Membership Inference Attacks on Deep Learning Models

By

SHAHBAZ REZAEI
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Xin Liu, Chair

Hao Chen

Zubair Shafiq

Committee in Charge

2023

To all innocent lives taken away by Islamic Republic.

Contents

Membership Inference Attacks on Deep Learning Models

Abstract

Recently, deep learning models have been extensively adopted in numerous applications, from health care to finance and entertainment industry. This wide-spread deployment of deep models raised concern over the privacy of data used to train deep models. This is a huge concern particularly for data-sensitive applications, such as health records, personal data, bio-metric data, etc. As a result, a new direction of research focusing on possible attacks aiming to identify training data of deep models emerged, called *membership inference*.

Membership inference (MI) attacks identify which samples have been used during training and which samples have not. The first generation of membership inference attacks mainly used deep models' prediction confidence as a feature to identify training samples. The intuition is that deep models are more confident on samples they have seen during training than non-training samples.

Despite their sound intuition and apparent successful reports, we, along a few other parallel studies, showed that the first generation of membership inference attacks are ineffective in practice for multiple reasons. First, they could not significantly outperform a naive baseline that labels a sample as a member (training sample) if it is correctly classified by the deep model and as a non-member (non-training sample) otherwise. Second, the confidence distribution of correctly classified samples, which cover the majority of a dataset, are not distinguishable between train and non-train samples. Only a small portion of mis-classified samples exhibit discrepant distribution. Third, all these membership inference attacks report average-case success metric (e.g., accuracy or ROC-AUC). However, privacy is not an average case-metric, and it should be treated similar to other security and privacy related problems. Similar to other security problems, the attack is reliable if it can identify a few training samples while almost on non-training samples are falsely labeled as a training sample. In other words, a reliable membership inference attack should have a decent true-positive rate (TPR) at low false-positive rates (FPR).

In this dissertation, we aim to move the membership inference research in a more practical direction, either by showing the limitations of the current attacks or by proposing more reliable attacks. As stated earlier, we first show that the current generation of membership inference

attacks are not reliable in practice. Then, we propose several new membership inference attacks that achieve more reliable performance in more realistic scenarios. The first attack focuses on the model's behavior in the entire sub-population, instead of a single sample in vacuum. More specifically, we compare the model's confidence on a target sample and other samples from the same sub-population. If the confidence of a sample is significantly higher than the average confidence on that sub-population, that is an indication of a training sample. We show that this attack can achieve moderate true positive with very low false positive. Additionally, we propose a BiGAN architecture to generate samples from the same sub-population, in case it is not available. The second attack aims to focus on user-level MI attack instead of the record-level MI attack. In this scenario, we identify if a user's data has been used during training instead if identifying which samples from the user have been used. Not only this attack is more realistic in privacy domain, but we show that we can achieve the state-of-the-art accuracy if multiple samples from a user are used to draw the membership inference. In another study, we show that MI attacks are generally more successful when deep ensemble is used. We show that deep ensemble shifts the distribution of train and non-train samples in a different way where they become significantly more distinguishable. Finally, we show that are a few simple aggregation mechanisms instead of ensemble averaging that can improve the accuracy and privacy of deep models in deep ensemble context.

Finally, we illustrate a fundamental issue with current MI attacks, including the state-of-the-art attacks, that limits their applications in certain scenarios. We elaborate the issues with a practical scenario where membership inference attacks are used by an auditor (investigator) to prove to a judge/jury that the auditee unlawfully used sensitive data during training. Although the current SOTA attacks can identify some training samples with low false positive ratio in a common experimental setting extensively used for MI attacks, an auditee can generate unlimited number of samples on which MI attacks catastrophically fail. This can be used in court to easily discredit the allegation of the auditor and make the case dismissed. Interestingly, we show that auditee does not need to know anything about the auditor's membership inference attack to generate those challenging samples. We called this problem, discredibility. Currently, there is no attack immune to discredibility. We hope that our research sheds light on this newly-discovered issue and encourage researchers to investigate it.

Acknowledgments

I would like to express my special appreciation and thanks to my supervisor Professor Xin Liu. She offered me the great opportunity to work on the topic that I am passionate about. I would like to thank you for encouraging my research and for allowing me to grow as a researcher. I would like to thank Professor Zubair Shafiq who helped me through the project and gave me invaluable feedback. I would like to express my appreciation for the helpful comments and guidance from my dissertation committee members at the University of California, Davis: Professors Hao Chen and Zubair Shafiq. In addition to them, I want to thank Professor Sam King and Professor Lifeng Lai from my qualifying exam committee for their advice. A special thanks to my family. Words cannot express how grateful I am to have their support.

CHAPTER 1

Introduction

With abundance of data and more accessible machine learning implementations, machine learning models and deep neural networks (DNN) are extensively adopted for various applications, from health-care to finance. Moreover, the emergence of machine learning as service (MLaaS) platforms facilitates the use of DNN models even for applications and businesses. With the widespread adaptation of deep neural networks (DNN), their security challenges have received significant attention from both academia and industry, especially for mission critical applications, such as road sign detection for autonomous vehicles, face recognition in authentication systems, and fraud detection in financial systems.

Recently, various privacy and security aspects of DNNs have been studied leading to the exposure of many vulnerabilities, including adversarial attacks [**1**,**10**,**97**,**99**,**136**,**140**,**144**], membership inference attacks [**41**, **44**, **71**, **92**, **110**, **137**, **141**], model extraction/stealing attacks [**5**, **21**, **57**, **118**], functionality stealing attack [**93**], model inversion attacks [**26**, **27**, **78**, **79**, **142**], property inference attacks [**2**, **28**, **79**, **94**, **124**], poisoning attacks [**13**, **48**, **53**, **86**, **109**, **145**], hyper-parameter stealing attacks [**122**], etc. In this dissertation, we focus on membership inference attacks where the goal is to identify samples used during the training of a DNN model.

