34], and Song et al. [35], respectively. Meanwhile, *Adversarial Regularization* (AdvReg) [23] is a regularization that is focused on mitigating MIAs. To conduct our experiment, we chose AdvReg from this type of attack since it mitigates the MIA risk the best.

AdvReg is based on a game theoretic framework similar to GANs [11]. Specifically, we train a model F we want to protect and a pseudo attacker A alternatively. The aim of the A is to distinguish member data from non-member data. It corresponds to a discriminator of a GAN, and the gain of the A is added to the loss of F as a regularization term.

AX (Adversarial eXample)-based Method: This method exploits an AX [38] to mitigate the MIA risk, where AX is a technique for deceiving ML by adding small noise to the input of the ML. We used *MemGuard* [17] in our experiments. MemGuard adds AX noise to the output of F, which we want to protect. Then, an attacker who uses an NN to attack F is deceived by the noise and cannot accurately determine the membership of the target data.

KT (Knowledge Transfer)-based Methods: These methods exploit KT to mitigate the MIA risk. Here, KT means knowledge distillation (explained in Section 3.1) or its variants. There are three known KT-based methods: *DMP* [33], *PATE* [26], and an improved variant of PATE, *PATE with confident-GNMax* [27]. We used DMP and PATE with confident-GNMax in our experiments for image data. However, we used only DMP for tabular data because PATE with confident-GNMax requires GANs.

Details on DMP have already been given in Section 2.3. Meanwhile, PATE trains multiple teacher models with *disjoint* subsets of private training data, gives public data hard labels chosen by noisy voting among the teachers, and finally trains a student model using labeled public data. A noisy voting mechanism provides differential privacy guarantees with respect to the training data. Confident-GNMax is a new noisy aggregation method for improving the original PATE. To achieve a smaller privacy budget ε, instead of labeling all public data, it selects the samples among public data to be labeled by checking if the result of a noisy plurality vote crosses a threshold. Once the threshold and noise parameters are determined, ε can be computed. We train a student model using semi-supervised learning with GANs [32].

4.6 ML Setups and Hyperparameter Choosing

4.6.1 ML Setups

In all experiments, we used a batch size of 64, the SGD optimizer with a momentum of 0.9 and weight decay of 10^{-5}, and the ReduceLROnPlateau scheduler with default hyperparameters. The model that recorded the best validation accuracy in five trials was evaluated to test the accuracy and risks against the four types of MIAs. We conducted all experiments using the PyTorch 1.7 framework on a Tesla V100 GPU with 32-GB memory.

Table 2 shows how we split the above datasets in our experiments. Here, *validation dataset* is a dataset used to select the best model parameters that does not overlap with the training dataset in our experiment. Following the previous studies [23, 24], we considered strong attackers who know the non-member dataset and a subset of the member dataset of the target model as their prior knowledge (see Section 4.3). We used the rest of the training/testing data as the target data to execute an MIA. The amounts of known data and target data are also depicted in Table 2.

4.6.2 Hyperparameter Tuning

Unprotected, AdvReg, MemGuard, DMP, and KCD: Using Optuna [25], we optimized hyperparameters for each scheme.

- For unprotected models, we chose learning rates that maximize validation accuracies.
- For AdvReg, MemGuard, DMP, and KCD, we tuned the learning rates and their specific parameters, i.e., the penalty parameter λ (AdvReg), learning rate β of a pseudo attacker, the weights c_2, c_3 of the loss function (MemGuard), the size of the public reference data[4] (DMP), and the intensity α of the distillation in Algorithm 2 (KCD), respectively. We optimized their hyperparameters toward a high validation accuracy and low MIA risk.

The hyperparameters of the defenses were basically chosen to have almost the same accuracy as the unprotected model and a considerably low MIA risk, except for some defenses whose accuracy inevitably drops no matter which hyperparameters we chose for them with low MIA risk.

- In Tables 3, 4, and 5, we chose hyperparameters for AdvReg, that enable a better privacy-utility trade-off (i.e., relatively small validation accuracy drop and mid-level MIA resistance) since almost the same validation accuracy as the unprotected model (making the MIA risk similar to a random guess, resp.) results in an MIA risk that is the same as that of an unprotected model (deterioration of validation accuracy, resp.).
- For MemGuard, we fixed $\varepsilon = 1.0$ and tuned the other parameters toward a low MIA risk.
- The hyperparameters of DMP were chosen to replicate the performance of the original paper [33].
- For KCD, we chose a model whose validation accuracy is close to that of DMP.

