An Optimal Knowledge Distillation for Formulating an Effective Defense Model Against Membership Inference Attacks

Thi Thanh Thuy Pham[1], Huong-Giang Doan[2]
Faculty of Information Security, Academy of People Security, Ha Noi, Viet Nam[1]
Faculty of Control and Automation, Electric Power University, Ha Noi, Viet Nam[2]

***Abstract*—A membership inference attack (MIA) on machine learning models aims to determine the sensitive data that has been used to train machine learning models. Machine learning-based applications (MLaaS—machine learning as a service) in finance, banking, healthcare, etc. are facing the risks of private data leaks by MIA. Several solutions have been proposed for mitigating MIA attacks, such as confidence score masking, regularization, knowledge distillation (KD), etc. However, the utility-privacy trade-off problem is still a major challenge for existing approaches. In this work, we explore the KD-based approach to defending against MIA attacks. This approach has received increasing attention in the research community on machine learning safety recently as it aims at effectively addressing the above-mentioned challenge of mitigating MIA attacks. An efficient KD-based defense framework that includes multiple teacher and student models is proposed in this work for alleviating MIA attacks. Three main phases are deployed in this framework: (1) teacher model training; (2) knowledge distillation from the teacher model to the student model based on prediction augmentation and aggregation from the teacher model; and (3) repeated knowledge distillation among student models. The experimental results on standard datasets show the outperforms in both model utility and privacy of the proposed framework compared to other state-of-the-art solutions for mitigating MIA.**

***Keywords*—Knowledge distillation; membership inference attack; teacher model; student model; privacy-utility trade-off**

I. Introduction

A membership inference attack (MIA) is one kind of AI security attack in which the attackers try to determine if the sensitive information used in training a machine learning model. In some AI-based applications, protecting the privacy of training data is an important requirement, such as individuals' bank account numbers, credit/debit card details, transaction data or patients' medical records. In the common MIA attack scenario, two machine learning models are considered: (1) the target model, which is trained on the dataset that needs to be kept private, and (2) a MIA model, which is trained by the attacker. Based on MIA model, the attacker can predicts whether a particular data sample is a member or non-member of the private training set. The extent of MIA attacks on machine learning models depends on the information obtained by the attackers. This can be (i) the shadow data, which is the one that has the same distribution as the data used to train a target model; (ii) the knowledge of the target model, including the model architecture, the learned parameters like weights or coefficients, and the learning algorithm. The white-box attacks rely on the knowledge of the target model and the training data distribution of the target model. In a black-box attack, the attackers can only approach the trained target model (e.g., a target classifier) and get the prediction outputs from this model.

Several solutions have been proposed to mitigate the MIA attacks. They can be classified into four main approaches: confidence score masking, regularization, differential privacy, and knowledge distillation. In the first approach, the confidence scores of class predictions in the output vector of the target model are masked to prevent information leakage from these [1]. This technique is mainly deployed for black-box attacks on the classification models. Therefore, it is easily deployed without any intervention inside the target model. The defensive intervention only happened with model output. However, this defense method can still be breached by attack methods such as label-only attacks [2] or metric-based attacks [3]. The regularization technique aims at preventing model overfitting, which is a key factor in the success of MIA attacks. Several solutions to this approach are proposed, such as L2-norm regularization, data augmentation, and dropout [4], Adversarial Regularization [5]. The regularization technique not only interferes with the output of target models but also their internal parameters. Therefore, it can be applied to both black-box and white-box MIA attacks. Although the regularization technique is widely applied and effective against MIA attacks, the accuracy of the target models is inversely proportional to the privacy level that this technique provides. This means the regularization technique brings high privacy to the target models, but it can also reduce their accuracy [6]. In the defense method of differential privacy, the personal information is added to the noise. This will make it difficult for MIA attackers to identify the original data. However, the challenge when applying this method is to find a reasonable way to balance the effectiveness between the overall accuracy and its privacy against MIA attacks [7]. The last defense approach to MIA attacks is Knowledge Distillation (KD). KD was introduced in [8] as one of the transfer learning methods. The fundamental concept of the KD is derived from the process of human learning, in which information is transferred from a teacher with greater knowledge to a student with less understanding. The teacher models are much larger than the student models. However, based on the knowledge distilled from the teacher model, the student model still achieves almost the same performance as the teacher model. The KD-based defense models for mitigating MIA attacks require two datasets named private and reference datasets. The private dataset is used to train the

teacher model, which is considered the unprotected model. The reference dataset is soft labeled based on the predictions of the trained teacher model. The soft-labeled reference dataset is utilized to train the student model, which is considered a protected model. The reference dataset can be the unlabeled public dataset [9] or the private one [10]. The main challenge for the KD-based defense models is the private-utility trade-off of the protected model. In addition, the student/protected model is desired to have as high accuracy as the teacher model.

Among the above-mentioned approaches, the KD-based method against MIA attacks has been attracting the research community recently because of its higher defense capacity than many other solutions while still ensuring the model's performance. However, the private-utility trade-off is still an open issue with this approach. Focusing on this, in this work, we propose a new framework based on KD for mitigating MIA attacks. It is different from other available KD-based approaches, in this framework, we deploy (1) soft labeling of the reference dataset by prediction augmentation and aggregation from the teacher model and (2) repeated knowledge distillation among multiple student models. The prediction augmentation is executed through teacher model calibration with several temperature parameters. This will output several class probability distributions for each input sample. The prediction aggregation from the teacher model is done based on an optimal selection of the prediction probabilities from the teacher model. This helps to create uniform distribution predictions over all classes and contains no useful information for MIAs while still maintaining the classification performance of the target model. In addition to knowledge distillation from the teacher model to a student model as other works, in this work, we first conduct knowledge transfer from one student model to another student model multiple times. This creates multi-layer masking for the target dataset and helps strengthen the defense ability of the target model against MIAs. The experimental results on standard public datasets show the outperformance of our contributions on not only classification performance but also the defense ability of the target model against MIAs compared to other related state-of-the-art (SOTA) methods.

The remainder of this paper is organized as follows: In Section II, we briefly survey recent related works based on KD for mitigating MIA attacks. The proposed methodology is presented in Section III. The experimental results are analyzed in Section IV. Finally, Section V concludes the paper and states research directions for future work.

II. Related Work

The knowledge distillation technique was originally designed to reduce computational cost and memory requirements while maintaining the performance of deep learning models. This enables deep learning models to be deployed on devices with limited computing and storage capacity. Recently, the KD approach has also been exploited in cyber security with KD-based defense against MIA attacks.

