Learning-Based Difficulty Calibration for Enhanced Membership Inference Attacks

Haonan Shi, Tu Ouyang and An Wang
Case Western Reserve University
{*haonan.shi3, tu.ouyang, an.wang*}*@case.edu*

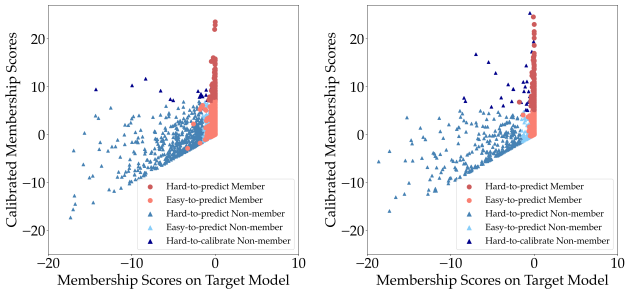
***Abstract*—Machine learning models, in particular deep neural networks, are currently an integral part of various applications, from healthcare to finance. However, using sensitive data to train these models raises concerns about privacy and security. One method that has emerged to verify if the trained models are privacy-preserving is Membership Inference Attacks (MIA), which allows adversaries to determine whether a specific data point was part of a model's training dataset. While a series of MIAs have been proposed in the literature, only a few can achieve high True Positive Rates (TPR) in the low False Positive Rate (FPR) region ($0.01\% \sim 1\%$). This is a crucial factor to consider for an MIA to be practically useful in real-world settings. In this paper, we present a novel approach to MIA that is aimed at significantly improving TPR at low FPRs. Our method, named *learning-based difficulty calibration for MIA (LDC-MIA)*, characterizes data records by their hardness levels using a neural network classifier to determine membership. The experiment results show that *LDC-MIA* can improve TPR at low FPR by up to 4x compared to the other difficulty calibration-based MIAs. It also has the highest Area Under ROC curve (AUC) across all datasets. Our method's cost is comparable with most of the existing MIAs, but is orders of magnitude more efficient than one of the state-of-the-art methods, LiRA, while achieving similar performance.**

1. Introduction

Machine learning has become increasingly important in many mission-critical domains, such as healthcare, finance, manufacturing, and cybersecurity. However, these applications often rely on the use of sensitive data as the training dataset for ML models. For instance, large-scale medical images containing private patient information are used to train CNN models for the recognition of body organs [38] and brain tumor segmentation [10]. Another example is that Fu *et al.* trained a CNN model using real credit card transaction data from a commercial bank to detect fraudulent behaviors [9]. While machine learning has proven to be highly effective in these domains, researchers have cautioned that overfitting can lead to the memorization of training data, potentially resulting in the leakage of sensitive information. To this end, Membership Inference Attacks (MIA) have been developed to determine whether a target sample belongs to the training dataset of a target model.

In most MIAs, the attackers take advantage of the fact that the target model produces more accurate results on the data records in their training dataset compared to those drawn from the same distribution but not included in the training dataset [28], [31], [39]. Shokri *et al.* proposed training a shadow model to mimic the behavior of the target model by learning from the output of the shadow model when exposed to member and non-member data records [31]. Later, Yeom *et al.* discovered that an attacker can calculate a membership score, such as the entropy loss value, of a target sample from the target model and use a threshold to determine membership [39]. However, previous works [3], [22], [37] point out that the score-based approach fails to distinguish between members and non-members with high precisions when the non-member data records also have low loss values.

To tackle this issue, Watson *et al.* proposed using a reference model [37], which is trained on data from the same distribution as the target model's training dataset. By calculating the difference between the loss values obtained from the target and reference models, the reference model helps calibrate the target model's behavior on a data record. This approach is an example of difficulty calibration-based attacks, one of the most advanced MIAs. To further improve the attack performance, Carlini *et al.* designed a Likelihood Ratio Attack (LiRA) [3]. LiRA utilizes multiple shadow models to estimate the distribution



(a) Cat (b) Airplane

Figure 1: The figure shows data records of Airplane and Cat classes in the CIFAR-10 dataset. Each record is represented by a marker indicating its membership type. The y-axis shows the calibrated membership scores, and the x-axis shows the membership scores on the target model. The membership score is determined as the negative of cross-entropy loss values. The calibrated membership score is the difference between a data record's membership score on the target model and the reference model. In traditional MIAs, data samples with higher membership scores are more likely to be members, while in difficulty calibration-based MIAs, data samples with higher calibrated membership scores are more likely to be members.

2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P) | 979-8-3503-5425-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/EUROSP60621.2024.00012

DOI 10.1109/EuroSP60621.2024.00012

of loss values on a target data record for models that are either trained or not trained on this data sample.

Existing difficulty calibration-based MIAs rely on a single metric, such as membership or calibrated membership scores, to differentiate between members and non-members. However, such approaches have certain limitations since single metrics may display variations or anomalies that make it difficult to distinguish members from non-members. Therefore, we propose using multiple metrics and information to analyze from different perspectives simultaneously. Our proposed attack[1] achieves this by adopting a learning-based approach that utilizes multiple features. We aim to achieve high TPRs at low FPRs and high AUC while minimizing the cost for attackers.

We consider two types of costs in our attack: the training cost and the data cost. The training cost is mainly dependent on the number of attacking models and their complexities, while the data cost involves the amount of data needed to train them. We strive to minimize both in our design. As mentioned by Carlini *et al.* [3], MIA can also be used as an auditing tool for ML models. For example, a company may use MIA to examine all the training data to detect privacy leakage before releasing a commercial model. Cost is an important factor in such scenarios.

In our discussions, we classify data records based on their difficulty levels. We refer to data records whose class labels can be easily predicted correctly by both the target and the reference models **easy-to-predict** samples, while those whose labels are difficult to predict correctly are called **hard-to-predict** samples. To illustrate this, we provide an example in Figure 1. There are also some hard-to-calibrate samples in this figure, which we will explain in Section 3.2. Based on the figure, we can make a few important observations. First, suppose we only consider the membership scores of the target samples on the target model (as shown on the x-axis). In that case, many non-members overlap with members because they are easy-to-predict samples and thus have low membership scores. The members in this overlap can either be easy-to-predict or hard-to-predict samples. Using the calibrated membership scores (as shown on the y-axis) increases the gap between the easy-to-predict non-members and hard-to-predict members. However, if we only consider the calibrated membership scores, the easy-to-predict members and the hard-to-predict non-members may overlap since both groups may have low calibrated membership scores. Therefore, both membership and calibrated membership scores are useful in distinguishing members from non-members. Second, if we use a fixed threshold value on the calibrated membership scores to differentiate between members and non-members, as Watson *et al.* did, it may not work well for both classes since they have different optimal cut-off points. This indicates that the hardness levels of data records are not universal across different classes. Therefore, adopting a more intelligent approach to determining the threshold values is necessary. Third, data distribution also plays an important role in determining how difficult it is for a data record to be correctly classified, in addition to its intrinsic characteristics. As revealed by Long *et al.* [22], the more neighbors a data record has in the training dataset, the easier it is for it to be correctly classified. This also means that the data record is more likely to be determined as a member by attackers, regardless of its membership.

Based on these observations, we propose developing a classifier that can learn to calibrate difficulty based on the membership score on the target model, the calibrated membership score, the label of the target data record, and its neighborhood information. To train this classifier, we can use a shadow target model and a reference model trained with data records that share the same distribution as those belonging to the target model training dataset. The shadow target model would mimic the target model's behavior in classifying members and non-members. We call the proposed attack *LDC-MIA*. The main contributions of this paper are threefold. (1) The proposed attack significantly improves the TPR at low FPR while minimizing the cost for attackers. We only require one shadow model and one reference model to improve the TPR. The classifier we build is a simple model with three fully connected layers. (2) We conduct a comprehensive characterization of the data records' hardness levels and use these characters to train a neural network for determining membership. This learning-based calibration approach can be easily extended to integrate other features without requiring significant retraining efforts. (3) Through extensive evaluations, we provide insights into each character's contributions to the success of our proposed attack.

We conduct extensive experiments to evaluate the performance of *LDC-MIA* on various datasets. Specifically, we measure the TPR at low FPRs ranging from 0.01% to 1%. This metric helps us evaluate the model's ability to correctly identify positive instances while minimizing the number of false positive predictions for practical use. Our results show that our proposed attack achieves the highest AUC across all datasets compared to state-of-the-art MIAs and improves TPR up to 4x. In addition, we measure the precision-recall curve to analyze how well the model performs across different recall levels while maintaining high precision. The results indicate that *LDC-MIA* consistently produces the highest precision values for different recall values across all datasets. For instance, *LDC-MIA* identifies 52.72% of the members with a precision of 80%, significantly higher than other MIAs can achieve.

2. Background

2.1. Machine Learning

In the machine learning classification tasks, for a dataset $\boldsymbol{X}$ that contains data across n classes, a neural network model f_θ trained on $\boldsymbol{X}$ is a function capable of mapping an input data sample x to a probability distribution across n classes. We denote by $f_\theta(x)$ the output vector from f_θ, where this vector represents the prediction posteriors of x across n class, where $f_\theta(x)_y$ indicates the prediction posterior value of x for a specific class y.

During the training process of a machine learning model, for training data (x, y), the loss function $\mathcal{L}(f_\theta(x)_y, y)$ is typically defined to calculate the error between the prediction posterior $f_\theta(x)_y$ of the training

1. The implementation of *LDC-MIA* is available at https://github.com/horanshi/LDC-MIA.

data and its ground truth label y. For classification tasks, the cross-entropy loss is a commonly used loss function:

$$\mathcal{L}(f_\theta(x)_y, y) = -log(f_\theta(x)_y) \tag{1}$$

The training of neural network models utilizes stochastic gradient descent [19] to minimize the loss function:

$$\theta_{i+1} \leftarrow \theta_i - \lambda \sum_{(x,y)\in B} \nabla_\theta \mathcal{L}(f_{\theta_i}(x), y) \tag{2}$$

where B is a batch of training data from $\boldsymbol{X}$, λ is the learning rate for updating the parameters θ of the neural network. In this paper, we will denote a trained model as f. Training a machine learning model involves running multiple epochs to achieve high generalizability. Also, various techniques are utilized in the training model process, such as data augmentation [6], [36] and tuned learning rates [15], [23], which enhance the model's ability to generalize from the training data to unseen data, thereby ensuring the model's usefulness in practical applications.

