HP-MIA: A Novel Membership Inference Attack Scheme for High Membership Prediction Precision

Shi Chen[a], Wennan Wang[b,*], Yubin Zhong[a], Zuobin Ying[d], Weixuan Tang[c,*] and Zijie Pan[d]

[a] *School of Mathematics and Information Science, Guangzhou University Guangzhou China*
[b] *School of Economics, Xiamen University Xiamen China*
[c] *Institute of Artificial Intelligence and Blockchain, Guangzhou University Guangzhou China*
[d] *Faculty of Data Science, City University of Macau Macau China*

ARTICLE INFO

Keywords:
Machine Learning
Deep Learning
Privacy Protection
Membership Inference Attack

ABSTRACT

Membership Inference Attacks (MIAs) have been considered as one of the major privacy threats in recent years, especially in machine learning models. Most canonical MIAs identify whether a specific data point was presented in the confidential training set of a neural network by analyzing its output pattern on such data point. However, these methods heavily rely on overfitting and are difficult to achieve high precision. Although some recent works, such as difficulty calibration techniques, have try to tackle this problem in a tentative manner, identifying members with high precision is still a difficult task.

To address above challenge, in this paper we rethink how overfitting impacts MIA and argue that it can provide much clearer signals of non-member samples. In scenarios where the cost of launching an attack is high, such signals can avoid unnecessary attacks and reduce the attack's false positive rate. Based on our observation, we propose High-Precision MIA (HP-MIA), a novel two-stage attack scheme that leverages membership exclusion techniques to guarantee high membership prediction precision. Our empirical results have illustrated that our two-stage attack can significantly increase the number of identified members while guaranteeing high precision.

1. Introduction

Machine learning models have demonstrated its effectiveness in various fields, ranging from image classification to speech recognition. A sophisticated machine learning model usually requires large amounts of data for training. However, in most cases, these training data contains sensitive information, which brings a major concern of whether the model will reveal sensitive information about the training data. Unfortunately, recent researches [5, 39, 44] have shown that attackers can infer sensitive information from training data, if given access to machine learning models. In particular, among existing privacy attacks, Membership Inference Attack (MIA) [39] causes the most serious privacy leakage. In MIA, the adversary aims to infer whether a record exists in the training set of the target model. Since MIA is easy to achieve and powerful, MIA has been considered as a major security threat in many scenarios.

Nevertheless, existing MIAs methods tend to predict non-member samples as member samples and suffer from a high false positive rate (FPR) [35]. FPR shows how often a MIA method mislabels non-member samples as member ones. Therefore, previous attacks [38, 39, 40] do not work well in scenarios where the false positive cost is high. Some recent works [3, 37, 46] have considered leveraging difficulty calibration to mitigate the high FPR problem.

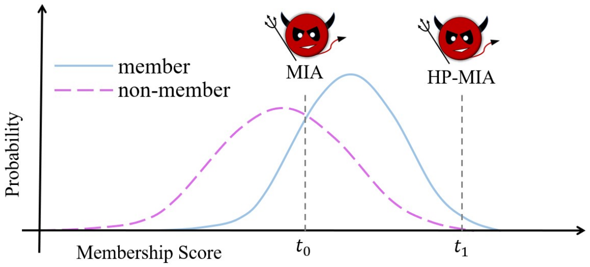


Figure 1: The difference between HP-MIA and MIA.The aim of MIA is to find a suitable threshold that achieves the highest accuracy in distinguishing members from non-members, while the goal of HP-MIA is to find the threshold that identifies members with a high prediction rate.

For example, in a target CNN that achieve 98.69% accuracy at MNIST, the C-Conf attack proposed by Watson et al.[46] can identify 52 out of 10000 members with 100% precision, which far less than an ideal result. Although this result is already a big improvement, identifying more members with high precision is still a difficult task.

Deep neural networks are equipped with strong learning abilities [52]: Consider a seriously-overfitted target model, its loss could still be maintained at a relatively low level, even if the training samples contain sample-specific noise. Particularly, suppose the Loss of all members in the training set is less than a constant ϵ. Since MIA is built upon ovefitting to a large extent, we can mark all records with a loss greater than a ϵ as non-members and achieve membership exclusion with 100% precision. In fact, easy-to-predict non-members may have very low loss, and neural networks are prone to overconfidence in records outside the training data. Therefore, overfitting output pattern may not only provide a valid basis for membership inference directly, but can also

*Corresponding author
wwwennan@xmu.edu.cn (W. Wang); tweix@gzhu.edu.cn (W. Tang)
ORCID(s):

provide an explicit non-membership signal to an adversary. MIA has been used in some work to construct more powerful privacy theft attacks, so it is significant to implement high precision MIA. In addition to this, MIA has now been developed as a privacy analysis technique for machine learning models[27, 40], but previous work has mostly been limited to discussing privacy leakage at the model level. High-precision MIA can capture the most vulnerable samples and thus help us understand the privacy leakage problem of machine learning models at the data level.

To this end, in this paper we propose High Precision MIA (HP-MIA), a novel MIA pipeline that identifies more member samples than previous works. As illustrated in Figure 1, while previous MIAs mainly focus on determining an optimal threshold for distinguishing members from non-members with high accuracy, our work focus on identifying members with low FPR. We tackle the construction of a high prediction rate MIA as an optimization problem with constraints. Specifically, instead of using overfitting signals of a neural network for direct membership inference, we use them to perform membership exclusion. HP-MIA consists of two stages, (i) membership exclusion stage, which exclude non-membership samples from the target dataset by overfitting signals, and (ii) membership inference stage, which leverages calibrated attack to identify the true members.

Moreover, MIA has been used in some works[4, 5] to construct more powerful privacy attacks. MIA has also now been developed as a privacy analysis technique for machine learning models, but most previous works[12, 27, 30, 40] has been limited to discussing privacy leakage at the model level. High-precision MIA can capture the most vulnerable samples and contribute to understanding why privacy leakage exists in machine learning models at data level.

Our attack is also evaluated broadly on various attack scenarios and various empirical results have confirmed the effectiveness of our proposal. In different datasets, HP-MIA is able to correctly infer $2 \sim 10$ times more member samples than previous works when the precision is close. Since our attack only requires training a small number of machine learning models, it has a smaller computational cost than other recent work[3, 49].

Contribution

We summarize our contributions and key finding as follows :

- Based on new observations on overfitting, we propose a novel perspective on designing MIAs. Particularly, we find that overfitting provides the adversary with a far more reliable non-membership signal than membership signal. In scenarios where the cost of attack is high, such signals can help the adversary avoid unnecessary losses.

- We propose a Two-stage High Precision MIA (HP-MIA), which consists of sample exclusion stage and inference stage. We improve the performance of high-precision inference via preemptive exclusion. Unlike previous MIA attack, we leverage overfitting signals to perform exclusion on non-members rather than directly identifying members.

- We deploy our attack on various datasets and models. Our empirical results show that HP-MIA is able to identify more memberships than other attacks while guaranteeing high precision. In addition, we further investigated how example difficulty affects member privacy risk, and the results suggest that hard-to-predict examples may be easier to cause privacy leakage.

Organization

We present the background of the membership inference attack and some recent work on difficulty calibration techniques in Section 2. Section 3 presents our HP-MIA framework and a new two-stage attack. Experimental results are given in Section 4. In Section 5 we discuss how and why our attack is successful. In Section 6 we discuss some related work. In Section 7 we conclude this work.

2. Background

In this section, we give the definition of membership inference attack (MIA) and introduce the threshold-based MIA in Section 2.1. Then, we describe difficulty calibration techniques used to mitigate the high FPR problem of MIA in Section 2.2. In Section 2.3, we introduce the Likelihood Ratio Attack.

2.1. Membership Inference Attack

Definition 1 (Membership Inference Attacks[39]). *Given a machine learning model h that has completed training on the training set $D \sim Q^n$, and a target sample $z = (x, y)$, where x represents input data, y represents the label, and Q denotes the probability distribution of the data points. The membership inference attack can be formalized as a binary classifier:*

$$\mathcal{A} : Z \times H \longrightarrow \{0, 1\} . \tag{1}$$

where 0 means z does not belong to the training set D, otherwise it is 1. Z denotes the set of all samples $z \sim Q$ and H denotes the set of all classifiers trained on examples from a data distribution Q.

Most of the previous work[28, 38, 39, 40, 46] assumed that the adversary only has black-box access to the target model and infer membership information from posterior probability vector. In addition to this, the adversary trains a shadow model to mimic the behavior of the target model. Shokri et al.[39] use neural networks to construct the attack model and train it based on the inputs and outputs of the shadow model.

A common binary classification in membership inference problems is the threshold model, which distinguishes members from non-members by computing a particular score $s(h, z)$ and setting a threshold t.

$$\mathcal{A}_{score}(h, z, s, t) = I\left[s(h, z) > t\right] , \tag{2}$$

where $z = (x, y)$ denotes the target example and h denotes the target model, the indicator function $I[x]$ equals to 1 if x is true and 0 otherwise. $s(h, z)$ is referred to as the "membership score"[46] and can usually be calculated using loss(Loss)[50], confidence(Conf)[38], modified entropy(Mentr)[40], etc. The formulas for the above three methods are as follows:

$$s_{Loss}(h, z) = -l(h, z) = log(h(x)_y), \tag{3}$$

$$s_{Conf}(h, z) = \max_i log(h(x)_i), \tag{4}$$

$$s_{Mentr}(h, z) = (1-h(x)_y)log(h(x)_y)+\sum_{i\neq y} h(x)_i log(1-h(x)_i), \tag{5}$$

where $h(x)_y$ represents the confidence of the probability vector outputted by the h on y, l denotes cross-entropy loss.

2.2. Calibrated Membership Inference Attack

If non-member examples have low prediction difficulty, the target model may demonstrate a high level of confidence. Most early MIA [38, 39, 50]mistakenly assumed that the prediction difficulty of members and non-members is the same. This assumption is thought to be the primary cause of the high FPR problem. To tackle this issue, Watson et al.[46] proposed the difficulty calibration technique.

