Practical Blind Membership Inference Attack via Differential Comparisons

Bo Hui[†*], Yuchen Yang[†*], Haolin Yuan[†*], Philippe Burlina[‡], Neil Zhenqiang Gong[§] and Yinzhi Cao[†]
[†]The Johns Hopkins University [‡]The Johns Hopkins University Applied Physics Laboratory [§]Duke University

***Abstract*—Membership inference (MI) attacks affect user privacy by inferring whether given data samples have been used to train a target learning model, e.g., a deep neural network. There are two types of MI attacks in the literature, i.e., these with and without shadow models. The success of the former heavily depends on the quality of the shadow model, i.e., the transferability between the shadow and the target; the latter, given only blackbox probing access to the target model, cannot make an effective inference of unknowns, compared with MI attacks using shadow models, due to the insufficient number of qualified samples labeled with ground truth membership information.**

In this paper, we propose an MI attack, called BLINDMI, which probes the target model and extracts membership semantics via a novel approach, called differential comparison. The high-level idea is that BLINDMI first generates a dataset with nonmembers via transforming existing samples into new samples, and then differentially moves samples from a target dataset to the generated, non-member set in an iterative manner. If the differential move of a sample increases the set distance, BLINDMI considers the sample as non-member and vice versa.

BLINDMI was evaluated by comparing it with state-of-the-art MI attack algorithms. Our evaluation shows that BLINDMI improves F1-score by nearly 20% when compared to state-of-the-art on some datasets, such as Purchase-50 and Birds-200, in the blind setting where the adversary does not know the target model's architecture and the target dataset's ground truth labels. We also show that BLINDMI can defeat state-of-the-art defenses.

I. INTRODUCTION

Machine learning (ML), especially Deep Learning (DL), has achieved, or even surpassed, human-level performance on many critical areas, such as medical diagnosis [5], [6], image and speech recognition [17], [20], [23], [51], self-driving cars [3], and natural language translation [29]. Despite this success, one major issue of DL models like deep neural networks (DNNs) has been their vulnerability to a variety of attacks [12], [38], [46], [47]. A type of privacy-related attack—i.e., the focus of the paper—is the membership inference (MI) attack [40], [41], [43], [49], whereby an adversary infers whether a specific sample belongs to the training set of a given learning model, defined as a membership. For example, an adversary can infer whether a specific disease image from a given hospital was used to train an artificial intelligent (AI) diagnostic system, thus potentially violating patients' protected health information (PHI) and Health Insurance Portability and Accountability Act (HIPAA) provisions. For another example, the inference of location data used in an AI recommendation system may leak users' past physical location, violating their privacy.

The high-level intuition behind membership inference attacks is that the output probability distributions of a DNN model from, say for example, a Softmax layer, may vary between members and a non-members. While intuitively simple, one major *challenge* to this idea is that an adversary, when only given blackbox access (i.e., only having access to the output probability distribution), needs to collect enough samples with output probabilities and labeled as either members or non-members to classify a new data sample with unknown membership. On one hand, many existing MI attacks—e.g., the DNN-based from Shokri et al. [43], the loss function-based from Yeom et al. [49], and another DNN-based with feature selections from Salem et al. [41]—all adopt an offline shadow model trained from a surrogate dataset, that provides ground truth information on whether a given sample is a member. However, such shadow models differ from the real target model and thus the output probability distributions, though being similar, as noted by prior studies on model transferability [10], [21], [50], are still different. Therefore if the shadow model is drastically different from the target, the attack performance will degrade significantly as shown by Salem et al. [41] and as also confirmed by our own experiments.

On the other hand, researchers have also proposed MI attacks without shadow models. For example, the unsupervised and adversary binary attacks of Salem et al. [41] consider a sample as a member if the probability of the predicted class is larger than a threshold learned from one thousand randomly generated samples. Another approach, the label-only attack of Yeom et al. [49], infers membership by comparing the ground truth against the predicted label. However, both existing shadow-model-free attacks rely binary comparisons, e.g., comparing the predicted probability or label with a pre-determined threshold or the ground-truth label. Such a "one-size-fit-all" inference cannot model the complex decision boundary between members and nonmembers in the hyper-dimensional space induced by an inference neural network in shadow-model-dependent attacks. The root reason in lacking such modeling ability goes back to the aforementioned challenge: a powerful MI attack needs enough labeled output probability distributions of members and non-members to learn the decision boundary, but the ground truth information of members and nonmembers for the target model is unavailable given only blackbox access.

[*] The first three authors have equal contributions to the paper.

Network and Distributed Systems Security (NDSS) Symposium 2021
21-24 February 2021
ISBN 1-891562-66-5
https://dx.doi.org/10.14722/ndss.2021.24293
www.ndss-symposium.org

In this paper, we propose a novel MI attack, called BLINDMI, which probes the target model and then infers membership directly from the probing results instead of shadow models. BLINDMI exploits two insights: The first is that although an adversary does not have both member and nonmember labels of the target model, the adversary can easily obtain one-class labels, i.e., nonmember labeled samples, by producing newly-constructed samples that can be considered as non-members with high probability given the very large input space of possibilities. Such one-class semantics can be learned by existing ML classifiers, like a one-class SVM, thus leading to an MI attack defined as BLINDMI-1CLASS. This BLINDMI-1CLASS serves as a baseline approach if we only exploits the first insight of BLINDMI.

The second insight is that the removal of a non-member from a dataset containing both members and non-members, will move the entire set away from non-members in the hyper-dimensional space, and conversely, the addition moves it towards it. Therefore, assume that we have two datasets: one is closer to nonmembers and the other to members. If we move one sample from the latter to the former and the distance between two sets decreases, the moved sample can be considered a member; otherwise, if it increases, the sample can be considered a non-member. This approach is called *differential comparison* in this paper as it compares the differential distance between two sets. One advantage of this approach is that it only needs two small-size sets as opposed to a considerable amount of data for a one-class classifier, while achieving a comparatively higher inference performance.

Specifically, we design an attack, called BLINDMI-DIFF, which performs differential comparison to infer membership. Following upon the first insight, BLINDMI-DIFF obtains a dataset with nonmembers. Then, BLINDMI-DIFF differentially compares the dataset with a given set of data samples, called a target dataset, following the second insight to remove all the nonmembers from the target. The entire differential comparison procedure is iterated until convergence, i.e., the move of any samples between two sets only decreases the distance. Then, the remaining samples in the target dataset are considered as members.

We implement a prototype of BLINDMI[1] including BLINDMI-DIFF and BLINDMI-1CLASS. Our evaluation shows that BLINDMI outperforms state-of-the-art membership inference attacks in terms of F1-score in different settings, e.g., even when the adversary knows the target model's exact architecture and hyper-parameters. Furthermore, we evaluate BLINDMI and other attacks under realistic assumptions following Bargav et al. [25] to adjust the nonmember-to-member ratio in the target dataset and show that even if the ratio is as high as 39, BLINDMI still has an over 50% F1-score as opposed to below 30% of the state-of-the-art MI attacks. We also test BLINDMI against existing defenses, including Adversarial Regularization [36], MemGuard [26], Mixup + MMD [30] and differential privacy [1], and show that BLINDMI can break these defenses by achieving reasonable F1-score with different privacy-utility budgets.

[1]The default version of BLINDMI and BLINDMI-DIFF, without specification, is BLINDMI-DIFF with generated non-members, called BLINDMI-DIFF-w/.

TABLE I. DIFFERENT THREAT MODELS AND THEIR ASSUMPTIONS.

	output distr.	model arch.& hyper-parameter	targets' true labels
Blind (default)	✓	✗	✗
Blackbox	✓	✗	✓
Graybox	✓	✓	✓
Graybox-Blind	✓	✓	✗

II. OVERVIEW

In this section, we first present our threat model and then describe overarching assumptions and principles used throughout the paper.

A. Threat Model

Our threat model assumes an adversary trying to infer whether each sample in a given input dataset, called the target dataset, belongs to—i.e., is a member of—the training set of a deep learning (DL) model, called the target model. The adversary can probe the target DL model with samples to obtain the probability distribution of output classes. There are four different variations of the threat model based on the adversary's capability as described below and shown in Table I.

- Blackbox-Blind, or called Blind for short. The blind setting only grants an adversary blackbox access to the target model without details of its architecture, network weights, or hyper-parameters. Further, the adversary does not have ground truth class labels of the target dataset, which usually takes a considerable amount of manual effort sometimes even from specialized experts, e.g., a highly trained ophthalmologist and retinal specialist in labeling the existence of certain diseases for the EyePACS dataset.
- Blackbox. The blackbox setting is similar to the blind, but assumes that the adversary has the ground-truth information of all the samples in the target dataset via, e.g., manual labelling. Note that some existing attacks, e.g., Yeom et al. [49], only work if such ground-truth information is available.
- Graybox. The graybox setting gives full knowledge to the adversary in terms of the model details. Specifically, except for the training data, the adversary knows almost everything about the model, such as the architecture (e.g., VGG, ResNet, and DenseNet) and the hyper-parameters used for training (e.g., learning rate and maximum number of epochs). Note that the adversary cannot know the training data (called a whitebox), because MI attacks are unnecessary in such a setting.
- Graybox-Blind. The graybox-blind setting is similar to the graybox one, but also assumes that the adversary does not have ground-truth information of the target dataset.

Note that our *default* threat model setting is blind unless otherwise noted, because the blind setting is the most strict and practical for membership inference attack. We also adopted other settings in comparison with prior works, e.g., blackbox with Yeom et al. [49] and graybox(-blind) with Shokri et al. [43] and Salem et al [41].

B. Problem Formulation and Notations

The attack problem considered in this paper is as follows: given a target model, i.e., F_m (see recapitulation of notations

TABLE II. NOTATIONS OF SYMBOLS IN THE PAPER

Notation	Description
S_{target}	The target dataset of membership inference attack
F_m	The target DL model with m prediction classes
S^{prob}_{target}	$S^{prob}_{target} = \{y = F_m(x) \mid x \in S_{target}\}$
$G_{projection,k}$	$G_{projection,k} : \mathbb{R}^m \rightarrow \mathbb{R}^k$
$S^{prob,k}_{target}$	$S^{prob,k}_{target} = \{y' = G_{projection,k}(y) \mid y \in S^{prob}_{target}\}$
S_{nonmem}	A generated dataset with non-members of F_m
S^{prob}_{nonmem}	$S^{prob}_{nonmem} = \{y = F_m(x) \mid x \in S_{nonmem}\}$
$S^{prob,k}_{nonmem}$	$S^{prob,k}_{nonmem} = \{y' = G_{projection,k}(y) \mid y \in S^{prob}_{nonmem}\}$

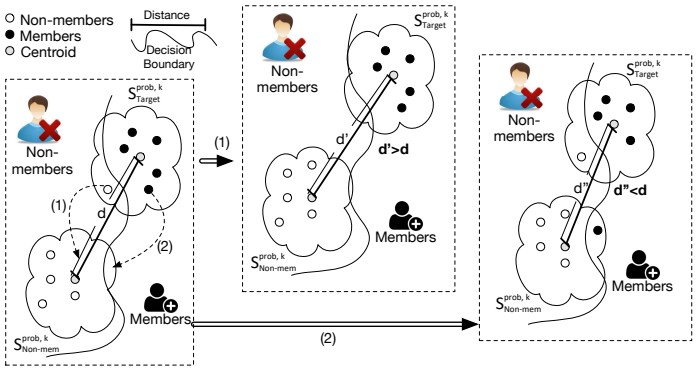


Fig. 1. A High-level Idea of the Key, Atomic Step in Differential Comparisons. BLINDMI measures the distance d between $S^{prob,k}_{target}$ and $S^{prob,k}_{nonmem}$ and moves a sample from $S^{prob,k}_{target}$ to $S^{prob,k}_{nonmem}$. Then, BLINDMI recalculates the distance d' and compares d' with d. If d' is larger than d, BLINDMI considers the moved sample as a nonmember; otherwise, BLINDMI considers it as a member. This is an iterative process until convergence.

in Table II), with m classes and a target dataset S_{target}, BLINDMI is tasked with inferring whether each sample in S_{target} belongs to the training set of F_m. The adversary will feed the set of samples S_{target} into F_m, obtain the set of output probability distributions, i.e., S^{prob}_{target}, and then applies a projection function $G_{projection,k}$ that converts all data samples from m dimensions to k dimensions for inference. The converted samples form into a new set called $S^{prob,k}_{target}$. One example $G_{projection,k}$ is the selection of top three probabilities in $y \in S^{prob}_{target}$ plus the one corresponding to the ground truth class.

Another important dataset is a generated dataset, S_{nonmem}, which is a reference dataset to determine whether samples in S_{target} are members. Elements in S_{nonmem} are all—or mostly—non-members of the target model training dataset. Similarly, the adversary will also obtain S^{prob}_{nonmem} and $S^{prob,k}_{nonmem}$ for the comparison with S_{target}.

C. Differential Comparison Intuition

We now introduce the high-level idea, i.e., differential comparison, in Figure 1. We depict two datasets, $S^{prob,k}_{nonmem}$ and $S^{prob,k}_{target}$, in the output probability distribution sub-space. The curve dividing the space is the boundary between member (right) and non-member (left). $S^{prob,k}_{nonmem}$ is located more towards the left because it consists exclusively of samples with high probability of being non-members, while $S^{prob,k}_{target}$ more or less in the middle between members and non-members.