PATE: For PATE, we trained four ensembles of teachers, i.e., 3, 5, 10 and 25, and selected five different privacy levels (ε, δ) for each ensemble (where δ is fixed to 10^{-4} as the order of the size of the public reference dataset is 10^4 [27]). Since our interest is empirical MIA resistance, not DP guarantees, we chose various values for ε. For example, for three teachers, we chose $\varepsilon = 229, 1473, 6291, 36849, 83535, 141923$ (These cannot be round numbers because these values are automatically computed after we choose the thresholds and noise parameters). Epsilon 141923 was the minimum value that maximized the validation accuracy (i.e., corresponding to the non-private case), and epsilon 229 was the minimum value that provided enough labeled public data to train a student. Using Optuna, we optimized the learning rates toward a high validation accuracy for each ε.

In Table 5, we chose the hyperparameters “3 teachers, $\varepsilon = 141923$, $\delta = 10^{-4}$,” which maximized the valida-

[4] There are three privacy-utility trade-off hyperparameters depicted in the DMP paper [33], temperature, entropy criterion, and the number of reference data as explained in Section 2.3. We chose the number of pieces of reference data from them for our experiments since this number shows the best trade-off in our environment.

	Train.			Ref.	Val.	Test.		
Dataset	All	Known	Target			All	Known	Target
Purchase	10000	5000	2500	10000	5000	5000	2500	2500
Texas	10000	5000	2500	10000	5000	5000	2500	2500
CIFAR10	25000	12500	2500	25000	5000	5000	2500	2500

Table 2. Dataset splits. "All": All data used to train or test, "Known": Known data that attacker can exploit to execute MIA, "Target": Target data for which attacker attempts to infer membership.

tion accuracy because all the trained models had almost the same MIA risks.

4.6.3 Choice of Loss Function

The loss functions for most of the defenses were chosen from the original studies. The exception is DMP with synthetic reference data [33]; we chose the mean squared error (MSE) as a loss function since "synthetic" DMP with this loss function performed better than the original loss function, KL divergence, in our experiments. We chose a suitable loss function on the basis of known facts about distillation: the KL loss at a high temperature T asymptotically approaches the MSE, and which of these performs well is an empirical question [12] (the loss with $T = 1$ is KL loss). Therefore, we examined the KL divergence loss at various T for "synthetic" DMP and found that a higher T leads to better performance and that MSE loss is the best. By doing similar experiments for our defense, KCD, we determined the suitable loss to be the MSE for the Purchase100 and CIFAR10 datasets and KL divergence with $T = 1$ for the Texas100 dataset.

4.6.4 Notes on Implementation of DMP

The published code[5] does not include reference data selection but nonetheless achieved good results. Therefore, we did not implement entropy-based criteria for DMP in our experiments.

For DMP with synthetic reference data, we trained (unconditional) DCGAN as in the original study [33]. We trained them to obtain generated images in accordance with the implementation of PyTorch examples[6]. Since the resulting images (Figure 6) were natural and showed large diversity, we considered them to be sufficient for the reference dataset of DMP.

5 Experimental Results

5.1 Tabular Dataset

Tables 3 and 4 show the accuracies and MIA attack accuracies of our KCD and known defenses for two tabular datasets, Purchase100 and Texas100. Here, KCD is compared with the best defenses chosen in Section 4.5 from each of the three categories described in the same section. We stress that one can succeed in an MIA with a probability of 50% by random guessing. Hence, the baseline of the attack accuracies of these tables is 50%.

Note that the values for the attack accuracy of MemGuard on these tables are much higher than the values reported in the original paper [17]. This is because the setting we consider, described in Section 4.3, is more advantageous for attackers than that of [17].

Figure 2 shows the privacy-utility trade-off of KCD and DMP. The results of our experiments for the tabular datasets Purchase100 and Texas100 are summarized as follows.