Specifically, we assume that the adversary has a reference dataset D_{ref} with the same distribution as the training set of the target model. He trains some reference models on the D_{ref} before performing the attack. Then, the calibrated membership scores should be calculated based on the results of the target example obtained on the target and reference models. Formally, we define the calibrated membership scores as:

$$s_{cal}(h, z) = s(h, z) - E_{g\leftarrow T(D_{ref})}[s(g, z)], \tag{6}$$

where T denotes the randomized training algorithm,$g \leftarrow T(D_{ref})$ denotes the machine learning model g generated by algorithm T through dataset D_{ref}. The calibrated attack is performed by setting a threshold on the calibrated score.

The goal of this calibration technique is to eliminate the interference of the example's own characteristics with the MIA, similar approaches have been used in the work of Sablayrolles et al.[37] and Carlini et al.[5]

2.3. Likelihood Ratio Attack

Likelihood Ratio Attack(LiRA) is a hypothesis testing based attack proposed by Carlini et al.[3] to strengthen the threat of MIA in low FPR scenarios. Let $\phi(h) = log(\frac{h}{1-h})$, for an attack sample $z = (x, y)$, where x represents input data and y represents the label. The probability distribution of $\phi(h(x)_y)$ computed by the model h containing z in the training set is denoted by Q_{in}, while Q_{out} denotes the probability distribution of the computed result without z in the training set. Based on the Neyman-Pearson Lemma[33], the threshold of online LiRA is given by Carlini et al.[3] as follows:

$$s_{LiRA}(h, z) = \frac{p(\phi(h(x)_y)|Q_{in})}{p(\phi(h(x)_y)|Q_{out})}, \tag{7}$$

where p denotes the conditional probability density function. Carlini et al.[3] assume that the $Q_{in/out}$ is Gaussian distribution, and in order to fit $Q_{in/out}$, a large number of reference models need to be trained for each attack sample. To avoid the huge computational effort, Carlini et al.[3] proposed offline LiRA, an attack based on one-sided hypothesis testing, where only a batch of models need to be trained for fitting the probability distribution Q_{out}. Specifically, this approach uses the following membership score:

$$s_{LiRA}(h, z) = 1 - Pr[X > \phi(h(x)_y)], X \sim Q_{out}. \tag{8}$$

3. Methodology

In this section we introduce our two-stage High Precision Membership Inference Attack (Two-stage HP-MIA). In Section 3.1 we describe HP-MIA formally using a game and illustrate the setup of this paper on adversary knowledge. In Section 3.2 and Section 3.3 we introduce the attack procedure and technical details of Two-stage HP-MIA.

3.1. Threat Model

This paper assumes that the adversary has only black-box access to the target model and an adversary dataset derived from the same distribution as the target model's training set. Through adversary dataset, the adversary can train shadow models and reference models to mimic the behavior of the target model. In addition to this, we assume that the adversary knows the stochastic training algorithm used by the victim in training the target model and the structure of the target model. Our assumptions about the adversary's knowledge are similar to most prior work[28, 38, 39, 40].

In contrast to the definition in Section 2.1, for HP-MIA, the adversary is more interested in the precision of the attack. Inspired by some previous work[3, 28, 50], we describe this process by defining the following game:

Definition 2 (HP-MIA game $G(Q, \mathcal{A}, T, n)$). *Let Q be a distribution over data points, $\mathcal{A}$ be an attack, T be a randomized training algorithm, n be a positive integer. The game proceeds as follows:*

1. *The challenger chooses a secret bit $b \leftarrow \{0, 1\}$ uniformly at random, and samples a training dataset $D \sim Q^n$.*
2. *If $b = 1$, the challenger randomly selects a record z in the training set D. Otherwise, the challenger samples a record z from the distribution Q(such that $z \notin D$).*
3. *The challenger trains a model $h \leftarrow T(D)$ on D and sends h and z to the adversary.*

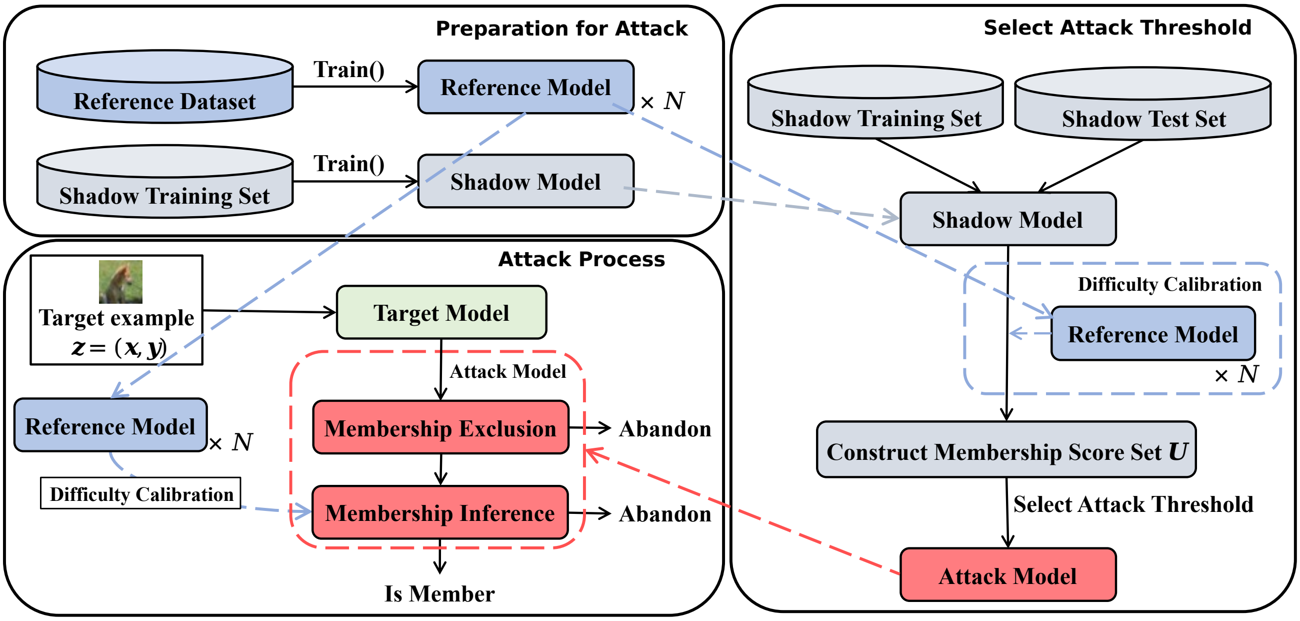


Figure 2: Overall Procedure of Two-stage HP-MIA

4. *The adversary tries to infer the secret bit as b', and performs an attack only if* $b' = 1$.
5.
$$G(Q, \mathcal{A}, T, n) = \begin{cases} 1 & b' = 1 \text{ and } b = 1. \\ 0 & b' = 1 \text{ and } b = 0. \end{cases}$$

Note the difference between Definition 2 and the game defined by Yeom et al[50]. We consider an "extremely cautious adversary": in order to achieve high precision, the adversary only considers an example as a member when the confidence level is high. Otherwise, the adversary abandons the attack. Therefore, we only consider the case $b' = 1$. In the HP-MIA game, the adversary wins when he launches an attack and successfully identifies a membership, and loses when he identifies a membership incorrectly. In reality, MIA may be used as the first step in some more powerful attacks[4, 5], so an error in MIA may lead to unnecessary losses. In this scenario, a high-precision attack is essential.

Given a target dataset D_{target}, $D_{adv} \bigcap D_{target} = \emptyset$, the adversary aims to identify members of the target model from the target dataset D_{target} with as high a precision as possible. For the above attack setup, the two important metrics we focus on are the precision(Pr) and recall(Recall) of the attack. We give a formal definition of HP-MIA by a constrained optimization problem:

Definition 3 (High-Precisionα Membership Inference Attack)**.** *Given a target model h, a target dataset D_{target}, $\alpha \in$* [0, 1] *is a precision constraint value. We call $\hat{\mathcal{A}}$ is a High-Precision α Membership Inference Attack (HPα-MIA) for h if $\hat{\mathcal{A}}$ satisfies:*

$$\begin{aligned} \hat{\mathcal{A}} &= \underset{\mathcal{A}_{score}}{argmax}\, Recall(\mathcal{A}, D_{target}), \\ &s.t. Pr(\mathcal{A}, D_{target}) \geqslant \alpha. \end{aligned} \tag{9}$$

where $\mathcal{A}_{score}$ denotes the threshold attack model, as shown in (2), $Pr(\mathcal{A}, D_{target})$ denotes the precision of $\mathcal{A}$ on D_{target}.

The formulae for Pr and Recall we will give in Section 4.1.4. We outline the whole process of Two-stage HP-MIA in Figure 2. The adversary needs to build reference model and shadow model before conducting an attack. Then, the adversary select the attack threshold based on the shadow model. When performing membership inference, the attacker first excludes some examples by membership exclusion attack, and then uses high-precision membership inference for the remaining examples.

3.2. Preparation for Attack

Before the attack, the adversary needs to build a series of models to imitate the behavior of the target model, which will be used for membership score calibration and attack model training. These models can be divided into reference model and shadow model because of their different uses. We assume that the adversary has a adversary dataset D_{adv} which from the same distribution as the training set of the target model, so we can construct the reference datasetD_{ref} for training the reference model and the shadow datasetD_{shadow} for training the shadow model. Note that the two data sets should be disjoint, i.e. $D_{ref} \bigcap D_{shadow} = \emptyset$.

The reference model is used to construct the calibrated MIA, The calculation of the calibrated membership score is shown in (6). The basic idea of calibrated attack is to judge whether the target model has learned the target example by comparing the prediction of the target model with that of the reference model. In general, the higher the number of reference models, the better the difficulty correction. We will discuss the effect of the number of reference models on the attack performance in Section 4.3.3.