2.2. Membership Inference Attacks

In membership inference attacks, the attacker aims to identify whether a given target sample is part of the target model's training dataset. MIA was first introduced by Shokri *et al.* [31], with the trend of increasingly sensitive data being used to train machine learning models, MIA has received considerable attention in many scenarios [4], [25], [26].

Definitions. Given a target model f and target sample x, the process of MIA can be defined as:

$$\mathcal{A} : x, f \longrightarrow \{0, 1\} \tag{3}$$

where $\mathcal{A}$ is the attack function, if the target sample x is in the training dataset of f, the attack function $\mathcal{A}$ outputs 1(i.e., member), otherwise the output of $\mathcal{A}$ is 0(i.e., non-member).

There are some MIAs [28], [39] use the membership score of the target sample on the target model as the basis for determining whether it is a member. This membership score can be the loss, confidence, etc. In this paper, we will use the cross-entropy loss of the target sample on target model to calculate the membership score, the membership score of target sample (x, y) is defined as:

$$s(f, (x, y)) = -\mathcal{L}(f(x)_y, y) = log(f(x)_y) \tag{4}$$

where f is the target model.

Difficulty Calibration. One category of the state-of-the-art MIAs is based on difficulty calibration [3], [21], [37]. These attacks are designed to accurately identify members by first identifying the easy-to-predict non-members and then separating them from hard-to-predict members. The key to their success is their detailed analysis of the sample hardness of each target sample [3], [37]. To achieve this, they often use a reference model or shadow model(s) to compare the membership scores of each target sample on different models where they are either members or non-members of the training dataset. A larger score indicates that the sample is likely to be a member, while a smaller score indicates that it is more likely to be a non-member. The intuition behind this approach is that a member data record may lead to very different outputs on a model where they are part of the training dataset compared to another one where they are not in the training dataset. These differences can be represented by different values, such as calibrated membership score [37], likelihood ratio [3]. Among these, the calibrated membership score, proposed by Watson *et al.*, is the easiest to obtain. Given a target sample x, its calibrated membership score can be calculated using the following equation:

$$s^{cal}(h, g, (x, y)) = s(h, (x, y)) - s(g, (x, y)) \tag{5}$$

where y denotes its predicted label, h represents the target model, and g represents a reference model that shares the same model architecture as the target model. To determine whether a target sample is a member, a pre-defined threshold is applied on the calibrated membership score. The specific attack process is illustrated by the equation below:

$$A(h, g, (x, y)) = \mathbb{1}\left[s^{cal}(h, g, (x, y)) > \tau\right] \tag{6}$$

where $\mathbb{1}$ is an indicator function, τ is the decision threshold. In other words, if the calibrated membership score exceeds the threshold τ, the target sample is determined as a non-member; otherwise, it is determined as a member. This approach allows the proposed MIA to identify members with high TPRs at low FPRs.

3. Attack Methodology

3.1. Adversary knowledge

As with previous MIAs, we assume an attacker using *LDC-MIA* has access to certain adversarial knowledge. Firstly, the attacker has black-box access to the target model. Secondly, the attacker has an auxiliary dataset with the same data distribution as the target model's training dataset. This auxiliary dataset may or may not overlap with the target model's training dataset, and the attacker does not need to know which part of the auxiliary dataset is included in the target model's training dataset. Our proposed attack method for MIA is also different from many existing ones, as it does not require knowledge of the target model's architecture and the training algorithm of the target model.

3.2. Design intuition

Recent state-of-the-art attacks [3], [37] have explored the difficulty level of each data record and applied parametric calibration to improve the attack performance in the low FPR region. These attacks are similar in design to our proposed method in that each target data record is individually considered when performing attacks. However, these works provided limited discussions on the impact of calibration on different types of data records with varying intrinsic properties. Inspired by this gap, we categorize member and non-member data records into five categories based on their difficulty levels for label predictions and calibrations, as shown in Figure 3: hard-to-predict member/non-member, easy-to-predict member/non-member, and hard-to-calibrate non-member. The definition of easy-to-predict and hard-to-predict samples can be found in section 1. We refer to

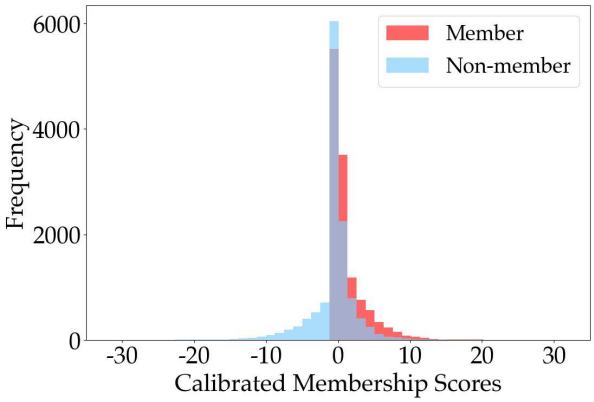


Figure 2: Histogram of the calibrated membership scores of members and non-members. The calibrated membership scores correspond to those in Figure 3.

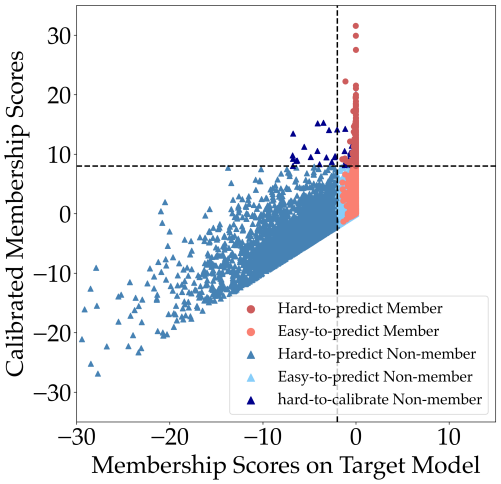


Figure 3: Based on the calibrated membership scores and membership scores on the target model (VGG-16), we group the target samples from the CIFAR-10 dataset into five categories: hard-to-predict member/non-member, easy-to-predict member/non-member, and hard-to-calibrate non-member.

the non-members that have high calibrated membership scores as **hard-to-calibrate** samples, as their membership cannot be accurately determined by the existing difficulty calibration based MIAs. This figure is similar to Figure 1, and we have divided it into four regions using a fixed membership score threshold and a fixed calibrated membership score threshold. By analyzing this figure, we can get the following insights.

Membership scores on the target model are still useful. MIAs based on loss utilize the gap in cross-entropy loss to differentiate between members and non-members. The basic idea is that members would have smaller loss values (higher membership scores) while non-members would have larger ones. Other score-based attacks also follow a similar approach. However, this type of attack can only accurately identify non-members that are difficult to predict. It cannot identify members with high precision. To solve this problem, difficulty calibration based MIAs use calibrated membership scores instead. The higher the calibrated score, the more likely it is that a data record is member data. These attacks can improve TPR in the low FPR region, as they can better identify hard-to-predict members.

Even though FPR can be reduced by carefully selecting a threshold for the calibrated scores, it is difficult to eliminate all of them. In Figure 3, it can be seen that many non-members overlap with members along the y-axis. Difficulty calibration based MIAs have improved traditional MIAs, specifically the TPR in the low FPR region. This is achieved by identifying hard-to-predict members more accurately by carefully selecting threshold values for calibrated membership scores. However, decreasing the number of false positives remains challenging since many non-members overlap with members, as demonstrated in Figure 3. This can be seen more clearly in Figure 2, which shows the distribution of calibrated membership scores for members and non-members. In this figure, the calibrated membership scores are on the x-axis and share the same values with that of Figure 3. The y-axis shows the number of members and non-members with corresponding calibrated scores. The figure highlights that many members have low calibrated scores, making them easy-to-predict members. These members overlap with non-members near the line where the calibrated score equals 0. Within this region, the easy-to-predict members overlap with both hard-to-predict and easy-to-predict non-members, as both groups have similar outputs on the target and the reference models.

Two important observations can be made from the figure. Firstly, attackers are likely to encounter easy-to-predict members in real-world attacks. Therefore, identifying these samples can significantly improve TPR. Secondly, existing difficulty calibration based MIAs may fail to isolate such members from non-members by only considering the calibrated membership scores. However, Figure 3 shows that adjusting the thresholds for membership and calibrated membership scores simultaneously can make it easier to differentiate between members and non-members. This suggests that the membership scores on the target model are still useful in addition to the calibrated membership scores. Based on this observation, we utilize membership scores on the target model in *LDC-MIA* to exclude the hard-to-predict non-members. This not only helps identify hard-to-predict members but also easy-to-predict members, thus improving TPR in all FPR regions.

Neighbor information is also important. In Figure 3, it is clear that there are some hard-to-calibrate non-members. These samples are not easy-to-predict as they have low membership scores on the reference model, nor are they hard-to-predict as they have high membership scores on the target model. The most plausible explanation for this phenomenon is that these non-members have more neighbors in the members than other non-member samples. Neighbors are determined by the similarity between two data samples. Specifically, we can input two data samples into a model and compute the cosine similarity of the output vectors of the last layer before the softmax layer. If their cosine similarity exceeds a certain threshold, then they are considered neighbors.