In [9] a KD-based defense solution against MIAs, named DMP (Distillation for Membership Privacy) is proposed. DMP requires two datasets: a private dataset and a reference dataset. The private dataset is the labeled dataset and needs to be protected from attacks. The reference data is sensitive and unlabeled. It is drawn from the same distribution as the private training dataset and used to train the target model. These datasets are utilized in three phases of DMP, including the pre-distillation phase, the distillation phase, and the post-distillation phase. In the first phase, an unprotected model is trained on a private dataset. This model is then used in the second phase to generate a reference dataset that minimizes membership privacy leakage and transfers its knowledge to the protected model. In the final phase, the protected model is trained on the reference data with both ground truth and predictions from the unprotected model. DMP is the first method based on KD. In comparison with other previous approaches against MIAs, it improved not only the defense capacity but also the model's performance on some benchmark datasets. However, obtaining a large amount of publicly available reference data with the same distribution as private data is challenging in practice. Moreover, the reference data generation by DC-GAN as conducted in [9] seems to be a more reasonable solution for this challenge, but it reduces the performance of the model.

The solution proposed in [11] to address the challenge raised in [9]. The reference dataset in [11] is a part of the private dataset, not the public one as in [9]. In order to overcome the overfitting that can occur with this selection, the authors in [11] proposed KCD (Knowledge Cross-Distillation) for membership privacy. KCD uses multiple teacher models to transfer knowledge to the student model (target model). The private dataset is divided into several parts. The knowledge transfer process is done several times. At each time, consider one part of the private dataset as a reference dataset and other parts as a private dataset. The private dataset is used to train the teacher model, and the reference dataset part is soft labeled by the trained teacher model. Finally, we get soft-labeled reference data parts and utilize them to train the target model. Similarly, the work in [10] proposed a multi-teacher architecture to transfer knowledge to the student model. The private dataset is split into K disjoint partitions of the same size. The teacher models are trained on these partitions in the manner of K-fold cross validation. The soft targets are generated from these trained teacher models, and they are used to train the student model in the distillation phase. In general, in comparison with [9], the multi-teacher knowledge distillation decreases the attack accuracy and improves the classification performance of the target model. However, experimental results on widely used datasets show that the testing accuracy of the proposed target models is only less than 86%. It is still necessary to increase the classification performance of the target model and ensure data privacy against MIA attacks.

In this work, we propose an efficient KD-based framework for mitigating MIA attacks. It is similar to the approach of [11], [10]; in this framework, the sensitive dataset is split into two parts: one for training the teacher model, and the other is softly labeled by the teacher model and used for training the student model. The teacher model are trained in the manner of two-fold cross validation. However, it is different from the above approaches in that soft labeling for the reference dataset is done by prediction augmentation and aggregation from the teacher model. Furthermore, in this research, we add an additional layer of knowledge distillation that is repeatedly implemented by the student models. Other related works only stop at

transferring knowledge from one or more teacher models to a student model and using this student model as a defensive model against MIA attacks. However, in our work, an optimal defense model will be selected from the student models. This aims at creating multi-layer masking for privacy data and then helps strengthen the defense ability of the target model against MIA attacks. The details of the proposed framework will be discussed in the next section.

III. Methodology

A. The Overall Framework

The overall defense framework against membership inference attacks is shown in Fig. 1. There are three main blocks in this framework: (1) teacher model training; (2) knowledge distillation from the teacher model to the student model (Teacher-Student KD) based on prediction augmentation and aggregation from the teacher model; and (3) repeated knowledge distillation (Repeated Student KD) from the student model θ_S^{n-1} to the θ_S^n, with n is the number of times the student model θ_S is executed.

Inspired by the idea of [11], in this work, we also deploy the sensitive private dataset for our proposed KD-based defense system. The data scenario for the training teacher model and knowledge distillation from the teacher to the student model is shown in Fig. 2.

We have a sensitive private dataset D, and we split it into two parts, D_1 and D_2. We first use D_1 for training teacher model θ_T. The trained θ_T will be utilized for soft labeling D_2. Secondly, we train the teacher model θ_T on D_2 and use the trained model θ_T to soft label D_1. The datasets with soft labels named D_1' and D_2' will be used to train the student model for the first time (θ_S^1). In order to express this generally (in Fig. 1), we refer to the parts of the dataset used for training teacher model θ_T as D_{Pri} and the ones for soft labeling as D_{Ref}:

- $D_{Pri} = \{(x_{1_P}, y_{1_P}), .., (x_{N_P}, y_{N_P})\}$ ($N_P = |D_{Pri}|$)
- $D_{Ref} = \{(x_{1_R}), .., (x_{N_R})\}$ with the corresponding hard labels $\{(y_{1_R}), .., (y_{N_R})\}$

In block 1, we train the teacher model on the D_{Pri}. The D_{Pri} is split to D_{Pri}^{train} which is used to train the teacher model θ_T, a test split D_{Pri}^{test} and a validation split D_{Pri}^{val}. The teacher model θ_T is trained using D_{Pri}^{train} until the training converges to minimize the loss

$$\sum_{(x_P, y_P) \in D_{Pri}^{train}} L\left(\theta_T\left(x_P\right), y_P\right)$$

In block 2, we utilize the trained θ_T to soft label the $D_{Ref} = \{x_{1_R}, .., x_{N_R}\}$, and D_{Ref} is labeled by θ_T: $y_{N_R^0} = \theta_T(x_{N_R})$. The soft labeled data $D_{Ref}^0 = \{(x_{1_R}, y_{1_R}^0), .., (x_{N_R}, y_{N_R}^0)\}$ will be utilized as ground truth for training the student model θ_S^1: $\theta_S^1\left(x_{N_R}, \theta_T\left(x_{N_R}\right)\right)$ until the training converges to minimize the loss:

$$\begin{aligned} &\alpha \sum_{(x_R, y_R^0) \in D_{Ref}^0} L\left(\theta_S^1\left(x_R\right), y_R^0\right) + \\ &(1-\alpha) \sum_{(x_R, y_R) \in D_{Ref}} L\left(\theta_S^1\left(x_R\right), y_R\right) \end{aligned} \tag{1}$$

where y_R^0 is soft label returned by θ_T for the input x_R, and y_R is the hard label of x_R.

The soft labeling is implemented based on prediction augmentation and aggregation from the teacher model. The details for this will be represented in the next subsection.