Algorithm 1: Attack Process of Two-stage HP-MIA

```
Input: target model $h$, target record $z$, membership score used $s_0$ in the first stage and its threshold $t_0$, membership
  score $s_1$ used in the second stage and its threshold $t_1$
if $\underline{s_0(h, z) < t_0}$ then
    // Membership Exclusion
    return $\underline{\emptyset}$
else if $\underline{s_1(h, z) < t_1}$ then
    // Membership Inference
    return $\underline{\emptyset}$
else
    return $\underline{1}$
```

The shadow model is used to mimic the behavior of the target model just like the reference model, but it is used for the determination of the attack model threshold. The shadow training set is constructed from the adversary dataset, does not overlap with the reference training set due to reliance on the reference model for difficulty correction in shadow training. For convenience, we refer to the setup of Salem et al.[38] which only trains a shadow model for the attack.

3.3. Two-stage High Precision MIA

We propose a novel attack framework, which is divided into two steps: exclusion and inference. We adopt simple threshold model as the implementation of this attack, and therefore need to determine the membership scores(s_1 and s_2) and thresholds(t_1 and t_2) for both steps. The method for determining the thresholds will be shown in Section 3.4.

We use the cross-entropy Loss as the membership score s_1 in the first stage, which is calculated in (3). We excluded examples with large Loss values as non-members in the first stage. This is a completely opposite approach to Yeom et al.[50]. Our intuition is that the neural network is able to fit the training data well, so examples with large Loss values have a high probability of being non-members.

For the examples that are not excluded in the first stage, we further perform HP-MIA on them. Note that a score-based attack without considering the example difficulty makes it difficult to achieve a high-precision MIA. Therefore, we use the calibrated Loss as the membership score s_2 in the second stage, which is calculated in (6).

Algorithm 1 demonstrates the process of performing Two-stage HP-MIA on a single target example. Note that in this paper, we consider an extremely cautious adversary who only judges the target example as a member when he is very sure, otherwise he will abandon the attack.

3.4. Select Attack Threshold

Two-stage HP-MIA needs relies on two thresholds, the threshold t_0 for membership exclusion attack and the threshold t_1 for membership inference attack. Algorithm 2 shows our process of choosing the optimal threshold. Function "Two-stage threshold" demonstrates the process of selecting the threshold for Two-stage HP-MIA. The adversary needs to set the value of α, where a higher α indicates a higher requirement for algorithm precision. The value of β does not need to be set, as the algorithm will automatically search for the optimal β value.

In practice, we do not have access to the training set of the target model and thus cannot solve the optimization problem (9) on the real dataset D_{target}. According to our assumptions on adversary knowledge in Section 3.1, we can construct the member dataset D^{in}_{shadow} and non-member dataset D^{out}_{shadow} of the shadow model for supervised training of the attack model. As for the score-based attacks, the process of constructing a attack is actually finding an optimal threshold, and we choose the appropriate threshold in the following membership score set U:

$$U(h, s, D_{shadow}) = \left\{ u_i = \frac{s(h, z_i) + s(h, z_{i+1})}{2} \ : \ z_i \in D_{shadow} \right\}, \tag{10}$$

where $s(h, z_i) \leqslant s(h, z_{i+1}), i = 1, 2, ..., m$,and m is the amount of members of the shadow data set. Specifically, we iterate through all elements in U and calculate the attack precision and recall corresponding to each element, select the subset U' that satisfies precision $\geqslant \alpha$, and return the element in U' that corresponds to the largest recall.

We denote the precision constraint value of the membership inference attack as α and the precision constraint value of the membership exclusion attack as β. α is set by the adversary according to his requirements.

Our method to obtain the two thresholds for Two-stage HP-MIA relies on the "Membership Exclusion threshold" and "Membership Inference threshold" functions. These two functions calculate the optimal threshold t_0 and $t_1$of Membership Exclusion and Membership Inference for a given precision constraint. As shown in Algorithm 2, for a given β and D_{shadow}, the attacker first obtains the threshold t_0 by the "Membership Exclusion threshold" function, and excludes some non-members according to t_0 to obtain the remaining target sample $D_{remaining}$. After that, according to the a specified α givenby the adversary, the optimal threshold t_1 and the number of correctly identified members TP for $D_{remaining}$ is obtained by the "Membership Inference threshold" function.

The adversary constructs the optimal attack by continuously adjusting the value of β using the trained shadow model. They select the optimal thresholds t^{opt}_0 and t^{opt}_1 that maximize TP. Using the two determined thresholds t_0^{opt} and t_1^{opt}, the adversary proceeds to attack the victim model.

4. Experiments

In this section, we first show the experimental setup in Section 4.1, including dataset, target model architecture, training setup, and evaluation metrics of the attack model. Then, we evaluate our attack and compare it with the previous MIA in Section 4.2. Finally, we analyze the impact of some factors on the attack performance in Section 4.3.

4.1. Experimental Setup

4.1.1. Dataset

We conducted experiments on several baseline datasets of different complexity: MNIST[26], Fashion-MNIST (F-MNIST)[47], CIFAR10[24], Purchase100[1], and Texas100[2]. We randomly divide each of these datasets into six datasets, two of which are used as the training set D_{target}^{in} and test set D_{target}^{out} for the target model, two of which are used as the training set D_{shadow}^{in} and test set D_{shadow}^{out} for the shadow model, and the remaining two datasets are used as the reference training set for the training of the reference model.

For MNIST and F-MNIST, the target model test set has 10,000 images, and the remaining datasets have 12,000 images each. For CIFAR10, each dataset has 10,000 images. For Purchase100, each dataset has 20,000 records, and for Texas100, each dataset has 10,000 records.

4.1.2. Model architectures

The target model for MNIST and Fashion-MNIST is a small CNN with two convolutional layers and a maximum pooling layer, two convolutional layers with 24 and 48 output channels, and a kernel size of 5, followed by a fully connected layer with 100 neurons as the classification head, and we use Tanh as the activation function. For CIFAR10, we use AlexNet(CIFAR10-A)[25] and Wide-ResNet (CIFAR10-W, with depth 28 and width 2)[51] as the structure of the target model. For Purchase100 and Texas100, we refer to the work of Song et al.[40] and use a multilayer perceptron (MLP) as the target model with four hidden layers with the number of neurons of 1024, 512, 256, and 128, respectively, and use Tanh as the activation function.

4.1.3. Model tranning

We use Adam optimizer[22] to train the target model. For the target models of MNIST, F-MNIST, CIFAR10-A, CIFAR10-W, Purchase100 and Texas100, we set the learning rates to 0.001, 0.001, 0.0003, 0.001, 0.0001 and 0.0002, respectively, the corresponding batch sizes to 100, 100, 100, 50, 200, and 50, respectively, and the number of iterations to 100, 100, 200, 200, 200 and 200, respectively. To reduce the degree of overfitting, for MNIST, FMNIST, Purchase100 and Texas100, we use L2 regularization and set $\lambda = 0.0005$. For CIFAR10, we use data augmentation(Random Horizontal Flip and Random Rotation) to improve the accuracy of the model and prevent overfitting. Table 1 shows the performance of several target models.

The shadow and reference models are trained using the same training algorithm and hyperparameters as the target model. For each dataset, we train a shadow model on the shadow dataset for attack model construction and 20 reference models on the reference dataset for calculating the calibrated score.

4.1.4. Success metrics

We consider an extremely cautious MIA adversary who wants to identify as many members as possible with high precision. We use the following metrics to evaluate the attack performance: number of correctly identified members (TP), recall (Recall) and precision (Pr). Recall and Pr are calculated as follows:

$$\text{Recall} = \frac{\text{TP}}{|D_{target}^{in}|}, \qquad \text{Pr} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

where FP denotes the number of non-members identified as members by the attack model. We will show the PR curves to present the experimental results more comprehensively. Referring to some recent work[3, 46, 49], we will also show the ROC curve and calculate the AUC (area of the ROC curve).

4.2. Attack Evaluation

4.2.1. Effectiveness of two-stage HP-MIA

To highlight the effect of Two-stage attack, we use two calibrated attacks, C-Loss and C-Conf, to compare with our attacks. These two attacks use Loss and Confidence as membership scores, respectively, and use difficulty calibration[46] to remove the effect of example difficulty. It is worth noting that C-Loss can be viewed as a direct HP-MIA without using the membership exclusion technique.

In addition, we also consider Attack R from the work of Ye et al. and LiRA proposed by Carlini et al. Attack R proposed by Ye et al. also belongs to a threshold attack, but the process of finding the threshold is different. The adversary needs to be given a confidence requirement r. The threshold is computed using the r-percentile of the loss histogram of the target example on the reference model. The target example is judged to be a member if the loss value computed on the target model is less than the threshold. For the attack of Ye et al. we use linear interpolation method to calculate the continuous percentile. For Carlini et al.'s attack, considering the computational cost, we only implement offline LiRA and calculate the membership scores according to Equation (8) to achieve HP-MIA. For MNIST, F-MNIST, Purchase100 and Texas100, we train 100 reference models to implement Ye et al.'s attack and offline LiRA, and for CIFAR10-A and CIFAR10-W, we train 60 reference models.