According to Long *et al.*, certain data records are are more vulnerable to being identified as members by attackers if they have fewer neighbors in the training data [22]. This is because such records may display unique characteristics that the target model can overfit to, making them easier to identify. On the other hand, data records

with more neighbors may lead to incorrect membership inferences by MIAs. Therefore, a non-member data record with many neighbors in the target model's training dataset is more likely to be misidentified as a member than another non-member sample with fewer neighbors. This means that if we have two samples with similarly high membership scores, the one with fewer neighbors is more likely to be a member. Motivated by this, *LDC-MIA* includes neighbor information to differentiate the hard-to-predict members and hard-to-calibrate non-members in the upper right region in Figure 3.

We aim to lower the calibrated membership score for hard-to-calibrate non-members, and to achieve this, we rescale the calibrated membership scores with neighborhood information. However, there are two challenges we need to address when calculating the neighborhood information. Firstly, attackers do not have access to the training dataset of the target model and thus are unable to compute cosine similarities with data samples that are members. Secondly, attackers have no access to the output vectors of the last layer before the softmax layer in the target model, making it impossible to exploit the target model to compute neighborhood information. To address these challenges, we make two important assumptions. Firstly, we assume that if a data sample has more neighbors in the auxiliary dataset, it will also have more neighbors in the training dataset of the target model. This assumption is reasonable since the auxiliary dataset follows the same distribution as the training dataset. Secondly, we assume that if two data samples are neighbors using the output vectors obtained from the target model, they will also be neighbors using the output vectors obtained from the reference model. This intuitive assumption holds when the target and reference models share the same architecture. Based on these assumptions, we can then leverage the reference model and the auxiliary dataset to approximate the neighborhood information of a target data record x:

$$NI(x) = \frac{1}{\sum_{i=1}^{n} [cosine\_similarity(\mathbf{v}_x, \mathbf{v}_{\text{aux}_i}) > \theta]} \quad (7)$$

, where $\mathbf{v}_x$ is the output vector of the target data record, $\mathbf{v}_{aux_i}$ is that of the data records in the auxiliary dataset, n is the size of the auxiliary dataset, and θ is the similarity threshold value to determine neighbors. We use $\theta = 0$ in our attack. Through our experiments in Section 4, we verify that setting θ to 0 works well for most datasets. The intuition behind this is that the value of cosine similarity is greater than 0 when two data records are positively related. Then, we enhance the membership scores proposed by Watson *et al.* [37] with neighborhood information as follows:

$$s^{\text{cal}}(h, g, (x, y)) = [s(h, (x, y)) - s(g, (x, y))] \cdot NI(x) \quad (8)$$

, where h is the target model, and g is the reference model.

We compare the effect of the enhanced membership score and that of the membership score proposed by Watson *et al.* in Figure 4. The MIA follows Watson *et al.*'s approach of using a threshold on calibrated membership scores to determine membership. Based on the figure, it can be observed that the new score can better distinguish members from hard-to-calibrate non-members manifested in improved TPRs at the same low FPRs. It is important to note that we only compared the TPR in the low FPR region because this is the region where MIAs are considered practical. The comparison results suggest that neighborhood information is a valuable component in membership scores.

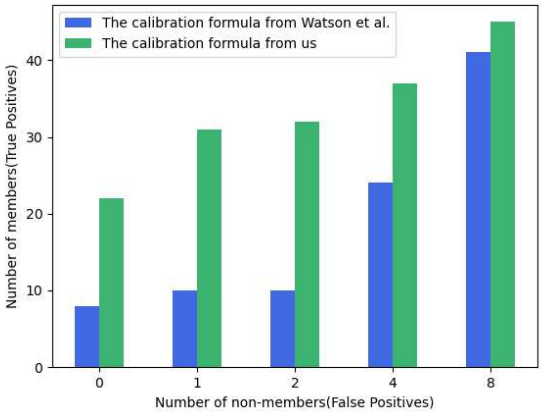


Figure 4: Compare TPR in the low FPR region ($< 0.01\%$) for attacks on the CIFAR-10 dataset using different membership scores.

Different MIA score thresholds are needed for accurately classifying samples of different labels. One of the main objectives of an attacker is to achieve high precision in identifying member data records in the training dataset. To achieve this goal, the attacker strives to differentiate between members and non-members as much as possible. Many existing MIAs rely on a threshold value of the membership scores to distinguish between members and non-members. However, the divergence of membership scores in a target model is influenced not only by the hardness and neighborhood information of a data record but also by its assigned label.

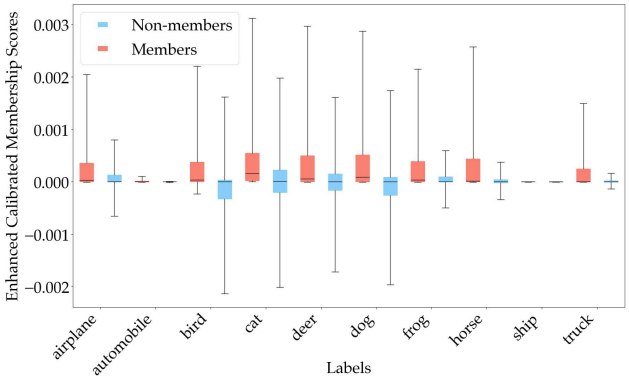


Figure 5: Membership scores of data records with different labels in CIFAR-10.

This could be attributed to the different distributions of easy and hard data records across various classes. As a result, we believe that determining the most suitable threshold of membership scores for each class can further enhance the accuracy of MIAs. To illustrate this, we present an example of the membership scores of data records in CIFAR-10 belonging to different classes in Figure 5, using the enhanced calibrated membership score calculated by Eq 8. In this example, we use the VGG-16 as both our target and reference models. Both models are trained until they achieve their highest accuracy values on the test dataset.

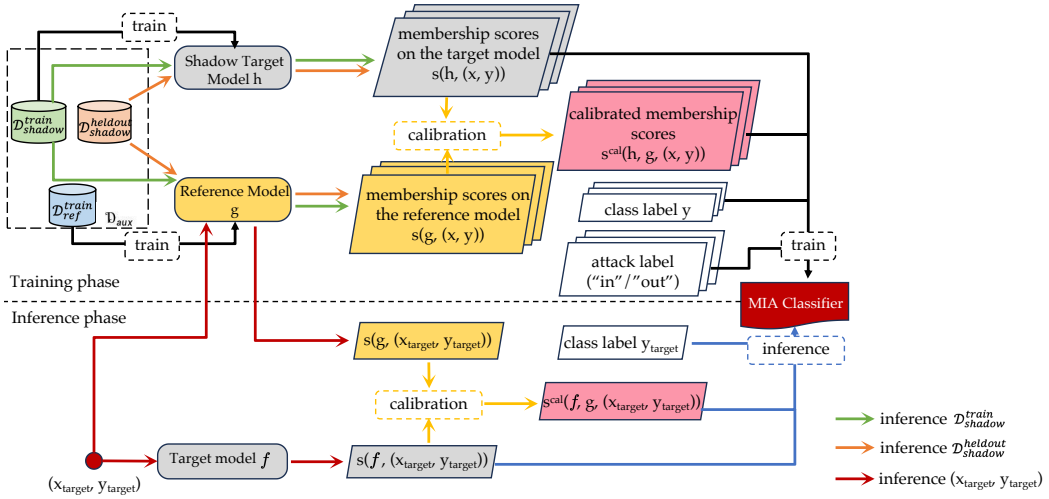


Figure 6: Workflow of *LDC-MIA*. During the training phase, $\mathcal{D}_{shadow}^{train}$ and $\mathcal{D}_{shadow}^{heldout}$ serve as members and non-members for the shadow model, respectively. They are input into the shadow target model h and the reference model g to obtain membership scores $s(h,(x,y))$ and $s(g,(x,y))$. The calibrated membership score $s^{\text{cal}}(h,g,(x,y))$ is obtained by using Eq 8. Finally, we use $s(h,(x,y))$, $s^{\text{cal}}(h,g,(x,y))$, and the class labels as features to train the MIA classifier. For $\mathcal{D}_{shadow}^{train}$ and $\mathcal{D}_{shadow}^{heldout}$, their ground truth labels in the classifier training are "in" or "out", respectively. During the inference phase, we follow the same procedure to obtain the features of the target data sample (x_{target}, y_{target}) except that the target model is used rather than the shadow target model.

The figure displays the membership score of different data records on the $y-$axis, while the $x-$axis shows their labels. The scores for member and non-member data records are shown in two different colors. It is evident that the average membership score is higher for members. This means that if an attacker sets a reasonable threshold value, they can identify more members accurately, resulting in high precision. For instance, choosing a threshold of 0.001 can help identify airplanes with high precision without increasing many false positives. However, this may lead to low precision for other classes like deer or cats, which can harm overall precision. Therefore, to maintain overall precision, the attacker should carefully select a threshold value for each class.

3.3. Attack Framework

Based on the aforementioned intuitions, we propose to build a classifier that can determine the membership of data records. Our ultimate goal is to utilize the discriminatory abilities of neural network models to perform membership inference attack based on the intuitions we identified in Section 3.2. The proposed attack consists of two phases. The first phase is the training phase, in which a shadow target model h, a reference model g, and an attack classifier are all trained. The second phase is the inference phase, in which we obtain the features of each target data record from the trained reference model and the target model f and use the features for classification using the attack classifier. The proposed attack workflow is illustrated in Figure 6.

In the proposed attack, we use an auxiliary dataset $\mathcal{D}_{aux}$ that has the same distribution as the data used for training the target model. During the training phase, we first split $\mathcal{D}_{aux}$ into three distinct parts. 1. $\mathcal{D}_{shadow}^{train}$, which is used to train the shadow target model. 2. $\mathcal{D}_{shadow}^{heldout}$, which contains non-member data records for the shadow model. 3. $\mathcal{D}_{ref}^{train}$, which is used to train the reference model. This way, we can keep a clear separation between the data used for training the different models.