In this section, we briefly introduce a rudimentary knowledge regarding DNNs' security and privacy. Next, we introduce the three major categorize of attacks on DNNs. Lastly, we focus on membership inference attacks, current state-of-the-art, and our contributions.

1.1. Background

In machine learning security, an attack has a threat model that defines the goal, capabilities (or knowledge), and target model. The attacker's goal can be categorized in terms of security violation: 1) violation of *availability* that aims to reduce the confidence of a model for normal inputs, 2) violation of *integrity* that aims misclassification on certain inputs without affecting normal inputs,

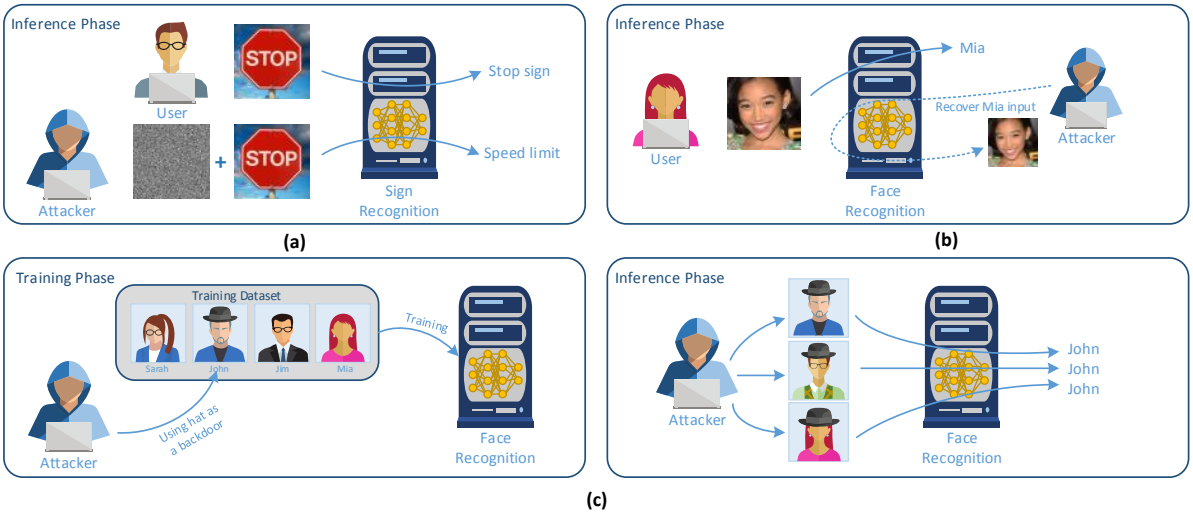


FIGURE 1.1. Typical attacks on machine learning: (a) Adversarial attack, (b) model inversion attack, and (c) backdoor attack.

and 3) violation of *privacy* that aims to obtain confidential information about the model, training or inference-time data and users, or even hyper-parameters used during training (hyper-parameter stealing attack).

The life-cycle of a typical machine learning model with offline training data consists of training and inference phases, which indicate attacker's capabilities and knowledge. Training phase capabilities are *data injection*, where the attacker injects new data points to the training dataset, *data poisoning*, where the attacker modifies the existing data points in the training dataset, and *logic corruption*, where the attacker interferes with the learning algorithm.

In the inference phase, the model is assumed to be fixed and the attacker cannot change the model. However, the attacker can still craft data inputs that fool the model to provide incorrect outputs. Hence, the attacker's capability is defined based on how much information she has about the model, ranging from *white-box*, where everything is possibly known including the entire model and training data, to *black-box* attacks, where minimum knowledge about the model, training data and algorithm is known. Any attack model that lays between while-box and black-box attack in terms of available information about the model is called *gray-box* attack.

1.2. Security Attacks on DNNs

In machine learning security, attacks are often categorized into three attack types based on the threat model:

Evasion attack (adversarial attack): The goal of an evasion attack is to manipulate the input data such that the model misclassifies. Although one can technically manipulate training data using evasion attack methods during training phase (often for adversarial retraining as a defense mechanism), evasion attack is an inference phase attack that violates the integrity. Figure 1.1(a) illustrates the adversarial attack where the attacker add a small perturbation, imperceptible to human eye, to the stop sign image to cause the model to misclassify.

Data poisoning: This is a training phase attack where the attacker inject or manipulate training data to either create a backdoor to use at inference time (without compromising the model performance on normal input data) or to corrupt the training process. Hence, it can violate availability or integrity depending on the goal. A typical example is to create a backdoor for face recognition task where the attacker injects a set of training samples with a specific object in a target person's training data. The aim is to force the model to associate the specific object with the target class. Then, any face image with the object is classified as the target class even if it belongs to another person. For instance, in Figure 1.1(c), the attacker inject faces of John with a special hat during training. Then, at the inference phase, any face that has the hat is classified as John by the model.

Exploratory attacks: The aim of the attack is to violate the privacy at inference phase. It covers several types of attacks, including *model extraction*, to extract model parameters, *membership inference attack*, to examine whether a data point is used during the training phase, *model inversion*, to infer something about input by observing the model output. In Figure 1.1(b), the attacker aims to recover the input image of Mia by observing the output and the model. Although exploratory attacks have been widely studied for classical machine learning algorithms, there are only a few works on deep learning models. For example, it has been recently shown that sensitive and out-of-distribution sentences, such as "My social security is —-", can be leaked from commercial text-completion neural networks [**8**].