In block 3, D_{Ref} will be soft labeled by θ_S^1: $y_{N_R}^1 = \theta_S^1(x_{N_R})$. The soft-label data $D_{Ref}^1 = \{(x_{1_R}, y_{1_R}^1), .., (x_{N_R}, y_{N_R}^1)\}$ will be utilized as ground truth for training the student model θ_S^2: $\theta_S^2\left(x_{N_R}, \theta_S^1\left(x_{N_R}\right)\right)$. The soft labeling is implemented based on prediction from θ_S^1 until the training converges to minimize the loss

$$\begin{aligned} &\alpha \sum_{(x_R, y_R^1) \in D_{Ref}^1} L\left(\theta_S^2\left(x_R\right), y_R^1\right) + \\ &(1-\alpha) \sum_{(x_R, y_R) \in D_{Ref}} L\left(\theta_S^2\left(x_R\right), y_R\right) \end{aligned} \tag{2}$$

The soft data labeling and knowledge distillation steps are implemented repeatedly from θ_S^{n-1} to θ_S^n. The final student model θ_s^n will be considered the protected model or a defense model against MIA attacks. The target model θ_S^n is trained until it converges to the loss

$$\begin{aligned} &\alpha \sum_{(x_R, y_R^n) \in D_{Ref}^{n-1}} L\left(\theta_S^n\left(x_R\right), y_R^{n-1}\right) + \\ &(1-\alpha) \sum_{(x_R, y_R) \in D_{Ref}} L\left(\theta_S^n\left(x_R\right), y_R\right) \end{aligned} \tag{3}$$

In the proposed system, we believe that the soft labeling for D_{Ref} by prediction augmentation from the teacher model in block 1 will create uniform distribution predictions over all classes and contain no useful information for MIAs. In addition, the knowledge distillation from the teacher model θ_T to the first student model θ_S^1 is implemented by the combination of learning from ground-truth labels and teacher predictions. Based on this, the student model θ_S^1 can learn more effectively not only from the behavior of θ_T on x_R but also from the D_{Pri}. This helps the student model θ_S^1 have the competitive classification performance with the teacher model. Moreover, in the block 3, the repeated knowledge distillation from θ_S^{n-1} to θ_S^n creates multi-layer masking for the D_{Ref} dataset. This will help strengthen the defense ability of target model θ_S^n against MIAs on the original sensitive private dataset D.

B. Prediction Augmentation and Aggregation from the Teacher Model

The prediction augmentation and aggregation from the teacher model θ_T for soft labeling D_{Ref} are shown in Fig. 3.

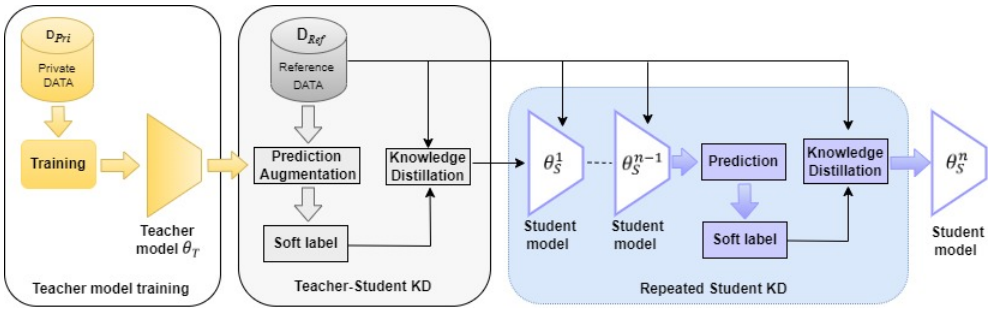

Fig. 1. The overall defense framework for membership inference attacks based on knowledge distillation from prediction augmentation of teacher model and repeated knowledge distillation of student models.

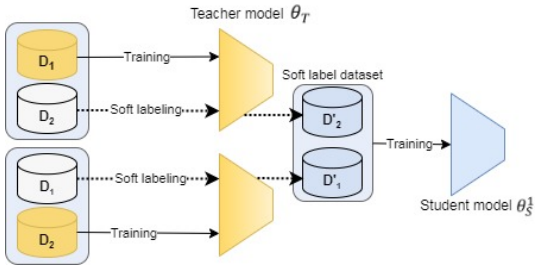

Fig. 2. The data scenario for training teacher and student models.

It should be noted in this figure that the repeated KD from one student model to another is done by prediction augmentation.

We have an unlabeled dataset $D_{Ref} = \{x_{1_R}, .., x_{N_R}\}$ that needs to be labeled by the teacher model θ_T. Given an input x_R, θ_T estimates the probability that $P(y_R = c \mid x_R)$ for each class value of $c = 1, .., C$. Thus, θ_T will output a C-dimensional vector whose elements sum to 1, or give out C estimated probabilities:

$$p_i = softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \quad for\ i = 1, 2, ..., C \tag{4}$$

where $z, p \in \mathbb{R}^C$ and $\mathbf{z}$ is the output vector of the last layer of the teacher model; $0 < p_i < 1$ and $\sum_i p_i = 1$. Using the temperature parameter in softmax for controlling the softness of the probability distribution, we have the probabilities as follows:

$$p_i = softmax_T(z_i) = \frac{e^{z_i/T}}{\sum_{j=1}^{n} e^{z_j/T}} \tag{5}$$

where T is called the temperature parameter. When T gets lower, the biggest value in x_R get more probability, when T gets larger, the probability will be split more evenly on different elements. In this work, we conduct prediction augmentation through teacher model calibration with K temperature parameters. This means, for a single input x_R, the teacher model θ_T will output K probability distributions p_j^k according to K temperature parameters ($k = 1, 2, .., K$); j is the number of the classes ($j = 1, 2, .., C$), as follows:

$$\begin{aligned} p_j^1 &= \left[p_1^1, p_2^1, .., p_C^1\right] \\ p_j^2 &= \left[p_1^2, p_2^2, .., p_C^2\right] \\ &... \\ p_j^K &= \left[p_1^K, p_2^K, .., p_C^K\right] \end{aligned} \tag{6}$$

where p_j^k is calculated as follows:

$$p_j^k = \frac{e^{z_j/T_k}}{\sum_{j=1}^{n} e^{z_j/T_k}} \tag{7}$$

where T_k is a temperature hyper-parameter ($k = 1, 2, .., K$).

In order to avoid the leakage of D_{pri} from MIA attacks, there should be a uniform distribution over all classes for x_R, but we must still ensure the classification accuracy of the model. This means we need to have an uniform probability distribution of the classes but still keep a maximum probability which assigns to a certain class by each output probability distribution respect to each T_k. In order to achieve this goal, we firstly consider the predictions of θ_T in case of the smallest value of T_k, which is equivalent to p_j^k with $k = 1$ or p_j^1. In the set of $p_j^1 = \left\{p_1^1, p_2^1, .., p_C^1\right\}$, we examine two subsets of the prediction probabilities. One contains high probability values (HP), and the other includes low probability values (LP). HP contains the $\underset{j=1 \div C}{max} \left\{p_j^1\right\}$ and its neighborhoods N_ϵ that are significantly lower than $\underset{j=1 \div C}{max} \left\{p_j^1\right\}$, as follows:

$$HP(p) = \left[\underset{j=1 \div C}{max} \left\{p_j^1\right\}, N_\epsilon\right] \tag{8}$$

with N_ϵ is represented as follows:

$$N_\epsilon\left(\underset{j=1 \div C}{max}\{p_j^1\}\right) = \left\{p \in p_j^1 \mid d\left(p, \underset{j=1 \div C}{max}\{p_j^1\}\right) < \epsilon\right\} \tag{9}$$

LP contains the remaining probability values in p_j^1: $LP(p) = p_j^1 \cap HP(p)$.