We can only implement HP-MIA that satisfies the precision constraint on the shadow model, so the precision on the target model may be biased, and the bias size depends on

[1] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data
[2] https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm

Algorithm 2: Select Attack Threshold

```
function Membership Exclusion threshold :
    Input: shadow dataset $D_{shadow}$, target model $h$, membership score $s$ and precision constraint value $\beta$.
    Initialize $U'$ and $TPset$ to $\emptyset$
    Construct the set $U(h, s, D_{shadow})$ according to Equation (10), $m \leftarrow |U|$
    for $i \leftarrow 1 : m$ do
        // Find the optimal threshold $t_0$ in $U(h, s, D_{shadow})$ at the specified precision $\beta$
        Judge examples in $D_{shadow}$ with membership scores less than $u_i$ as non-members
        Calculate the number of correctly identified non-members ($TP$) and the precision ($Pr$) of non-member
          identification
        if $Pr \geqslant \beta$ then
            $U' \leftarrow U' \cup u_i$
            $TPset \leftarrow TPset \cup TP$
    if $U' == \emptyset$ then
        return fail
    $k \leftarrow \max_{k=1,2,...,n} TPset$
    $t_0 \leftarrow U'_k$
    return $t_0$
function Membership Inference threshold :
    Input: shadow dataset $D_{shadow}$, target model $h$, membership score $s$ and precision constraint value $\alpha$.
    Initialize $U'$ and $TPset$ to $\emptyset$
    Construct the set $U(h, s, D_{shadow})$ according to Equation (10), $m \leftarrow |U|$
    for $i \leftarrow 1 : m$ do
        // Find the optimal threshold $t_1$ in $U(h, s, D_{shadow})$ at the specified precision $\alpha$
        Judge examples in $D_{shadow}$ with membership scores greater than $u_i$ as members
        Calculate the number of correctly identified members ($TP$) and the precision ($Pr$) of member identification
        if Pr $\geqslant \alpha$ then
            $U' \leftarrow U' \cup u_i$
            $TPset \leftarrow TPset \cup TP$
    if $U' == \emptyset$ then
        return fail
    $TP, k \leftarrow \max_{k=1,2,...,n} TPset$
    $t_1 \leftarrow U'_k$
    return $t_1, TP$
function Two-stage threshold :
    Input: shadow dataset $D_{shadow}$, target model $h$, membership score used $s_0$ in the first stage, membership score
      $s_1$ used in the second stage and precision constraint value $\alpha$.
    Initialize $TP^{opt}$, $t_0^{opt}$ and $t_1^{opt}$ to 0
    for $\beta \leftarrow 0, 1; step = 0.001$ do
        // Adjust the $\beta$ value to get the optimal recall
        $t_0 \leftarrow$ Membership Exclusion-threshold($D_{shadow}, h, s_0, \beta$)
        $D_{remaining} \leftarrow \left\{ z_i : z_i \in D_{shadow}, s(h, z_i) \geqslant t_0 \right\}$
        $t_1, TP \leftarrow$ Membership Inference-threshold($D_{remaining}, h, s_1, \alpha$)
        if $TP > TP^{opt}$ then
            $TP^{opt} \leftarrow TP$
            $t_0^{opt} \leftarrow t_0$
            $t_1^{opt} \leftarrow t_1$
    return $t_0^{opt}, t_1^{opt}$
```

Table 1
Accuracy of the target model

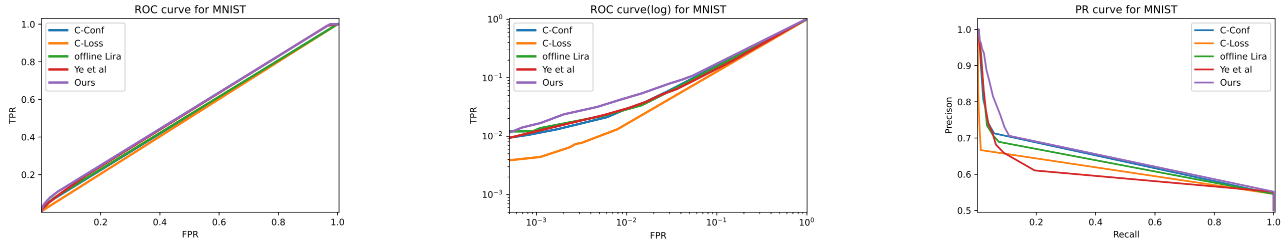
	MNIST	F-MNIST	CIFAR10-A	CIFAR10-W	Purchase100	Texas100
Model	CNN	CNN	AlexNet	WideResNet	MLP	MLP
Train_Acc	100%	99.85%	100%	99.53%	99.98%	98.94%
Test_Acc	98.69%	88.71%	70.61%	80.22%	83.16%	45.32%

Table 2
Evaluation of various data sets, model structures, and MIA methods, $\alpha = 98\%$

		MNIST	F-MNIST	CIFAR10-A	CIFAR10-W	Purchase100	Texas100
C-Loss	TP	17	27	2	3	15	107
	Recall	0.14%	0.23%	0.02%	0.03%	0.08%	1.07%
	Pr	**100%**	93.10%	**100%**	**100%**	93.75%	**97.27%**
C-Conf	TP	79	187	139	91	7	203
	Recall	0.66%	1.56%	1.39%	0.91%	0.04%	2.03%
	Pr	98.75%	92.12%	**100%**	96.81%	87.80%	94.86%
offline LiRA	TP	43	10	**1803**	**770**	**32**	31
	Recall	0.36%	0.08%	**18.03%**	**7.70%**	**0.16%**	0.31%
	Pr	**100%**	**100%**	95.40%	84.34%	94.12%	86.11%
Two-stage(Ours.)	TP	**86**	**233**	481	99	20	**520**
	Recall	**0.72%**	**1.94%**	4.81%	0.99%	0.10%	**5.20%**
	Pr	98.85%	88.59%	98.57%	95.19%	**100%**	97.20%
$\alpha = 98\%$							

Table 3
Evaluation of various data sets, model structures, and MIA methods,$\alpha = 100\%$

		MNIST	F-MNIST	CIFAR10-A	CIFAR10-W	Purchase100	Texas100
C-Loss	TP	17	27	2	3	15	27
	Recall	0.14%	0.23%	0.02%	0.03%	0.08%	0.27%
	Pr	**100%**	93.10%	**100%**	**100%**	93.75%	96.43%
C-Conf	TP	52	104	139	56	7	**203**
	Recall	0.43%	0.87%	1.39%	0.56%	0.04%	**2.03%**
	Pr	**100%**	93.70%	**100%**	94.92%	87.80%	94.86%
offline LiRA	TP	43	10	**1803**	**770**	**32**	31
	Recall	0.36%	0.08%	**18.03%**	**7.70%**	**0.16%**	0.31%
	Pr	**100%**	**100%**	95.40%	84.34%	94.12%	**86.11%**
Two-stage(Ours.)	TP	**85**	**114**	110	90	20	**145**
	Recall	**0.71%**	**0.95%**	1.10%	0.90%	0.10%	**1.45%**
	Pr	**100%**	**100%**	94.23%	94.74%	**100%**	**100%**
$\alpha = 100\%$							

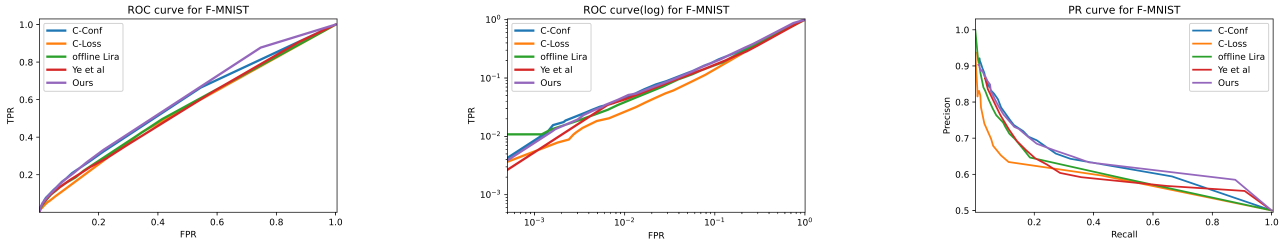


(a) ROC curve

(b) ROC curve(log)

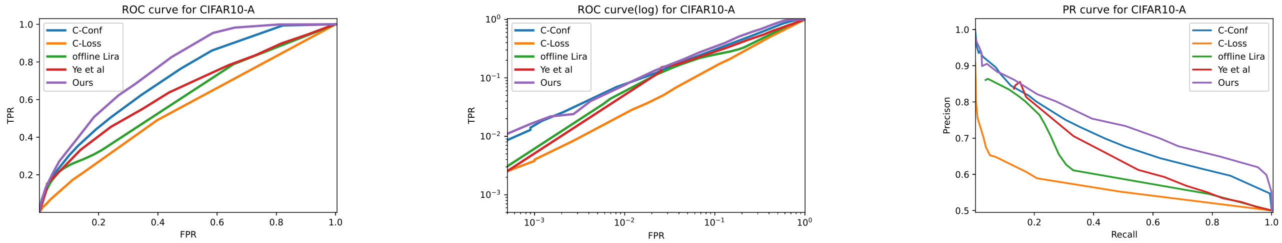
(c) PR curve

Figure 3: ROC curve and PR curve for MNIST



(a) ROC curve (b) ROC curve(log) (c) PR curve

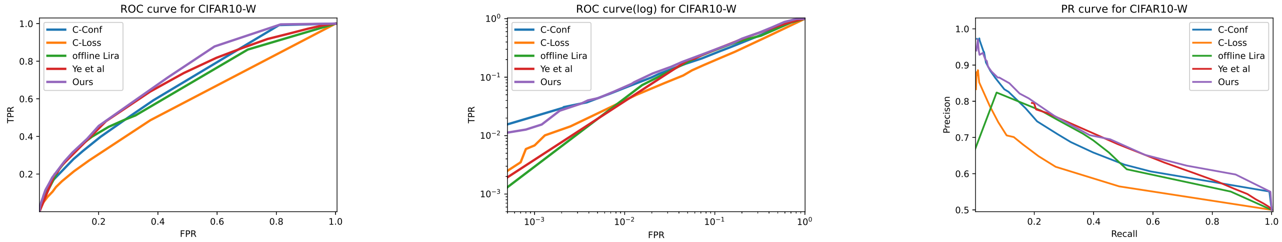
Figure 4: ROC curve and PR curve for F-MNIST