Intuitively, the idea of differential comparison is to move one sample from $S^{prob,k}_{target}$ to $S^{prob,k}_{nonmem}$. If the moved sample is a non-member like Case (1) in Figure 1, $S^{prob,k}_{nonmem}$ moves further towards the left and $S^{prob,k}_{target}$ to the right. Therefore, the distance between $S^{prob,k}_{target}$ and $S^{prob,k}_{nonmem}$ increases from the original d to d'. If the moved sample is a member like Case (2), the distance will decrease from d to d''since both sets are now comprised of a mixture of samples. Such a change in d can then be used to infer whether the moved sample is a member.

While intuitively simple, the distance between $S^{prob,k}_{nonmem}$ and $S^{prob,k}_{target}$ changes over time after each move. That is, what we described in the previous paragraph is a key, atomic step of differential comparison. In practice, this atomic step is repeated until no more samples can be moved: The series of moves with a fixed d is defined as one iteration. Then, differential comparison will update d based on new $S^{prob,k}_{target}$ and $S^{prob,k}_{nonmem}$ for another iteration until the distance d does not change across iterations, called convergence.

There are two things worth noting here. First, differential comparison moves one sample instead of removing it so as to maximize the distance change. Removal of a sample only changes the position of $S^{prob,k}_{target}$ with regards to the decision boundary in the hyper-dimensional space (like Figure 1); as a comparison, moving the sample changes the positions of both $S^{prob,k}_{target}$ and $S^{prob,k}_{nonmem}$, thus improving the algorithm's sensitivity. Second, even after convergence, there may still exist some nonmembers left in $S^{prob,k}_{target}$, i.e., the moving of these samples does not increase the distance between $S^{prob,k}_{target}$ and $S^{prob,k}_{nonmem}$. This is a lower probability situation, as shown in our evaluation of differential comparison's performance, and that this is as to be expected due to an inherent ambiguity between members and non-members.

TABLE III. DIFFERENT VARIATIONS OF BLINDMI

Variations	Description
BLINDMI-DIFF	differential comparison version
BLINDMI-DIFF-W/ (*default*)	BLINDMI-DIFF with generated non-member set
BLINDMI-DIFF-W/O	BLINDMI-DIFF without generated non-member set
BLINDMI-1CLASS	one-class SVM version with generated non-members as training set

III. DESIGN

In this section, we describe a detailed design of BLINDMI.

A. Overall Attack Procedure

We now describe the overall procedure of BLINDMI in Figure 2. BLINDMI takes target samples with unknown membership and outputs a membership result for each individual sample. Specifically, BLINDMI first generates a non-member dataset and then queries the target DNN model with the target and the non-member datasets to obtain the output probabilities. Then, BLINDMI applies a projection function to select certain important features from the output probabilities. The next step depends on different variations of BLINDMI. BLINDMI-DIFF adopts differential comparison to classify members and non-members in the target dataset; BLINDMI-1CLASS trains a one-class model from the selected output probabilities of the non-member and classifies samples in the target dataset using the trained model. We list different variations of BLINDMI in Table III. Both the BLINDMI-DIFF-W/ and BLINDMI-1CLASS require a generated nonmember set as opposed to BLINDMI-

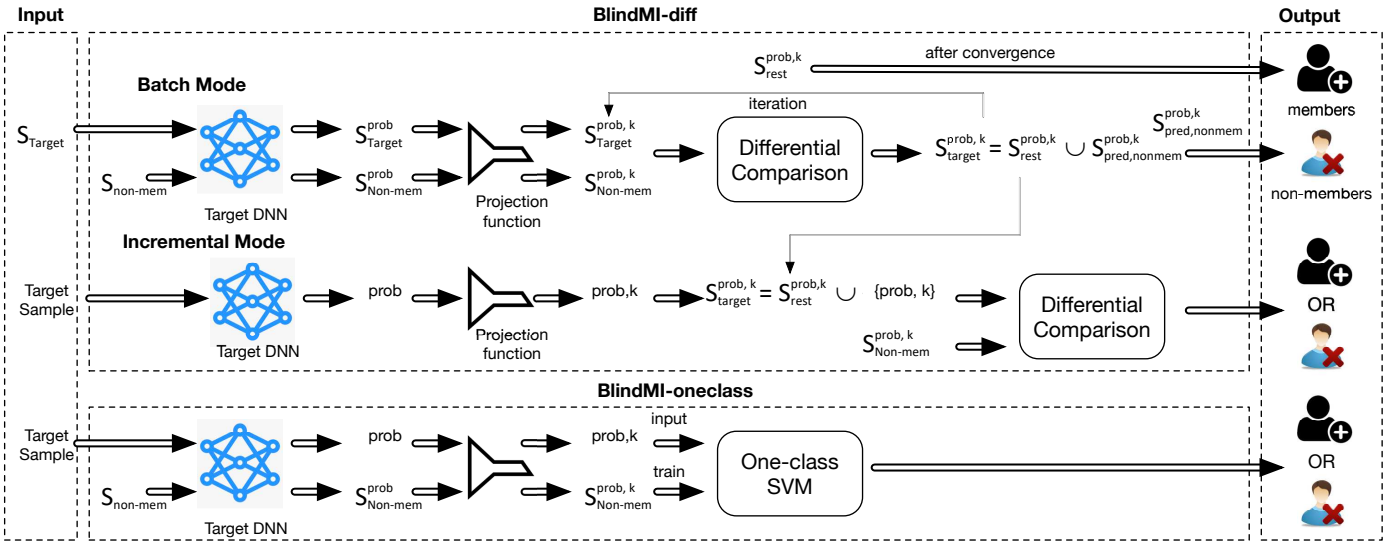


Fig. 2. Overall Attack Procedure of BLINDMI

TABLE IV. METHODS IN GENERATING NON-MEMBERS

Options	Description
Sample transform	Laplace, Sobel, Scharr, and Canny
Random perpetuation	Gaussian or Salt Pepper noise
Random generation	Random feature values
Cross domain	Samples from a different domain

DIFF-w/o that does not require it. The default BLINDMI is BLINDMI-DIFF-w/. We introduce BLINDMI-DIFF-w/o since it may be hard to create an effective nonmember set in some cases and the created set could be contaminated; at the same time, we introduce BLINDMI-1CLASS more as a baseline to separate the effectiveness of differential comparison and the generation of a nonmember set.

Next, we introduce two different modes of BLINDMI: batch and incremental. The batch mode takes and classifies a batch of target samples iteratively and the incremental mode takes one sample, adds to a previous batch, and then classifies the target. It is worth noting that both modes give the same inference results for given samples. They are designed to handle different scenarios: The batch mode for a cluster of samples and the incremental mode for each individual sample.

B. Dataset Preparation for Differential Comparison

In this subsection, we describe how to generate non-members for a target model. There are two general directions of non-member set generation: (i) generating new samples or transforming existing ones and (ii) roughly separating a set with existing samples into two with members and non-members.

1) Generation of Non-members: Let us start from the first direction. This is applicable if an adversary can probe the target model with arbitrary samples. From a high-level perspective, because the input space is usually much larger than the training set, the adversary can generate a new sample, which is likely not in the training set. We now discuss four generation methods below and in Table IV.

- Sample Transformation. An adversary applies an operator, e.g., Laplace, Sobel, Scharr, and Canny, on an existing sample to obtain a new one. Take Sobel for example: the adversary transforms an image to one emphasising edges. The advantage of this method is that it usually preserves some semantics, thus being effective to be distinguishable from members. Additionally, the generated sample is stealthy as all the operators are commonly used in image processing. The down-side of this method is that many operators are specific to the image domain.
- Random Perpetuation. An adversary adds random noises, like Gaussian and Salt Pepper, to an existing sample for the generation. This method is also effective as many semantics are preserved, but less stealthy because one may detect noise levels in the frequency domain.
- Random Generation. An adversary generates a sample with random features. This method is less effective as the generated samples, e.g., a random image, may not have any semantics, and less stealthy as any human can easily spot the generated sample as a noise.
- Cross-domain Samples. An adversary may adopt samples from another domain, e.g., a celebrity face dataset for a model trained with CIFAR-100. This is also effective but less stealthy, because the probed samples are apparently from a different domain.

Note that samples generated following Table IV are highly likely non-members. Therefore, these samples can be used in both BLINDMI-1CLASS and BLINDMI-DIFF, particularly the training set of BLINDMI-1CLASS and the comparison set of BLINDMI-DIFF.

2) Rough Sample Separation: We then describe the second direction. This is applicable when the adversary does not have free probing access to the target model, but only obtain the output probability distribution of a limited dataset. This scenario may happen if the adversary is only allowed to probe samples from a certain source, e.g., disease images acquired at a specific hospital. There are two different methods used in such separation: (i) a clustering algorithm like k-means and agglomerative clustering, and (ii) a separation based on the highest probability score. The first method is to apply clustering to roughly divide the target dataset into two, one as members and the other as non-members. The second is to

roughly select those with high probability score as members and those with low as non-members. This generation is only applicable to BLINDMI-DIFF, because the non-member set may be noisy.

C. Probability Score Projection

We now discuss our probability score projection function $G_{projection,k}$, which applies on S^{prob}_{target} to obtain k different elements. The high-level idea is that class types, e.g., a bird vs. a tree, are less important features for an MI attack, but the ranking of values in different classes determines the membership. Based on this insight, we design three different projection functions.

- All probability scores in an order. This projection function ranks all probability scores from the largest to the smallest, which removes the class information but only keeps the relative values.
- Top-k probability scores. This projection function selects the top-k probability scores to further remove some noisy ones with small values.
- Top-k + Ground Truth Class. This projection function—used in the blackbox setting—further includes the value corresponding to the ground truth class.

D. Differential Comparison

In this section, we describe one key technique, i.e., differential comparison, in this paper. The first task is to calculate the distance between two sets. Just like all general ML tasks, it is hard to differentiate member and non-members directly in the output probability distribution space. Therefore, BLINDMI maps all probabilities to the Reproducing Kernel Hilbert Space (RKHS) [4] and then calculates the distance between two centroids in the kernel space. Specifically, our distance, based on Maximum Mean Discrepancy (MMD) [18], is shown in Equation 1.

$$D(S^{prob,k}_{target}, S^{prob,k}_{nonmem}) = \left\| \frac{1}{n_t}\sum_{i=1}^{n_t}\phi(y_i) - \frac{1}{n_n}\sum_{j=1}^{n_n}\phi(y'_j) \right\|_{\nu} \quad (1)$$

where $y_i \in S^{prob,k}_{target}$, $y'_j \in S^{prob,k}_{nonmem}$, n_t and n_n are the size of $S^{prob,k}_{target}$ and $S^{prob,k}_{nonmem}$, ν is the dimension of the kernel space, and ϕ is a feature space map $k \mapsto \nu$, e.g., a Gaussian kernel function $k(y, y') = \langle \phi(y), \phi(y') \rangle = exp(-||y - y'||/(2\sigma^2))$.

The second task is to perform differential comparison between two sets. There are two variations of differential comparison, single- and bi-directional, which defines the direction in moving samples between two sets.

1) Single-directional Differential Comparison: This method iteratively moves samples from $S^{prob,k}_{target}$ to $S^{prob,k}_{nonmem}$, compares the distance before and after move, and then determines the moved sample's membership. Details of the method are shown in Algorithm 1. Lines 1–2 of Algorithm 1 prepare some initial variables and then Lines 3–14 are the iterative algorithm. Specifically, Line 5 first calculates the distance between two sets and Lines 6–12 go through all the elements in $S^{prob,k}_{target}$. If the updated distance after moving a sample (Line 7) is larger than the original (Line 8), BLINDMI-DIFF considers it as a non-member. After one iteration, BLINDMI-DIFF updates $S^{prob,k}_{target}$ (Line 13) and starts the entire process again.