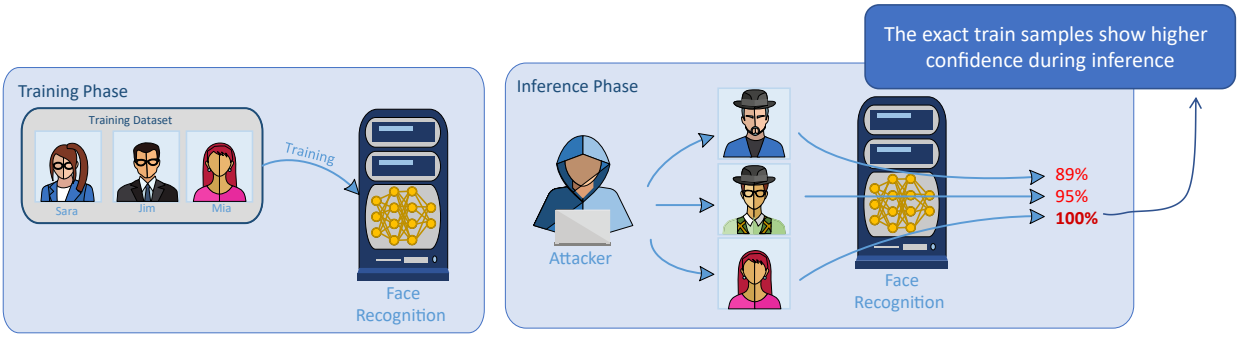


FIGURE 1.2. Membership inference attacks often rely on model's confidence to inference the membership status.

1.3. Membership Inference Attacks on DNNs

Given a data sample and a model, membership inference (MI) attempts to determine if the data sample had been used to train the model. Membership inference attacks have been proposed in various applications and scenarios, including natural language processing models [**9**, **41**, **49**, **84**, **108**], generative models [**4**, **11**, **32**, **37**, **38**, **43**, **73**], speech recognition [**82**, **107**, **120**], health/genomic data [**6**, **30**, **31**, **71**, **129**, **135**, **141**], recommendation systems [**17**, **125**, **137**], graph neural networks [**34**, **54**, **92**, **127**], etc. Although some attacks have been designed to work on a very specific scenario, majority of membership inference attacks proposed in literature are general purpose attacks that can be used on any deep neural network.

There is an extensive recent literature on membership inference (MI) attacks on deep learning models that achieve high MI attack accuracy [**71**, **75**, **76**, **105**, **110**, **114**, **119**, **134**]. These MI attack models often use confidence values of the target model to infer the membership of an input sample, as shown in Fig 1.2. High MI attack accuracy is often justified by claiming that deep learning models are more confident towards the training (member) samples than the samples they have not seen during training. Consequently, MI attack accuracy is reported to be highly correlated to model's overfitting or generalization gap [**105**, **110**, **114**] because an overfitted model should perhaps behave even more confident towards training samples.

The first membership inference attack on deep models is proposed by Shokri et al. in [**110**]. The key idea is to build a machine learning attack model that takes the target model's output (confidence values) to infer the membership of the target model's input. To train the attack

models, membership dataset containing (x_{conf}, y_{mem}) pairs is needed where x_{conf} represents the confidence values obtained by the target model for each sample and y_{mem} is a binary variable indicating whether the sample is used in target model's training or not. To build the membership dataset, a set of shadow models are trained for which the training and non-training samples are known. The attack is possible under two assumptions [**105**]: (a) the shadow models share the same structure as the target model, and (b) the training dataset used to train the shadow models share the same distribution as the one used to train the target model. To mitigate these limitations, Salem et al. [**105**] relax the second assumption by showing the attack is possible using different datasets and the first assumption by proposing a threshold-based attack that does not require a training procedure. To further relax these assumptions, several studies [**71**, **75**, **76**, **114**, **119**, **134**] introduce better dataset generating procedures for shadow models, and extend the experiments to various scenarios and datasets. Majority of studies share the same idea of using target model's output for membership inference. More recent state-of-the-art approaches [**7**, **98**, **126**] adopts some forms of difficulty calibration (see Section 2), but they still rely on confidence values.

Unrelated to the membership inference, a large body of research focuses on understanding decision boundary of deep models [**56**, **83**], geometry and space of deep models [**23**, **85**], and properties of loss surface [**29**, **58**] to often improve training, to understand adversarial examples, or to improve robustness. In [**83**], it is shown that the decision boundary gets closer to training samples as training progress, and misclassified samples are closer to decision boundary than correctly classified samples. In [**23**], experimental results corroborate the idea that the classification region of deep models are connected, that is, deep models consist of several large regions, each of which contains samples of one class. In [**58**], the effect of large batch versus small batch on the curvature of the minimum of loss function and generalization gap is studied. Although none of these analyses has been used for membership inference attack directly, it is worth investigating them. The only studies that investigate features rather than confidence values are in [**15**, **96**]. In [**15**], the authors proposed two attacks based on input transformation and distance to the boundary in a black-box setting. Similarly, in [**96**], they propose the sampling attack that randomly perturbs an input to obtain a set of random transformations of the input and uses the predicted labels to infer membership

status. In this dissertation, we study distance to the decision boundary, a set of gradient norms with respect to model's input, and a set of gradient norms with respect to model's weight.