1. Tables 3 and 4 show that KCD was *much better* than the known defenses that also do not use a public reference dataset, AdvReg and MemGuard, in all of the three categories of MIAs, the black-box MIA with confidence score [31, 34, 35], the label-only MIA [5], and the white-box MIA [24].
   For Purchase100, for instance, the testing accuracy of KCD was 11.5% higher than that of AdvReg and its attack accuracy was 13.3% smaller than that of MemGuard for "BB w/score" attacks.
2. Surprisingly, Tables 3 and 4 also show that, in both privacy and utility senses and for all of the three categories of MIAs, *KCD is comparable to the state-of-the-art MIA defense, DMP, with public reference data, although KCD does not use public reference data.* As mentioned in Section 1, the availability of

[5] https://github.com/vrt1shjwlkr/AAAI21-MIA-Defense

[6] https://github.com/pytorch/examples/blob/master/dcgan/main.py

	Defense		Purchase100				
			Train	Test	BB		WB
	Category	Name			w/score	label only	
$^{\nexists}$Public Ref.	Reg-based	AdvReg [23]	82.3%	64.2%	59.9%	58.9%	60.2%
	AX-based	MemGuard [17]	**100.0%**	**77.0%**	72.1%	(68.6%)	(74.3%)
	KT-based	**KCD**	93.8%	75.7%	**58.8%**	**58.7%**	**59.5%**
$^{\exists}$Public Ref.	KT-based	DMP [33]	89.3%	75.4%	57.1%	57.5%	57.3%
Unprotected			100.0%	77.0%	73.7%	68.6%	74.3%

Table 3. Comparisons on Purchase100.

	Defense		Texas100				
			Train	Test	BB		WB
	Category	Name			w/score	label only	
$^{\nexists}$Public Ref.	Reg-based	AdvReg [23]	60.5%	45.5%	59.5%	56.7%	58.0%
	AX-based	MemGuard [17]	**90.7%**	**52.5%**	68.6%	(69.7%)	(70.0%)
	KT-based	**KCD**	59.2%	52.0%	**56.2%**	**53.6%**	**55.8%**
$^{\exists}$Public Ref.	KT-based	DMP [33]	65.1%	51.9%	56.3%	56.1%	56.5%
Unprotected			90.7%	52.5%	69.9%	69.7%	70.0%

Table 4. Comparisons on Texas100.

	Defense		CIFAR10				
			Train	Test	BB		WB
	Category	Name			w/score	label only	
$^{\nexists}$Public Ref.	Reg-based	AdvReg [23]	84.9%	76.3%	54.6%	54.7%	55.2%
	AX-based	MemGuard [17]	**100.0%**	82.1%	64.3%	(55.6%)	(66.0%)
	KT-based	DMP [33] (synth. ref.)	81.1%	75.5%	**52.5%**	**52.5%**	**52.6%**
		KCD	94.0%	**82.2%**	55.8%	55.6%	56.2%
$^{\exists}$Public Ref.	KT-based	DMP [33]	84.2%	82.2%	51.1%	50.9%	51.4%
		PATE [27]	74.2%	72.8%	51.2%	50.2%	51.4%
Unprotected			100.0%	82.1%	65.9%	65.4%	66.0%

Table 5. Comparisons on CIFAR10

Explanatory notes on above three tables:

- Rows
  - "$^{\nexists}$Public Ref." (resp. "$^{\exists}$Public Ref.") means defense methods for ML models not using (resp. using) public reference data.
  - The bold means the best value in each column among the defenses of "$^{\nexists}$Public Ref."
  - In Table 5, "DMP [33] (synth. ref.)" is DMP with public reference data generated using DCGAN [29].
  - In Table 5, "PATE" is PATE with confident-GNMax [27].
- Columns
  - "Train" and "Test" are the training and testing accuracies.
  - "BB w/score" is the maximum attack accuracies of the following black-box MIAs using confidence scores, ML Leaks Adversary 1 [31] and five metric-based attacks [31, 35]. See Appendix A for the attack accuracy of each attack.
  - "BB label only" and "WB" are the attack accuracies of the BD attack [5] and NSH attack [24], respectively.
- Others
  - The values of MemGuard in "BB label only" were not obtained in the experiments. We included the same values as the unprotected models because, as explained in [5], MemGuard works in such a manner that a model's predicted labels are not changed using the defense; therefore, the attack accuracies of label-only attacks for MemGuard become the same as those of unprotected models.
  - Similarly, we included the same values as the unprotected models for MemGuard in "WB" because MemGuard is designed for black-box MIAs, and attackers using white-box MIAs can easily recover an unprotected model from MemGuard.