Algorithm 1 Single-directional Differential Comparison

Input: $S^{prob,k}_{nonmem}$, $S^{prob,k}_{target}$
Output: $S_{pred,nonmem}$, $S_{pred,mem}$
1: $S_{pred,nonmem} \leftarrow empty$
2: $flag \leftarrow true$
3: **while** flag **do**
4: $\quad flag \leftarrow false$
5: $\quad d \leftarrow D(S^{prob,k}_{nonmem}, S^{prob,k}_{target})$
6: $\quad$ **for** $y \in S^{prob,k}_{target}$ **do**
7: $\quad\quad d' \leftarrow D(S^{prob,k}_{nonmem} \cup \{y\}, S^{prob,k}_{target} - \{y\})$
8: $\quad\quad$ **if** $d' \geq d$ **then**
9: $\quad\quad\quad S_{pred,nonmem} \leftarrow S_{pred,nonmem} \cup y$
10: $\quad\quad\quad flag \leftarrow true$
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: $\quad S^{prob,k}_{target} \leftarrow S^{prob,k}_{target} - S_{pred,nonmem}$
14: **end while**
15: $S_{pred,mem} \leftarrow S^{prob,k}_{target}$

Algorithm 2 Bi-directional Differential Comparison

Input: $S^{prob,k}_{target1}$, $S^{prob,k}_{target2}$
Output: $S_{pred,nonmem}$, $S_{pred,mem}$
1: $flag \leftarrow true$
2: **while** flag **do**
3: $\quad flag \leftarrow false$
4: $\quad d \leftarrow D(S^{prob,k}_{target1}, S^{prob,k}_{target2})$
5: $\quad$ **for** $y \in S^{prob,k}_{target1}$ **do**
6: $\quad\quad d' \leftarrow D(S^{prob,k}_{target2} \cup \{y\}, S^{prob,k}_{target1} - \{y\})$
7: $\quad\quad$ **if** $d' \geq d$ **then**
8: $\quad\quad\quad S^{prob,k}_{target1} \leftarrow S^{prob,k}_{target1} - \{y\}$
9: $\quad\quad\quad S^{prob,k}_{target2} \leftarrow S^{prob,k}_{target2} \cup \{y\}$
10: $\quad\quad\quad flag \leftarrow true$
11: $\quad\quad\quad d \leftarrow d'$
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad$ **for** $y \in S^{prob,k}_{target2}$ **do**
15: $\quad\quad d' \leftarrow D(S^{prob,k}_{target1} \cup \{y\}, S^{prob,k}_{target2} - \{y\})$
16: $\quad\quad$ **if** $d' \geq d$ **then**
17: $\quad\quad\quad S^{prob,k}_{target2} \leftarrow S^{prob,k}_{target2} - \{y\}$
18: $\quad\quad\quad S^{prob,k}_{target1} \leftarrow S^{prob,k}_{target1} \cup \{y\}$
19: $\quad\quad\quad flag \leftarrow true$
20: $\quad\quad\quad d \leftarrow d'$
21: $\quad\quad$ **end if**
22: $\quad$ **end for**
23: **end while**
24: $S_{pred,mem}, S_{pred,nonmem} \leftarrow S^{prob,k}_{target1}, S^{prob,k}_{target2}$

2) Bi-directional Differential Comparison: This method works on a roughly divided two datasets, say $S^{prob,k}_{target1}$ and $S^{prob,k}_{target2}$, and moves samples in both directions, i.e., $S^{prob,k}_{target1} \rightarrow S^{prob,k}_{target2}$ and $S^{prob,k}_{target2} \rightarrow S^{prob,k}_{target1}$. More specifically, the method details are shown in Algorithm 2. BLINDMI-DIFF first moves samples from $S^{prob,k}_{target1}$ to $S^{prob,k}_{target2}$ in Lines 5–13, and then $S^{prob,k}_{target2}$ to $S^{prob,k}_{target1}$ in Lines 14–22. Then, BLINDMI-DIFF iterates the entire procedure until it converges.

Note that one major challenge here is to decide whether $S^{prob,k}_{target1}$ or $S^{prob,k}_{target2}$ contains non-members, as those two sets are symmetric and look the same. The intuition here is that the average prediction confidence score of members is higher than the one of non-members. Therefore, BLINDMI-DIFF compares the average confidence score for a decision in the end.

TABLE V. A DESCRIPTION OF DIFFERENT DATASETS USED IN THE EVALUATION.

Dataset	# of classes	Description	Resolution	# Epochs (target model)	Training set (target model)	Training set (shadow model)	Target set
Adult	2	census income records	N/A	100	16,280	16,280	32,560
EyePACS	5	retina images with diabetic retinopathy	150×150	(pre-trained + 15) or 150	10,000	10,000	20,000
CH-MNIST	8	histological images of colorectal cancer	64×64	(pre-trained + 15) or 150	2,500	2,500	5,000
Location	30	mobile users' location check-in records	N/A	100	2,505	2,505	5,010
Purchase-50	50	shoppers' purchase histories	N/A	100	10,000	10,000	20,000
Texas	100	inpatients stays in health facilities	N/A	100	10,000	10,000	20,000
CIFAR-100	100	object recognition dataset	32×32	(pre-trained + 30) or 150	10,000	10,000	20,000
Birds-200	200	photos of birds species	150×150	(pre-trained + 15) or 150	5,894	5,894	11,788

E. Batch Division and Size Optimization

In this part, we discuss how BLINDMI divides the target dataset into small batches with an appropriate size especially when the size of nonmember dataset is small. The high-level idea of determining the size is that BLINDMI needs to maximize the distance change in differential comparison when moving one sample. Specifically, BLINDMI starts from a batch size consistent with the size of nonmember dataset. Such an algorithm keeps BLINDMI sensitive while still maintaining a small size of non-members.

IV. DATASETS, PRIOR ATTACKS AND IMPLEMENTATION

In this section, we describe the datasets used in the experiments, target and shadow models, existing state-of-the-art attacks, and our implementation of BLINDMI.

A. Datasets

We use eight datasets as shown in Table V to evaluate BLINDMI on different application scenarios.

1) UCI Adult: UCI Adult, or Adult for short, has 48,842 records with census attributes, such as age, gender, education, marital status, and working hours. The classification task is to predict whether a person earns over $50,000 per year based on given attributes. We follow a well-known preprocessing method[2] to obtain a target datasets with 32,560 records—half are used as the training set of the target model and half as the training set of the shadow model. The target dataset contains all the samples.

2) EyePACS: The EyePACS dataset from Kaggle's was used for a Diabetic Retinopathy Detection challenge[3]. The dataset includes 88,703 high-resolution retina images taken under a variety of imaging conditions and each image has a label ranging from 0 to 4, representing the presence and severity of diabetic retinopathy. We adopt the preprocessing method from Kaggle[4]. We select 10,000 random images as the training set of the target model, 10,000 disjoint images as the training set of the shadow model, and 20,000 images—i.e., 10,000 members and 10,000 non-members—as the target set for inference.

3) CH-MNIST: CH-MNIST [27] is a benchmark dataset of 5,000 histological images of human colorectal cancer including 8 classes of tissues. We obtain a version[5] of CH-MNIST from TensorFlow Datasets, in which each image's resolution is 150×150. We resize all images to 64×64 to increase the diversity of image resolution, and then randomly select two sets of 2,500 images as training data of the target and the shadow models. The target dataset has all 5,000 images, i.e., 2,500 members and 2,500 non-members for the target model. Note that due to the small size of CH-MNIST, the training sets of shadow and target models have overlap.

4) Location: This dataset is from the publicly available set of mobile users' location "check-ins" in the Foursquare social network[6]. We obtain a processed version of the dataset from a prior work [43], which has 5,010 record with 446 binary features and is clustered into 30 classes, each representing a different geosocial type. The task is to predict the user's geosocial type given his or her record. We use the whole dataset and randomly chose samples to create two sets, each with 2,505 samples, to train the target model and the shadow model respectively. There are overlapping samples in both target and shadow models' training sets since the dataset is small.

5) Purchase-50: Purchase-50 dataset is from Kaggle's "Acquired Valued Shoppers Challenge"[7] and contains purchase histories of many shoppers. We obtain a simplified version with 197,324 records from R.Shokri et al. [43], where each record contains 600 binary features representing whether the customer has purchased an item. We cluster the dataset into 50 classes, in which each class represented a different purchase habit. The training datasets of target and shadow models are disjoint with 10,000 samples each; The target dataset has 20,000 samples, i.e., 10,000 members and 10,000 nonmembers.

6) Texas hospital stays: Texas hospital stays, or Texas for short, is the inpatient stays records in several health facilities based on the Hospital Discharge Data released by Texas Department of State Health Services from 2006 to 2009. We follow the same preprocessing method and classification task as prior work [43]. The training datasets of target and shadow models are disjoint with 10,000 samples each; We also select 20,000 records for the target dataset, i.e., 10,000 members and 10,000 nonmembers.

7) CIFAR-100: CIFAR-100 is a popular benchmark dataset that is used to evaluate image recognition algorithms. The dataset has 60,000 images evenly distributed over 100 classes. We randomly select two sets of 10,000 images evenly distributed over 100 classes as the training datasets of the target model, and another disjoint 10,000 images as the training datasets of the shadow model. The target dataset has 20,000 images: 10,000 members and 10,000 nonmembers.

[2] https://github.com/rupampatir/TrainingDataSynthesizer/blob/master/classifiers/income/income_classifier.py
[3] https://www.kaggle.com/c/diabetic-retinopathy-detection/data
[4] https://www.kaggle.com/ratthachat/aptos-eye-preprocessing-in-diabetic-retinopathy
[5] https://www.tensorflow.org/datasets/catalog/colorectal_histology
[6] https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[7] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data

TABLE VI. TARGET AND SHADOW MODELS' ARCHITECTURE AND HYPER-PARAMETER SETTING

Model arch.	# of layers	Target model		Shadow model (blackbox)	
		Max. epochs	LRN rate	Max. epochs	LRN rate
ResNet50	50	p*+m**	$5e^{-5}$	p+0.2m	$5e^{-5}$
ResNet101	101	p+m	$5e^{-5}$	p+0.3m	$1e^{-4}$
VGG16	16	p+m	$5e^{-5}$	p+0.6m	$5e^{-5}$
DenseNet121	121	p+m	$5e^{-5}$	p+m	$1e^{-4}$
VGG19	19	p+m	$5e^{-5}$	p+1.5m	$5e^{-5}$
Standard CNN	2	m	$5e^{-5}$	0.5m	$1e^{-4}$
MLP	[3–7] dense	m	$5e^{-5}$	[0.3–2]m	$1e^{-4}$/$5e^{-5}$

* p: the epoch of a pre-trained weight on the ImageNet dataset;
** m: the maximum epoch of target model for each dataset in Table V.

TABLE VII. A LIST OF CONDITIONS OF BASELINE MI ATTACKS AND DIFFERENT VARIATIONS OF BLINDMI

Attacks	True labels	Shadow	Threat model	Target Model Probes
NN [43]	✗	✓	all	Target set
Top3-NN [41]	✗	✓	all	Target set
Top1-Thre [41]	✗	✗	all	Target set + 1,000 samples
Loss-Thre [49]	✓	✓	blk, gray	Target set
Label-Only [49]	✓	✗	blk, gray	Target set
Top2+True	✓	✓	blk, gray	Target set
BLINDMI-DIFF-w/	✗✓	✗	all	Target set + 20 samples
BLINDMI-DIFF-w/o	✗✓	✗	all	Target set
BLINDMI-1CLASS	✗✓	✗	all	Target set + 1,000 samples

* ✗✓: The approach works either with or without the condition, e.g., the ground truth labels.

8) Caltech-UCSD Birds 200: Caltech-UCSD Birds 200 [48], or for short Birds-200, is an image dataset with photos of mostly North American birds species. The dataset has 11,788 images from 200 classes. In our experiments the training dataset of target and shadow models each has 5,894 samples; The target dataset has 5,894 members and 5,894 nonmembers

B. Target and Shadow Models

In this part, we describe the architectures and hyper-parameters of target and shadow models of our evaluation in Table VI. We adopt seven different popular DNN architectures with pre-set maximum epochs and learning rate. Note that all popular DNNs, e.g., ResNet, VGG, and DenseNet, are the standard architectures with pre-trained parameters from ImageNet; we adopt the same standard CNN architecture and hyperparameters as prior blackbox MI attack [43]; the multilayer perceptron (MLP) model has at most seven dense layers with size of 8192, 4096, 2048, 1024, 512, 256, and 128 and an additional Softmax layer. Now we describe how we select and train target and shadow models.

- Target model. Given a dataset, we randomly select a model architecture from the target model column of Table VI and train the model with the specified hyperparameters.
- Shadow model (blackbox and blind settings). Given a target model and a dataset, we randomly select and train a model with the architecture and hyperparameters specified in the shadow model column of Table VI.
- Shadow model (graybox and graybox-blind settings). Given a target model and a dataset, we select the same architecture and hyperparameters as the target model.

C. State-of-the-art Attacks

In this part, we describe state-of-the-art attacks in the literature as shown in Table VII. We follow the descriptions in prior work to implement each attack for the comparison with BLINDMI. Generally speaking, there are two categories, those without ground-truth labels and those with ground-truth labels.

1) Attacks without Ground-truth Labels: We describe three prior attacks without ground-truth labels. Presumably, those attacks work under all settings, but their performance are the same, with or without ground-truth label information, i.e., under blind and blackbox settings.

- Neural network (NN). The NN-based MI attack trains a NN from all features from the output probability distributions of a shadow model. We follow both Shokri et al. [43] and Salem et al. [41] for the implementation.
- Neural network with top three features (Top3-NN). This MI attack proposed by Salem et al. [41] , which trains an NN based on the top three features from the output probability distributions of a shadow model.
- Threshold based on top one feature (Top1-Thre). This MI attack, which is also proposed by Salem et al. [41] as their Adversary Three, compares the top feature from the output probability distribution with a threshold and classifies the sample as member if the top feature is larger than the threshold.

2) Attacks with Ground-truth Labels: We describe three attacks that specifically require ground-truth labels: They may or may not need a shadow model. That is, these attacks only work under settings where ground-truth labels are available, i.e., blackbox and graybox settings.

- Threshold based on a loss function (Loss-Thre). This MI attack from Yeom et al. [49], which requires a shadow model, computes a cross-entropy loss, $loss = -log(F_T(x)_y)$, where $F_T(x)_y$ is the probability of the true label y of the data sample x, and classifies x as a member if $loss$ is smaller than the average loss of all training samples in the shadow model.
- Discrepancy between predicted and ground-truth class (Label-Only). This MI attack from Yeom et al. [49], which does not requires a shadow model, classifies a sample as a member if the predicted class is the same as the ground-truth one.
- Neural network with top two feature plus the feature with ground-truth label (Top2+True). This MI attack is an improved version of the NN attack from Shokri et al. [43] and Salem et al. [41] with the consideration of the ground-truth label. We add this attack as a baseline for the comparison purpose.