1.4. Contributions

In this dissertation, we focus on practical membership inference attacks on deep neural networks. We first study the practicality of the current membership inference attacks. We then propose a few attacks and cases where more reliable membership inference is possible. Lastly, we introduce a new concept, called discredibility, to warn the unreliability of even the SOTA membership inference attacks for certain scenarios. The detailed contribution of the dissertation is as follows:

(1) We show that traditional classification performance metrics, such as accuracy, precision, and recall are not enough to give us a clear picture of how MI attacks perform in practice, particularly on negative (non-member) samples. A better comprehensive evaluation should include false alarm rate (FAR) or true positive at low false positive rate as suggested in [**7**]. Moreover, we study the performance of correctly classified samples and misclassified samples separately. We show that membership inference of correctly classified samples, to which the majority of training samples belong, is a very difficult task. we extensively analyze and use other information available from the target model, including values from intermediate layers, the gradient w.r.t input, gradient w.r.t to model weights, and distance to the decision boundary. In some cases, these types of information slightly leak more membership status than confidence values, but they are still not sufficient for a reliable MI attack in practice.

(2) We propose a fundamentally different MI attack that achieves the same accuracy as SOTA while significantly reducing the shadow training computational overhead. Here, instead of comparing victim model's confidence on the target sample versus the average confidence of typical models, which requires training numerous shadow models, we compare the victim model's confidence on the target sample versus the victim model's confidence on similar samples from the same subpopulation as the target sample. Hence, we obviate the need to train multiple shadow models. However, in practice, the attacker may not have access to samples from the same subpopulation. To tackle this issue, we develop a BiGAN-like

architecture to train a generator that craft samples from the subpopulation of a given image. In other words, our attack only needs training a single generator model once and then it can be used for even unseen samples.

(3) Moreover, we introduce a user-level MI attack against metric embedding learning using properties of clusters in latent space. In user-level MI attack, the goal is to identify if any sample (image) from a target user has been used in the training. Here, the attacker might not have access to the exact training samples, but she can obtain other samples from the same user. This attack is more relevant in tasks where a user's identity is in danger of leaking, such as person re-identification. To launch the attack, we use average distance to the cluster's center and average pair-wise distance as features. We show that our attack achieves high accuracy even when the target model is probed with images of a training user that have not been used in the training, and therefore, we make the user-level MI attack viable.

(4) Furthermore, we perform a systematic empirical study of MI attacks on deep ensemble models. We start with an in-depth analysis of the most common ensembling technique and MI attack, and then we extend the results to various ensembling techniques and MI attacks. First, we show that when deep ensembles improve accuracy, it also leads to a more effective membership inference. We show that common defense mechanisms in membership inference literature, including differential privacy, MemGuard, MMD+Mixup, L1 and L2 regularization, as well as other ensemble training approaches, such as bagging and partitioning, can be used to mitigate effectiveness of MI attacks but at the cost of accuracy. We solve this trade-off issue by changing the fusing mechanism of deep ensembles which improves the accuracy and privacy, simultaneously.

(5) Additionally, we introduce a useful potential application of MI attacks for the purpose of auditing. Here, an auditor aims to prove to the judge/jury that private data has been unlawfully used by the auditee under investigation. The auditor uses an MI attack, and report the performance of the MI attack along the samples labeled as members to the judge. We show that the auditee can provide an unlimited number of non-member samples to the judge for which the MI attack model constantly fails, without knowing anything about the

MI attack or having query access to it. We call this process *discredibility*. Discredibility allows auditee to seriously damage the credibility of MI attack model used by the auditor and, consequently, get the case dismissed. We demonstrate that all contemporary membership inference attacks suffer from discredibility. We generalize our findings beyond this auditing application and argue about the inaccuracy of current attacks for record-level membership inference. Our findings suggest that current attacks may better suited for subpopulation-based membership inference or user-level membership inference.

CHAPTER 2

Related Work

Membership inference aims to identify samples used during the training of a target model, referred to as a *victim model*. Samples that has been used during the training are referred to as *members* or *train* samples, and other samples as *non-members*, *non-train* or *test* samples. We categorize the membership inference attacks into two classes, similar to [**126**]: MI attacks without sample difficulty calibration and MI attacks with sample difficulty calibration. Difficulty calibration aim to adjust the membership score of the attack so that it takes the difficulty of the classification of that particular sample into account. First generation of MI attacks do not use difficulty calibration and it has been shown to perform poorly in practice. Most state of the art attacks adopted some form of sample difficulty calibration. In this section, we introduce both classes of attacks.

2.1. MI Attacks without Difficulty Calibration

First generation of membership inference attacks were built upon the intuition that the confidence output of a victim model exhibits different distribution between train and non-train samples [**100**]. Simply put, the victim model is more confident on train samples than on non-train samples. Hence, the first membership inference attack on deep models was proposed in [**110**] using this idea. They train a membership inference attack model that takes the confidence output of a model as an input and predicts its membership status. The training dataset of the MI attack model is constructed by training *shadow models* and taking their confidence output. Shadow models are trained on the same task as the victim model, but with different dataset. Since the training set of the shadow models is known to the attacker, the attacker can easily construct the labels for the MI attack dataset. Many papers use the same idea with different variations of less restrictive assumptions [**69**, **71**, **75**, **76**, **100**, **105**, **114**, **119**, **134**, **146**].