Training phase. Once all the models are trained, we feed all the data records in $\mathcal{D}_{shadow}^{train}$ and $\mathcal{D}_{shadow}^{heldout}$ to the shadow target model and the reference model. We can then obtain membership scores from both the shadow target model and the reference model on members and non-members, i.e., members' $s(h,(x,y))$, non-members' $s(h,(x,y))$, members' $s(g,(x,y))$, and non-members' $s(g,(x,y))$. Note that $\mathcal{D}_{shadow}^{train}$ contains data records of the members of the training dataset of the shadow target model, while $\mathcal{D}_{shadow}^{heldout}$ contains data records of non-members. This means that the attacking classifier can observe how both members and non-members behave on the shadow target model and the reference model, allowing it to discriminate different behaviors. The reference model is introduced to calibrate the membership scores. Members' membership scores on the shadow target model are paired with those on the reference model, and Eq (8) is used to calculate the calibrated membership score $S^{cal}(h,g,(x,y))$. To obtain the neighborhood information of a data record, we exclude the interference of the training data of the reference model by using the data records from D_{shadow}^{train} and $D_{shadow}^{heldout}$ only. These data records consist of v_{aux} in Eq (7), which are obtained from the reference model g. Then, the same process is carried out for non-members to obtain the calibrated membership score.

The membership scores obtained from the target model, as mentioned in Section 3.2, can still be useful in MIAs. We include these scores as one of the features to train the attacking classifier, along with the ground truth label of the data records and the calibrated membership score. With the help of the ground truth membership information, the classifier can learn to predict the membership of a given data record using these features. After training the classifier, we apply it to the actual attack during the

inference phase.

Inference phase. During the inference phase, an attacker can only access the target model as a black box. This means that they can only access the membership score of a target data record (x_{target}, y_{target}) by using the prediction results of the target model. Just like in the training phase, we use D_{shadow}^{train} and $D_{shadow}^{heldout}$ to obtain neighborhood information. Then, we provide (x_{target}, y_{target}) to the same reference model that was used during the training phase and calculate the calibrated membership score $S^{cal}(f, g, (x, y))$. Finally, we feed the membership score, calibrated membrship score, ground truth label y_{target} to the classifier to predict the membership of (x_{target}, y_{target}). Our classifier is an MLP model that consists of two hidden layers with ReLU activation functions, followed by a softmax layer.

4. Evaluations

In this section, we conduct a series of experiments to evaluate the performance of the proposed attack on the most widely used datasets and various target model architectures. Additionally, we compare *LDC-MIA* with several other representative black-box MIA methods [29], [37], [39].

4.1. Experimental Setup

4.1.1. Datasets. In our experiments, we use the following datasets that have been often used for image classifications:

- **CIFAR-10 [17].** The CIFAR-10 is a benchmark dataset used for image classification tasks. Each image is 32×32×3, and there are 60k images categorized into 10 classes with equal distribution per class.
- **CIFAR-100 [17].** The CIFAR-100 dataset consists of 100 classes of images, with 32×32×3 sized images and a total of 60k images. Similar to the CIFAR-10 dataset, it is also used for image classifications.
- **CINIC-10 [7].** CINIC-10 is also a dataset used for image classifications, which includes images from CIFAR-10 and ImageNet [8]. In this dataset, there is a total of 27k images across 10 classes, each with a size of 32×32×3.

In addition to the image dataset, we also use the following datasets for our experiments:

- **Adult [2].** The adult dataset contains information on people's income, with 2 classes and 14 features for each of the 48842 data records.
- **Credit [13].** The Credit dataset is often used for binary classification tasks involving credit scoring. It contains 1000 data records with each consisting of 20 features. There are two classes of data records in the dataset.

In our evaluations, we split each dataset into six parts: D_{target}^{train}, $D_{target}^{heldout}$, D_{shadow}^{train}, $D_{shadow}^{heldout}$, D_{ref}^{train} and D^{test}. D_{target}^{train} is used to train the target model, while $D_{target}^{heldout}$ is made up of non-members of the target model. Similarly, D_{shadow}^{train} is used to train the shadow target model, and $D_{shadow}^{heldout}$ contains non-members for the shadow target model. D_{ref}^{train} is the training dataset for the reference model, and D^{test} is the test dataset for all the models. The sizes of all the datasets used in our experiments are listed in Table 1.

TABLE 1: Datasets division.

Datasets	D_{target}^{train}	$D_{target}^{heldout}$	D_{shadow}^{train}	$D_{shadow}^{heldout}$	D_{ref}^{train}	D^{test}
CIFAR-10	12500	12500	7500	7500	10000	10000
CIFAR-100	12500	12500	7500	7500	10000	10000
CINIC-10	22500	22500	13500	13500	18000	90000
Adult	8140	8140	4884	4884	6513	16281
Credit	200	200	120	120	160	200

4.1.2. Models. To demonstrate the effectiveness of *LDC-MIA*, we select two models of different sizes as target models on the CIFAR-10 and CIFAR-100 datasets. For the CIFAR-10 dataset, we choose WideResNet28-10 [40] and VGG-16 [32]. For the CIFAR-100 dataset, we select DenseNet-121 [14] and SmallNet. Using SmallNet allows us to make a fair comparison with the existing MIAs. For the CINIC-10 dataset, we choose VGG-16 [32]. For the Adult and Credit datasets, we employ a multi-layer perceptron (MLP) model as the target model. This model consists of one hidden layer with ReLU activation function, followed by a softmax layer. The training and testing accuracy of the target models are shown in Table 2.

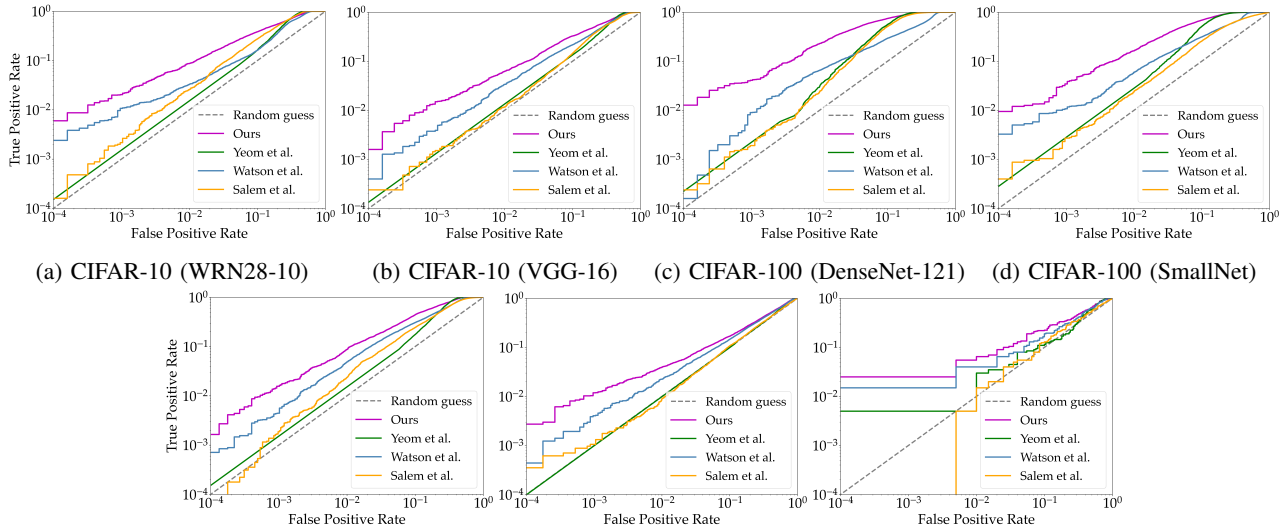
TABLE 2: Accuracy of target models on different datasets.

Dataset	Target Model	Train Accuracy	Test Accuracy
CIFAR-10	WideResNet28-10	98.87%	79.63%
CIFAR-10	VGG-16	97.62%	71.71%
CIFAR-100	DenseNet-121	99.88%	45.04%
CIFAR-100	Smallnet	94.53%	31.27%
CINIC-10	VGG-16	96.16%	60.56%
Adult	MLP	92.04%	83.29%
Credit	MLP	90.62%	83.23%

We use the same network architecture as the target model for both the reference model and the shadow target model. We also explore employing different model architectures for the reference model and shadow target model (results are detailed in Section 5). When training the shadow target model, its validation loss does not need to be similar to the target model (this non-requirement is useful in practice since our method does not assume that the attacker know the target model's validation loss). The reference model is trained until it reaches the maximum validation accuracy. Our proposed MIA classifier is of multi-layer-perceptron architecture, that consists of two hidden layers with ReLU activation functions, followed by a softmax layer. We utilize stochastic gradient descent (SGD) with a learning rate of 0.1, Nesterov momentum of 0.9, and a cosine learning rate schedule for the training. The duration of training for each model varies between 20 to 200 epochs, depending on the complexity of the models and the size of the datasets. All experiments are conducted on general-purpose machines equipped with Intel Xeon Silver 4208 CPU@2.10 GHz, Quadro RTX 5000 GPU, and 16 GB RAM.

4.1.3. Metrics. In our experiments, we use the following metrics to evaluate the results of the MIAs:

- **Full Log-scale ROC.** In evaluating the accuracy of MIAs, precision is an important metric. Carlini *et al.* [3] suggest that the TPR should be emphasized in low FPR regions, as higher TPR in these regions indicates higher precision of the MIA method. A full log-scale receiver operating characteristic (ROC) curve can be used for a clearer comparison of TPR among different MIAs in low FPR regions.



(a) CIFAR-10 (WRN28-10) (b) CIFAR-10 (VGG-16) (c) CIFAR-100 (DenseNet-121) (d) CIFAR-100 (SmallNet)

(e) CINIC-10 (f) Adult (g) Credit

Figure 7: Full log-scale ROC curves of MIAs(*LDC-MIA*, Yeom *et al.* [39], Watson *et al.* [37] and Salem *et al.* [29]) on different datasets.