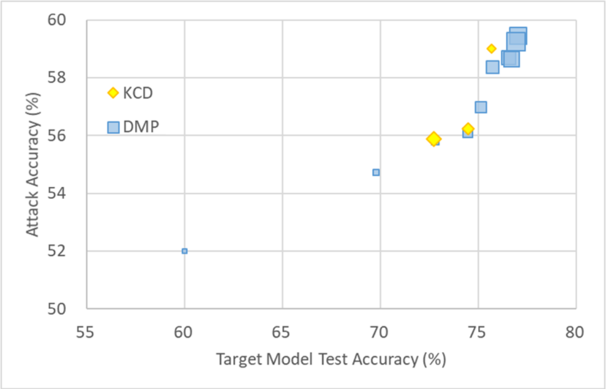

Fig. 2. Privacy-utility trade-off of DMP and KCD for Purchase100.

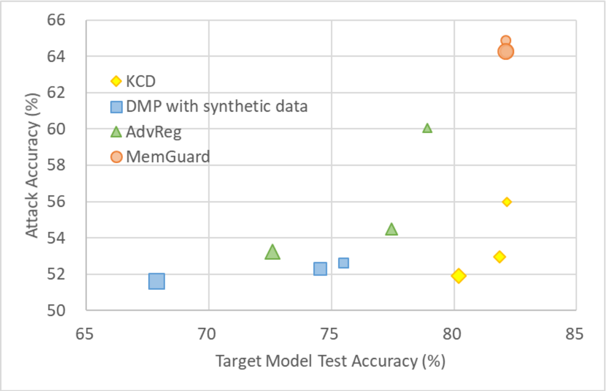

Fig. 3. Privacy-utility trade-offs among defenses for CIFAR10.

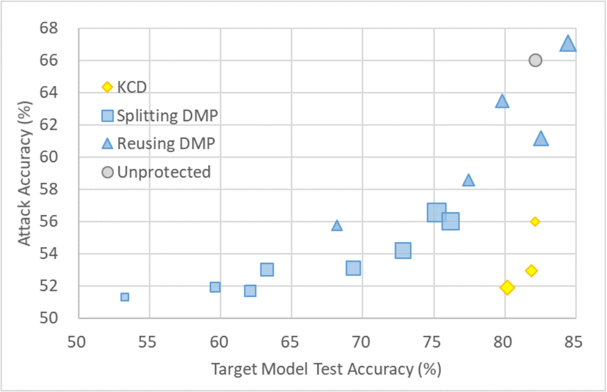

Fig. 4. Privacy-utility trade-off of our KCD, “splitting” DMP, “reusing” DMP, and unprotected model for Purchase100.

Explanatory notes on above three figures:

- The points towards the bottom right are better defenses, i.e., more accurate and more private ones.
- A larger point means a larger parameter.
- The vertical axis “Attack Accuracy” means “BB w/score” from Table 3.
- The privacy-utility trade-off hyperparameters are as follows.
  - KCD: α of Algorithm 2 ($\alpha = 0.6, 0.8, 1.0$ for Purchase100, $\alpha = 0.25, 0.5, 1.0$ for CIFAR10)
  - DMP: the number of pieces of reference data ($n = 2000, 4000, \ldots, 20000$); this is the best privacy-utility trade-off parameter in our experiment as described in Section 4.6.2 and our α cannot be used directly for DMP as described at the end of Section 3.2.
  - DMP with synthetic data: the number of pieces of synthetic reference data ($n = 12500, 37500, 50000$)
  - AdvReg: the penalty parameter $\lambda = 2.623, 3.019, 8.847$
  - MemGuard: the distortion budget $\varepsilon = 0.1$, 0.5, 1.0
  - “Splitting” DMP and “Reusing” DMPs: the percentages θ of training data used to train the student models ($\theta = 7\%$ to 50% for “splitting” DMP, $\theta = 20\%$ to 100% for “reusing” DMPs). Note that the accuracy of “reusing DMP” with $\theta = 100\%$ is better than the unprotected model. This kind of increasing accuracy is observed in knowledge distillations [12].

public data is not guaranteed [33]. The above results show that KCD could avoid this problem without sacrificing privacy or utility in these experiments.