At other $T_k, (k = 2, .., K)$, we have the probability distributions for each class $j, (j = 1, .., C)$. The final probability

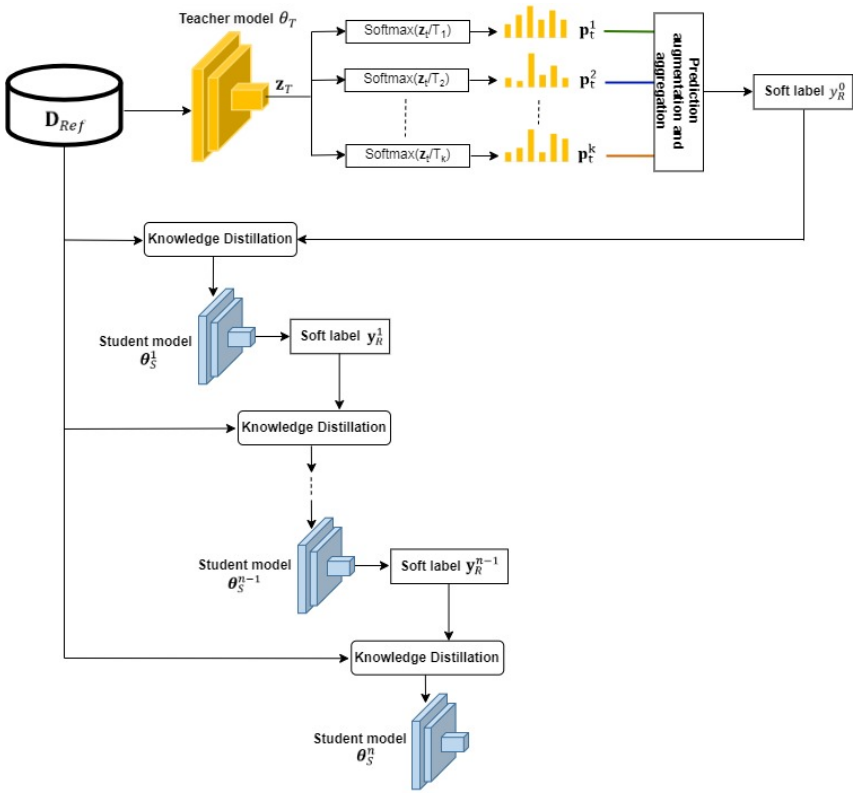

Fig. 3. The prediction augmentation and aggregation from the teacher model for the soft labeling of the reference dataset and the repeated knowledge distillation from one student model to another.

distribution for an input x_R with prediction augmentation from θ_T model by K temperature parameters will be aggregated as follows:

$$p_{j,x_R} = \left\{ \min_{j=1 \div C} \{p_j^k\} \mid p_j^k \in HP(p) \right\} \cup \left\{ \max_{j=1 \div C} \{p_j^k\} \mid p_j^k \in LP(p) \right\} \quad (10)$$

We then label the samples $\{x_{1_R}, .., x_{N_R}\}$ of reference dataset D_{Ref} according to the maximum probability element in p_{j,x_R} predicted by the teacher model θ_T.

IV. Experiment and Result

A. Experimental Datasets and Teacher, Student model Structures

In this work, several datasets are utilized for experiments: Purchase100[1], Texas100[2], CIFAR10, CIFAR100 [12], MNIST[3], MS-COCO [13], and ImageNet [14].

The Purchase100 dataset used in this work is set as in [6]. It contains 197,324 records of the user's product transactions each year. Each record contains 600 binary features that represent whether the user has purchased the product or not. The records are grouped into several classes, each representing a different purchase style. The Purchase100 dataset is set for 5 different classification tasks with a different number of classes: 2, 10, 20, 50, 100. The classification task is to predict the purchase style of a user given the 600-feature vector.

Texas100 dataset as used in [6] for the classification task. The dataset contains 100 classes of patient records with 67,300 binary feature vectors with a dimension of 6,170. Each dimension corresponds to symptoms and its value states if the corresponding patient has the symptom or not; the label represents the treatment given to the patient.

CIFAR10 and CIFAR100 are popular image classification datasets. CIFAR10 contains 60,000 RGB images with the size of 32 × 32 pixels for each. Each image is labeled in one of 10 classes. CIFAR100 has 100 classes containing 600 images each There are 20 super classes out of 100 in the CIFAR100. Each image is labeled with the superclass and the class to which it belongs.

MNIST dataset contains 70,000 grey-scale images of handwritten digits. There are 10 classes, one for each digit '0' to '9'. MS-COCO dataset is a mainstream dataset for object detection, with 118,000 training images and 5,000 validation images from 80 categories. ImageNet is a benchmark dataset for image classification, with nearly 1.3 million training images and 50,000 images for validation. The images come from 1,000 categories.

In this work, we use the same architecture for teacher and student models. The dataset split for experiments and the

[1]https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data.
[2]https://www.dshs.texas.gov/THCIC/Hospitals/ Download.shtm
[3]url = https://yann.lecun.com/exdb/mnist

TABLE I. THE TEACHER/STUDENT MODELS AND THE SPLITS OF THE EXPERIMENTAL DATASETS

Dataset	Model	Dpri			Dref	Attack train		Attack test	
		Train	Test	Val		Member	Non-member	Member	Non-member
Purchase100	FC	10,000	5,000	5,000	10,000	10,000	5,000	5,000	2,500
Texas100	FC	10,000	5,000	5,000	10,000	10,000	5,000	5,000	2,500
MNIST	FC	30,000	5,000	5,000	30,000	30,000	5,000	5,000	2,500
CIFAR10	Wide ResNet-28 Alexnet VGG16 DenseNet121	25,000	5,000	5,000	25,000	25,000	5,000	5,000	2,500
CIFAR100	Wide ResNet-28 Alexnet VGG16 DenseNet121	25,000	10,000	5,000	25,000	25,000	5,000	5,000	2,500

teacher/student model structures are shown in Table I. For example, in the Purchase100 dataset, 10,000 samples are set for each D_{Pri} and D_{Ref}; 5,000 samples are used for validation and 5,000 for testing the model. The amount for attack model training is 10,000 member and 5,000 non-member samples, while the amount for attack model testing is 5,000 and 2,500, respectively.

As in [15], the teacher/student model for Purchase100 is a 4-layer fully connected neural network (FC) with layer sizes [1024, 512, 256, 100] and a 5-layer fully connected neural network with layer sizes [2048, 1024, 512, 256, 100] for the Texas100 dataset. In this work, we also use a 5-layer fully connected neural network with layer sizes [2048, 1024, 512, 256, 100] for the MNIST dataset. For CIFAR10 and CIFAR100 four models of Wide ResNet-28 [16], Alexnet [17], VGG16 [18], DenseNet121 [19] are deployed for teacher/student model.