- **TPR at Low FPR.** We also analyze the TPR of various MIA methods at a few low FPR points, including 1%, 0.1%, and 0.01%. These values enable numerical comparisons between different MIA methods.
- **Precision-Recall (PR) Curve** In real-world scenarios, attackers are more likely to encounter members that are easy to predict into the model than those that are hard to predict, as indicated in Figure 2. Therefore, most MIAs can achieve high recall by identifying such members. However, recall only measures the effectiveness of the model in capturing most of the positive instances, it does not reflect how accurately the model predicts positives. Therefore, it is also important to measure precision values at relatively high recall. We can do this by looking at the precision values when the recall value ranges between 0.2 and 0.7. This metric helps us measure how effectively the model balances between precision and recall.
- **Balanced accuracy and AUC.** As with previous MIA methods [25], [31], [37], we measure the overall performance of *LDC-MIA* using Balanced accuracy and AUC. When working with imbalanced datasets, accuracy alone can be misleading. Hence, balanced accuracy is an important metric often used to evaluate the performance of a classification model by considering the arithmetic mean of TPR and TNR. On the other hand, AUC quantifies the overall performance of the model by measuring the area under the ROC curve.

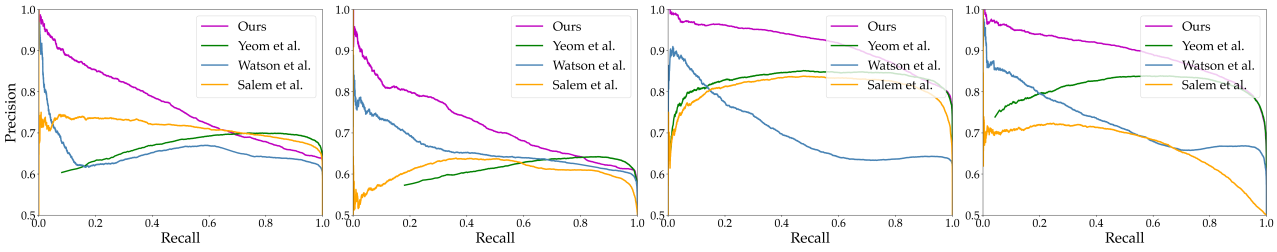
4.1.4. Baselines. In our evaluations, we compared *LDC-MIA* with four other MIA methods. Salem *et al.* [29] used the posteriors of target data records obtained from the target model and trained a shadow model to mimic the target model's behaviors. They proposed three different adversary models, and we compared *LDC-MIA* with Adversary 1, which had the best performance. Yeom *et al.* [39] performed the attack without any auxiliary model by using the loss values of the target data records on the target model. Watson *et al.* [37] used a reference model for difficulty calibration when performing MIA. LiRA [3] aims to exploit statistical differences between data points labeled as members and non-members to infer membership status. It requires to train a large number of shadow models for each target sample. We compare our work with Salem *et al.* [29], Yeom *et al.* [39] and Watson *et al.* [37] in Section 4.2 and with LiRA [3] in Section 4.3.

4.2. Main Results

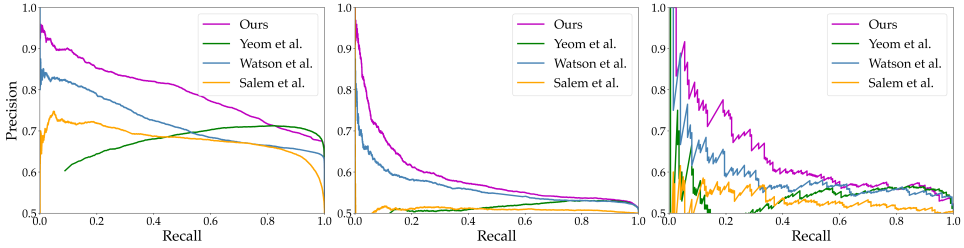
In our evaluations, all the attacks are carried out in the black-box scenario, and we demonstrate and analyze the results using various metrics as mentioned in Section 4.1. To ensure a fair comparison, we used the same auxiliary dataset and two auxiliary models – one shadow model and one reference model – in our proposed attack. For other MIA methods, we use two reference or shadow models if they employ any.

4.2.1. TPR at low FPR regions. In this experiment, we compare the TPR-FPR tradeoff of our method *LDC-MIA* and three other MIA methods across five datasets. The ROC curves for all methods over five datasets are depicted in Figure 7. The figure shows that *LDC-MIA* achieves higher TPR at almost all FPRs than other MIA methods across all datasets. Further, we compare the TPR values in the low FPR region (i.e., between 0.01% and 1%) with results obtained from an average of 5 runs in Table 4, The results show that *LDC-MIA* achieves better TPRs in the low FPR region. On CIFAR-100 and CINIC-10 datasets, *LDC-MIA* outperforms the state-of-the-art difficulty calibration-based MIA method proposed by Watson *et al.*, having 4x higher TPRs in low FPRs.

4.2.2. Precision-Recall curve. We compare MIA methods' effectiveness by examining the precision and recall tradeoffs made by our method and three others. The Precision-recall curves of all MIA methods across five datasets are depicted in Figure 8. Note that when the recall is 0, the precision values of all MIAs are also 0 in the figure. Our method *LDC-MIA* reaches the highest precision values over most of the recall value range



(a) CIFAR-10 (WRN28-10) (b) CIFAR-10 (VGG-16) (c) CIFAR-100 (DenseNet-121) (d) CIFAR-100 (SmallNet)



(e) CINIC-10 (f) Adult (g) Credit

Figure 8: Precision-Recall curves of MIAs (*LDC-MIA*, Yeom *et al.* [39], Watson *et al.* [37] and Salem *et al.* [29]) on different datasets.

TABLE 3: TPR at Low FPR regions of MIAs across datasets across MIA methods.

Attack Method	CIFAR-10						CIFAR-100						CINIC-10			Adult			Credit		
	WRN28-10			VGG-16			DenseNet-121			SmallNet											
(FPR)	0.01%	0.1%	1%	0.01%	0.1%	1%	0.01%	0.1%	1%	0.01%	0.1%	1%	0.01%	0.1%	1%	0.01%	0.1%	1%	0.01%	0.1%	1%
Salem *et al.* [29]	0.01%	0.2%	2.5%	0.02%	0.1%	1.2%	0.02%	0.2%	2.3%	0.04%	0.3%	2.3%	0.00%	0.2%	2.3%	0.00%	0.1%	1.0%	0.00%	0.0%	1.5%
Yeom *et al.* [39]	0.00%	0.0%	0.0%	0.00%	0.0%	0.0%	0.00%	0.0%	2.9%	0.04%	0.2%	1.0%	0.00%	0.0%	0.0%	0.00%	0.0%	0.0%	0.50%	0.5%	3.0%
Watson *et al.* [37]	0.24%	1.0%	3.3%	0.04%	0.4%	3.4%	0.01%	0.9%	6.5%	0.32%	1.1%	5.9%	0.07%	0.4%	4.0%	0.04%	0.4%	2.0%	1.50%	1.5%	4.0%
Ours	**0.66%**	**2.3%**	**9.1%**	**0.21%**	**1.6%**	**6.7%**	**1.28%**	**4.2%**	**23.2%**	**0.95%**	**3.8%**	**16.4%**	**0.16%**	**1.6%**	**8.7%**	**0.20%**	**1.2%**	**3.9%**	**2.50%**	**2.5%**	**6.5%**

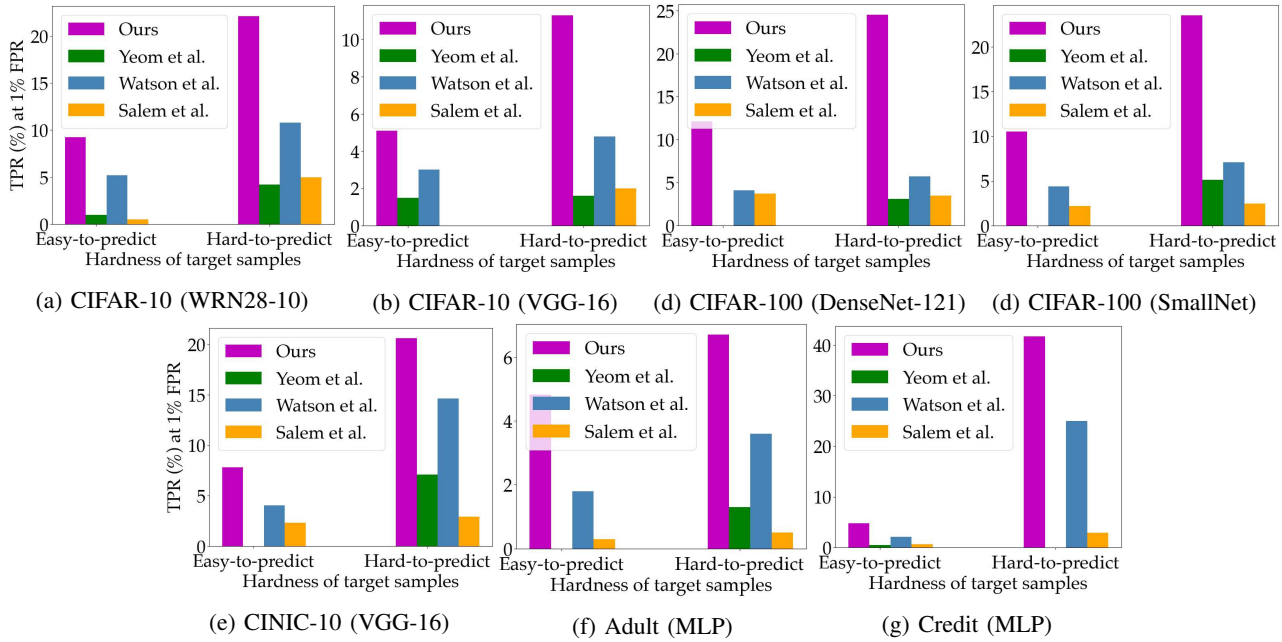
TABLE 4: Balanced accuracy and AUC of MIAs on different datasets.