The effectiveness of the first generation of MI attack has been seriously challenged when it has been shown that they can barely outperform a trivial baseline, called *gap attack* [**15**] or *naive attack*

[**66**,**100**]. Gap attack labels a sample as member if it is correctly classified by the victim model, and non-member otherwise. We show that the seemingly intuitive assumption that was the basis of these attacks generally do not hold. In other words, the distribution of confidence output of member and non-member samples are not significantly different, particularly when correctly classified samples are looked upon separately which constitute the majority of samples. Furthermore, Carlini et al. [**7**] argue that using average-case metric is not suitable for security-related applications and suggest using the true positive rate at a low false positive rate.

2.2. MI Attacks with Difficulty Calibration

The main challenge for the first generation of membership inference attacks was distinguishing between hard member samples (for which the confidence is low) from easy non-member samples (for which the confidence is high). As suggested in [**126**], MI attacks should have an adaptable reference point to which it compares the confidence of the target sample, called sample calibration. Most SOTA MI attacks that can perform well in a low false positive rate solve this issue by calibrating the confidence so that it takes the difficulty of the target sample into account [**7**, **98**, **103**, **126**].

In the Watson attack, the attacker excludes the target sample from the training set, and then train multiple shadow models. As a result, the attacker can obtain the average confidence output of a model in the absence of the target sample in the training data. By comparing the confidence output of the victim model with the average confidence output of the shadow models, the attacker can predict the membership status of the sample. In [**7**,**103**], a variation of this idea was used with one main difference. These attacks include two set of shadow models: one where the training set excludes the target sample and one where the training set includes the target sample. The main limitation of these attacks is that it needs to train hundreds of models for each sample it wants to investigate.

To tackle the huge training cost of these attack, we propose a slightly different and more efficient kind of calibration where it does not require training shadow models. In this attack, the attacker use a BiGAN architecture to craft samples from the same subpopulation as the target sample. Then, the attacks compares the confidence output of the target sample versus the subpopulation. If the target sample has higher confidence than its subpopulation, it is an indication of a member

sample. This attack achieves similar accuracy as other calibration-based attacks, while decreasing the training computation cost of shadow models significantly.

2.3. Uncategorized MI Attacks

Majority of membership inference attacks in literature use confidence output of the victim model as the main feature. There are, however, a few approaches that utilize other features [**15**, **96**, **100**]. Assuming a white-box access to the victim model, we [**100**] study a set of attacks using distance to the decision boundary, gradient w.r.t model weight, and gradient w.r.t input. However, none of the features significantly outperform the confidence output. A similar MI attack approach based on distance to the decision boundary has also appeared in [**15**] in the black box setting. Another MI attack approach is to compare the prediction of the target sample with the prediction label of its transformed versions [**15**] or randomly perturbed versions [**51**, **96**]. The intuition is that deep models are more robust to training samples. Hence, the transformed/perturbed sample is less likely to be mislabeled by the victim model. Despite their moderate accuracy and black-box nature of some of these novel MI attacks, they perform poorly in terms of achieving low false positive rates.

2.4. Membership Inference Defenses

Defense mechanisms against membership inference attacks can be summarized into two categories [**96**]:

Generalization-based: Shokri [**110**] was the first to correlate membership inference success with overfitting. Since then, many standard regularization techniques have been used to alleviate overfitting, such as L1 regularization [**15**], L2 regularization [**15**, **52**, **89**, **110**, **119**], differential privacy [**15**, **96**], dropout [**52**], and adversarial training [**88**]. Interestingly, ensemble learning has also been proposed as a defense mechanism. In [**105**], they proposed a combination of partitioning and stacking as a defense mechanism. The intuition is that training each model with different subset of data makes the entire ensemble model less prone to overfitting.

Confidence-masking: These defense mechanisms aim to reduce the amount of information that can be obtained from the output of a target model by perturbing [**52**, **68**] or limiting the dimensionality of the output [**15**, **110**, **119**]. Most confidence-masking approaches manipulate confidence values post-training. As a result, the output values of these models do not reliably represent the

"confidence" of the model. These approaches are built under the assumption that accurate prediction of confidence is not needed. However, many applications require accurate estimation of confidence. Moreover, if the accurate prediction of confidence is not required, then the trivial MI defense would be to only output the class label and avoid these confidence-masking defenses altogether.

CHAPTER 3

On the Difficulty of Membership Inference Attacks

In this chapter, we show that the way the MI attack performance has been reported is often misleading because they suffer from high false positive rate or false alarm rate (FAR) that has not been reported. FAR shows how often the attack model mislabel non-training samples (non-member) as training (member) ones. The high FAR makes MI attacks fundamentally impractical, which is particularly more significant for tasks such as membership inference where the majority of samples in reality belong to the negative (non-training) class. Moreover, we show that the current MI attack models can only identify the membership of misclassified samples with mediocre accuracy at best, which only constitute a very small portion of training samples.

We analyze several new features that have not been comprehensively explored for membership inference before, including distance to the decision boundary and gradient norms, and conclude that deep models' responses are mostly similar among train and non-train samples. We conduct several experiments on image classification tasks, including MNIST, CIFAR-10, CIFAR-100, and ImageNet, using various model architecture, including LeNet, AlexNet, ResNet, etc. We show that the current state-of-the-art MI attacks cannot achieve high accuracy and low FAR at the same time, even when the attacker is given several advantages. The source code is available at https://github.com/shrezaei/MI-Attack.

3.1. Introduction

There is an extensive recent literature on membership inference (MI) attacks on deep learning models that achieve high MI attack accuracy [**71**, **75**, **76**, **105**, **110**, **114**, **119**, **134**]. These MI attack models often use confidence values of the target model to infer the membership of an input sample. High MI attack accuracy is often justified by claiming that deep learning models are more confident

TABLE 3.1. Complete evaluation of CIFAR-100 with three different target models. Almost all papers report the third section that includes accuracy, precision, recall, and F1 score. The second section, including train and test accuracy of the target victim model, is missing in many papers in literature despite its usefulness in evaluating the generalization gap (and degree of overfitting). The last section that includes balanced accuracy and FAR has never been reported, but it is of paramount importance for understanding the performance of attack models in practice.