B. Attack Scenario

In this work, black-box and white-box attacks as in [11] are deployed to evaluate the defense performance of the proposed framework. The black-box attack scenarios is shown in Fig. 4. We put the sets of non-member data (the non-training data of the target model) and member data (the training data of the target model) into the target model θ_S^n. It will output the corresponding confidence scores or labels of the inputs. These results are then used for training the attack model θ_A. Given the input target data, the attack model will infer the membership status of the target data. In this work, we evaluate two types of black-box attacks. The first one belongs to the case that the attack classifier knows only the predicted labels from the target model but not confidence scores. Inversely, in the second case, the attack classifier knows only confidence scores but not predicted labels. We deploy the Boundary Distance (BD) attack with HopSkipJump [20] for black-box attack with labels only and ML Leaks Adversary 1 attack [21] for black-box attack with confidence scores. In Boundary Distance (BD) attack with HopSkipJump, a testing sample is inferred as member if the L2 norm of the smallest adversarial perturbation of this sample is larger than a predetermined threshold.

The attack model for both types of black-box attacks is a binary classifier with a multilayer perceptron (a 64-unit hidden layer and a softmax output layer), as in [21].

The white-box attack scenario deployed in this work is the same as the one in [22]. In this case, the inputs for the training attacker classifier are confidence score of target data, as in the case of black-box attack, and the target model parameters and

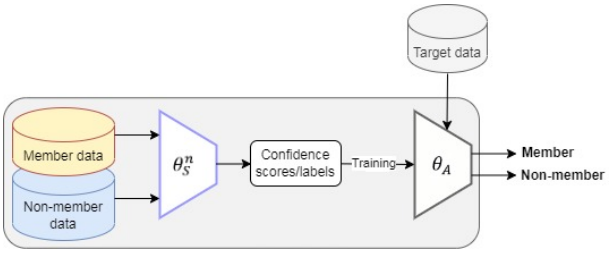

Fig. 4. The black-box attack scenario.

structure. As shown in Fig. 5, we put member data and non-member data to the target model θ_S^n and the outputs for this are confidence scores/labels. In addition, the target data is also an input of θ_S^n to give out the gradient for the model parameter of θ_S^n. Confidence scores/labels and gradient are used to train the attack model θ_A. Based on the trained θ_A model, the attacker can infer the member data or non-member data of the target data.

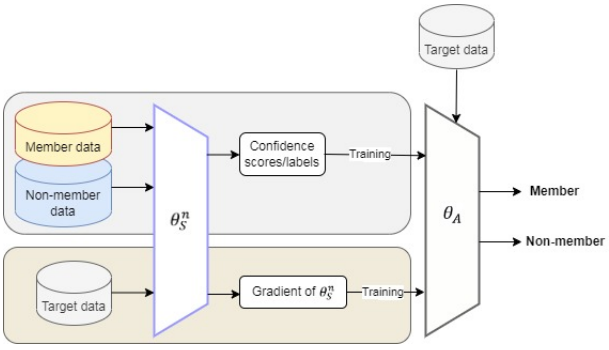

Fig. 5. The white-box attack scenario.

The data sets for training and testing the attack models is shown in Table I. The member samples are a portion of the training data of the target model, and the non-member samples are not included in the training set of the target model.

C. Defense Scenario

In order to evaluate the defense performance of the proposed system, we compare it to the popular defenses for MIA privacy, including AdvReg (Adversarial Regularization) [15] and MemGuard [23]. Furthermore, our defense solution is also compared to the SOTA KD-based methods of DMP [9] and KCD [11].

TABLE II. THE EXPERIMENTAL RESULTS FOR KNOWLEDGE DISTILLATION FROM THE TEACHER MODEL TO THE FIRST STUDENT MODEL (ST1) AND AMONG THE STUDENT MODELS, FROM ST1 TO ST6, ON DIFFERENT DATASETS AND MACHINE LEARNING MODELS

2*Dataset	2*Model	2*Teacher Acc (%)	st1 Acc (%)		st2 Acc (%)		st3 Acc (%)		st4 Acc (%)		st5 Acc (%)		st6 Acc (%)	
			(-)Pred	(+)Pred	(-)Pred	(+)Pred	(-)Pred	(+)Pred	(-)Pred	(+)Pred	(-)Pred	(+)Pred	(-)Pred	(+)Pred
Purchase100	FC	94.09	93.95	89.81	94.21	90.39	93.81	90.16	92.69	90.76	91.96	89.08	91.48	89.75
Texas100	FC	94.16	93.91	91.59	93.25	90.98	93.08	90.18	92.81	89.95	92.58	89.51	91.98	89.06
MNIST	FC	96.70	96.88	93.21	95.89	92.33	96.03	91.27	95.32	91.89	94.11	91.61	94.28	89.93
4*CIFAR10	Wide ResNet-28	94.80	95.36	88.98	94.88	86.29	94.11	84.37	93.78	83.15	92.56	81.06	91.61	80.11
	Alexnet	89.40	90.04	86.72	90.31	87.06	89.85	86.85	89.91	86.19	89.17	86.35	88.75	85.89
	VGG16	93.04	93.22	87.16	93.07	87.01	93.69	87.34	92.81	86.59	92.39	86.28	91.58	85.63
	DenseNet121	95.80	96.28	90.74	97.35	91.08	97.61	91.38	96.74	91.59	96.41	92.08	96.05	91.94
4*CIFAR100	Wide ResNet-28	79.04	79.94	76.64	80.04	77.12	79.35	76.72	78.56	76.83	77.18	75.51	75.59	75.94
	Alexnet	65.72	70.66	67.91	71.15	68.37	70.39	67.48	69.81	65.94	68.29	65.31	67.33	63.78
	VGG16	73.32	75.40	71.16	76.19	71.03	75.82	72.15	76.68	71.84	74.24	70.59	72.57	69.64
	DenseNet121	80.66	81.92	79.06	82.11	78.86	83.56	77.69	83.08	78.48	82.18	76.44	81.95	75.39

The AdvReg method is a regularization that attempts to prevent overfitting in machine learning models. Overfitting phenomena can allow an attacker to perform MIAs. In [15], a min-max privacy game between the defense mechanism and the inference attack is proposed. This aims to simultaneously minimize the classification loss of the model and the maximum gain of the MIA against it. An adversarial regularization parameter, which is the gain of the inference attack, is added to the loss function of the target model to protect the privacy of the data and control the trade-off between membership privacy and classification error.

If the AdvReg method tries to tamper with the training process of the target model, MemGuard attempts to interfere with the confidence score vectors predicted by the target model for the input data samples. In a black-box attack setting, an attacker has the data samples and puts them into the target model to gain confidence score vectors. These vectors will be inputs to train the attack model. The trained attack classifier will be used to predicts a data sample is a member or not of the target model's training dataset. In order to protect the training data privacy, MemGuard adds a carefully crafted noise vector to a confidence score vector to turn it into an adversarial example that misleads the attacker classifier.