Attack Method	CIFAR-10				CIFAR-100				CINIC-10		Adult		Credit	
	WRN28-10		VGG-16		DenseNet-121		SmallNet							
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Salem *et al.* [29]	69.41%	0.731	65.19%	0.679	82.28%	0.885	66.54%	0.679	70.14%	0.745	51.17%	0.514	54.75%	0.539
Yeom *et al.* [39]	68.80%	0.723	**69.81%**	0.696	82.91%	0.900	84.82%	0.889	**77.12%**	0.780	55.07%	0.542	**60.25%**	0.581
Watson *et al.* [37]	66.20%	0.707	66.18%	0.705	71.10%	0.741	73.78%	0.772	71.80%	0.768	54.86%	0.574	59.25%	0.607
Ours	**71.62%**	**0.794**	67.82%	**0.752**	**82.94%**	**0.933**	**85.12%**	**0.918**	75.71%	**0.832**	**56.27%**	**0.592**	59.50%	**0.640**

compared to other MIA methods, across all datasets. For example, *LDC-MIA* achieves recall of 77.2% and 49.1% on DenseNet-121 and SmallNet, with 90% precision, respectively for the CIFAR-100 dataset. For the CINIC-10 dataset, *LDC-MIA* identifies 52.72% of the members with a precision of 80%.

4.2.3. Balanced accuracy and AUC. The balanced accuracy is the arithmetic mean of sensitivity and specificity, the higher the better. AUC value indicates the overall discriminatory power of the model over all possible TPR-FPR tradeoffs, the higher the better. Table 4 shows the balanced Accuracy and AUC of all the MIA methods averaged out of 5 runs. The highest metric values and the metric values of *LDC-MIA* have been highlighted. *LDC-MIA* achieves the highest AUC values across all the datasets. Regarding balanced accuracy values, *LDC-MIA* has close if not better results compared to the best-performing MIAs.

4.2.4. Improvement on TPR at various hardness levels. Data records of different hardness levels may have different vulnerabilities to membership inference attacks. Therefore, we categorize data from different datasets into two groups based on their membership scores on the reference model: the hard-to-predict and easy-to-predict samples. Note that neither the member nor the non-member data records are in the training dataset of the reference model. To determine which group each data record belongs to, we use a threshold value. For non-binary class datasets, if a data record's membership score falls in the range of $(-10, 0]$, it is classified as an easy-to-predict sample; otherwise, it is classified as a hard-to-predict sample. For binary class datasets such as Adult and Credit, the ranges for easy-to-predict and hard-to-predict samples are $(-5, 0]$ and $(-\infty, -5]$, respectively. The TPR values of different MIAs are shown in Figure 9. Note that all the figures share the same label on $y-$axis: TPR at 1% FPR. We can see that *LDC-MIA* significantly improves the TPR for



(a) CIFAR-10 (WRN28-10) (b) CIFAR-10 (VGG-16) (d) CIFAR-100 (DenseNet-121) (d) CIFAR-100 (SmallNet)

(e) CINIC-10 (VGG-16) (f) Adult (MLP) (g) Credit (MLP)

Figure 9: Improvement of MIA performance on target samples of various hardness levels.

both easy-to-predict and hard-to-predict samples across all datasets. However, for threshold-based MIA, such as the one proposed by Yeom *et al.*, it could easily misclassify all member samples. In this experiment, the identification of both hard-to-predict and easy-to-predict members has been improved by the combination of calibrated membership score and membership score on the target model in the classifier.

4.3. Comparison with LiRA

LiRA [3] is a state-of-the-art MIA that can achieve high TPRs in the low FPR regions. To determine membership, LiRA calculates the likelihood ratio, which represents the ratio of the likelihood of a data point being in the target model's training dataset to the likelihood of it being from a different, unknown dataset. The likelihood ratio is computed based on the output probabilities of N shadow models and membership is determined based on the likelihood ratio. In our experiments, we compare *LDC-MIA* with online LiRA. WideResNet28-10 is the target model and is trained on the CIFAR-10 dataset. We then gradually increase the number of shadow or reference models used by LiRA and *LDC-MIA* and compare the TPRs at 1% FPR as well as AUCs. The results are shown in Table 5. From

TABLE 5: Performance of LiRA and *LDC-MIA* on CIFAR-10 (WRN28-10).

Auxiliary Models	TPR at 1%FPR		AUC	
	LiRA	*LDC-MIA*	LiRA	*LDC-MIA*
N=2	2.38%	**9.07%**	0.599	**0.794**
N=4	3.21%	**8.21%**	0.676	**0.806**
N=8	5.46%	**9.02%**	0.656	**0.802**
N=16	7.12%	**9.05%**	0.688	**0.807**
N=32	**15.81%**	8.76%	0.757	**0.805**
N=64	**18.33%**	9.01%	0.771	**0.804**
N=128	**20.75%**	8.59%	0.781	**0.801**
N=256	**21.32%**	8.97%	0.792	**0.806**

the results, we can see that the performance of *LDC-MIA* does not change significantly as the number of auxiliary models increases, while LiRA's performance improves with more auxiliary models. However, the AUC of *LDC-MIA* outperforms the LiRA's, with different numbers of auxiliary models, from 2 to 256. The better AUC across the board indicates that our method *LDC-MIA*, which uses multiple features, can help to distinguish members from non-members better on average. When the number of auxiliary models is no more than 16, the TPRs of *LDC-MIA* at an FPR of 1% are higher than those of LiRA. Note that LiRA achieves better TPRs in the low FPRs when more than 32 auxiliary models are used; however, as discussed in the next section, this could lead to cost concerns.

4.4. Attack Cost

In real-world attacks, the cost of the attack is another key factor to consider alongside the attack performance. A cost-efficient approach, sometimes even a less performant one, may work better for some use cases. Let's consider a few scenarios as follows. First, attackers have a limited budget for an MIA. Not having access to powerful computation hardware disallows them from training many shadow models or reference models for a high-cost attack. Second, MIA is used as an auditing tool to evaluate data privacy [3]. where a model provider needs to evaluate many of the training data to ensure a sufficiently low likelihood of privacy leaks before releasing the model. For this use case, an MIA approach whose cost is sub-linear to the number of sample data is preferred. Third, the aforementioned model provider might need to frequently update the model training dataset for retraining [11], [24]. Running MIA for data leakage audit every time after retraining asks for a balanced tradeoff between the utility and the cost of the MIA approach. Lastly, a less costly MIA approach may serve better with a tight timeline. For example, a model provider must release an updated model by a deadline and finish the MIA data leakage audit on time. A less costly MIA usually requires less computing and thus is faster.

We compare the cost of MIAs by comparing the cost of computing and the cost of data. We approximate the compute cost by calculating the training time and inference time of all auxiliary models used in the MIA (including shadow and reference models). Note that all computing experiments run on the same hardware configuration. We approximate the data cost of different MIAs as the number of auxiliary models used because the MIAs we evaluate take a similar amount of training data for auxiliary models. Therefore, the more auxiliary models are used, the larger the data cost for the MIA. Both *LDC-MIA* and Salem *et al.* [29] need to train a neural network-based MIA classifier in addition to the auxiliary models. These two classifiers employ a shallow multilayer perception (MLP) architecture. It takes around 22 seconds to train such a classifier in all the datasets we experimented with. This is a significantly small cost compared to training one shadow model for image classification. Therefore, these MIA classifiers' training and inference time are not included in the discussion below, where we use the task of attacking CIFAR-10 as an example to demonstrate the attack costs of different MIAS. Based on the results in Section 4.3, *LDC-MIA* does not have much performance gains from employing additional auxiliary models. Hence, we calculate the cost of *LDC-MIA* by including only one shadow model and one reference model.

Table 6 shows the costs of attacking WideResNet28-10 on the CIFAR-10 dataset. We perform the MIAs on 25k target samples and show the sum of attack costs of all the samples. The training cost of the attack does not change as the number of target samples to attack increases, except for LiRA. The training cost for LiRA increases linearly as the number of target samples changes because LiRA requires training multiple shadow models for **each** target sample for a reliable estimate of the likelihood ratio.

TABLE 6: The attack costs of different MIAs.

Attack Method	Auxiliary Models (N)	TPR at 1%FPR	AUC	Training cost (minutes)	Inference cost (s) per attack
Salem *et al.* [29]	1	2.5%	0.731	19	24
Yoem *et al.* [39]	-	0.0%	0.723	-	19
Watson *et al.* [37]	1	3.3%	0.707	19	38
Watson *et al.* [37]	10	3.7%	0.742	189	221
LiRA [3]	16	7.12%	0.688	304	348
LiRA [3]	256	21.32%	0.792	4867	4942
Ours	2	9.1%	0.794	39	43

Yeom *et al.* [39] has the least cost among all, as it requires no auxiliary models and only needs one-time inference on the target model per target sample. Coming with the low cost are the low AUC and TPR of this approach. Our method *LDC-MIA* has a similar cost to Salem *et al.* [29]. Both attacks require training a few auxiliary models that, once ready, can be used for all target sample attacks. Then, per target sample, the attack takes one or two inferences on the target model. The total cost is in the order of tens of minutes. Watson *et al.* [37] method can opt-in to use multiple reference models for better attack performance, though with higher training cost, which is linear to the number of the auxiliary models, as illustrated in Table 6.

In contrast, for LiRA, the training cost is linear to the product of the number of auxiliary models and the number of target samples. For every target sample, LiRA must train a different set of auxiliary models tailored to that particular sample. This is different from all other MIAs, including *LDC-MIA*. Multiple target samples can reuse the same auxiliary models in all other MIAs. Thus, the training cost can be amortized across a batch of target sample attacks.