(a) ROC curve (b) ROC curve(log) (c) PR curve

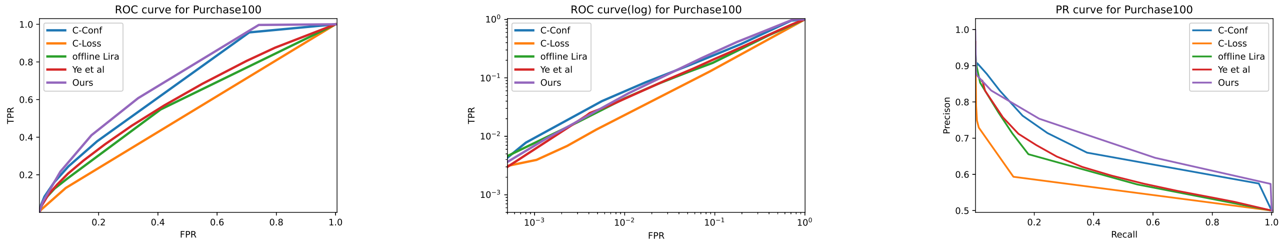
Figure 5: ROC curve and PR curve for CIFAR10-A

how close the shadow model is to the target model. Most of the time, as the precision constraint value rises, the accuracy of the attack on the target model becomes higher. Since the method of Ye et al. is not suitable for the direct construction of HP-MIA, we compare it with it here. Table 2 and Table 3 show the attack performance of the three attacks when α is set to 0.98 and 1. Bolded characters indicates the best result for a specific metric (e.g., TP, Recall) among different methods. The Two-stage attack consistently identifies the



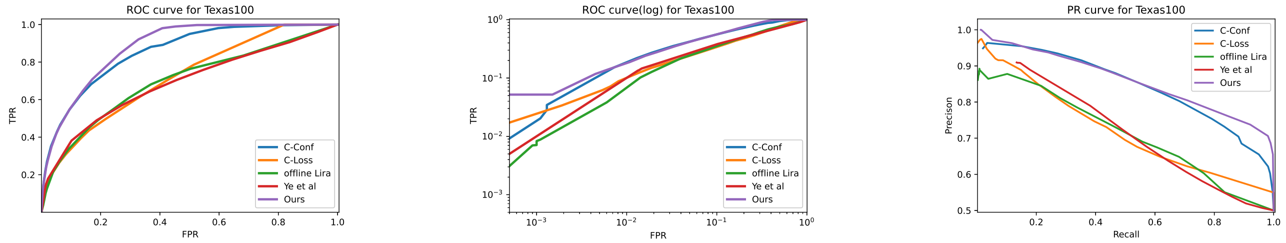
(a) ROC curve (b) ROC curve(log) (c) PR curve

Figure 6: ROC curve and PR curve for CIFAR10-W



(a) ROC curve (b) ROC curve(log) (c) PR curve

Figure 7: ROC curve and PR curve for Purchase100



(a) ROC curve

(b) ROC curve(log)

(c) PR curve

Figure 8: ROC curve and PR curve for Texas100

Table 4
AUC of different attacks on different datasets

	MNIST	F-MNIST	CIFAR10-A	CIFAR10-W	Purchase100	Texas100
C-Loss	0.5027	0.5439	0.5538	0.5705	0.5207	0.7170
C-Conf	0.5151	0.5831	0.7124	0.6660	0.6728	0.8540
offline LiRA	0.5175	0.5529	0.6057	0.6478	0.5820	0.6975
Ye et al	0.5351	0.5500	0.6419	0.6831	0.6052	0.6887
Two-stage(Ours.)	**0.5394**	**0.6005**	**0.7673**	**0.7122**	**0.7070**	**0.8786**

most members on all models at various precision constraint settings.

When we set $\alpha = 0.98$ and $\alpha = 1$, we find that Two-stage HP-MIA can always identify more members with higher accuracy. For example, for MNIST, when the precision is 100%, our attack has identified the most samples. For the Texas100, other attacks cannot achieve 100% accuracy, while Two stage HP-MIA recognizes more samples while achieving 100% precision. Unfortunately, when the same α is set, the accuracy of different attacks is different, which is not convenient for us to compare better.

Referring to the previous work[3, 49], we further show the ROC curve and PR curve of the attack, and use the ROC (log) curve to show the attack effect under low FPR. Note that some attacks are difficult to achieve high precision attacks on some datasets, e.g., Ye et al.'s method struggles to achieve precision above 0.95 on any dataset other than MNIST. Therefore, the curves of some attacks are incomplete. As shown in Figure 3 - 8, We found that the curve area of Two stage HP-MIA was the largest in most cases, and reached high TPR at a fast speed. Table 4 shows the AUC values of different attacks on different target models, and it can be found that our methods have the largest AUC.

4.2.2. *Failure of the direct Overfitting-based MIA*

Throughout this paper, we refer to MIAs that attack using only uncorrected membership scores as Overfitting-based MIAs. This type of attack does not consider the impact of sample characteristics on privacy leakage. Overfitting-based MIAs fails under the requirement of high precision, so we did not compare these methods directly with our attacks in Section 4.1. We construct HP-MIA using three membership scores, Loss[50], Conf[38, 39] and Mentr[40], respectively, and Table 5 shows the performance of these attacks on different datasets. Note that Algorithm1 will return a threshold that achieves the maximum accuracy when it finds that it cannot find a threshold that satisfies the accuracy requirement on the shadow model. We find that these attacks are completely unable to achieve the precision we require, even though we only set $\alpha = 0.9$.

4.3. Ablation Study

4.3.1. *Membership exclusion precision constraint value*

Compared to other score-based attacks, Two-stage HP-MIA has two thresholds and thus takes more time in building the attack model. Some score-based MIAs provide an empirical threshold, for example, Waston et al.[46] point out that the empirical threshold for calibration attacks is a value slightly greater than 0. The adversary needs to adjust the precision constraint value β of the membership exclusion attack to achieve the most powerful attack when constructing Two-stage HP-MIA, and we would like to know if β has an empirical value as a reference. We conduct experiments on the MNIST and CIFAR10 datasets to observe the performance of Two-stage HP-MIA when different β are set.

Unfortunately, we find that there is no relatively general precision constraint value for the membership exclusion attack. Figure 9 shows our experimental results, and we find that the value of β has different effects on the two datasets.

Table 5
Performance of overfitting-based MIA under high-precision constraints, All attacks on Purchase100 result in $TP + FP = 0$

	MNIST	F-MNIST	CIFAR10-A	CIFAR10-W	Purchase100	Texas100
Loss	52.63%	33.33%	65.64%	60.48%		74.34%
Conf	60.26%	33.33%	61.08%	53.94%		74.34%
Mentr	52.63%	33.33%	59.80%	52.90%		74.49%
α = 90%						

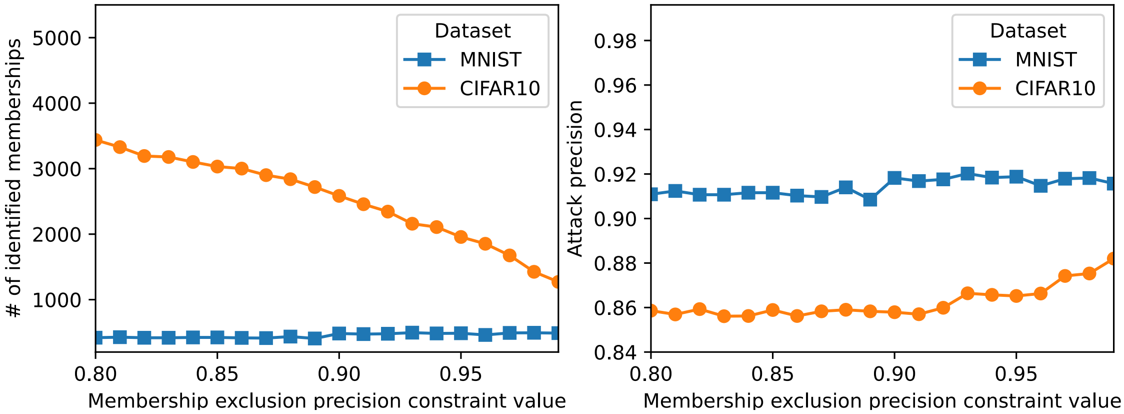


Figure 9: Effect of precision constraint value for membership exclusion. We found that there is no more general precision constraint value for the membership exclusion technique, and the impact of different different precision constraint values on CIFAR10 and MNIST is not the same.

the number of exposed examples on the MNIST dataset increases slowly with larger β, while the number of identified examples on CIFAR10 decreases rapidly with larger β. HP-MIA require capturing more detailed model features and example characteristics, so it is difficult to have a general reference value. In order to construct more robust attacks, it is necessary to spend more time to optimize the thresholds.

4.3.2. l_2 regularization

l_2 regularization is a relatively simple defense technique for member inference attacks[21, 28, 39]. We assume that the adversary is unknown to the defense used by the victim, and both the shadow model and the reference model are trained using the original algorithm. Table 6 shows the performance of the target model with the regularization technique and the inference effect of the Two-stage attack. As a common method to overcome overfitting, regularization can prevent the leakage of membership privacy to some extent.

In general, the number of memberships that can be inferred by Two-stage decreases significantly as λ grows. It is worth noting that lower levels of l_2 regularization may not reduce the attack precision as well. For CIFAR10, the attack precision at $\lambda < 0.001$ is instead higher than that without the l_2 regularization method.

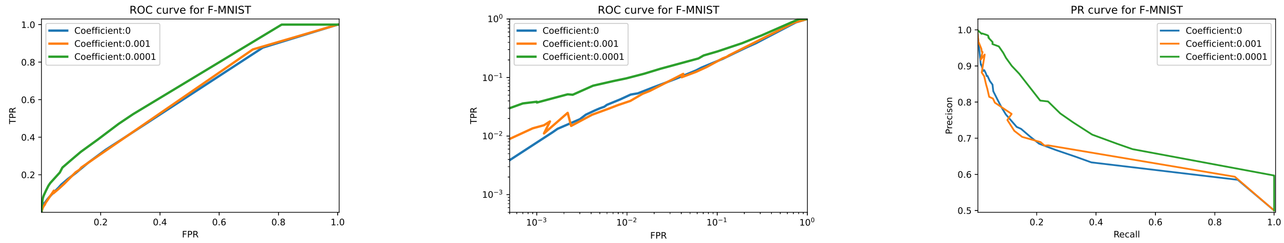
Figure 10 and 11 show the ROC curves and PR curves under different regularization factor settings. Our attack requires the use of two thresholds, and the first stage of the attack depends on the degree of overfitting of the target model. Therefore, the attack curves vary widely under different degrees of regularization. We believe that L2 regularization can be an effective defense against MIA, but it is not absolute. For F-MNIST, we find that the target model using regularization has a greater degree of privacy leakage instead.