3. Figure 2 shows that, for Purchase100 and for the “BB w/score” attack, *the privacy-utility trade-off of KCD was also comparable to that of the state-of-the-art MIA defense requiring public reference data, DMP.*

   We also executed similar experiments for “label only” and “WB.” They showed that the privacy-utility trade-off of KCD was comparable to that of DMP for these two types of attacks as well.

5.2 Image Dataset

Our experiments for the image dataset CIFAR10 were conducted in a similar manner as the above experiment. We additionally compared KCD with two more defenses. The first one was PATE (with confident-GNMax) [27]. The second one was DMP with synthetic reference data [33] generated using deep convolutional GANs (DCGANs) [29]. Note that these two defenses were not used in our experiment with the tabular datasets because they use GANs.

The results of these experiments are summarized as follows.

1. Table 5 and Figure 3 show that the privacy-utility trade-off of KCD was *much better* than that of the known defenses without public reference data, AdvReg and MemGuard, and DMP with synthetic reference data for the “BB w/score” attack.

   We also executed similar experiments for “label only” and “WB.” They showed that the privacy-utility trade-off of KCD is much better than the known defenses without public reference data as well.
2. Table 5 also shows that KCD was comparable to DMP and performed much better than PATE in terms of testing accuracy. DMP and PATE were better in terms of privacy, but KCD is better in the sense that it does not require public reference data.

6 Discussions and Limitations

6.1 Discussions

Best Number n of Teacher Models: Figure 5 shows the performance of our KCD for various teacher models on CIFAR10. Generally, a higher number of teacher models implies better privacy and utility, and they perform the best when $n = 15$.

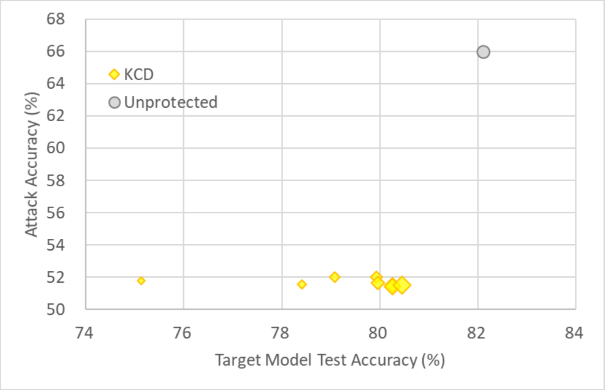

Fig. 5. Effect of number n of teacher models on performance of our KCD for CIFAR10. We examined $n = 2, 3, 5, 7, 9, 11, 13$, and 15 (a larger point means a larger n). Note that points indicating $n = 11$ and 13 are too close to distinguish.

The computational cost of KCD is greater than those of DMP and PATE but less than those of some of the other defense methods, such as AdvReg. A large computational cost may limit applications for training large models with limited computational resources. However, we believe that the advantage of KCD, “public reference dataset not necessary,” makes other applications possible.

We stress that, for inferring, KCD incurs only the same computational cost as an unprotected target model, unlike MemGuard.

Comparison with Naïve Ideas: To clarify the effect of our “knowledge cross-distillation” idea for KCD in terms of privacy and utility, we compared KCD with two naïve improvements to DMP to make it “without reference data.”

The first naïve improvement, *“splitting” DMP*, is as follows. Split the training dataset into two distinct parts; the former and the latter parts contain $(100 - \theta)\%$ and $\theta\%$ of training data, respectively. Then, train the teacher model using the former part as a training dataset and train the student model through distillation by using the remaining one as a reference dataset.

The second naïve improvement, *“reusing” DMP*, is as follows. Train a teacher model using all of the training data, take a subset containing $\theta\%$ of training data, and reuse this subset as the reference dataset to train a student model.

Figure 4 shows that, for the CIFAR10 dataset, the privacy-utility trade-off of our KCD was better than those of these two variants of DMP in our experiments. Our KCD contains two ideas, “splitting training dataset” and “reusing training data for reference data.” The above result shows that the performance of our KCD is achieved only when both of these ideas are used, and it cannot be achieved with only one of these ideas.

6.2 Limitations

Duplication in Dataset: If certain data appear twice in the training dataset, KCD cannot ensure defense against MIAs for such a pair of data. In fact, the defense against MIAs as depicted in Algorithm 2 is ensured because inputs $x \in D_i$ to F_i are not contained in dataset $D \setminus D_i$ used in the training of F_i. However, this is not the case when the same data fall into D_i and $D \setminus D_i$, respectively.