Dataset Model	Cifar-100 AlexNet	Cifar-100 ResNet	Cifar-100 DenseNet
Target Model Train Acc.	92.48%	95.80%	99.98%
Target Model Test Acc.	43.87%	74.14%	82.83%
Attack Acc.	82.62%	79.13%	87.74%
Attack Precision	91.90%	87.3%	86.97%
Attack Recall	86.92%	87.85%	98.29%
Attack F1	89.23%	87.45%	92.26%
Attack Bal. Acc.	74.02%	61.70%	66.65%
Attack FAR	38.89%	64.45%	65.00%

towards the training (member) samples than the samples they have not seen during training[1]. Consequently, MI attack accuracy is reported to be highly correlated to model's overfitting or generalization gap [**105**, **110**, **114**] because an overfitted model should perhaps behave even more confident towards training samples.

In this chapter, we show that the way the previous papers report the attack performance do not reveal how exactly these attacks perform in practice and can be misleading. First, many papers do not provide the train and test accuracy of the target victim models. Hence, it is not clear whether the target model is well-trained. We only find a handful of papers that report such metrics [**50**, **52**, **75**, **89**, **110**]. Even in these cases, one can clearly spot impractical target models where generalization gap is sometimes larger than 35% [**75**, **110**], 50% [**89**], or 80% [**50**]. Clearly, such extremely overfitted models have no practical use and the results on such models should not be generalized to well-trained models. Second, all papers we have examined limit their reporting to accuracy, precision, and recall. Such a reporting does not reveal the performance of attack models on negative samples (non-member), especially how many negative samples are misclassified as positive (false positive). For binary classification tasks, this is of crucial importance, especially when the negative class can significantly outnumber the positive class (e.g., all possible images

[1]In this dissertation, we use training samples, member samples, and positive samples interchangeably. Likewise, we also use non-training, test, non-member, and negative samples interchangeably.

vs. the limited number of images used to train an image classification model). A good practice, which is also common in other fields such as intrusion detection systems [**128**], is to report false positive rate (FPR) or false alarm rate (FAR) alongside the other metrics. A good attack model should have a low FAR.

To evaluate the feasibility of MI attack, we tried to reproduce the results in [**89**] for CIFAR-100 dataset, presented in Table 3.1[2]. Although the generalization gap of AlexNet is high ($\sim 48\%$), we keep the results for the sake of comparison. As it is shown, commonly used metrics, including accuracy, precision, and recall, do not reveal how attack models really perform on non-member samples. The high FAR of these attacks make them unreliable. Interestingly, even the attack on an extremely overfitted model such as AlexNet still suffers from high FAR.

In this chapter, we first elaborate on why previous reporting practices are misleading in membership inference research. Second, we provide a comprehensive evaluation of membership inference attacks on deep learning models. We give as much advantage as possible to an attacker and we show that a reliable MI attack with high accuracy and low FAR is hard to achieve. We show that the reason MI attacks often fail is not because attack models are trained poorly. The reason is that the statistical properties of the features used in MI attacks are not clearly different and distinguishable for training and non-training sample.

To provide an insight on why membership inference of some samples are possible, we separate datasets into two parts: correctly classified samples (by the target victim model) and misclassified samples. In general, we find that membership inference of correctly classified samples, independent of what dataset or model is used, is a more difficult task than the membership inference of misclassified samples. This is because deep learning models often behave similarly on train and non-train samples when they are correctly classified. This observation sheds light on the difficulty of membership inference on deep models.

Our contributions are summarized as follows:

- We show that attack accuracy, precision, and recall are not enough to show the performance of MI attacks, particularly on negative (non-member) samples. Instead, we should also report FAR (or other substitutes explained in Sec.3.2) and train/test accuracy of

[2]In literature, we have only found two papers with public source codes [**105**, **114**]. We run their implementation on CIFAR-10 and observed the same problem. They both suffer from high FAR, which has not been reported before.

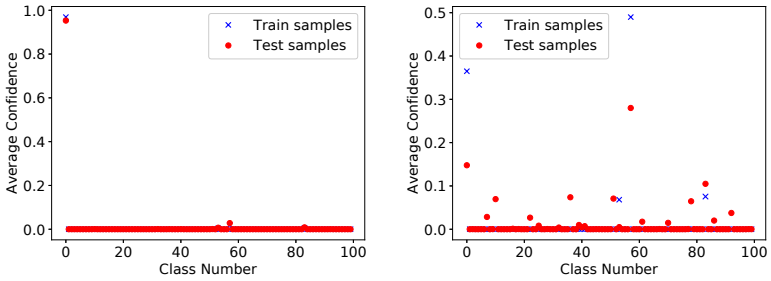
TABLE 3.2. Performance evaluation of an MI attack model when balancedness of the evaluation set is changed.