4.3.3. Number of reference models

Figure 14 shows the TP and Pr of Two-stage HP-MIA with different number of reference models ($\alpha = 0.9$). we use the datasets MNIST, F-MNIST and CIFAR10. in general, the attack precision receives little effect from the number of reference models, and the difference between the maximum and minimum precision on MNIST, F-MNIST and CIFAR10 are 0.97%, 0.89% and 0.43%. Besides, using fewer reference models may lead to a lower number of identified memberships. The TP of Two-stage HP-MIA using only one reference model is the least on the target model of three datasets. However, we found that the increase in the number of reference models did not significantly improve the TP except for CIFAR10. For MNIST,the highest number of identified members was for the attack using 8 reference models, with 621, while the attack using 20 reference models identified 590 memberships. For F-MNIST, the highest number of

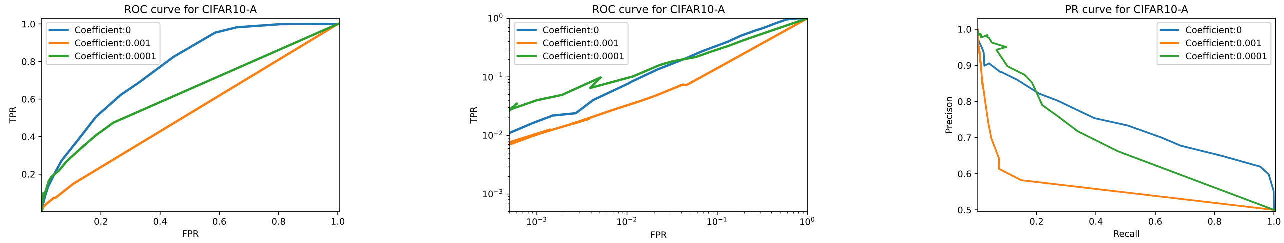
Table 6
Two-stage experimental results on the target model using regularized defense, the datasets are F-MNIST and CIFAR10-A

Dataset	λ	Train_Acc	Test_Acc	TP	Pr
F-MNIST	0	100%	88.74%	1406	89.21%
	0.0001	100%	88.77%	283	84.99%
	0.0003	100%	88.16%	111	84.09%
	0.0005	99.96%	88.14%	83	81.37%
	0.0007	99.93%	88.34%	75	81.52%
	0.001	99.68%	87.87%	47	79.66%
	0.005	95.12%	88.50%	0	0
	0.01	89.68%	86.50%	0	0
CIFAR10-A	0	100%	70.61%	3866	85.34%
	0.0001	99.92%	67.88%	2044	92.57%
	0.0003	99.74%	69.13%	1781	90.77%
	0.0005	100%	68.85%	1531	92.79%
	0.0007	99.88%	70.23%	583	88.74%
	0.001	99.50%	66.59%	505	90.02%
	0.005	98.91%	69.26%	23	85.19%
	0.01	92.26%	62.26%	10	76.92%



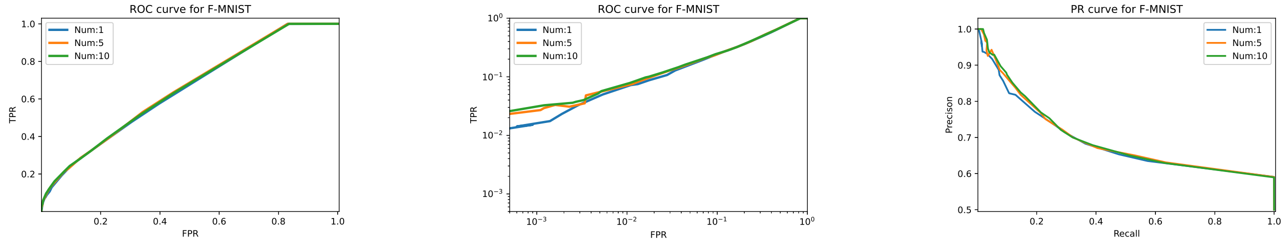
(a) ROC curve (b) ROC curve(log) (c) PR curve

Figure 10: ROC curve and PR curve for F-MNIST, Target models using different regularization factors.



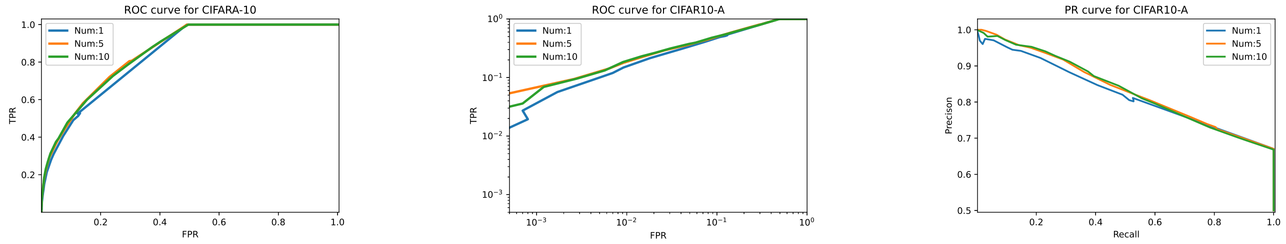
(a) ROC curve (b) ROC curve(log) (c) PR curve

Figure 11: ROC curve and PR curve for CIFAR-A, Target models using different regularization factors.



(a) ROC curve (b) ROC curve(log) (c) PR curve

Figure 12: ROC curve and PR curve for F-MNIST(Two-stage HP-MIA using different number of reference models)



(a) ROC curve

(b) ROC curve(log)

(c) PR curve

Figure 13: ROC curve and PR curve for CIFAR-A(Two-stage HP-MIA using different number of reference models)

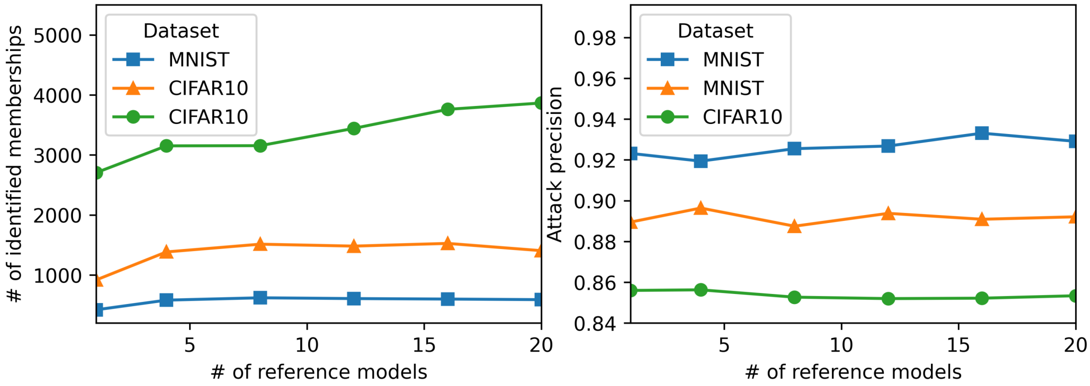


Figure 14: The TP and Pr of Two-stage HP-MIA with different number of reference models. Attack precision receives little effect from the number of reference models ($\alpha = 0.9$). Using few reference models (e.g., one) may result in a low number of identified memberships.

identified memberships was for the attack using 16 reference models, with 121 more members identified than when using 20 reference models. We do not recommend training too many reference models for calculating the calibrated score when not planning to spend too much time to deploy the attack.

Figure 12 and Figure 13 show the results of our experiments using ROC curves and PR curves. We find that there will be some reduction in TPR at low FPR when fewer reference models are used. But overall, we think that the number of reference models does not have a significant impact on the attack.

4.3.4. The methodology used in the second stage

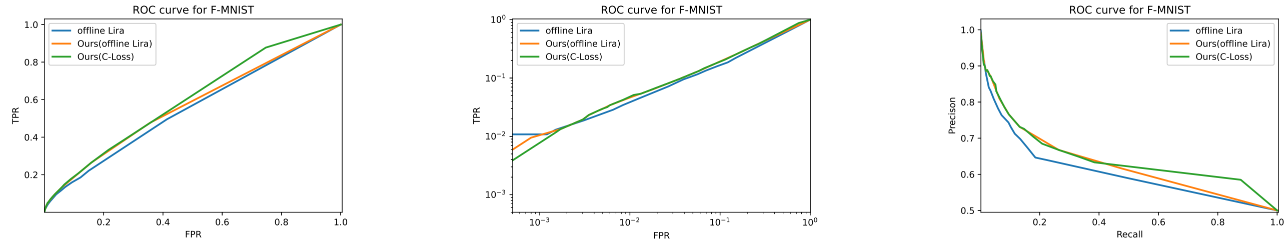
Two-stage HP-MIA is a generalized framework that allows an adversary to use a variety of different attacks in two stages. Due to the need for high precision, we recommend using methods that consider sample difficulty in the second stage. Combining exclusion attacks with methods other than C-Loss can construct new attacks. In the second stage of Two-stage HP-MIA, we use the offline Lira method proposed by Carlini et al.[3] and test the effectiveness of this attack on F-MNIST and CIFAR10-A. Our experimental setup and model structure on both datasets are consistent with the prior.

Figure 15 and Figure 16 show the experimental results of this method with the method using C-Loss in the second stage phase and the offline Lira method. According to the ROC curve and PR curve, we found that the performance of offline Lira combined with the exclusion attack has been improved to some extent, especially the AUC value (area of the ROC curve) has increased significantly. However, the AUC value of this method is still smaller than the previous method using C-Loss in the second stage. It should be noted that the computational cost of offline Lira is higher than that of C-Loss. Therefore, we prefer to use C-Loss in the second stage.