Similarly, a training dataset that contains two pieces of data that are not the same but very similar would affect the privacy-utility trade-off of KCD. Investigating and solving this is for future work.

Outlier Data, Imbalanced Dataset: Long et al. [21, 22] showed that an ML model became weaker against MIAs when the target data were outliers or selected carefully by an attacker, even if the ML model was well-generalized.

We selected the target data uniformly at random in our experiments. Hence, KCD, as well as other known defense methods, may have weak MIA resistance against carefully selected data.

Truex et al. [41] showed that MIAs against minority classes of imbalanced data were more likely to be successful. Here, imbalanced data means a dataset with skewed class proportions. Minority classes mean the classes that make up a smaller proportion. Hence, KCD, as well as other defense methods, may also have weak protection against MIAs in this case.

7 Conclusion

We proposed a new defense against MIAs, *knowledge cross-distillation (KCD)*, which does not require any public or synthetic reference data to protect ML models unlike the state-of-the-art defense, DMP.

Our experiments showed that the privacy protection and accuracy of our defense were comparable to those of DMP for the tabular datasets Purchase100 and Texas100, and our defense had a much better privacy-utility trade-off than those of the existing defenses for the CIFAR10 image dataset.

Our defense is a feasible method for protecting the privacy of ML models in areas where public reference data are scarce. Future work includes ensuring the privacy of duplicated or similar data in a dataset, investigating privacy for outlier and/or imbalanced data, and guaranteeing the privacy of KCD theoretically.

Acknowledgments

We would like to thank Reza Shokri, Tomoyuki Yoshiyama, and Kazuya Kakizaki for useful comments. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In IEEE Symposium on Security and Privacy, pages 39–57. IEEE Computer Society, 2017.

[2] Bill Text - AB-375 Privacy: personal information: businesses. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml ?bill_id=201720180AB375, 2018.

[3] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. arXiv preprint arXiv:1912.11279, 2019.

[4] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In 2020 ieee symposium on security and privacy (sp), pages 1277–1294. IEEE, 2020.

[5] Christopher A Choquette Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. arXiv preprint arXiv:2007.14321, 2020.

[6] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: A new paradigm to machine learning. Archives of Computational Methods in Engineering, pages 1–22, 2019.

[7] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, Automata, Languages and Programming, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[8] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor,

Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings, volume 4004 of Lecture Notes in Computer Science, pages 486–503. Springer, 2006.

[9] Ethics Guidelines for Trustworthy AI. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai, 2019.

[10] 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). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679, 2016.

[11] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: membership inference attacks against generative models. Proc. Priv. Enhancing Technol., 2019(1):133–152, 2019.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In NIPS Deep Learning and Representation Learning Workshop, 2015.

[13] HIPAA. https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf, 1996.

[14] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS Genet, 4(8):e1000167, 08 2008.

[15] Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. CoRR, abs/2103.07853, 2021.

[16] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. In 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021. The Internet Society, 2021.

[17] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang 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, pages 259–274, 2019.

[18] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[19] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In Srdjan Capkun and Franziska Roesner, editors, 29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020, pages 1605–1622. USENIX Association, 2020.

[20] Zheng Li and Yang Zhang. Label-leaks: Membership inference attack with label. CoRR, abs/2007.15528, 2020.

[21] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. CoRR, abs/1802.04889, 2018.

[22] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, XiaoFeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In IEEE European Symposium on Security and Privacy, EuroS&P 2020, Genoa, Italy, September 7-11, 2020, pages 521–534. IEEE, 2020.

[23] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pages 634–646, 2018.

[24] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In IEEE Symposium on Security and Privacy, pages 739–753. IEEE, 2019.

[25] Optuna. https://optuna.org/.

[26] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In ICLR. OpenReview.net, 2017.

[27] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[28] Kaggle's Acquire Valued Shoppers Challenge. https://www.kaggle.com/c/acquire-valued-shoppers-challenge, 2013.

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[30] Alexandre Sablayrolles, Matthijs Douze, Yann Ollivier, Cordelia Schmid, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In ICML 2019-36th International Conference on Machine Learning, volume 97, pages 5558–5567, 2019.

[31] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In Network and Distributed Systems Security Symposium 2019. Internet Society, 2019.