4.5. Effect of data augmentation

Data augmentation is a technique that can be used to increase the size of a dataset by applying different transformations to existing training data. This technique is often used to improve the generalization and robustness of models. By exposing the model to a wider range of data variations, it can learn to handle different input scenarios, resulting in better performance. In this experiment, we applied data augmentation during the training of the target model and the shadow models (if any). Then, we evaluated its effect on the MIAs by measuring its AUC and TPR at 1% FPR values. Since attackers may not have access to the data augmentation techniques used in the target model in real-life situations, we train the shadow target model with random data augmentations. Specifically, we use horizontal flipping for the shadow (target) model training and random cropping and rotation for the target model training. When querying the target model, we use the original target sample. The experiments are conducted on the CIFAR-10 and CIFAR-100 datasets. The target model for CIFAR-10 is VGG-16, and the target model for CIFAR-100 is SmallNet. The experiment results are shown in Table 7.

TABLE 7: The impact of data augmentation

Attack Method	CIFAR-10				CIFAR-100			
	w/o aug		w/ aug		w/o aug		w/ aug	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Salem *et al.* [29]	0.679	1.2%	0.605	0.2%	0.697	2.3%	0.625	0.6%
Yeom *et al.* [39]	0.696	0.0%	0.602	0.0%	0.889	1.0%	0.802	0.0%
Watson *et al.* [37]	0.705	3.4%	0.651	1.1%	0.772	5.9%	0.709	2.7%
Ours	**0.752**	**6.7%**	**0.712**	**3.8%**	**0.918**	**16.4%**	**0.869**	**12.4%**

From the results, it can be observed that both AUC and TPR decrease when data augmentation is applied. This is because data augmentation reduces overfitting, thereby reducing the effect of MIAs [39]. However, *LDC-MIA* has not been affected as much as the other MIA methods. This suggests that our proposed attack is robust against data augmentation in the target model training.

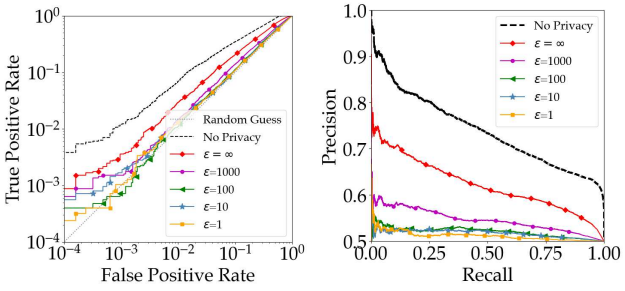
5. Ablation Study

5.1. Differential Privacy

In order to evaluate the robustness of our proposed attack, we utilize the concept of differential privacy during the training of the target model. This technique adds noise or randomization to the data, which helps protect individual privacy in datasets. It can be useful in limiting the effectiveness of many existing MIAs [3], [37], [39]. We use DP-SGD [1], one of the state-of-the-art DP mechanisms, for training the target model in our experiments. DP-SGD adds carefully calibrated noise to the gradients computed during each iteration. The amount of noise

added depends on the sensitivity of the gradients and the desired privacy budget ε. While a smaller ε provides stronger privacy guarantees, it can also result in noisier updates. The other two important parameters in DP-SGD are the clipping bound C and the noise multiplier σ.

The clipping bound is a threshold value applied to the gradients computed during training. This operation limits the influence of any single data point on the model's parameters. The noise multiplier is a parameter that determines the amount of noise added to the gradients during each iteration of the training. In practice, to achieve a specific privacy budget ε, one can adjust the noise multiplier σ and the total number of iterations. In our experiments, we set C to 10 and vary σ from 0.0 to 1.0 to adjust ε. We evaluate the performance of the proposed attack at different ε values (∞, 1000, 100, 10, and 1). The PR curve and the ROC curve are shown in Figure 10, and the AUC and TPR are shown in Table 8.



(a) ROC Curve (b) Precision-Recall Curve

Figure 10: Effectiveness of using DP-SGD against our attack with different privacy budgets.

TABLE 8: Performance of *LDC-MIA* against DP-SGD for Smallnet trained on CIFAR-10.

σ	ε	Model acc	AUC	TPR at 0.1%FPR
0	∞	64.51%	0.646	0.3%
0.2	1000	59.35%	0.560	0.2%
0.3	100	52.56%	0.529	0.1%
0.6	10	43.65%	0.524	0.2%
1	1	28.91%	0.513	0.1%

Figure 10 and Table 8 show that as the desired privacy budget ε increases, both AUC and TPR decrease. However, in practice, there is a trade-off between privacy (ε) and utility (the accuracy of the trained model). When higher privacy is required, adding more noise can significantly affect the model's utility. This is evident from the model accuracy column in Table 8. In practice, to preserve model accuracy, reasonable values of ε, such as 100 or 1000, are more likely to be used. It is observed that AUC and TPR are not significantly reduced in these cases. Moreover, even with small ε values, the proposed attack can still achieve high precision values ($> 90\%$) at low recall, as seen from Figure 10b.

5.2. Overfitting Level of the Target Model

Previous studies [29], [31] have demonstrated that the performance of MIAs is closely related to the overfitting level of the target model. Overfitting occurs when the model fits the training data too well, even with noise and unique patterns to its training dataset. Several factors can affect a model's overfitting level, such as dataset size and quality, training rounds, model complexity, and regularization. Typically, increasing the training dataset's size helps reduce overfitting, while decreasing it has the opposite effect. Because a larger and more diverse dataset allows the model to observe a broader range of variations and generalize better with less memorization of specific data points.

In our experiments, we adjust the training dataset size between 6500 and 12500 to vary the target model's overfitting level. Meanwhile, we keep the size of the training dataset of the reference model $\mathcal{D}_{ref}^{train}$ and that of the shadow target model $\mathcal{D}_{shadow}^{train}$ fixed. We then measure the impact of the overfitting level by evaluating AUC and TPR of the proposed attack. The results are shown in Table 9.

TABLE 9: The effect of overfitting on the target model of VGG-16 trained on CIFAR-10.

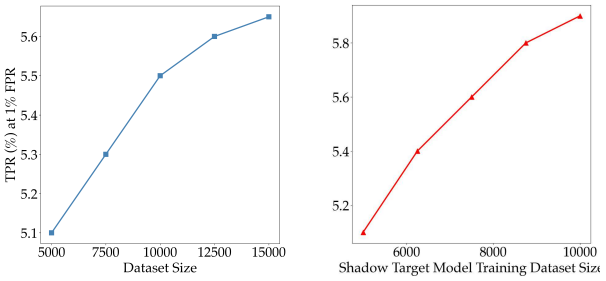
Training dataset size	Train Test Acc Gap	*LDC-MIA* AUC	*LDC-MIA* TPR at 1%FPR
6500	37.91	0.787	8.1%
8000	36.28	0.783	7.6%
9500	34.08	0.770	7.3%
11000	29.96	0.761	6.1%
12500	26.91	0.754	5.7%

Table 9 depicts that the overfitting level of the target model increases as the size of its training dataset decreases, by looking at the gap between the training accuracy and test accuracy of the target model. The larger the gap between these two, the more the model is overfitting. We note that in our experiment for our method *LDC-MIA* the AUC and TPR at 1% FPR of the MIA improve slightly as the overfitting level of the target model increases. Membership inference attack benefits from a higher level of overfitting, which could mean a higher level of memorization. The above results align with the findings on other MIAs in the literature.

5.3. Training Dataset Sizes for the Shadow Target and the Reference Models

The size of the datasets used to train the shadow target and reference models is an important factor in our proposed attack. As discussed in Section 5.2, the training dataset sizes affect the performance of the trained shadow target and reference models, leading to performance variance in the proposed attack. To evaluate this factor, we divide an auxiliary dataset $\mathcal{D}_{aux}$ consisting of 25k data records into two parts: $\mathcal{D}_{shadow}^{train}$ for training the shadow target model and $\mathcal{D}_{ref}^{train}$ for training the reference model. We set up two configurations to vary the sizes of $\mathcal{D}_{shadow}^{train}$ and $\mathcal{D}_{ref}^{train}$. In the first configuration, we fix $\mathcal{D}_{shadow}^{train}$ to be $1/5$ of $\mathcal{D}_{aux}$ and vary the size of $\mathcal{D}_{ref}^{train}$. In the second configuration, we do the opposite by fixing $\mathcal{D}_{ref}^{train}$ to be $1/5$ of $\mathcal{D}_{aux}$ and vary the size of $\mathcal{D}_{shadow}^{train}$. We then evaluate the TPR at 1% FPR of *LDC-MIA* on the VGG-16 target model trained with the CIFAR-10 dataset. The results are shown in Figure 11. Note that the two subfigures have the same label on $y-$axis.

Figure 11 shows that increasing the size of the datasets, $\mathcal{D}_{shadow}^{train}$ and $\mathcal{D}_{ref}^{train}$, improves the TPR of *LDC-MIA*. Increased size of $\mathcal{D}_{shadow}^{train}$ improves the generaliza-



(a) Fixed $\mathcal{D}_{shadow}^{train}$ size (b) Fixed $\mathcal{D}_{ref}^{train}$ size

Figure 11: The impact of the training dataset sizes of the shadow target and the reference models.

tion of the shadow target model. The MIA classifier benefits from the better shadow target model due to exposure to a broader range of membership scores and calibrated membership scores. On the other hand, the increased size of $\mathcal{D}_{ref}^{train}$ improves the reference model, which leads to the improvement of the MIA classifier through more accurate calibrated membership scores. Figure 11 also shows that the effect on the MIA classifier performance of the size of $\mathcal{D}_{shadow}^{train}$ is similar to that of the size of $\mathcal{D}_{ref}^{train}$.

5.4. Model Architectures

Let's consider a real-world threat scenario where the attackers do NOT have knowledge about the architecture of the target model, so they guess an architecture for auxiliary models. We want to analyze the impact of having different network architectures in the shadow target model on the performance of *LDC-MIA*. To do so we randomly select two models from VGG-16, ResNet-18, ResNet-34, and Smallnet as the target and shadow target models while keeping the reference model's architecture the same as the shadow target model. We evaluate the performance of *LDC-MIA* using AUC and TPR at 1% FPR as metrics, on the CIFAR-10 dataset. The results are shown in Figure 12.