D. Implemenatation

We implemented BLINDMI with 811 lines of code (LoC) based on TensorFlow 2.1.0. Specifically, our implementations of BLINDMI-1CLASS, BLINDMI-DIFF-w/, and BLINDMI-DIFF-w/o are of 227, 261 and 323 Lines of Python 3.7 code respectively. The non-member generation module has 72 LoC, the differential comparison module 182 LoC. We also implement prior attacks with 344 LoC. Our implementations of BLINDMI and prior attacks are

open-source and available at this anonymous repository: https://github.com/hyhmia/BlindMI.

V. Evaluation

We first introduce the evaluation metrics and several research questions (RQs). Then, we show the performances of MI attacks under different settings based on different RQs, and explain what we learn from the results in details.

A. Evaluation Metrics, Experimental Setting and Research Questions

We mainly use F1-score, the harmonic mean of precision and recall, as our evaluation metrics, because F1-score represents a trade-off between precision and recall. Specifically, Precision represents the ratio of real-true members predicted among all the positive membership predictions made by an adversary, and Recall demonstrates the ratio of true members predicted by an adversary among all the real-true members. We adopt the batch mode for BlindMI in our experiments. Following the prior work [41], in the Blind and Graybox-Blind settings, we select the top three feature values for all variations of BlindMI; in the Blackbox and Graybox settings, we select the top two feature values plus the value of the ground-truth class. All the experiments are performed using the GeForce RTX 2080 graphics cards (NVIDIA).

Our evaluation aims to answer the following RQs.

- *RQ1 [All Settings]:* What is the performance of all variations of BlindMI compared with state-of-the-art MI attacks under different settings?
- *RQ2 [Blackbox Setting]:* How does BlindMI perform under existing defenses against MI attacks?
- *RQ3 [Blind Setting]:* What is the performance of BlindMI for different quality and size of the non-member set?
- *RQ4 [Blind Setting]:* How do different initial classifiers and kernel functions affect the performance of BlindMI-diff?
- *RQ5 [Blind Setting]:* How long and how many moves and iterations are needed for BlindMI-diff to converge?
- *RQ6 [Blackbox Setting]:* What is the performance of BlindMI under different real-world settings, e.g., nonmember-to-member ratio and number of target model's classes?

B. RQ1: Attack Performance With Different Settings

In this subsection, we evaluate and compare the Precision, Recall, and F1-score of BlindMI and existing attacks in Section IV-C. Our setting for this RQ is that the nonmember dataset size of BlindMI-diff-w/ is 20, the nonmember dataset size of BlindMI-1class is 1,000, and BlindMI-diff-w/o does not need additional nonmembers. The target dataset sizes depending on the problem domain are shown in Table V. Each attack is performed ten times with a new target and shadow model with different training datasets, model architectures and hyperparameters each time. Then, we obtain the average values of F1-score together with the standard error of the mean among the ten attacks.

Table VIII shows the Precision, Recall and F1-score of different attacks under four adversarial settings. The best performances of all attacks under different settings are highlighted with different colors (blue for recall, green for precision, and red for F1-score.) Note that if the performance of a prior attack, e.g., NN-based, is the same with and without ground truth label, we only show the attack once under the blind and graybox-blind settings. We do show BlindMI multiple times under different settings for ease of comparison. Next, we introduce several observations from our experiments.

[Observation RQ1-1] BlindMI *significantly outperforms state-of-the-art MI attacks under all settings in terms of F1-score.*

The first observation is that BlindMI outperforms state-of-the-art MI attacks under all settings: The reason is that BlindMI extracts membership semantics directly from the target model via probing. Sometimes, the performance boost is over 20%, e.g., for the Adult and BIRDS-200 datasets under the blind setting. As a comparison, no single prior attack dominates the performance in F1-score. Consider the blind setting for example. Top1-Thre is the best for the EyePACS dataset except for BlindMI; NN is the best for the CH-MNIST dataset except for BlindMI; and Top3-NN outperforms all methods for the Purchase-50. The reason is that no prior attacks extract enough membership semantics as BlindMI does.

[Observation RQ1-2] *The introduction of ground-truth labels improves attack performance, but to a limited degree for* BlindMI.

The second observation is about how the introduction of ground-truth labels affects attack performance. The performance boost is sometimes significant for prior attacks. Take Purchase-50 for example. The best average F1-score under the blind setting is 59.6%, but the average F1-score increases to 72.1%, a 12.5% increase, under the blackbox setting.

As a comparison, the best performance boost of BlindMI with the ground truth label is 2.9% for the EyePACS dataset. That said, although ground-truth labels introduce additional membership semantics, the semantics introduction is limited in terms of F1-score improvement.

[Observation RQ1-3] *Shadow model quality plays an important role in some existing attacks.*

The third observation is about how different shadow models affect the attack performance. First, BlindMI does not need a shadow model and therefore BlindMI's performance is the same with or without shadow model. Second, the performance of some existing attacks varies a lot given different shadow models. Take the NN attack for BIRDS-200 under the blind setting for example. The average F1-score is 58.3, but the standard error is 15.0 with a confidence of 68.3%. That said, the choice of shadow models is crucial in the performance of existing attacks with shadow models.

[Observation RQ1-4] BlindMI-diff-*w/ performs the best among all three variations in terms of F1-score, while* BlindMI-diff-*w/o does not need additional probes to the target model.*

Table VIII shows that BlindMI-diff-w/ is the best comparing with BlindMI-diff-w/o and BlindMI-1class. At the same time, BlindMI-diff-w/ does require 20 additional probes to the target model to generate a non-member set. If the adversary's access to the target model is restricted to the target dataset, BlindMI-diff-w/o is an alternative option as opposed to BlindMI-diff-w/.

TABLE VIII. PRECISION, RECALL, F1-SCORE (%) WITH STANDARD ERROR OF THE MEAN OF PRIOR ATTACKS AND BLINDMI UNDER FOUR DIFFERENT ADVERSARIAL SETTINGS (BLUE INDICATES THE HIGHEST RECALL, GREEN THE HIGHEST PRECISION, AND RED THE HIGHEST F1-SCORE FOR EACH SETTING).