D. Evaluation Metrics

In this work, two evaluation metrics are used for evaluating the performance of the target models against MIA attacks. The first one is Generalization Error (**GE**). **GE** [24] expresses the absolute difference between the train accuracy and test accuracy of the target model θ_S^n. It reflects the overfitting level of the target model. A larger GE means a higher privacy risk of membership inference attacks [6]. The second evaluation metric is attack accuracy which is the fraction between samples correctly classified as members of the training dataset and the total samples classified as members.

E. Experimental Results

The experiments are conducted to evaluate (1) the performance of the proposed framework in knowledge distillation from the teacher model θ_T to the student model θ_S^n; (2) the defense performance of θ_S^n against black-box and white-box attacks (as mentioned in Section IV-B) and compare this to other SOTA methods (as indicated in Section IV-C)

1) Evaluation of the knowledge distillation performance: In this section, we evaluate the knowledge distillation performance from the teacher model θ_T to the first student model θ_S^1, and repeated knowledge distillation among the student models (from θ_S^1 to θ_S^n). The evaluations are implemented in two experimental scenarios: (1) knowledge distillation from teacher model to student model with the augmented and aggregated predictions from the teacher model (+$Pred_{a\&a}$), and (2) knowledge distillation from teacher model to student model without the augmented and aggregated predictions from the teacher model (-$Pred_{a\&a}$).

The parameters of the experimental models are as follows:

- Full connected model (FC): Batch size equals 32; 50 epochs to 100 epochs for training model; Adam optimizer; Cross entropy lost function; Learning rate is from 10^{-4} to 10^{-6};
- Wide ResNet-28, Alexnet, VGG16, DenseNet121: batch size is 32; epoch number for training is 200; trained with Adam optimizer; Lost function is Cross entropy; Learning rate is from 10^{-5} to 10^{-6}.

The temperature values are $T_k = \{2, 3, 4, 5\}$; $\alpha = 0.5$; $n = 6$.

Table II shows the experimental results for knowledge distillation from teacher model θ_T to the first student model θ_S^1 (st1) and from θ_S^1 (st1) to θ_S^6 (st6) on different datasets and machine learning models. In general, in scenario 2 (-$Pred_{a\&a}$), the classification results of the student model 1 (θ_S^1) are higher than the ones of the teacher model, except for the FC model with the Purchase100 and Texas100 datasets. In the scenario 1 (+$Pred_{a\&a}$), the classification results of the θ_S^1 are lower than the ones of teacher model, except for the case of CIFAR100 with the Alexnet model. These results are also lower than the case of (+$Pred_{a\&a}$) of θ_S^1. The classification results also gradually decrease from the student model 1 to the student model 6 in both cases of (-$Pred_{a\&a}$) and (+$Pred_{a\&a}$).

2) Evaluation of the defense ability of the student model against MIAs:

-Generation error evaluation:

Fig. 6 represents the generation error (GE) evaluation of student models on the CIFAR10 dataset with Wide ResNet-28, Alexnet, VGG16, DenseNet121 models, respectively. The result for the scenario of (+)$Pred_{a\&a}$ is denoted as (+)GE, and for the scenario of (-)$Pred_{a\&a}$ is (-)GE.

It can be seen from Fig. 6 that the minimum values of (-)GE and (+)GE obtained from the student model 6 (st6) and the student model 1 (st1) on Wide ResNet-28 are 2.41% and

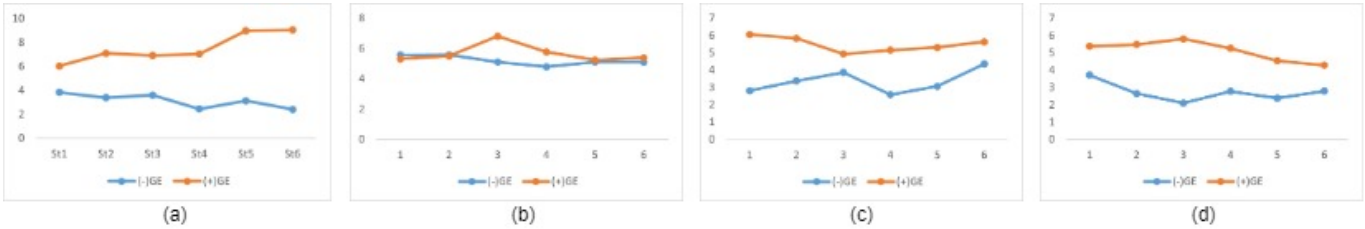

Fig. 6. Generation error for the CIFAR10 dataset with (a) the Wide ResNet-28 model, (b) the Alexnet model, (c) the VGG16 model, and (d) the DenseNet121 model.

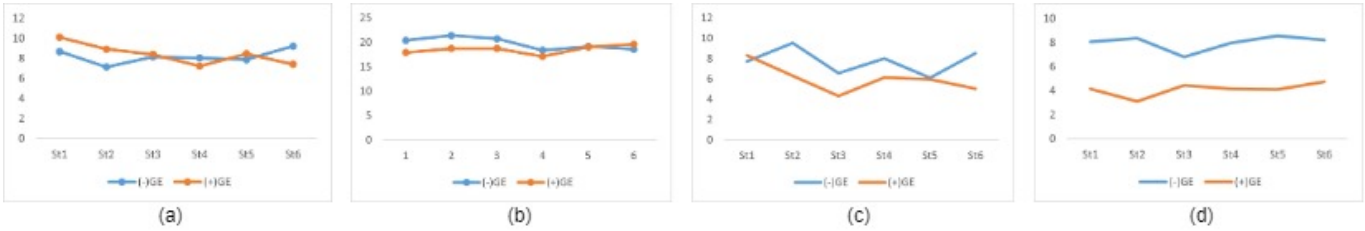

Fig. 7. Generation error for the CIFAR100 dataset with (a) the Wide ResNet-28 model, (b) the Alexnet model, (c) the VGG16 model, and (d) the DenseNet121 model.

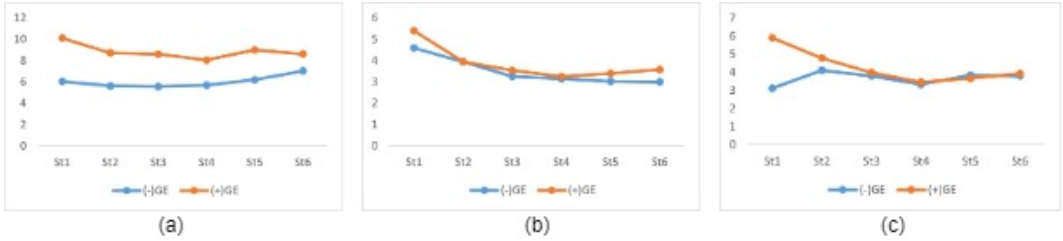

Fig. 8. Generation error with the FC model for (a) the Purchase100 dataset, (b) the Texas100 dataset, and (c) the MNIST dataset.