[32] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016.

[33] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pages 9549–9557. AAAI Press, 2021.

[34] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine

learning models. In 2017 IEEE Symposium on Security and Privacy (SP), pages 3–18. IEEE, 2017.

[35] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. arXiv preprint arXiv:2003.10595, 2020. (Accepted in USENIX Security 2021.).

[36] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019, pages 241–257. ACM, 2019.

[37] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):1929–1958, 2014.

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[39] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. Mitigating membership inference attacks by self-distillation through a novel ensemble architecture. arXiv preprint arXiv:2110.08324, 2021.

[40] Hospital Discharge Data Public Use Data File. https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm.

[41] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Wenqi Wei, and Lei Yu. Effects of differential privacy and data skewness on membership inference vulnerability. In First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2019, Los Angeles, CA, USA, December 12-14, 2019, pages 82–91. IEEE, 2019.

[42] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. IEEE Transactions on Services Computing, pages 1–1, 2019.

[43] Michael Veale, Reuben Binns, and Lilian Edwards. Algorithms that remember: model inversion attacks and data protection law. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376(2133):20180083, 2018.

[44] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF), pages 268–282. IEEE, 2018.

A Appendix

	Category	Defense	Leaks 1 [31]	Top 1 [31]	Corr. [35]	Conf. [35]	Entr. [35]	m-Entr. [35]
$^{\nexists}$Public Ref.	Reg-based	AdvReg [23]	57.0%	56.8%	58.9%	59.9%	55.3%	59.7%
	AX-based	MemGuard [17]	66.6%	71.9%	61.3%	72.1%	70.1%	72.1%
	KT-based	**KCD**	54.9%	55.0%	58.8%	57.0%	53.7%	57.3%
$^{\exists}$Public Ref.	KT-based	DMP [33]	53.9%	53.9%	57.1%	55.8%	52.8%	55.7%
Unprotected			72.8%	72.0%	61.3%	73.6%	71.2%	73.7%

Table 6. BB attacks with confidence scores on Purchase100

	Category	Defense	Leaks 1 [31]	Top 1 [31]	Corr. [35]	Conf. [35]	Entr. [35]	m-Entr. [35]
$^{\nexists}$Public Ref.	Reg-based	AdvReg [23]	52.5%	52.1%	56.7%	58.8%	53.2%	59.5%
	AX-based	MemGuard [17]	57.7%	58.0%	68.6%	68.2%	57.7%	68.2%
	KT-based	**KCD**	54.8%	54.9%	53.1%	56.2%	54.8%	55.4%
$^{\exists}$Public Ref.	KT-based	DMP [33]	51.2%	51.5%	56.1%	56.3%	51.0%	56.1%
Unprotected			58.8%	58.6%	68.6%	69.7%	59.4%	69.9%

Table 7. BB attacks with confidence scores on Texas100

	Category	Defense	Leaks 1 [31]	Top 1 [31]	Corr. [35]	Conf. [35]	Entr. [35]	m-Entr. [35]
$^{\nexists}$Public Ref.	Reg-based	AdvReg [23]	53.1%	52.7%	54.6%	54.6%	51.9%	54.6%
	AX-based	MemGuard [17]	63.0%	63.4%	58.6%	64.3%	63.1%	64.3%
	KT-based	DMP [33](synth. ref.)	51.0%	51.2%	52.5%	51.8%	50.3%	52.0%
		KCD	52.1%	52.2%	55.6%	55.3%	51.3%	55.8%
$^{\exists}$Public Ref.	KT-based	DMP [33]	50.8%	50.7%	50.7%	50.4%	51.1%	50.2%
		PATE [27]	50.0%	49.8%	50.5%	50.4%	50.0%	51.2%
Unprotected			64.2%	63.8%	58.6%	65.6%	63.9%	65.9%

Table 8. BB attacks with confidence scores on CIFAR10

The above tables show the attack accuracies of each black-box MIA with confidence scores. “Leaks 1” means ML Leaks Adversary 1 [31]. “Top 1,” “Corr.,” “Conf.,” “Entr.,” “m-Entr.,” mean five metric-based attacks [31, 35], *Top 1*, *correctness*, *confidence*, *entropy*, and *m-entropy attacks*, respectively.

Fig. 6. Images generated by (unconditional) DCGAN