	Attack	Metric	Adult	EyePACS	CH-MNIST	Location	Purchase-50	Texas	CIFAR-100	Birds-200
Blind	NN	Precision	49.9 ± 0.30	56.6 ± 3.80	62.5 ± 0.62	68.9 ± 5.09	61.0 ± 1.64	62.5 ± 3.02	73.2 ± 12.1	95.1 ± 1.90
		Recall	35.1 ± 10.1	90.5 ± 9.58	85.8 ± 8.22	93.6 ± 6.39	67.1 ± 19.9	99.8 ± 0.15	99.5 ± 0.28	53.5 ± 27.5
		F1-Score	40.6 ± 7.32	69.1 ± 0.02	71.7 ± 3.53	78.4 ± 3.23	59.4 ± 11.9	76.7 ± 2.20	83.1 ± 3.53	58.3 ± 27.4
	Top3-NN	Precision	49.8 ± 0.37	55.7 ± 3.69	62.1 ± 0.19	69.1 ± 4.98	61.0 ± 1.73	62.5 ± 2.80	70.3 ± 9.33	98.1 ± 0.08
		Recall	20.2 ± 7.34	93.3 ± 6.58	84.8 ± 9.70	92.8 ± 7.17	67.6 ± 19.9	**99.9 ± 0.08**	**99.6 ± 0.11**	60.3 ± 23.1
		F1-Score	26.7 ± 7.25	69.5 ± 1.04	70.9 ± 4.03	78.1 ± 3.39	59.6 ± 12.1	76.8 ± 2.07	81.7 ± 6.66	68.6 ± 21.3
	Top1-Threshold	Precision	0.51 ± 0.23	**99.9 ± 0.03**	**74.7 ± 24.9**	14.1 ± 2.93	61.6 ± 12.1	0.34 ± 0.20	**97.2 ± 1.70**	**99.9 ± 0.06**
		Recall	31.3 ± 13.6	55.2 ± 0.51	40.9 ± 13.7	63.7 ± 1.60	63.2 ± 1.97	24.6 ± 14.2	86.2 ± 3.55	57.9 ± 0.81
		F1-Score	1.01 ± 0.44	71.1 ± 0.42	52.8 ± 17.6	22.7 ± 3.87	53.5 ± 7.26	0.67 ± 0.38	92.8 ± 1.72	71.4 ± 0.65
	BlindMI-Diff-w/	Precision	50.0 ± 0.03	66.2 ± 2.80	65.4 ± 2.52	75.9 ± 1.44	**66.0 ± 0.55**	**76.1 ± 1.06**	90.7 ± 0.86	95.5 ± 0.08
		Recall	**90.4 ± 5.95**	94.4 ± 3.35	88.8 ± 2.04	99.9 ± 0.08	95.2 ± 1.38	97.4 ± 1.64	97.4 ± 0.37	98.2 ± 0.26
		F1-Score	**64.2 ± 1.59**	**77.7 ± 0.80**	**75.1 ± 1.49**	**86.2 ± 0.90**	**78.0 ± 0.31**	**85.5 ± 0.80**	**93.9 ± 0.63**	**96.8 ± 0.09**
	BlindMI-Diff-w/o	Precision	**50.5 ± 0.05**	60.3 ± 1.26	63.3 ± 1.60	**71.5 ± 0.87**	61.7 ± 0.38	69.2 ± 4.09	86.1 ± 0.78	94.5 ± 0.24
		Recall	84.2 ± 4.05	**99.3 ± 0.10**	92.4 ± 2.68	**99.9 ± 0.04**	**99.5 ± 0.25**	97.6 ± 1.83	98.9 ± 0.60	98.0 ± 0.80
		F1-Score	62.7 ± 1.12	75.0 ± 1.40	75.1 ± 1.89	83.3 ± 0.57	76.2 ± 0.25	80.7 ± 2.37	92.1 ± 1.15	96.2 ± 0.26
	BlindMI-1Class	Precision	49.9 ± 0.07	64.7 ± 2.41	59.0 ± 1.13	63.4 ± 0.44	61.1 ± 1.38	70.9 ± 1.11	83.5 ± 0.40	83.8 ± 11.9
		Recall	56.5 ± 5.85	94.4 ± 0.01	**94.8 ± 1.41**	99.7 ± 0.27	98.1 ± 0.71	97.2 ± 2.07	98.9 ± 0.69	**99.1 ± 0.60**
		F1-Score	52.6 ± 2.47	76.8 ± 1.70	72.8 ± 1.27	77.6 ± 0.33	77.1 ± 0.50	81.9 ± 0.07	90.6 ± 0.52	90.3 ± 6.77
Blackbox	Top2+True	Precision	49.8 ± 0.10	58.9 ± 1.45	62.4 ± 1.92	72.0 ± 3.56	62.3 ± 1.69	71.7 ± 1.95	69.4 ± 11.3	94.9 ± 3.55
		Recall	59.6 ± 14.5	97.4 ± 2.53	95.3 ± 1.63	99.3 ± 0.74	71.1 ± 18.3	**99.9 ± 0.05**	**99.9 ± 0.08**	63.8 ± 35.9
		F1-Score	52.1 ± 6.27	73.4 ± 0.41	75.4 ± 1.84	83.3 ± 2.24	62.9 ± 10.7	83.4 ± 1.29	80.9 ± 7.85	69.5 ± 25.6
	Loss-Threshold	Precision	**64.4 ± 2.07**	58.5 ± 0.72	**64.0 ± 1.45**	54.1 ± 26.6	61.6 ± 3.20	**79.4 ± 12.4**	76.0 ± 8.08	**98.4 ± 0.15**
		Recall	49.9 ± 0.04	99.9 ± 0.02	82.6 ± 8.06	66.3 ± 2.03	52.8 ± 23.3	63.1 ± 8.94	99.2 ± 0.64	57.6 ± 17.1
		F1-Score	56.2 ± 0.77	73.8 ± 0.57	71.8 ± 4.01	47.7 ± 19.7	48.1 ± 18.6	69.6 ± 9.60	85.6 ± 5.09	71.2 ± 13.7
	Label-Only	Precision	46.5 ± 3.04	57.2 ± 0.10	55.1 ± 1.90	60.3 ± 0.16	56.4 ± 0.08	66.3 ± 0.70	75.3 ± 0.47	76.1 ± 1.26
		Recall	71.7 ± 9.54	**99.9 ± 0.01**	**99.8 ± 0.10**	99.3 ± 0.47	**99.9 ± 0.04**	99.9 ± 0.08	99.1 ± 0.47	**99.9 ± 0.01**
		F1-Score	56.2 ± 5.28	72.8 ± 0.09	70.9 ± 1.54	75.3 ± 0.12	72.1 ± 0.07	79.7 ± 0.50	85.5 ± 0.47	86.4 ± 0.81
	BlindMI-Diff-w/	Precision	50.0 ± 0.01	**70.8 ± 3.23**	64.0 ± 2.50	**78.1 ± 1.03**	**66.7 ± 0.78**	77.0 ± 0.53	**91.0 ± 0.17**	95.3 ± 0.49
		Recall	**97.0 ± 1.18**	93.7 ± 0.47	97.9 ± 0.88	98.9 ± 0.93	**99.8 ± 0.07**	99.2 ± 0.54	98.9 ± 0.49	99.2 ± 0.59
		F1-Score	**66.0 ± 0.28**	**80.6 ± 1.90**	**77.2 ± 1.83**	**87.3 ± 0.70**	**79.9 ± 0.57**	**86.7 ± 0.37**	**94.8 ± 0.14**	**97.2 ± 0.03**
	BlindMI-Diff-w/o	Precision	50.0 ± 0.07	62.9 ± 0.10	62.3 ± 0.53	74.4 ± 0.61	63.6 ± 0.27	74.1 ± 1.62	88.2 ± 0.93	93.9 ± 0.50
		Recall	90.1 ± 1.61	99.6 ± 0.17	92.1 ± 1.29	**99.9 ± 0.01**	99.5 ± 0.02	97.4 ± 1.83	98.8 ± 0.68	99.0 ± 0.76
		F1-Score	64.2 ± 0.27	77.1 ± 0.13	74.3 ± 0.80	85.3 ± 0.39	77.6 ± 0.19	84.1 ± 0.42	93.2 ± 0.82	96.4 ± 0.09
	BlindMI-1Class	Precision	50.0 ± 0.08	67.3 ± 2.38	62.5 ± 2.80	67.3 ± 1.66	64.4 ± 0.39	71.5 ± 1.57	90.8 ± 0.68	95.8 ± 0.53
		Recall	69.9 ± 0.04	96.6 ± 0.62	96.2 ± 1.02	99.9 ± 0.03	99.0 ± 0.22	99.3 ± 0.06	97.6 ± 0.39	97.1 ± 1.88
		F1-Score	58.3 ± 0.07	79.3 ± 1.44	75.6 ± 1.90	80.4 ± 1.19	78.0 ± 0.32	83.4 ± 1.03	94.0 ± 0.47	96.4 ± 0.33
Gray-Blind	NN	Precision	50.2 ± 0.14	62.1 ± 0.82	62.2 ± 0.87	74.8 ± 1.14	64.2 ± 0.57	72.4 ± 1.75	90.4 ± 1.02	95.7 ± 0.56
		Recall	64.1 ± 16.7	96.1 ± 2.23	90.2 ± 4.21	99.8 ± 0.04	96.4 ± 1.65	98.5 ± 1.22	96.2 ± 0.27	97.8 ± 0.05
		F1-Score	54.3 ± 5.50	72.3 ± 0.08	73.5 ± 1.99	85.6 ± 0.71	77.0 ± 0.36	83.4 ± 0.83	93.2 ± 0.46	96.8 ± 0.28
	Top3-NN	Precision	50.2 ± 0.21	60.7 ± 1.10	62.0 ± 0.77	75.0 ± 1.06	64.1 ± 0.47	72.0 ± 1.65	90.4 ± 0.86	89.9 ± 0.08
		Recall	69.9 ± 25.7	97.5 ± 1.56	90.7 ± 3.87	99.9 ± 0.05	97.2 ± 1.14	**99.1 ± 0.69**	96.3 ± 0.15	96.8 ± 0.14
		F1-Score	56.4 ± 9.27	74.8 ± 0.37	73.6 ± 1.80	85.7 ± 0.69	77.2 ± 0.34	83.4 ± 0.90	93.2 ± 0.80	93.2 ± 0.03
	Top1-Threshold	Precision	0.51 ± 0.23	**99.9 ± 0.03**	**74.7 ± 24.9**	14.1 ± 2.93	61.6 ± 12.1	0.34 ± 0.20	**97.2 ± 1.70**	**99.9 ± 0.06**
		Recall	31.3 ± 13.6	55.2 ± 0.51	40.9 ± 13.7	63.7 ± 1.60	63.2 ± 1.97	24.6 ± 14.2	86.2 ± 3.55	57.9 ± 0.81
		F1-Score	1.01 ± 0.44	71.1 ± 0.42	52.8 ± 17.6	22.7 ± 3.87	53.5 ± 7.26	0.67 ± 0.38	92.8 ± 1.72	71.4 ± 0.65
	BlindMI-Diff-w/	Precision	50.0 ± 0.03	66.2 ± 2.80	65.4 ± 2.52	**75.9 ± 1.44**	**66.0 ± 0.55**	**76.1 ± 1.06**	90.7 ± 0.86	95.5 ± 0.08
		Recall	**90.4 ± 5.95**	94.4 ± 3.35	88.8 ± 2.04	99.9 ± 0.08	95.2 ± 1.38	97.4 ± 1.64	97.4 ± 0.37	98.2 ± 0.26
		F1-Score	**64.2 ± 1.59**	**77.7 ± 0.80**	**75.1 ± 1.49**	**86.2 ± 0.90**	**78.0 ± 0.31**	**85.5 ± 0.80**	**93.9 ± 0.63**	**96.8 ± 0.09**
	BlindMI-Diff-w/o	Precision	**50.5 ± 0.05**	60.3 ± 1.26	63.3 ± 1.60	71.5 ± 0.87	61.7 ± 0.38	69.2 ± 4.09	86.1 ± 0.78	94.5 ± 0.24
		Recall	84.2 ± 4.05	**99.3 ± 0.10**	92.4 ± 2.68	**99.9 ± 0.04**	**99.5 ± 0.25**	97.6 ± 1.83	**98.9 ± 0.60**	98.0 ± 0.80
		F1-Score	62.7 ± 1.12	75.0 ± 1.40	75.1 ± 1.89	83.3 ± 0.57	76.2 ± 0.25	80.7 ± 2.37	92.1 ± 1.15	96.2 ± 0.26
	BlindMI-1Class	Precision	49.9 ± 0.07	64.7 ± 2.41	59.0 ± 1.13	63.4 ± 0.44	61.1 ± 1.38	70.9 ± 1.11	83.5 ± 0.40	83.8 ± 11.9
		Recall	56.5 ± 5.85	94.4 ± 0.01	**94.8 ± 1.41**	99.7 ± 0.27	98.1 ± 0.71	97.2 ± 2.07	98.9 ± 0.69	**99.1 ± 0.60**
		F1-Score	52.6 ± 2.47	76.8 ± 1.70	72.8 ± 1.27	77.6 ± 0.33	77.1 ± 0.50	81.9 ± 0.07	90.6 ± 0.52	90.3 ± 6.77
Graybox	Top2+True	Precision	50.0 ± 0.04	63.7 ± 1.53	62.8 ± 1.65	75.4 ± 0.85	65.1 ± 0.38	75.1 ± 0.28	90.6 ± 0.68	95.7 ± 0.48
		Recall	96.8 ± 2.22	98.4 ± 1.38	93.4 ± 2.95	99.9 ± 0.04	98.3 ± 0.66	**99.9 ± 0.01**	97.3 ± 0.42	98.1 ±0.13
		F1-Score	66.0 ± 0.50	77.3 ± 0.69	75.1 ± 2.03	86.0 ± 0.55	78.4 ± 0.25	85.7 ± 0.18	93.8 ±0.53	96.9 ± 0.18
	Loss-Threshold	Precision	**66.4 ± 2.38**	**96.0 ± 3.93**	**84.4 ± 4.83**	77.2 ± 10.5	**79.1 ± 5.41**	75.4 ± 9.79	84.9 ± 8.05	75.4 ± 1.26
		Recall	50.0 ± 0.07	64.1 ± 2.70	64.5 ± 2.07	75.6 ± 1.79	66.1 ± 0.85	79.5 ± 0.64	90.8 ± 3.06	98.2 ± 0.23
		F1-Score	57.0 ± 0.84	76.8 ± 0.68	73.0 ± 2.90	75.9 ± 4.96	71.8 ± 2.70	76.5 ± 4.81	87.1 ± 3.39	85.3 ± 0.89
	Label-Only	Precision	46.5 ± 0.30	57.2 ± 0.10	55.1 ± 1.90	60.4 ± 0.16	56.4 ± 0.08	66.3 ± 0.90	75.3 ± 0.47	76.1 ±1.26
		Recall	71.7 ± 9.54	**99.9 ± 0.01**	**99.8 ± 0.10**	99.3 ± 0.47	**99.9 ± 0.04**	99.9 ± 0.08	**99.1 ± 0.47**	57.9 ± 0.81
		F1-Score	56.2 ± 5.28	72.8 ± 0.09	70.9 ± 1.54	75.3 ± 0.12	72.1 ± 0.07	79.7 ± 0.50	85.5 ± 0.47	86.4 ± 0.81
	BlindMI-Diff-w/	Precision	50.0 ± 0.01	70.8 ± 3.23	64.0 ± 2.50	**78.1 ± 1.0**	66.7 ± 0.78	**77.0 ± 0.53**	**91.1 ± 0.17**	95.3 ± 0.49
		Recall	**97.0 ± 1.18**	93.7 ± 0.47	97.7 ± 0.88	98.9 ± 0.93	99.8 ± 0.07	99.2 ± 0.54	98.9 ± 0.49	**98.2 ± 0.26**
		F1-Score	**66.0 ± 0.30**	**80.6 ± 1.90**	**77.2 ± 1.83**	**87.3 ± 0.70**	**79.9 ± 0.57**	**86.7 ± 0.37**	**94.8 ± 0.14**	**97.2 ± 0.03**
	BlindMI-Diff-w/o	Precision	50.0 ± 0.07	62.9 ± 0.10	62.3 ± 0.53	74.4 ± 0.61	63.6 ± 0.27	74.1 ± 1.62	88.2 ± 0.93	93.9 ± 0.50
		Recall	90.1 ± 1.61	99.6 ± 0.17	92.1 ± 1.29	**99.9 ± 0.01**	99.5 ± 0.02	97.4 ± 1.83	98.8 ± 0.68	99.0 ± 0.76
		F1-Score	64.2 ± 0.27	77.1 ± 0.13	74.3 ± 0.80	85.3 ± 0.39	77.6 ± 0.19	84.1 ± 0.42	93.2 ± 0.82	96.4 ± 0.09
	BlindMI-1Class	Precision	50.0 ± 0.08	67.3 ± 2.38	62.5 ± 2.80	67.3 ± 1.66	64.4 ± 0.39	71.5 ± 1.57	90.8 ± 0.68	**95.8 ± 0.53**
		Recall	69.9 ± 0.04	96.6 ± 0.62	96.2 ± 1.02	99.9 ± 0.03	99.0 ± 0.22	99.9 ± 0.06	97.6 ± 0.39	97.4 ± 0.05
		F1-Score	58.3 ± 0.07	79.3 ± 1.44	75.6 ± 1.90	80.4 ± 1.19	78.0 ± 0.32	83.4 ± 1.03	94.0 ± 0.47	96.4 ± 0.33

[Observation RQ1-5] *The variation of* BLINDMI-1CLASS *is larger than the one of* BLINDMI-DIFF.

Take Birds-200 for example. The best performance of BLINDMI-1CLASS is higher than the one of BLINDMI-DIFF. The reason is that the performance of ML classifier depends on the training data: If many data samples lie along the decision boundary, the one-class model can learn the membership semantics and thus outperforms BLINDMI-DIFF.

C. RQ2: Defenses

In this subsection, we evaluate the performance of BLINDMI against state-of-the-art defenses of MI attacks. Note that we evaluate all the defenses under the blackbox setting because some attacks only work under the blackbox but not the blind setting. Now, we describe three general defense directions and representative works in each direction below.

- Output probability alteration based on adversarial example. Such a defense alters the output probabilities so that it becomes hard for an adversary to infer membership information. A representative approach in this category is called MemGuard [26], which changes the output probability distribution so that it looks like an adversarial example to the inference model built by the adversary. We adopt the original implementation of MemGuard.[8]
- Regularization-based fortification of ML model. Such a defense fortifies existing ML models, especially DNN, via regularization. Two representative approaches in this category are MMD+Mix-up [30] (which include two previous defenses, namely dropout [49] and L2-Regularizer [41]) and the adversarial regularization [36]. We implement our own version of MMD+Mix-up and adopt an open-source version of the adversarial regularization.[9] Note that the MMD+Mixup defense is adaptive with a regularizer based on BlindMI, i.e., minimizing the cluster distance between members and non-members during the training process. As a comparison, the adversarial regularization is based on the NN attack, because it requires that the MI attack be differentiable with gradients while BlindMI is not. (If we adopt a differentiable distance function in adversarial regularization, adversarial regularization boils down to the MMD+Mixup method.)
- Differential privacy-based protection. Such a defense adds noise to the output to fool an adversary. A representative approach in this category is DP-Adam [1] and we adopt an open-source version of DP-Adam.[10]

Note that in our experiment, we adopt a dataset that is at least included in the corresponding defense paper for our evaluation. That is, we choose CH-MNIST for MemGuard and DP-Adam, and CIFAR-100 for MMD+Mix-up and Adversarial Regularization. Because these defenses adopted different datasets in the paper and we follow what what adopted.

1) Attacks against MemGuard: We first evaluate the performance of MemGuard under existing MI attacks. The utility-loss budget is set up as [0, 0.1, 0.3, 0.5, 0.7, 1.0] for MemGuard, which represents the percentage of altered outputs. We show the evaluation results in Figure 3(a) and also make the following observations.

[8]https://github.com/jjy1994/MemGuard

[9]https://github.com/NNToan-apcs/python-DP-DL

[10]https://github.com/tensorflow/privacy

[Observation RQ2-1] *Attacks with ground-truth labels generally have a higher F1-score than those without when attacking MemGuard.*

Our first observation is that MemGuard is generally vulnerable to attacks that utilize ground-truth labels. For example, the worst performing attacks are Top1-Threshold and Top3-NN, which will likely remove the output probability of the ground-truth labels. By contrast, the best performed attacks are Label-only, Top2+True, and BLINDMI, which are all able to utilize the ground-truth label information. The reason is that although MemGuard alters the output probabilities, it does not change the prediction class because MemGuard does not want to influence the legacy performance of the model.