5. Discussion

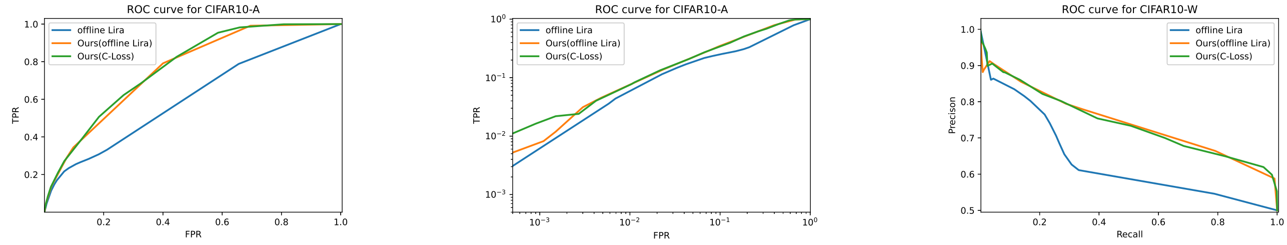
5.1. Why Our Attacks Work?

First, the use of difficulty calibration techniques is necessary to achieve a high precision attack. According to the experimental findings shown in Table 5 we find that direct attacks based on overfitting fail under high precision requirements, even for simple MLPs with an accuracy of only 43.89%. as different examples contribute differently to the model, it is difficult to achieve reliable MIA by simply considering the difference in prediction results between members and non-members of the model. Thus, it seems that the serious privacy crisis of being identified by the adversary



(a) ROC curve (b) ROC curve(log) (c) PR curve

Figure 15: ROC curve and PR curve for F-MNIST, Our (C-Loss) and Our (offline Lira) denote the two-stage HP-MIA using the C-Loss method and offline Lira method in the second phase, respectively.



(a) ROC curve (b) ROC curve(log) (c) PR curve

Figure 16: ROC curve and PR curve for CIFAR-A, Our (C-Loss) and Our (offline Lira) denote the two-stage HP-MIA using the C-Loss method and offline Lira method in the second phase, respectively.

with high precision is more related to the characteristics of the example itself.

More importantly, our attack no longer employs membership inference based on overfitting membership scores, but simply uses it as an exclusion technique. This method was inspired by the tendency of neural networks to overfit hard-to-predict example. And our experimental results amply demonstrate the effectiveness of this simple technique. Especially in scenarios where attacks are very costly, we believe that membership exclusion techniques are necessary to avoid launching unnecessary attacks. Comparing Two-stage HP-MIA with the exclusion technique and other attacks, Two-stage HP-MIA is able to identify more samples while ensuring high precision.

5.2. Which Examples are Dangerous and Which Examples are Safe?

The success of the difficulty calibration technique provides ample evidence that considering example characteristics is necessary for MIA, and that example specificity is closely related to membership privacy leakage. When looking at this problem from the perspective of a victim or defender, we may be concerned about those examples whose privacy is vulnerable to leakage. Since the current MIA considering example characteristics focuses only on the prediction difficulty of examples, we will only discuss the relationship between sample difficulty and privacy here, and we leave the exploration about the other example properties to future work.

We use the mean value of a example's Loss on the reference model as an indicator of its difficulty:

$$\tau(z) = E_{g\leftarrow T(D_{refernce})}\left[l(g,z)\right]$$

We conducted experiments on four datasets: MNIST, CIFAR10, Purchase100, and Texas100. We used the 20 reference models trained on each data from previous experiments to calculate the difficulty scores, using the average of the losses on the reference models as an estimate of the difficulty scores per sample. For CIFAR10, the model structure we used was AlexNet. The member datasets were divided into two parts, the exposed data identified by Two-stage HP-MIA with high prediction($\alpha = 90\%$) and the hidden data not identified, and the difficulty scores were calculated for each of the two types of data.

Figure 17 shows the frequency histograms of the difficulty scores in the four datasets. Interestingly, we found that for samples with very low Loss values (close to 0) on the reference model, our attacks were difficult to identify. While other difficult samples were identified with high accuracy. In general, sample security in the dataset is correlated with its prediction difficulty. The strong memory capability makes the neural network easy to show high confidence in the training set data, so if a sample has a high prediction difficulty but has a very low Loss value in can in the neural network, the adversary can confidently guess that it belongs to the training set.

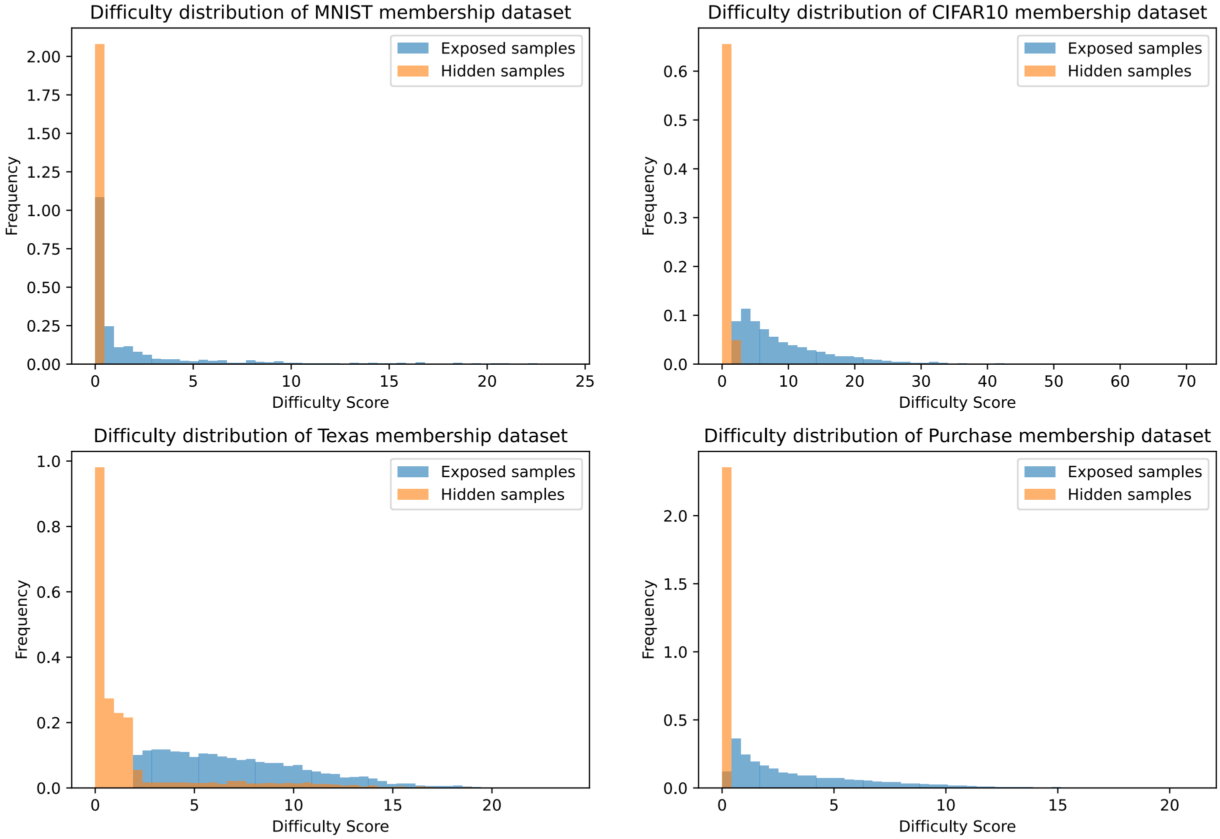


Figure 17: Difficulty score distribution of membership datasets of MNIST, CIFAR10, Purchase100 and Texas100. Examples with high predicted difficulty are more likely to be exposed, while simple samples are safer

Note that privacy leakage for hard examples is caused by a combination of example difficulty and model characteristics (overfitting). For the easy example, it is safer because the difference in predictive behavior for it in the model trained with it and the model trained without it is smaller.

It is worth mentioning that hard examples receive more attention than easy examples during the training of deep neural networks. When the model was trained using cross-entropy, hard examples contributed a larger gradient than easy examples in the back-propagation[36, 54]. In addition to this, Kishida et al.[23] found that hard examples contributed more to the generalization of the model than easy examples, suggesting that it is undesirable to remove hard examples from the dataset to guarantee privacy.

5.3. Comparison with Previous Works

The computational cost of the membership inference attack consists of two main components, one is the computational cost of training the shadow and reference models, and the other is the computational cost of optimizing the thresholds.

In our experiments, Ye et al.'s method does not require experimental shadow models, and all other attacks (including ours) train only one shadow model. Carlini et al.[3] and Ye et al.[49] design attack strategies from hypothesis testing, and their approaches require training a large number of reference models. For example, online LiRA train $2n$ reference models for each target record to fit the distribution Q_{in} and distribution Q_{out}. Offline LiRA and the method of Ye et al. train $2n$ reference models, and these $2n$ models can be used against all target records for the attack process. For m records, Online LiRA requires training $(m+1) \times n$ reference models, while Offline LiRA and the method of Ye et al. require training $2n$ reference models. Unlike their work, we do not use a large number of reference models to fit the probability distribution, but only a small number of models (e.g., $1 \sim 20$) for difficulty calibration. As a result, our attack incurs significantly lower (About $5 \sim 10$ times less) computing cost than the work of offline LiRA and Ye et al., and is close to that of Waston et al.[46]. As shown in figure 14, the effect of Two-stage HP-MIA did not receive a significant impact even with a smaller number of reference models. And unlike the work of Waston et al.[46], we used an overfitting-based attack to exclude non-member records, thus improving the precision of the calibrated attack.

Two-stage HP-MIA consumes more time in optimizing the thresholds since it has two thresholds. As shown in Algorithm 2, Two-stage HP-MIA needs to adjust the value of β several times and search for t_1 and t_2 after each adjustment of β, respectively. While other attacks only need to search a threshold value. Assuming that the adversary adjusts the β N (1001 in Algorithm 2) times while searching for the threshold, the cost of searching for the threshold in Two-stage HP-MIA is about $2N$ times as much as the other methods.

It should be noted that the cost of searching for thresholds is actually significantly less than the cost of training the reference model, so the overall computational cost of Two-stage HP-MIA is still less than the methods of Carlini et al. and Ye et al.