Balancedness	Attack Model	Positive (member) Class		Negative (non-member) Class		Balanced Accuracy	FAR
-	-	Precision	Recall	Precision	Recall	-	-
5:1	MI Attack	87.30%	87.85%	38.18%	35.55%	61.70%	64.45%
5:1	ZeroR	83.33%	100.0%	0.0%	0.0%	50.0%	100.0%
1:1	MI Attack	57.68%	87.42%	74.49%	35.42%	61.22%	64.82%
1:1	ZeroR	50.0%	100.0%	0.0%	0.0%	50.0%	100.0%
1:5	MI Attack	21.41%	87.82%	93.57%	35.73%	61.28%	64.42%
1:5	ZeroR	16.66%	100.0%	0.0%	0.0%	50.0%	100.0%

target models to better demonstrate the performance of MI attacks. Moreover, we study the performance of correctly classified samples and misclassified samples separately. We show that membership inference of correctly classified samples, to which the majority of training samples belong, is a very difficult task.

- We perform MI attack on various image datasets (including MNIST, CIFAR-10, CIFAR-100, and ImageNet), and models (LeNet, AlexNet, ResNet, DenseNet, InceptionV3, Xception, etc), some of which are studied for the first time in the MI context. We conduct experiments such that they give a lot more advantages to the attacker than in any previous work. Even in this case, we show that a meaningful membership inference attack with high accuracy and low FAR is often not achievable.
- In addition to confidence values of the target (victim) model, we extensively analyze and use other information available from the target model, including values from intermediate layers, the gradient w.r.t input, gradient w.r.t to model weights, and distance to the decision boundary. In some cases, these types of information slightly leak more membership status than confidence values, but they are still not sufficient for a reliable MI attack in practice. Surprisingly, all evidence suggests that deep models often behave similarly on train and non-train samples across all these metrics. The only considerable difference appears between correctly classified samples and misclassified samples, not between the train and non-train samples.

In summary, our experiments, including the reproduction of results in the literature, suggest membership inference of correctly classified samples, to which the majority of training samples



(a) Correctly labeled (b) Misclassified

FIGURE 3.1. Distribution of average confidence values for CIFAR 100 dataset (ResNet) (class 0)

belong, is a difficult task. Clearly more research is needed and we are hesitant to generalize our results to all scenarios, some of which are as follows:

- We mainly focus on vision tasks with high dimensional input. Membership inference for other tasks with low dimensional input may culminate in a different result.
- We do not extend our conclusion to generative models. High capacity generative models can often memorize training samples, which can be retrieved at inference time, as shown in [**8**]. However, there is no trivial method to retrieve memorized samples from a discriminative model even if it memorizes training samples.
- We do not extend the conclusion to any attack that can be launched during the training phase, such as data poisoning, model/training manipulation [**111**], etc. We only study membership inference on naturally trained models and natural datasets. Deep models may behave very differently if any unnatural manipulation appears during the training phase.
- In each dataset, there often exists outliers. The MI attack maybe more successful on these samples [**76**], whether they are classified correctly or not.

3.2. Better MI Attack Reporting

As discussed in Section I, the common approach of reporting accuracy, precision, and recall of MI attacks does not truly reveal the performance of them on negative (non-member) samples. By providing false alarm rate (FAR) in Table 3.1, we show that these attacks suffer from high false positive ratio. The reason why high false positive ratio does not significantly affect the reported

precision ($\frac{TP}{TP+FP}$) is due to the MI dataset imbalancedness. In a typical machine learning training, majority of samples are used for training and, consequently, the holdout (test) set is considerably smaller. For instance, the training:test (or member:non-member) ratio of CIFAR dataset is 5:1 and, consequently, the MI dataset to train/evaluate the MI attack model has the same imbalancedness ratio. As a result, the total number of FPs is small despite the high false positive ratio. This is clearly illustrated in Table 3.2 where the member:non-member ratio varies from 5:1, 1:1, to 1:5 and the precision on the positive class dropped from 87% to 21% for the same MI attack model. Note the MI attack model is the same for all experiments which was trained on a balanced dataset. We only change the balancedness of the evaluation set in Table 3.2.

We emphasize that reporting performance only on positive samples can be misleading. In Table 3.2, we report the precision and recall for both positive and negative classes. When the 5:1 ratio of balancedness is kept, the MI attack shows high precision and recall on positive samples, but low precision and recall on negative samples (which in general has not been reported). To stress this message, we also report ZeroR as a baseline. ZeroR is a classifier that always predicts the positive class (member). As shown in Table 3.2, ZeroR performance is high and close to the MI attack when **only** the precision and recall of positive samples are considered. This clearly shows that one should also report performance on negative samples as well.

The training:test ratio can significantly change the performance metrics of a same model, as shown in Table 3.2. So what is the right way to treat this ratio? Previous MI papers often keep the radio of 5:1, or in some cases 1:1 [**89**]. However, in practice, the ratio can reverse drastically: for example, a random sample, chosen from the distribution of all natural images, has a significantly higher probability of being a non-member sample than a member sample. However, there is no practical way to estimate the true value of this ratio. Furthermore, even if the ratio is known, due to the limited number of samples available for evaluation, it might not be practical to obtain more non-member samples to achieve the true ratio. As a result, in this chapter, we keep the ratio as 5:1, as in most existing papers in the evaluation set. Instead, we report performance metrics that are less sensitive to the balancedness ratio, i.e, balanced accuracy and FAR[3].

[3]A more elaborated argument for why FAR is important in tasks where one class hugely outnumber the other class is the base-rate fallcay [**3**].

There is another way to show the ineffectiveness of current MI attacks using a rather simple baseline. In [**66**], such a baseline is introduced, called naive attack. The idea is to predict a sample as member if it is correctly labeled by the target model, and to predict it as non-member if misclassified. Clearly, the FAR of naive attack is high because it classifies all correctly classified samples as members. However, this impractical attack can also achieve high accuracy on positive samples. Since this naive attack is obviously ineffective in practice, one can compare the accuracy gap between the naive attack and a MI attack to conclude the effectiveness of an MI attack. This approach has been used in [**66**]. We report the accuracy of this naive attack for completeness in Table 3.3, but we rely on accuracy/FAR pair to evaluate an attack's success.