6.03%, respectively. For Alexnet model, the minimum values of (-)GE and (+)GE are 4.8% and 5.24% for st4 and st5 models, respectively. The lowest (-)GE and (+)GE values of the VGG16 model are for the st4 model with 2.58% and the st3 model with 4.93%. For the DenseNet121 model, the st3 model has a minimum (-)GE value of 2.1% and the st6 model has a (+)GE minimum value of 4.29%.

The (+)GE and (-)GE results on CIFAR100 dataset with the models of Wide ResNet-28, Alexnet, VGG16, DenseNet121 are indicated in Fig. 7. The minimum results of (-)GE and (+)GE for the Wide ResNet-28 model are 7.15% from st2 model and 7.24% from st4 model, respectively. For the Alexnet model, the lowest (-)GE and (+)GE are 18.38% and 17.12% for the st4 model. The minimum results of (-)GE and (+)GE for the VGG16 model are 6.1% (st5) and 4.33% (st3). The lowest (-)GE and (+)GE results for DenseNet121 model are 6.81% for st3 model and 3.11% for st2 model.

The (+)GE and (-)GE evaluations on Purchase100, Texas100, and MNIST datasets with FC model are presented in Fig. 8. For the Purchase100 dataset, the lowest value of (-)GE is 5.57% for the st3 model, while the one of (+)GE is 8.05% for the st4 model. The (-)GE and (+)GE values for Texas100 dataset are lowest for the st6 model with 3%, and the st4 model with 3.26%. The smallest results of (-)GE and (+)GE on the MNIST dataset are 3.12% and 3.46% for the st1 and st4 models, respectively.

In general, GE results for both scenarios (-)$Pred_{a\&a}$ and (+)$Pred_{a\&a}$ at different datasets and experimental models change quite fluctuating across student models. We choose the optimal student models that have the smallest GE values in both (-)$Pred_{a\&a}$ and (+)$Pred_{a\&a}$ scenarios. They are the selected defense models, and they will be evaluated for their defense against MIA attacks in the next section.

-Black-box and white-box attacks on the defense student model:

Table III, Table IV, and Table V, represent the comparative results of our optimal defense models (as mentioned above) in both scenarios (-)$Pred_{a\&a}$ and (+)$Pred_{a\&a}$ to other SOTA methods of AdvReg [15], MemGuard [23], and KCD [11] on the datasets of Purchase100, Texas100, and CIFAR10. The model architectures are Wide ResNet-28 for CIFAR10, fully connected NNs with Tanh activation functions for Purchase100, Texas100, as in [11].

It can be seen from the Table III that, with the Purchase100 dataset, the DMP defense model [9] is the best one for mitigating MIA. The accuracy results of score, label only black-box attacks and white-box attack are the lowest ones with 57.1%,

TABLE III. THE RESULTS OF OUR DEFENSE MODEL AGAINST BLACK-BOX AND WHITE-BOX ATTACKS COMPARED TO OTHER SOTA DEFENSE METHODS ON THE PURCHASE100 DATASET. THE SCENARIOS WITH PREDICTION AUGMENTATION AND AGGREGATION FROM THE TEACHER MODEL ($+Pred_{a\&a}$) AND WITHOUT THIS ($-Pred_{a\&a}$) ARE EVALUATED FOR OUR METHOD

Defense method	Purchase100 dataset					
	Train Acc	Test Acc	Generation Error (GE)	Black-box attack Acc		White-box attack Acc
				Score	Label only	
AdvReg [15]	82.3%	64.2%	18.1%	59.9%	58.9%	60.2%
MemGuard [23]	100.0%	77.0%	23%	72.1%	68.6%	74.3%
DMP [9]	89.3%	75.4%	13.9%	**57.1%**	**57.5%**	**57.3%**
KDC [11]	93.8%	75.7%	18.1%	58.8%	58.7%	59.5%
Our method ($-Pred_{a\&a}$)	99.38%	93.81%	5.63%	74.56%	75.18%	76.7%
Our method ($+Pred_{a\&a}$)	98.81%	**90.76%**	8.05%	**58.04%**	**57.93%**	**58.6%**

TABLE IV. THE RESULTS OF OUR DEFENSE MODEL AGAINST BLACK-BOX AND WHITE-BOX ATTACKS COMPARED TO OTHER SOTA DEFENSE METHODS ON THE TEXAS100 DATASET. THE SCENARIOS WITH PREDICTION AUGMENTATION AND AGGREGATION FROM THE TEACHER MODEL ($+Pred_{a\&a}$) AND WITHOUT THIS ($-Pred_{a\&a}$) ARE EVALUATED FOR OUR METHOD

Defense method	Texas100 dataset					
	Train Acc	Test Acc	Generation Error (GE)	Black-box attack Acc		White-box attack Acc
				Score	Label only	
AdvReg [15]	60.5%	45.5%	15%	59.5%	56.7%	58.0%
MemGuard [23]	90.7%	52.5%	38.2%	68.6%	69.7%	70.0%
DMP [9]	65.1%	51.9%	13.2%	56.3%	56.1%	56.5%
KDC [11]	59.2%	52.0%	7.2%	**56.2%**	**53.6%**	**55.8%**
Our method ($-Pred_{a\&a}$)	95.61%	92.58%	3.03%	75.29%	74.81%	75.5%
Our method ($+Pred_{a\&a}$)	93.21%	**89.95%**	3.26%	**51.77%**	**52.28%**	**52.6%**

TABLE V. THE RESULTS OF OUR DEFENSE MODEL AGAINST BLACK-BOX AND WHITE-BOX ATTACKS COMPARED TO OTHER SOTA DEFENSE METHODS ON THE CIFAR10 DATASET. THE SCENARIOS WITH PREDICTION AUGMENTATION AND AGGREGATION FROM THE TEACHER MODEL ($+Pred_{a\&a}$) AND WITHOUT THIS ($-Pred_{a\&a}$) ARE EVALUATED FOR OUR METHOD

Defense method	CIFAR10 dataset					
	Train Acc	Test Acc	Generation Error (GE)	Black-box attack Acc		White-box attack Acc
				Score	Label only	
AdvReg [15]	84.9%	76.3%	8.6%	54.6%	54.7%	55.2%
MemGuard [23]	100.0%	82.1%	17.9%	64.3%	55.6%	66.0%
DMP [9]	84.2%	82.2%	2%	**51.1%**	**50.9%**	**51.4%**
KDC [11]	94.0%	82.2%	11.8%	55.8%	55.6%	56.2%
Our method ($-Pred_{a\&a}$)	96.23%	93.78%	2.45%	67.7%	62.5%	68.9%
Our method ($+Pred_{a\&a}$)	90.18%	**83.15%**	7.03%	**50.4%**	**50.6%**	**50.8%**