5.4. The Limitations of this Work

Based on the experimental results, we found that the precision of the final attack may be slightly less than α, which is due to the difference between the shadow model and the target model. In addition, as in previous work[3, 8, 39, 50], we assume that the training algorithm, model structure, and training set distribution used by the shadow model are independent identically distributed to the target model, which is a rather rigorous hypothesis in practice.

6. Related Work

6.1. Privacy Attacks against Machine Learning

Privacy attacks against machine learning model reveal the privacy risks of training data. Membership inference attacks(MIA)[39] aim to infer whether the target example belongs to the training set, and some current work[6, 8, 31, 34, 44] has generalized MIA to different application scenarios. Property inference attack was proposed by Ganju et al.[15] and this attack was demonstrated to be effective in extracting special attribute information from the model[45, 55]. Some work attempts to recover training data directly. Zhu et al.[56] proposed a method to recover data using gradients, which triggered some research on gradient information leakage[2, 9, 30, 53]. Carlini et al.[4, 5] showed that training data can be extracted by querying model in a black box scenario. Recent work[43] has found that it is possible to improve privacy attacks by poisoning the target model.

6.2. Privacy-Preserving Machine Learning

Differential privacy (DP) proposed by Dwork et al.[13, 14] can provide strong privacy guarantees for machine learning models. Abadi et al.[1] achieve DP training by cropping and adding noise to the gradients during training. Dong et al.[10] analyze the privacy leakage problem from the perspective of hypothesis testing and propose Gaussian DP. Unfortunately, differential privacy mechanisms can impair the performance of the target model[48]. Since MIA is considered as a fundamental privacy attack[3], many works[18, 19, 32] have developed corresponding privacy-preserving frameworks that utilize various strategies to defend against MIA and guarantee acceptable model accuracy.

6.3. Quantifying Privacy Risks of Machine Learning

Machine learning privacy attacks represented by MIA are often used as privacy metrics for models. Liu et al.[27] synthesized a variety of privacy attacks to evaluate the privacy leakage of the model through extensive experiments. Recently, Mireshghallah et al.[29] used MIA for privacy metrics of Masked Language Models. In terms of example privacy measures, Song et al.[40] define the privacy risk of a example in terms of Bayesian probability. Duddu et al.[12] use the Shapley value[16, 20] as a tool to measure a machine learning's memory for a single example.

6.4. Example Difficulty and Privacy Risk

Some recent works[23, 42] demonstrate some interesting properties of hard examples and easy examples, specifically that the difficulty of examples is stable across convolutional neural network structures. Sablayrolles et al.[37] and Watson et al.[46] use example difficulty to improve MIA, and Carlini et al.[3] further proposed a Likelihood Ratio Attack which is more powerful at low FPR. Our work further highlights the relevance of the example character to privacy risks. However, so far we still do not have a good understanding of how neural networks memorize data, and related work[11, 17, 41] on adversarial examples demonstrates the output of neural networks is easily controlled by deliberately designed noise. It remains a challenging problem to understand the relationship between sample difficulty and member privacy.

7. Conclusion

In this work, we rethink the relationship between overfitting and membership inference attacks and demonstrate that using an overfitting-based approach for membership exclusion can effectively improve the performance of HP-MIA. Our evaluation results show that our attack is able to identify more members while guaranteeing high accuracy compared to other attacks. In addition, our method has a smaller computational cost compared to the previous method.

We believe that our work in understanding membership privacy is preliminary and the relationship between example characteristics and privacy leakage needs to be further explored. In particular, we would like to know how adversaries should perform effective attacks on easy examples, and how victims and defenders have to work on the defense of hard examples.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No.62102107, No.62072132, No.62002075, No.92067104), Technology Planning Project of Changsha (No.kq2004004), FDCT of Macau under its General R&D Subsidy Program Fund (Grant No. 0038/2022/A), and Guangzhou University Provincial College Student Innovation Training Program (No. S202111078114).

This paper is the extended version of the paper "Two-stage High Precision Membership Inference Attack" in the Fourth International Conference on Machine Learning for Cyber Security, ML4CS 2022[7]. In this version, we added two new attacks and a new target model structure in Section 4, used regularization for other target models, and showed ROC and PR curves of attack results. We also added two new sections, Section 5 and Section 6, to discuss features of our method and related work.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. B. McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016.

[2] Mislav Balunovi'c, Dimitar I. Dimitrov, Robin Staab, and Martin T. Vechev. Bayesian framework for gradient leakage. ArXiv, abs/2111.04706, 2021.

[3] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas F. Terzis, and Florian Tramèr. Membership inference attacks from first principles. ArXiv, abs/2112.03570, 2021.

[4] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In 28th USENIX Security Symposium (USENIX Security 19), pages 267–284, 2019.

[5] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In USENIX Security Symposium, 2021.

[6] Hong Chang and R. Shokri. On the privacy risks of algorithmic fairness. 2021 IEEE European Symposium on Security and Privacy (EuroS&P), pages 292–303, 2021.

[7] Shi Chen and Yubin Zhong. Two-stage high precision membership inference attack. In Yuan Xu, Hongyang Yan, Huang Teng, Jun Cai, and Jin Li, editors, Machine Learning for Cyber Security, pages 521–535, Cham, 2023. Springer Nature Switzerland.

[8] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In ICML, 2021.

[9] Dimitar I. Dimitrov, Mislav Balunovi'c, Nikola Jovanovi'c, and Martin T. Vechev. Lamp: Extracting text from gradients with language model priors. ArXiv, abs/2202.08827, 2022.

[10] Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 84, 2022.

[11] Ranjie Duan, Xiaofeng Mao, Alex K. Qin, Yun Yang, Yuefeng Chen, Shaokai Ye, and Yuan He. Adversarial laser beam: Effective physical-world attack to dnns in a blink. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 16057–16066, 2021.

[12] Vasisht Duddu, Sebastian Szyller, and N. Asokan. Shapr: An efficient and versatile membership privacy risk metric for machine learning. ArXiv, abs/2112.02230, 2021.

[13] Cynthia Dwork. Differential privacy. In ICALP, 2006.

[14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci., 9:211–407, 2014.

[15] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018.

[16] Amirata Ghorbani and James Y. Zou. Data shapley: Equitable valuation of data for machine learning. ArXiv, abs/1904.02868, 2019.

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. CoRR, abs/1412.6572, 2015.

[18] Ismat Jarin and Birhanu Eshete. Miashield: Defending membership inference attacks via preemptive exclusion of members. ArXiv, abs/2203.00915, 2022.

[19] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019.

[20] R. Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, Dawn Xiaodong Song, and Uiuc Shanhai. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8235–8243, 2021.

[21] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. On the effectiveness of regularization against membership inference attacks. ArXiv, abs/2006.05336, 2020.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2015.

[23] Ikki Kishida and Hideki Nakayama. Empirical study of easy and hard examples in cnn training. In ICONIP, 2019.

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60:84 – 90, 2012.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proc. IEEE, 86:2278–2324, 1998.

[27] Yugeng Liu, Rui Wen, Xinlei He, A. Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. Ml-doctor: Holistic risk assessment of inference attacks against machine learning models. ArXiv, abs/2102.02551, 2021.

[28] 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. 2020 IEEE European Symposium on Security and Privacy (EuroS&P), pages 521–534, 2020.

[29] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and R. Shokri. Quantifying privacy risks of masked language models using membership inference attacks. ArXiv, abs/2203.03929, 2022.

[30] Fan Mo, Anastasia Borovykh, M. Malekzadeh, Hamed Haddadi, and Soteris Demetriou. Quantifying information leakage from gradients. ArXiv, abs/2105.13929, 2021.

[31] Milad Nasr, R. Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. ArXiv, abs/1812.00910, 2018.

[32] Milad Nasr, R. Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018.

[33] Jerzy Neyman and Egon S. Pearson. On the problem of the most efficient tests of statistical hypotheses. Philosophical Transactions of the Royal Society A, 1933.

[34] Yeachan Park and Myung joo Kang. Membership inference attacks against object detection models. ArXiv, abs/2001.04011, 2020.

[35] Shahbaz Rezaei and Xin Liu. On the difficulty of membership inference attacks. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7888–7896, 2021.

[36] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. Nature, 323:533–536, 1986.

[37] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In ICML, 2019.

[38] A. 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. ArXiv, abs/1806.01246, 2019.

[39] R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. 2017 IEEE Symposium on Security and Privacy (SP), pages 3–18, 2017.

[40] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In USENIX Security Symposium, 2021.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. CoRR, abs/1312.6199, 2014.

[42] Mariya Toneva, Alessandro Sordoni, Rémi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. ArXiv, abs/1812.05159, 2019.

[43] Florian Tramèr, R. Shokri, Ayrton San Joaquin, Hoang M. Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. Truth serum: Poisoning machine learning models to reveal their secrets. ArXiv, abs/2204.00032, 2022.

[44] Wei-Cheng Tseng, Wei-Tsung Kao, and Hung yi Lee. Membership inference attacks against self-supervised speech models. ArXiv, abs/2111.05113, 2021.

[45] Tianhao Wang. Property inference attacks on neural networks using dimension reduction representations. 2020.

[46] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In International Conference on Learning Representations, 2022.

[47] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv: Learning, 2017.

[48] Lei Xu, Chunxiao Jiang, Yi Qian, Jianhua Li, Youjian Zhao, and Yong Ren. Privacy-accuracy trade-off in differentially-private distributed classification: A game theoretical approach. IEEE Transactions on Big Data, 7:770–783, 2021.

[49] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, and R. Shokri. Enhanced membership inference attacks against machine learning models. ArXiv, abs/2111.09679, 2021.

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

[51] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. ArXiv, abs/1605.07146, 2016.

[52] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. ArXiv, abs/1611.03530, 2017.

[53] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. ArXiv, abs/2001.02610, 2020.

[54] Guangxiang Zhao, Wenkai Yang, Xuancheng Ren, Lei Li, and Xu Sun. Well-classified examples are underestimated in classification with deep neural networks. ArXiv, abs/2110.06537, 2021.

[55] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against gans. ArXiv, abs/2111.07608, 2022.

[56] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In NeurIPS, 2019.