Figure 12 suggests that the architecture of the shadow target model does NOT have much impact on the attack performance. TPR values of *LDC-MIA* are the best when the shadow target model shares the same architecture as the target model. However, AUC of *LDC-MIA* can achieve the highest value when the two model architectures differ in some cases, as depicted in Figure 12b. For instance, when the target model is VGG-16, the best-performing shadow target model uses ResNet-18. These results suggest that for *LDC-MIA* there is no absolute need to know the specific model architectures of the target model to launch an equivalently successful attack. Although we note that we use a limited number of pre-defined candidate architectures for guessing in this experiment, therefore the results are indicative but not comprehensive.

5.5. Model Learning Optimizers

There are more than one optimizers for training the machine learning models, Some widely-use examples include SGD, SGDM, and ADAM. Some optimizers provide better regularization, leading to better generalization and reduced overfitting. For instance, ADAM, a commonly used optimizer, is considered helpful in mitigating memorization. We use VGG-16 model on CIFAR-10 dataset in an experiment to investigate (1) What impact does the optimizer have on the attack performance? and (2) Does knowing which optimizer is used in the target model improve attack performance?

Figure 13 presents the TPR values at low FPR values, with the target model's optimizer shown on the x-axis, and different markers indicating the performance of various optimizers used in the shadow target model.

Figure 13 indicates that there is no significant effect on attack performance by varying optimizers for training the target model. The figure also shows that for every optimizer we test, the attacker can achieve the highest TPR by applying the same optimizer in the shadow target model as in the target model. Interestingly, in our experiment applying SGDM in the shadow target model consistently achieves better attack performance if the exact optimizer in the target model is unknown to the attacker.

5.6. Different Features

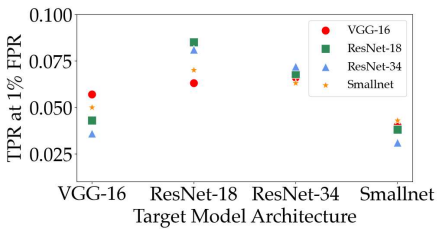
To evaluate the impact of the features introduced into the classifier in *LDC-MIA*, we remove one feature each time from the model training and compare the performance of the resulted classifiers that are trained on all-but-one features. The MIA classifier in *LDC-MIA* has three features — membership scores on the target model, calibrated membership scores, and labels. We compare the contribution of each feature to the attack by analyzing the full log-scale ROC curves. The target model in this experiment is VGG-16 trained on CIFAR-10. The results are shown in Figure 14. The figure indicates that removing any of the features results in degraded attack performance, and each feature contributes to the attack in different ways. Firstly, removing the label feature leads to a reduction in TPR at low FPR. This indicates that the label feature helps reduce false positives, leading to improved TPR, especially at low FPR. Secondly, removing the membership score reduces TPR in all FPR regions. This verifies what we discussed in Section 3.2 — that including the membership scores not only helps identify hard-to-predict members but also easy-to-predict members, thus improving TPR in all FPR regions. Finally, the performance of the proposed attack significantly degrades by excluding the calibrated membership scores. This is because the calibrated membership scores help separate the hard-to-predict members from the easy-to-predict non-members, which is a significant portion of the non-members, easily. Overall, all three features offer unique contributions to the success of our proposed attack.

6. Related Works

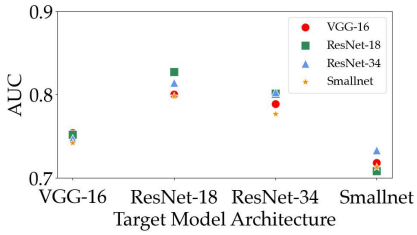
6.1. Membership Inference Attacks

Although membership Inference Attacks (MIA) can serve as an audit mechanism to verify the privacy of machine learning models [12], [20], [29], [33], they have become a major concern for privacy if used by miscreants, to leak sensitive data in the training dataset of the models.

In traditional MIAs, such as the work by Shokri *et al.* [31], attackers utilize the auxiliary dataset to train several shadow models to mimic the behavior of the target



(a) TPR at 1%FPR



(b) AUC

Figure 12: The impact of the architecture differences between the shadow target and the target models.

Figure 13: The impact of different training algorithms.

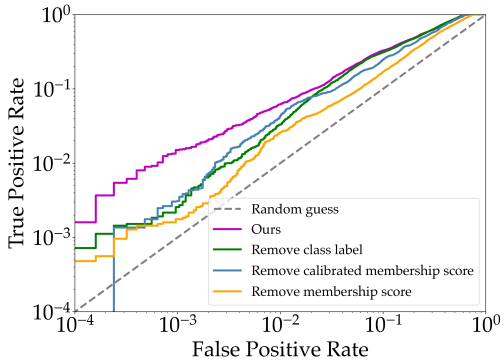


Figure 14: The ROC curve of *LDC-MIA* when different features are removed.

model. By analyzing the output generated by these shadow models, attackers can train a binary classifier that captures the difference in confidence scores for members and non-members of the shadow models. This binary classifier is then used to infer whether a target sample is a member or not based on its confidence score obtained from the target model. Salem *et al.* [29] proposed a similar attack using shadow model and classifiers but only using a single shadow model, which significantly reduces the cost associated with executing MIAs. These early techniques set the foundation for subsequent research by demonstrating the feasibility of MIAs.

Yeom *et al.* found that the success of MIAs is positively correlated with model overfitting, which they leverage to identify members by thresholding its membership score. If the score exceeds a pre-defined threshold, the sample is deemed a member. There are similar approaches for MIA by metrics thresholding [5], [28], [33]. Most of these works, set the threshold through simple statistics, while our method *LDC-MIA* uses a machine learning algorithm to identify more accurate thresholds that are learned by the algorithm from data.

More advanced MIAs use statistical methods, such as likelihood ratios and hypothesis testing, to distinguish subtle patterns in model behaviors trained with certain samples [3], [22], [37]. Some of these methods use auxiliary models to measure the differences in model behavior with or without a sample Difficulty calibration is introduced to better characterize the differences for different groups of instances based on their difficulty for MIA. Watson *et al.* [37] introduced a calibrated membership score that improves the attack performance by taking into account the hardness of individual samples. Carlini *et al.* [3] extended this concept by proposing Likelihood Ratio Attacks (LiRA) that sample dozens to hundreds of shadow models for each instance to characterize the differences between models trained with that instance and those without. In our work, we introduce several features to characterize the instances and leverage them for better difficulty calibration. To the best of our knowledge, LiRA achieves the highest TPRs at low FPRs. However, the online LiRA attack method requires training hundreds of auxiliary models for each target sample to achieve optimal attack performance. We consider it to be excessively expensive for real-world attacks. Our method *LDC-MIA* is orders of magnitude less expensive while achieving close performance in some of the datasets.

6.2. Defense Against MIA

Some defense methods mitigate MIAs by reducing the excessive memorization of training data by the target model. For example, training models with DP-SGD learning algorithm [1], which incorporates differential privacy related metrics in the learning objective. In our ablation study, we show that the use of DP-SGD in the target model indeed impacts the performance of our MIA method. The downsides of differential-privacy methods tend to lead to reduced target model accuracy. Additionally, regularization techniques such as dropout [34] and weight decay [18] defend against MIAs by lowering the model's overfitting. Recently, studies such as DMP [30], SELENA [35], and PATE [27] use knowledge distillation to defend MIA and demonstrate some success, while study in [16] shows that distillation alone provides only limited privacy across a number of domains.

7. Conclusion

In this paper, we delve into the difficulty calibration based MIAs and propose a novel learning-based attack, called *LDC-MIA*. This attack improves the performance of MIA, particularly the TPRs at low FPRs, by using features that characterize the hardness levels of data records. To achieve this, we leverage target samples' labels, neighborhood information, calibrated membership score, and membership score on the target model. Our experiments show that *LDC-MIA* can achieve state-of-the-art performance in terms of TPRs at low FPRs, AUC, and precision at high recall rates while keeping the attack cost relatively low.

8. Acknowledgement

We would like to thank our sheperd and the reviewers of Euro S&P'24 for their invaluable feedback. This work is partially supported by an NSF grant CNS-2008468 and an ONR grant N00014-23-1-2137.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[3] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[4] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.

[5] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[7] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang. Credit card fraud detection using convolutional neural networks. In *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part III 23*, pages 483–490. Springer, 2016.

[10] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[11] Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, et al. Applied machine learning at facebook: A datacenter infrastructure perspective. In *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 620–629. IEEE, 2018.

[12] Xinlei He, Zheng Li, Weilin Xu, Cory Cornelius, and Yang Zhang. Membership-doctor: Comprehensive assessment of membership inference against machine learning models. *arXiv preprint arXiv:2208.10445*, 2022.

[13] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[15] Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.

[16] Matthew Jagielski, Milad Nasr, Christopher Choquette-Choo, Katherine Lee, and Nicholas Carlini. Students parrot their teachers: Membership inference on model distillation. *arXiv preprint arXiv:2303.03446*, 2023.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[18] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Zheng Li, Yiyong Liu, Xinlei He, Ning Yu, Michael Backes, and Yang Zhang. Auditing membership leakages of multi-exit networks. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1917–1931, 2022.

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

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

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[24] Lucy Ellen Lwakatare, Aiswarya Raj, Ivica Crnkovic, Jan Bosch, and Helena Holmström Olsson. Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. *Information and software technology*, 127:106368, 2020.

[25] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

[26] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[27] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

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

[29] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

[30] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9549–9557, 2021.

[31] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[35] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1433–1450, 2022.

[36] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

[37] Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440*, 2021.

[38] Zhennan Yan, Yiqiang Zhan, Zhigang Peng, Shu Liao, Yoshihisa Shinagawa, Shaoting Zhang, Dimitris N Metaxas, and Xiang Sean Zhou. Multi-instance deep learning: Discover discriminative local anatomies for bodypart recognition. *IEEE transactions on medical imaging*, 35(5):1332–1343, 2016.

[39] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.