There are two more things worth noting. First, the performance of NN is actually better than the one of Top3-NN. The reason is similarly: NN adopts all the probability scores of the output, which contains the one corresponding to the ground-truth label for certain; by contrast, Top3-NN only adopts the top three probability scores, which may not contains the one corresponding to the ground-truth label.

Second, Top2+True performs better than Label-only when the privacy budget is small, but then degrades quickly when the privacy budget increases. The reason is that when the budget is small, top two probability scores will provide some membership information. When the budget increases, the top two of more samples are altered, which affects the performance of Top2+True.

[Observation RQ2-2] BLINDMI *still outperforms all existing attacks even if the output probabilities were adversarially altered.*

Our second observation is that BLINDMI still performs the best among all attacks. The underlying reasons are two-fold. First, although adversarial examples are close to the decision boundary, the decision boundary itself is a hyper-dimensional manifold and the projection of members and non-members on the manifold are still far from each other, thus being distinguishable. Second, although MemGuard alters output probability scores, sufficient information still exist, because MemGuard does not change the prediction results.

2) Attacks against DP-Adam: In this part, we evaluate all existing attacks against DP-Adam under the noise_multiplier as [0, 0.002, 0.004, 0.006, 0.008, 0.01]. The evaluation results are shown in Figure 3(b) and we make the following observations.

[Observation RQ2-3] *Attacks relying on binary comparison tend to have a low F1-score against DP-Adam.*

The reason behind this observation is that differential privacy (DP) perpetuates the probability outputs so that the boundary between members and non-members is blurred. Therefore, the performances of Top1-Threshold and Loss-Threshold are the worst. Consider Top1-Threshold for example: It is hard to differentiate members and non-members based on a single threshold of the highest output probability score due to the perpetuation enforced by DP.

[Observation RQ2-4] BLINDMI *has a higher performance than Label-only regardless of when the privacy-utility budget is small or large, while Label-only attack degrades slower with a mid-size budget.*

The reason is that Label-only attack depends on the performance gap of the target model on the training and testing datasets. Such a gap persists with a mid-size budget, but starts to shrink quickly for a large budget—and that is why Label-only's performance degrades finally with a large budget.

3) Attacks against MMD+Mixup: In this part, we evaluate all the MI attacks against MMD+Mixup with different privacy-utility budgets, i.e., the loss weight in the MMD as [0, 0.1, 0.5, 1, 2.5, 5]. Note that this budget controls the tradeoff between privacy and utility: A larger privacy-utility budget increases privacy protection, but at the same time decreases the model's utility. The evaluation results are in Figure 3(c)—BLINDMI clearly outperforms all existing attacks. We also make the following observation.

[Observation RQ2-5] *Attacks selecting more probability scores generally have a higher F1-score than those selecting less when attacking MMD+Mix-up.*

Specifically, Label-Only and Top1-Threshold are the worst when comparing with other attacks. The reason is that MMD+Mix-up also changes the target model's performance on the training dataset and therefore Label-Only and Top1-Threshold are heavily affected. As a comparison, other attacks also rely on the probability score of other classes, thus outperforming these two attacks.

4) Attacks against Adversarial Regularization: In this part, we evaluate all the MI attacks against the Adversarial Regularization [36] with different privacy-utility budget as [0, 0.3, 0.7, 1, 1.5, 2]. BLINDMI clearly outperforms all existing attacks as shown in Figure 3(d). Now we describe our observations.

[Observation RQ2-6] *Ground-truth Label plays an important role in defeating Adversarial Regularization and the results depend on how such labels are used in the attack.*

Specifically, Label-Only, Top2+True and BLINDMI, which all adopt ground-truth labels, are the best three MI attacks among all, while Loss-Threshold, which also adopts ground-truth labels, is the worst. The reason is that Loss-Threshold relies on the training data of a shadow model, which is drastically different from a model trained with adversarial regularization.

[Observation RQ2-7] *Simple MI attacks, except for* BLINDMI*, tend to have a better performance.*

Specifically, Label-Only and Top1-Threshold performs well compared with other attacks. The reason is that although the adversarial regularization fortifies the model via regularization, the output probability, especially the probability score of the predicted class, still contains abundant information.

D. RQ3: Nonmember Set Quality and Size

In this subsection, we evaluate how different generation methods and size of non-members affect the performance of BLINDMI, particularly BLINDMI-DIFF and BLINDMI-1CLASS. Without loss of generality, we use the EyePACS dataset and the blind setting: the target dataset consists of 20,000 samples and the size of non-member datasets changes from 20 to 10,000. Here are the settings used in the different non-member generation methods:

- Sample transformation. We adopt the Sobel operator.
- Random perpetuation. We adopt Gaussian noise with the mean value as zero and the variance as 0.001.
- Random generation. We adopt a uniform distribution in generating feature values.
- Cross-domain sample. We adopt samples from CH-MNIST.

We show our experiment results in Table IX and also make the following observations.

[Observation RQ3-1] *The performance of* BLINDMI-DIFF *stays mostly stable with a little increase as opposed to a big increase of* BLINDMI-1CLASS *as the size of nonmember datasets increases.*

Our first observation of RQ3 is on how the size of nonmembers affects F1-score. The F1-score of BLINDMI-DIFF is almost constant with around 1% boost as the size increases from 20 to 10,000. As a comparison, the F1-score of BLINDMI-1CLASS has between 5% and 12% increase except for random generated non-members.

The reason is that BLINDMI-1CLASS adopts a learning model, particularly one-class SVM, which needs some training data to learn the underlying semantics. As a comparison, BLINDMI-DIFF directly compares the distribution between the target and the non-member, which is effective in extracting membership semantics from just a few samples.

[Observation RQ3-2] *The quality of sample transformation is the best, while the random generation of non-members is the worst among all four methods.*

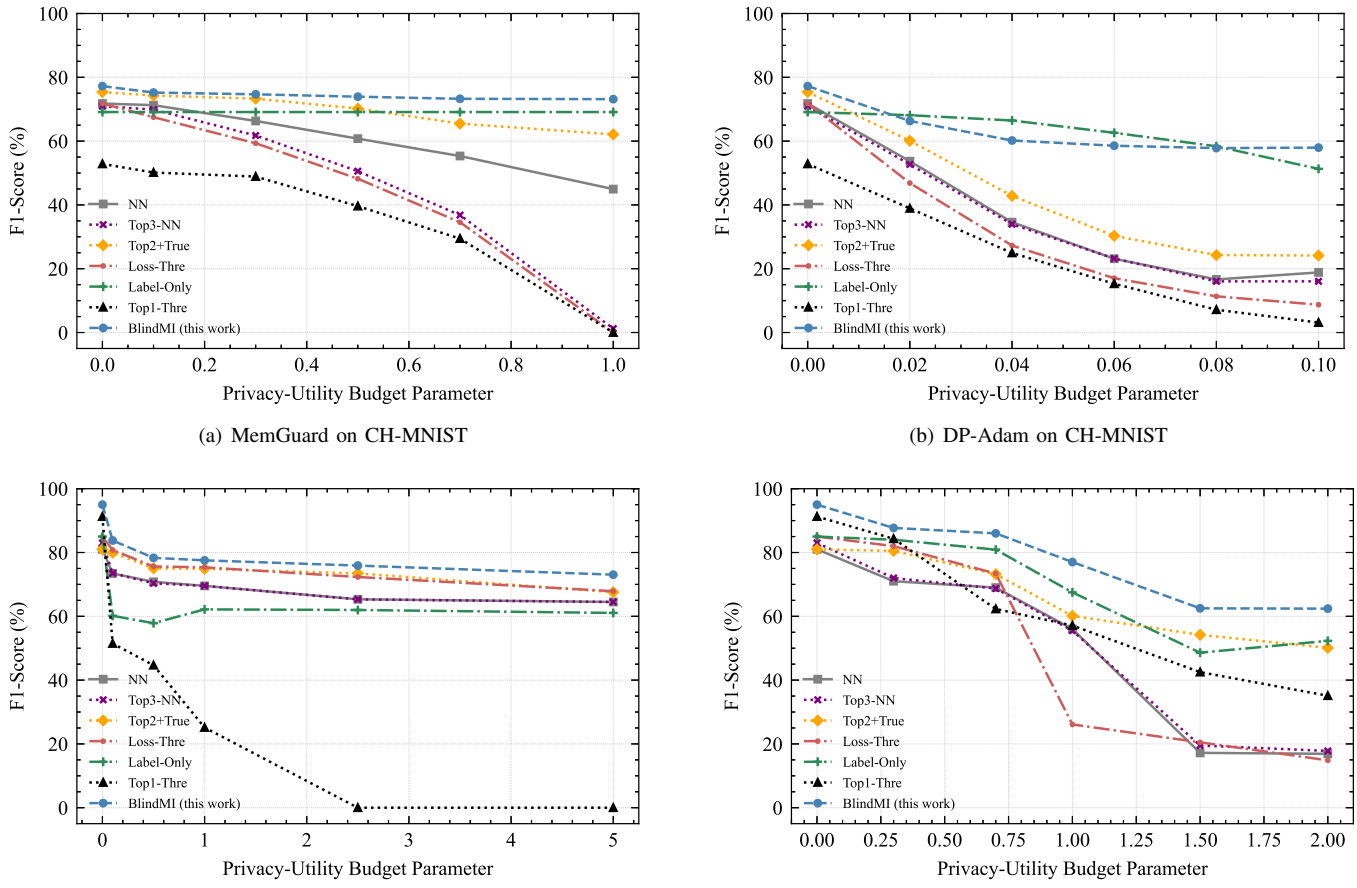
Our second observation is on how different nonmember generation methods affect attack performance. Table IX shows that sample transformation is the most effective method for both BLINDMI-DIFF and BLINDMI-1CLASS. Since the differences are relatively small, we perform two statistical tests, i.e., (i) the Mann-Whitney U test and the P-value, and (ii) the maximum mean discrepancy (MMD) tests, to demonstrate the statistical significance.

First, the U test value and P-value are shown in Table X. A large U test value and P-value indicates that two sets are similar, indicating statistical insignificance. Random generation is significantly different from all three other methods; Cross-domain sample selection is more similar to random perpetuation than sample transformation.

Second, we show the MMD value with standard error of the mean between generated samples and real-world non-members. A smaller MMD value indicates that the generated samples are close to real-world nonmembers. Clearly, nonmember generated by the sample transformation is the closest to real-world nonmembers; those generated randomly are the farthest—the MMD value is even larger than the one of members. The reason is that random generated samples follow a uniform distribution, which are far from the member and non-member boundary.

E. RQ4: BLINDMI-DIFF *with different classifiers and kernel functions*

In this subsection, we evaluate the performance of BLINDMI-DIFF with different internal parameters, e.g., BLINDMI-DIFF with different kernel functions and BLINDMI-DIFF-w/o with different initial separation classifiers. The performances of different kernel functions are in Table XII and the ones of different classifiers in Table XIII.



(a) MemGuard on CH-MNIST

(b) DP-Adam on CH-MNIST

(c) MMD+Mix-up on CIFAR-100

(d) Adversarial Regularization on CIFAR-100

Fig. 3. F1-Score of Different MI attacks, i.e., state-of-the-art and BLINDMI, against a Target Model together with Corresponding Defenses.

TABLE IX. F1-SCORE(%) OF BLINDMI-DIFF/BLINDMI-1CLASS WITH STANDARD ERROR OF THE MEAN OF WITH NONMEMBER DATASETS GENERATED VIA DIFFERENT METHODS OF VARIED SIZE.

Method \ Size	20	50	100	200	1,000	10,000
Sample transform	77.7±0.80 / 72.9±1.82	78.1±0.99 / 74.4±1.43	77.9±1.58 / 75.6±1.55	78.2±0.87 / 76.3±1.67	78.4±1.01 / 76.8±1.70	**78.7±0.63 / 77.5±1.31**
Random perpetuation	77.5±1.37 / 66.4±1.20	77.9±0.92 / 72.1±1.30	77.6±1.44 / 73.1±1.07	78.0±0.85 / 73.5±0.70	77.6±0.42 / 74.5±0.88	**78.2±0.30 / 75.7±0.69**
Random generation	75.5±2.51 / 71.6±1.98	75.7±1.93 / 71.6±2.31	75.3±2.03 / 71.3±2.38	75.6±1.79 / 71.4±2.00	75.7±1.59 / 71.8±2.03	**75.7±1.64 / 72.2±1.87**
Cross domain	77.9±1.26 / 64.9±1.99	78.0±1.38 / 71.4±1.46	**78.1±1.21** / 72.5±1.60	78.1±1.05 / 73.2±1.09	77.8±1.20 / 76.1±1.33	77.6±1.37 / **77.0±0.93**

TABLE X. MANN-WHITNEY U TEST VALUE (P-VALUE) OF F-1 SCORES OF BLINDMI-DIFF/BLINDMI-1CLASS WITH NONMEMBER SETS VIA DIFFERENT METHODS

	Sample transform	Random perpetuation	Random generation	Cross domain
Sample transform	18* (0.4678**) / 18 (0.4678)	–	–	–
Random perpetuation	7 (0.0455) / 7 (0.0463)	18 (0.4673) / 18 (0.4678)	–	–
Random generation	0 (0.0024) / 0 (0.0025)	0 (0.0023) / 7 (0.0461)	18 (0.4657) / 18 (0.4673)	–
Cross domain	9.5 (0.0981) / 7 (0.0463)	13 (0.2328) / 18 (0.4681)	0 (0.0023) / 10.5 (0.1303)	18 (0.4673) / 18 (0.4678)

*: the larger the U value is, the more similar two datasets are.
**: a p-value less than 0.05 indicates statistical significance.