TABLE VI. THE RESULTS OF OUR DEFENSE MODEL AGAINST BLACK-BOX AND WHITE-BOX ATTACKS ON CIFAR100 DATASET

Defense method	CIFAR100 dataset					
	Train Acc	Test Acc	Generation Error (GE)	Black-box attack Acc		White-box attack Acc
				Score	Label only	
Our method ($-Pred_{a\&a}$)	87.19	80.04	7.15	56.81	57.19	58.6%
Our method ($+Pred_{a\&a}$)	84.07	76.83	7.24	51.82	52.48	53.4%

57.5%, and 57.3%, respectively. The results obtained from our defense model with the $(+)Pred_{a\&a}$ scenario are only slightly lower than these results of DMP, with 58.04%, 57.93%, and 58.6%, respectively. However, the testing accuracy of our solution with $(+)Pred_{a\&a}$ is much higher than that of DMP (90.76% of ours compared to 75.4% of DMP). This is also much higher than the best MemGuard solution [23] (77%). Our method with $(-)Pred_{a\&a}$ has higher testing accuracy than the case with $(+)Pred_{a\&a}$. However, it also has much higher black-box and white-box attack accuracy than $(+)Pred_{a\&a}$ scenario.

In the experiments on Texas100 dataset, as shown in Table IV, our defense solution with the scenario of $(+)Pred_{a\&a}$ achieves the best performance against MIA. The black-box attack accuracy for score and label-only cases are 51.77% and 52.28%, respectively. The white-box attack accuracy is 52.6%. The classification accuracy of our method with $(+)Pred_{a\&a}$ is 89.95%, which is much higher than the best one of other solutions (52% of KDC method [11]). Although our method with the $(-)Pred_{a\&a}$ scenario achieves better classification results than the situation of $(+)Pred_{a\&a}$ (92.58% of $(-)Pred_{a\&a}$ compared to 89.95% of $(+)Pred_{a\&a}$), its defense ability is worse than the case of $(+)Pred_{a\&a}$ and other methods.

Table V presents the results for the CIFAR10 dataset. Our method with $(+)Pred_{a\&a}$ shows the best results for mitigating MIA attacks, with 50.4%, 50.6%, and 50.8% for black-box score-based, label-only and white-box attacks, respectively. These results are slightly better than those of the DMP method, with 51.1%, 50.9%, and 51.4%, respectively. The testing accuracy of our method with $(+)Pred_{a\&a}$ is also above that

TABLE VII. THE RESULTS OF OUR DEFENSE MODEL AGAINST BLACK-BOX AND WHITE-BOX ATTACKS ON MNIST DATASET

Defense method	MNIST dataset					
	Train Acc	Test Acc	Generation Error (GE)	Black-box attack Acc		White-box attack Acc
				Score	Label only	
Our method ($-Pred_{a\&a}$)	100	96.88	3.12	63.89	64.37	65.03%
Our method ($+Pred_{a\&a}$)	95.35	91.89	3.46	59.22	60.19	61.9%

of the DMP method, with 83.15% compared to 82.2% of the DMP. For the case of $(-)Pred_{a\&a}$), the testing accuracy is the best (93.78%), but it has the worst defense performance among others.

Tables VI and VII show the results of our solution for CIFAR100 and MNIST datasets in two scenarios of $(+)Pred_{a\&a}$ and $(-)Pred_{a\&a}$. The Wide ResNet-28 and FC models are implemented for the CIFAR100 and MNIST datasets, respectively. It can be seen from these tables that the classification performance of the $(-)Pred_{a\&a}$ scenario is better than the case of $(+)Pred_{a\&a}$. However, the resistance to MIA attacks of the $(-)Pred_{a\&a}$ case is not as good as the $(+)Pred_{a\&a}$ in both CIFAR100 and MNIST datasets.

The experimental results on different datasets with different models show the stable effectiveness of our proposed method in mitigating MIA attacks. By augmenting and aggregating the predictions from the teacher model to transfer to one student model ($+Pred_{a\&a}$), along with the knowledge transfer from one student model to another student model, we can choose the optimal student model as the efficient defense model against MIA attacks. We also see that, without prediction augmentation and aggregation from the teacher model ($-Pred_{a\&a}$), the classification performance of the defense model can be higher, but its attack accuracy is also higher than the case of $(+)Pred_{a\&a}$ and other solutions. With better classification efficiency than other SOTA solutions, our method with optimal student model and prediction augmentation and aggregation from the teacher model ($+Pred_{a\&a}$) can bring utility-privacy trade-off.

V. CONCLUSION AND FUTURE WORK

This work proposes a remarkable KD-based solution for mitigating MIA attacks. The knowledge is transferred from the teacher model to the student model based on the prediction augmentation and aggregation from the teacher model. The process of knowledge transfer also continues between student models to find out the optimal defense model against MIA attacks. The experimental results on the widely used datasets are promising and show better performance of our proposed method compared to SOTA methods.

Although the results are remarkable, there are still limitations in this study. The experiments have only been implemented with basic 2D CNN models and datasets. Knowledge transfer done iteratively across multiple models will be time-consuming. In the future, incremental learning mechanisms can be implemented in the proposed framework to take advantage of new information about added objects to further the concept of learning.

REFERENCES

[1] L. Hanzlik, Y. Zhang, K. Grosse, A. Salem, M. Augustin, M. Backes, and M. Fritz, "Mlcapsule: Guarded offline deployment of machine learning as a service," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3300–3309.

[2] G. Zhang, B. Liu, T. Zhu, M. Ding, and W. Zhou, "Label-only membership inference attacks and defenses in semantic segmentation models," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1435–1449, 2022.

[3] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2615–2632.

[4] S. Ben Hamida, H. Mrabet, F. Chaieb, and A. Jemai, "Assessment of data augmentation, dropout with l2 regularization and differential privacy against membership inference attacks," *Multimedia Tools and Applications*, pp. 1–30, 2023.

[5] H. Hu, Z. Salcic, G. Dobbie, Y. Chen, and X. Zhang, "Ear: An enhanced adversarial regularization approach against membership inference attacks," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[6] 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.

[7] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 7, pp. 48 901–48 911, 2019.

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

[9] V. Shejwalkar and A. Houmansadr, "Membership privacy for machine learning models through knowledge transfer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 11, 2021, pp. 9549–9557.

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

[11] R. Chourasia, B. Enkhtaivan, K. Ito, J. Mori, I. Teranishi, and H. Tsuchida, "Knowledge cross-distillation for membership privacy," *arXiv preprint arXiv:2111.01363*, 2021.

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

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[15] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 634–646.

[16] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *International conference on machine learning*. PMLR, 2016, pp. 1225–1234.