TABLE XI. MMD STATISTICAL TESTS OF BLINDMI-DIFF WITH NONMEMBER DATASETS GENERATED VIA DIFFERENT METHODS (EACH VALUE IS THE MMD WITH STANDARD ERROR OF THE MEAN BETWEEN CORRESPONDING SAMPLES AND REAL-WORLD NON-MEMBERS IN THE TEST DATASET.)

Sample trans	Random perp	Random generation	Cross domain	Training set
0.194 ± 0.009	0.438 ± 0.039	3.024 ± 1.024	0.225 ± 0.015	1.864 ± 0.022

[Observation RQ4-1] *The Gaussian kernel outperforms other kernels in most of the cases.*

As shown in Table XII, the Gaussian kernel outperforms other kernels in all the datasets of BLINDMI-DIFF-w/ and most of the datasets of BLINDMI-DIFF-w/o (except for CH-MNIST and CIFAR-100); the Laplacian kernel comes next due to its similarity to the Gaussian kernel (the former adopts L1-norm

TABLE XII. F1-SCORE (%) WITH STANDARD ERROR OF MEAN FOR DIFFERENT KERNEL FUNCTIONS OF BLINDMI-DIFF

		Gaussian (default)	Laplacian	Linear	Sigmoid	Polynomial
DIFF-w/	Adult	**64.2±1.59**	60.3±0.38	40.7±0.20	51.1±0.41	58.4±1.02
	EyePACS	**77.7±0.80**	67.3±0.31	71.8±0.93	72.8±0.87	73.9±0.88
	CH-MNIST	**75.1±1.49**	73.1±0.92	72.4±0.53	71.3±0.71	72.7±1.20
	Location	**86.2±0.90**	85.1±2.42	83.4±0.98	79.8±1.52	76.7±0.17
	Purchase-50	**78.0±0.31**	68.9±0.50	75.8±0.61	71.1±1.05	66.0±0.99
	Texas	**85.5±0.80**	83.6±0.47	81.2±0.29	80.9±0.49	81.9±1.72
	CIFAR-100	**93.9±0.63**	93.3±0.79	87.9±1.09	86.9±1.02	90.1±0.83
	Birds-200	**96.8±0.09**	91.9±1.32	95.7±1.06	94.4±1.31	93.9±0.96
DIFF-w/o	Adult	**62.7±1.12**	52.2±0.74	50.1±0.32	48.9±0.63	57.1±1.83
	EyePACS	**75.0±1.40**	72.9±0.65	69.4±0.19	69.2±0.28	70.1±0.53
	CH-MNIST	75.1±1.89	**75.7±2.22**	72.9±1.23	71.9±0.84	73.0±1.82
	Location	**83.3±0.57**	81.2±1.89	76.4±0.67	77.4±2.15	72.1±0.08
	Purchase-50	**76.5±0.25**	66.1±0.67	74.9±0.09	74.5±0.38	76.5±1.12
	Texas	**80.7±2.37**	76.2±1.24	74.1±0.80	74.7±0.79	75.8±1.02
	CIFAR-100	92.1±1.15	**92.8±1.32**	82.9±0.33	80.9±0.36	88.9±0.86
	Birds-200	**96.2±0.26**	96.0±0.34	95.7±0.83	94.1±0.51	94.4±1.02

and the latter L2-norm); the linear kernel, due to its simplicity, performs the worst.

[Observation RQ4-2] *The threshold classifier outperforms other initial sample separation classifiers for* BLINDMI-DIFF-*w/o.*

We evaluate three initial sample separation classifiers. The threshold classifier ("Threshold") is a separation based on the highest probability score, among which we select the 1,000 lowest ones as our nonmembers. The others are two different clustering algorithm including K-means and Agglomerative Clustering.

Table XIII shows that the "Threshold" is the worst for initial F1-score but the best after BLINDMI-DIFF-w/o. The reason is the "Threshold" only selects a few samples with a high probability to be nonmembers. Since "Threshold" left out many nonmembers, the initial F1-score is relatively low; at the same time, a high quality nonmember set also helps BLINDMI-DIFF-w/o to achieve a relatively good performance. The results of K-means and Agglomerative Clustering are similar. The initial F1-scores are higher than "Threshold"; however, since there does not exist a set with high quality nonmembers or members, the performance of BLINDMI-DIFF-w/ is relatively lower.

We also show the precision, recall and F1-score of BLINDMI-DIFF-w/ (with "Threshold" as the classifier) as the number of iterations increases in Figures 4, 5, and 6. The recall starts from a point that is very close to 1 and drops as the number of iterations; by contrast, the precision increases steadily together with the F1-score. It is worth noting that the recalls of Adult and CH-MNIST drop the most compared with other datasets because members are more similar to nonmembers in target models trained from these two datasets.

F. RQ5: Number of Moves, Iterations, and Execution Time of BLINDMI-DIFF

In this research question, we measure the time and the numbers of moves and iterations of BLINDMI-DIFF to finish the inference of the target dataset. Note that moves are atomic steps in which BLINDMI moves a sample from $S^{prob,k}_{target}$ to $S^{prob,k}_{nonmem}$. Then, iterations are when BLINDMI updates the

TABLE XIII. F1-SCORE (%) WITH STANDARD ERROR OF MEAN FOR DIFFERENT ROUGH SAMPLE SEPARATION CLASSIFIERS FOR BLINDMI-DIFF-W/O.

Dataset	Threshold (default)		K-means		Agg. Clustering	
	initial	+ diff-w/o.	initial	+ diff-w/o.	initial	+ diff-w/o.
Adult	60.2±0.04	**62.7±1.12**	55.1±1.75	60.1±1.02	58.7±0.90	59.4±0.23
EyePACS	70.6±0.58	**75.0±1.40**	70.0±1.15	74.9±0.23	70.0±1.15	73.0±0.50
CH-MNIST	73.2±0.71	**75.1±1.89**	70.3±0.18	72.0±2.46	69.8±0.21	76.3±1.41
Location	76.9±0.00	**83.3±0.57**	74.2±0.43	82.2±4.84	70.6±0.86	81.3±0.06
Purchase-50	69.0±0.00	**76.2±0.25**	73.6±0.28	74.2±1.23	72.7±0.90	73.3±0.66
Taxes	68.9±0.03	**80.7±2.37**	71.4±0.33	77.0±1.51	70.6±0.49	79.4±1.47
CIFAR-100	68.8±0.13	**92.1±1.15**	82.9±1.01	87.7±0.98	81.1±3.20	86.2±4.20
Birds-200	71.4±0.03	**96.2±0.26**	92.9±0.77	93.5±0.23	94.7±0.99	96.1±0.37

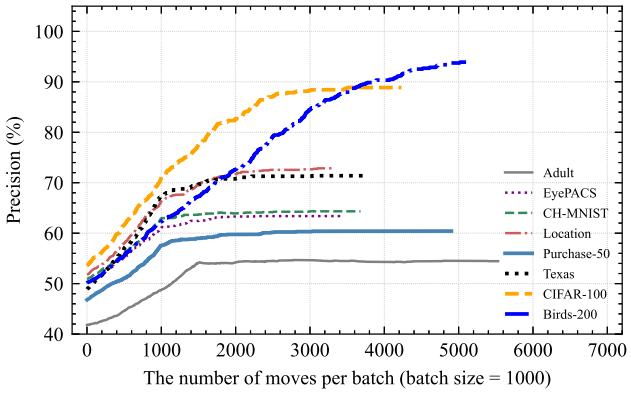


Fig. 4. Precision vs. # of moves per batch for BLINDMI-DIFF-w/o.

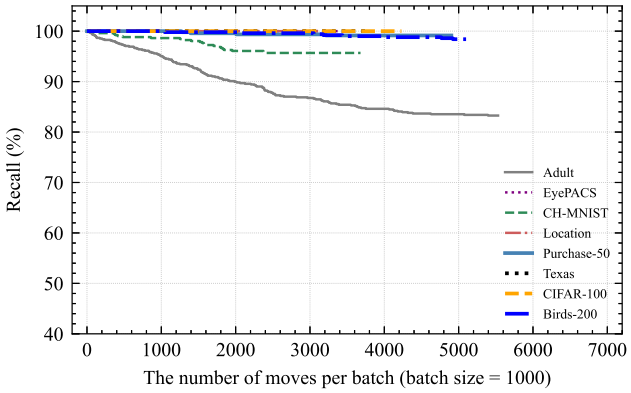


Fig. 5. Recall vs. # of moves per batch for BLINDMI-DIFF-w/o.

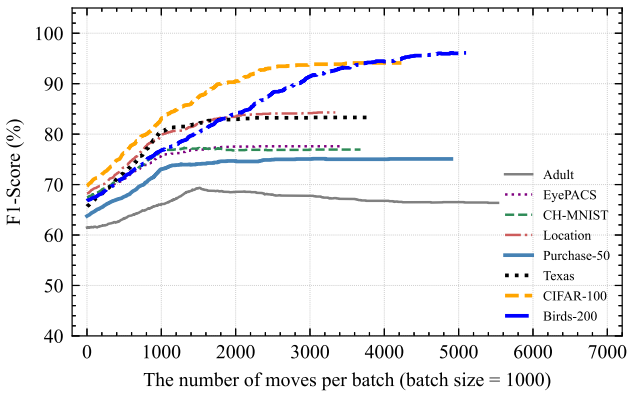


Fig. 6. F1-Score vs. # of moves per batch for BLINDMI-DIFF-w/o.

TABLE XIV. EXECUTION TIME (SECOND) AND # OF MOVES AND # OF ITERATIONS WITH STANDARD ERROR OF MEAN FOR BLINDMI-DIFF

Dataset	BlindMI-diff-w/.			BlindMI-diff-w/o.		
	Time (s)	Moves (#)	Iter. (#)	Time (s)	Moves (#)	Iter. (#)
Adult	494±23	63,124±616	7,012±98	2,530±28	202,407±694	405±14
EyePACS	224±16	44,838±858	2,818±14	751±31	94,247±448	120±9
CH-MNIST	73±11	12,181±386	983±29	293±28	25,061±429	30±2
Location	70±2	9,839±120	857±4	271±18	20,659±464	31±2
Purchase-50	370±6	48,943±373	4,336±114	1,215±45	97,243±761	127±3
Texas	313±5	47,428±903	3,086±65	781±4	67,379±746	110±7
CIFAR-100	238±15	41,128±358	3,051±101	984±59	104,006±310	168±9
Birds-200	183±18	30,261±647	2,067±25	842±68	70,109±325	107±2

distance between $S^{prob,k}_{target}$ and $S^{prob,k}_{nonmem}$. The evaluation results are shown in Table XIV.

[Observation RQ5-1] *The execution time and the number of moves and iterations depend on the size of the target dataset.*

Our first observation is that the execution time and number of moves and iterations depend on the size of the target dataset. The larger the target dataset is, the longer time and more moves and iterations it takes for BLINDMI-DIFF to finish the inference. The reason is that the larger size of the datasets increases the number of moves per interation, and thus increases the potential time and numbers of iterations taken by BLINDMI-DIFF.

[Observation RQ5-2] BLINDMI-DIFF-*w/o takes significantly longer time, and more moves, than* BLINDMI-DIFF-*w/.*

Our second observation is that BLINDMI-DIFF-w/o is generally slower than BLINDMI-DIFF-w. The reason is that BLINDMI-DIFF-w/o adopts bi-directional differential comparison: The moves are bi-directional and thus the number of BLINDMI-DIFF-w/o is larger than BLINDMI-DIFF-w/.

[Observation RQ5-3] *The total number of iterations depends on the batch size.*

Our third observation is that the batch size determines the number of iterations: That is why BLINDMI-DIFF-w/ with a batch size as 20 takes more iterations than BLINDMI-DIFF-w/o with a batch size as 1,000. Specifically, when the batch size is small, the number of batches is large, but the number of iterations per batch does not differ much, leading to a large number of iterations in total.

[Observation RQ5-4] *The distance between two sets increases as the number of moves per batch.*

Our last observation is on the distance between two sets vs. the number of moves per batch as shown in Figures 7 (BLINDMI-DIFF-w/) and 8 (BLINDMI-DIFF-w/o). Note that only a move that increases the distance is a valid one between two sets; otherwise, the sample is kept in the original set.

G. RQ6: BLINDMI *with Different Configurations*

In this subsection, we evaluate BLINDMI with different configurations, including different nonmember-to-member ratios (Bargav et al. [25]) and different prediction classes. The evaluations are performed under the blackbox setting as many attacks require ground-truth labels.

1) Different Nonmember-to-member Ratios: In this part, we evaluate the F1-score of BLINDMI and existing attacks when the nonmember-to-member in the target dataset changes. Specifically, we follow Bargav et al. [25] to adjust the

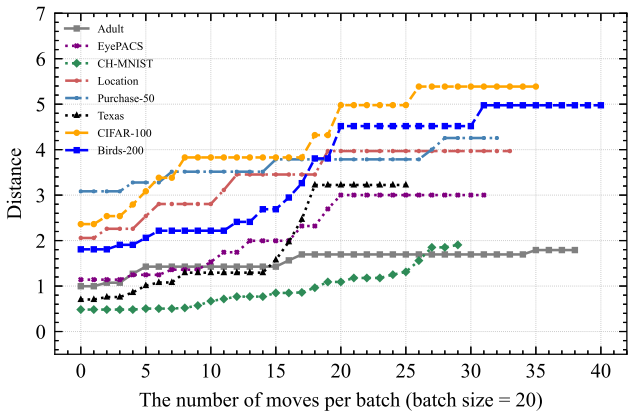


Fig. 7. Distance vs. # of iterations per batch for BLINDMI-DIFF-w/.

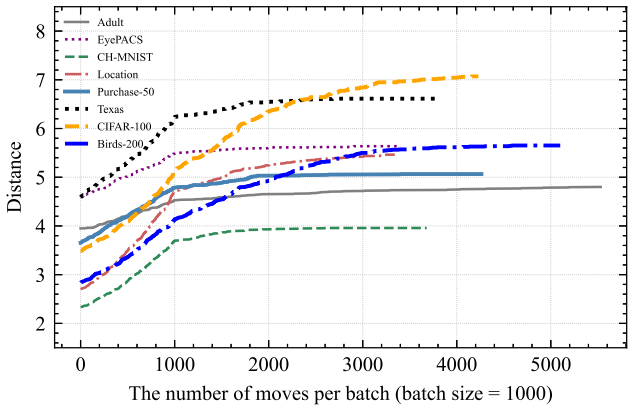


Fig. 8. Distance vs. # of iterations per batch for BLINDMI-DIFF-w/o.

Nonmember-to-Member ratio r and measure the F1-score. The underlying rational behind the introduction of r is that a practical target dataset usually has a small number of members and a large number of nonmembers. Our evaluation results based on CIFAR-100 are shown in Figure 9.

[Observation RQ6-1] *While the performance of all MI attacks degrades as the nonmember-to-member ratio (r) increases,* BLINDMI *is the slowest among all and significantly outperforms existing attacks at a large r value.*

This observation shows the practicability of BLINDMI under real-world settings. All other attacks in the literature drops logarithmically as r increases, while the performance decrease of BLINDMI is stable. That is, the performance of existing attacks drops below 50% when r is larger than 10, while the performance BLINDMI is still above 50%, i.e., 57.5% (35% than the state-of-the-art), when r equals to 39.

2) Different Prediction Classes: In this part, we evaluate the F1-score of BLINDMI and all other attacks when the number of classes in the target model increases. The experiment settings are as follows. We divide the entire CIFAR-100 datasets into subsets with 2, 10, 50, 70, and 100 classes and then launch MI attacks against target models trained from these subsets. The F1-scores of these attacks are shown in Figure 10 and our observation is as follows.

[Observation RQ6-2] *The performance of all MI attacks, including* BLINDMI*, increase as the number of classes in the*

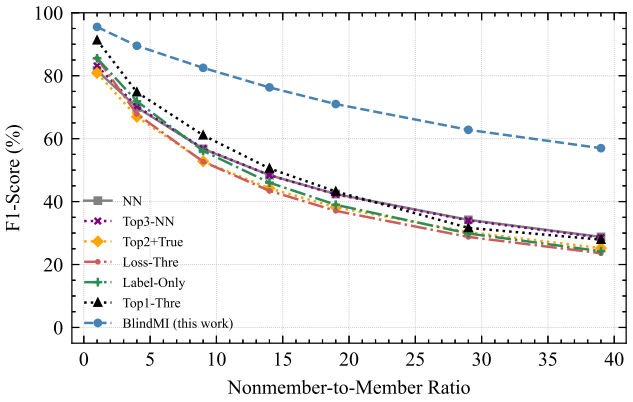


Fig. 9. F1-Score of Various Attacks vs. Nonmember-to-Member Ratio on CIFAR-100.

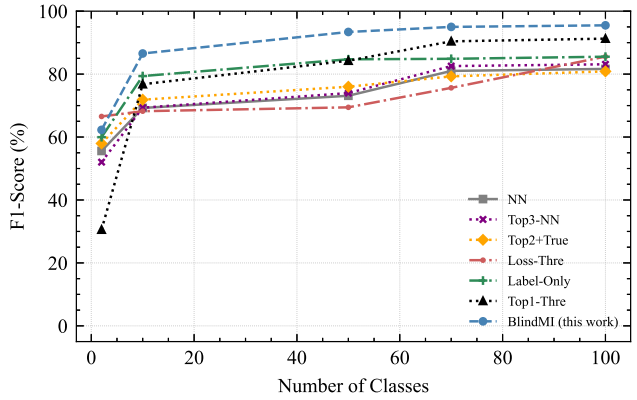


Fig. 10. F1-Score of Various Attacks vs. # of classes on CIFAR.

target model and this performance boost is more significant when the number of classes is small.

Figure 10 shows a steady improvement of all MI attacks as the number of classes. The reason is that when the dataset has more classes, the target model tends to generalize less, thus being more vulnerable to MI attacks. This can also explain why target models trained on CIFAR-100 and Birds-200 are more vulnerable compared to other datasets.

It is also worth noting that Top1-threshold performs the worst among all the MI attacks when the number of classes equals two, but the performance improves when the number becomes 100. That is, the top one probability score contains more information as the number of classes increases. We believe that this may be due to the fact that when the total probability is shared by more classes, one can infer more membership information from the top probability.

VI. A Discussion on Potential Defenses

In this section, we discuss potential defenses. There are two possible venues of defenses as we have seen in the literature.

- Limiting adversary's access to the target model. The first method is to limit the adversary's access to the target model: (i) restricting the number of probes and also which samples can be probed, and (ii) providing only the predicted class information as an output. The former will restrict BlindMI to BlindMI-diff-w/o, which performs a little bit worse than BlindMI-diff-w/. The latter will reduce BlindMI to the Label-only attack in the blackbox setting.
- Improving the robustness of the target model. The second method is to improve the model robustness to MI attacks via different methods evaluated in Section V-C. The differential privacy-based approach is likely the best method for defending against BlindMI in the literature. A combination of existing attacks may also be possible and this is left as a future work for our study.

VII. Related Work

Machine learning is vulnerable to different privacy attacks including model inversion [12], [13], membership inference [43], property inference [2], [14], as well as model and hyperparameter stealing [46], [47]. Our work studies membership inference (MI) attacks. We describe related work on MI attacks and defenses in Section VII-A and VII-B.

A. Existing Membership Inference (MI) Attacks

Membership inference attacks originate back to 2008, when, Homer et al. [22] first proposed a MI attack on biological data, whereby an adversary could infer whether a data sample belonged to a genome-based study knowing only parts the genome and summary statistics. Then, in 2017, Shokri et al. [43] proposed the first modern MI attack against deep neural networks with a shadow model and a binary attack classifier.

Prior attack methods include the following. Salem et al. [41] proposed several MI attacks. For example, the Top3-NN attack of Salem et al., a variant NN Attack, picks the top three largest values from all confidence scores to train an MI classifier. For another example, the Top1-Threshold attack of Salem et al. compares the top feature from the output probability distribution with a threshold and classified the sample as member if the top feature is larger than a threshold. Similarly, Yeom et al. [49] also proposed two attacks with the help of ground-truth labels: the first label-only attack compares the ground truth label with predicted, and the second loss-threshold attack computes cross-entropy loss and compares the computed loss with the average loss of all training samples. As a comparison, BlindMI is an attack that does not need a shadow model but also extracts complex membership semantics via probing only. Our evaluation shows that BlindMI outperforms existing attacks under different adversarial settings.

Researchers have also proposed theories on MI attacks. Sablayrolles et al. [40] proposed an optimal strategy for MI attacks using a probabilistic framework that consists of both Bayesian learning and noisy training. They showed that optimal attacks only depend on the loss function, and thus black-box attacks could be as good as whitebox attack. BlindMI actually proves the effectiveness of blackbox attacks.

In addition to attacks on classification models, researchers also have proposed MI attacks [19] on generative models [15] and those [37] on federated learning. As a comparison, BlindMI is an attack on single classification models rather than generative models or federated learning.

B. Existing Defenses

We now describe existing defenses of MI attacks [31], [32], [36], [44], especially those on classification models. Note that while existing defenses can prevent some existing MI attacks with a reasonable performance, our evaluation shows

that BLINDMI can still infer membership with a reasonable F1-score, e.g., over 60%.

1) Regularization: Researchers have proposed to improve privacy against MI attacks via different types of regularization. For example, Salem et al. [49] demonstrated two effective method of defending MI attacks, namely dropout and model stacking. The former randomly deletes a fixed proportion of edges in a fully connected neural network model to improve model robustness; the latter constructs a target model with multiple different machine learning models stacked together. For another example, Shokri et al. [43] adopted L_2-norm standard regularization with a polynomial in the model's loss function to penalize large parameters. Nasr et al. [36] introduced a min-max game mechanism to train models with membership privacy, which ensures indistinguishability between the predictions of a model on its training data and other data points from the same distribution. This strategy acts as an adversarial regularizer that generalizes the model. In addition, Li et al. [30] proposed to close the generalization gap by matching the training and validation accuracies. Specifically, they adopted a new set regularizer, called the Maximum Mean Discrepancy, between the softmax output empirical distributions of the training and validation sets during training.

2) Adversarial Example: Another direction is to borrow ideas from adversarial machine learning and generate an adversarial example for the inference model controlled by the adversary. For example, Jia et al. [26] introduced a new defense, called MemGuard, by adding noise to confidence score output from target models, thus fooling a binary classifier. Unlike previous adversarial examples [8], [16], [28], [33]–[35], [39], [45], MemGuard calculates the gradient of the loss function to find an appropriate noise and guarantee the utility loss to be zero.

3) Privacy Enhancement: Many differential privacy based defenses [9], [11], [24] add noise to the objective function that is used to learn a model or the gradients during optimizing the objective function. Shokri et al. [42] designed a differential privacy method for collaborative learning of DNNs. Cao et al. [7] showed that privacy-related data samples can be unlearned to improve model privacy.

VIII. CONCLUSION

In this paper, we present a novel MI attack, called BLINDMI, which adopts differential comparison moving samples in between two sets and making inference decisions. One of the key insights used here is that, moving a member from a mostly member dataset to a mostly non-member one will decrease the distance in feature space between two sets and vice-versa. We implement three versions of BLINDMI, BLINDMI-1CLASS (relying on one-class SVM), BLINDMI-DIFF-w/ (relying on generation of nonmembers), and BLINDMI-DIFF-w/o (relying on rough separations of members and non-members). Our evaluation shows that BLINDMI outperforms existing state of the art attacks, against not only a variety of DNN architectures, but also against DNNs with state of the art defenses deployed.

ACKNOWLEDGMENT

We want to thank anonymous reviewers for their helpful comments and feedback. This work was supported in part by the Johns Hopkins University Institute for Assured Autonomy with grant IAA 80052273, National Science Foundation (NSF) grant CNS-18-54000 and CNS-19-37786, as well as an IBM Faculty Award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF.

REFERENCES

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[2] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, 2015.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[4] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[5] P. Burlina, D. E. Freund, B. Dupas, and N. Bressler, "Automatic screening of age-related macular degeneration and retinal abnormalities," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 3962–3966.

[6] P. M. Burlina, N. Joshi, M. Pekala, K. D. Pacheco, D. E. Freund, and N. M. Bressler, "Automated grading of age-related macular degeneration from color fundus images using deep convolutional neural networks," *JAMA ophthalmology*, vol. 135, no. 11, pp. 1170–1176, 2017.

[7] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, 2015.

[8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[9] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization." *Journal of Machine Learning Research*, vol. 12, no. 3, 2011.

[10] T. Chen, I. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," 2016.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[12] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1322–1333. [Online]. Available: https://doi.org/10.1145/2810103.2813677

[13] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *USENIX Security Symposium*, 2014.

[14] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *CCS*, 2018.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[17] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[18] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[19] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: Membership inference attacks against generative models," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 1, pp. 133–152, 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[22] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays," *PLoS genetics*, vol. 4, no. 8, 2008.

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

[24] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang, "Towards practical differentially private convex optimization," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 299–316.

[25] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Revisiting membership inference under realistic assumptions," *arXiv preprint arXiv:2005.10881*, 2020.

[26] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 259–274.

[27] J. N. Kather, C.-A. Weis, F. Bianconi, S. M. Melchers, L. R. Schad, T. Gaiser, A. Marx, and F. G. Z"ollner, "Multi-class texture analysis in colorectal cancer histology," *Scientific reports*, vol. 6, p. 27988, 2016.

[28] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world. corr abs/1607.02533 (2016)," *arXiv preprint arXiv:1607.02533*, 2016.

[29] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[30] J. Li, N. Li, and B. Ribeiro, "Membership inference attacks and defenses in supervised learning via generalization gap," *arXiv preprint arXiv:2002.12062*, 2020.

[31] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," *arXiv preprint arXiv:1712.09136*, 2017.

[32] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "Understanding membership inferences on well-generalized learning models," *arXiv preprint arXiv:1802.04889*, 2018.

[33] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[34] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[35] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[36] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 634–646.

[37] ——, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 739–753.

[38] S. J. Oh, B. Schiele, and M. Fritz, "Towards reverse-engineering black-box neural networks," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

[39] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[40] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou, "White-box vs black-box: Bayes optimal strategies for membership inference," in *International Conference on Machine Learning*, 2019, pp. 5558–5567.

[41] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Network and Distributed Systems Security Symposium 2019*. Internet Society, 2019.

[42] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

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

[44] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 241–257.

[45] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[46] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *USENIX Security Symposium*, 2016, pp. 601–618.

[47] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 36–52.

[48] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

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

[50] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[51] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.