Despite their low performance, MI attacks still outperform the random guess. To shed light on why MI attacks are effective on some samples, we report the behavior of target models on correctly classified and misclassified samples, separately. We show that deep learning models often behave similarly when they correctly label a sample, whether it is a training (member) or test (non-member) sample. In comparison, deep learning models demonstrate slightly different behavior on misclassified samples, which can be exploited by MI attack models. Hence, we believe that separating correctly classified and misclassified samples, and reporting the MI attack on them separately provides a better insight on how MI attacks work.

In summary, the way MI attack has been reported in literature does not provide a complete picture of their performance. Relying only on precision and recall of the positive class can present a delusion of successful attack, particularly when the imbalancedness issue is ignored. Instead, we should report performance on both positive and negative samples, i.e., adding FAR, or precision and recall on negative samples. Furthermore, simple baselines, such as ZeroR and naive attack, can be used for comparison. Last, separating correctly labeled and incorrectly labeled samples provide a better insight on how these MI attacks work. Therefore, in this chapter, we report balanced accuracy and FAR, and we also report the performance on correctly classified and misclassified samples, separately, when possible.

3.3. Methodology

Threat model and assumptions. In this chapter, we give an attacker as much advantage as possible to show that even in such cases membership inference cannot significantly outperform the baseline. We assume a white-box access to the model and unlimited number of queries. Moreover, we give the membership status of up to 80% of training samples and test samples to the attackers and we only ask the attack model to predict the membership inference of the remaining samples. Hence, the attack performance we report in this chapter is as good as or better than any proposed attack based on shadow models [**75**, **110**] or transferred or synthesized data [**71**, **105**] [4]. In addition to confidence values of the target model, which have been used extensively for membership inference attack in the past, we also study the output of intermediate layers, distance to the decision boundary and a set of gradient norms to better understand if deep models behave differently on training and test samples.

Confidence values. Confidence values, or the output of Softmax layer, have been widely used for membership inference [**71**, **75**, **76**, **105**, **110**, **114**, **119**, **134**]. Figure 3.1 shows the distribution of average confidence for correctly classified samples and misclassified samples of a ResNet model trained on CIFAR-100. As it is shown, misclassified samples often show different distribution for training samples and non-training samples. However, correctly classified samples often saturate the true class confidence value and zero out other confidence values. We show in Section 3.4 that membership inference attack models are often fail to considerably outperform a coin toss for correctly labeled samples.

Output of intermediate layers. In deep models, first layers often extract general and simple features that are not specific to training samples. As suggested in [**89**], the last layer and layers close to the last one contain more sample-specific information. Hence, we also examine the output of the fully connected layers before the Softmax for membership inference attacks.

Distance to the decision boundary. Some research focuses on understanding decision boundary of deep models [**56**, **83**] or geometry and space of deep models [**23**, **85**] to often understand the nature of adversarial examples or to improve robustness. In this chapter, we investigate whether

[4]Note that the use of these methods are beneficial when the dataset is small for training a MI attack because one can potentially multiply the MI training dataset by obtaining multiple shadow models. However, in our study, such methods are not necessary since MNIST, CIFAR, and ImageNet datasets have abundant samples.

Algorithm 1 FGM-based algorithm to find distance to the decision boundary

Require: S (maximum number of steps), $\mathbf{x}$ (input sample) and y (the sample ground-truth), f_{cls} (an interface to the target model which returns predicted class), f_{conf} (an interface to the target model which returns the confidence value of the predicted class), L (target model loss function), θ (confidence threshold indicating when the algorithm stops optimizing):

procedure DISTANCE_TO_BOUNDARY($\mathbf{x}$, f)
  $\mathbf{x}_0 = \mathbf{x}$
  for t from 0 to S **do**
    $\mathbf{x}_{t+1} = \mathbf{x}_t + \varepsilon \frac{\nabla_x L(x_t, y)}{||\nabla_x L(x_t, y)||_2}$
    if $f_{cls}(\mathbf{x}_{t+1}) \neq f_{cls}(\mathbf{x}_t)$ **then**
      while $|f_{conf}(\mathbf{x}_{t+1}) - f_{conf}(\mathbf{x}_t)| > \theta$ **do**
        $\mathbf{x}_m = \frac{\mathbf{x}_{t+1} + \mathbf{x}_t}{2}$
        if $f_{cls}(\mathbf{x}_m) = f_{cls}(\mathbf{x}_t)$ **then**
          $\mathbf{x}_t = \mathbf{x}_m$
        else if $f_{cls}(\mathbf{x}_m) = f_{cls}(\mathbf{x}_{t+1})$ **then**
          $\mathbf{x}_{t+1} = \mathbf{x}_m$
        else
          return *Error!*
        end if
      end whilereturn $||\mathbf{x}_0 - \mathbf{x}_t||_2$
    end if
  end forreturn *Optimization failed!*
end procedure

the distance to boundary is a distinguishable feature for membership inference. To find the distance to the decision boundary, we use FGM [**20**] optimization procedure to craft an image on the other side of the decision boundary (Algorithm 1). Then, we perform a binary search to find an instance for which the model's confidence for two classes are almost equal, that is, the difference between two confidences is smaller than a small threshold, similar to [**56**]. Finally, we obtain the L_2 distance between the original sample and the crafted samples as a measure of distance to the boundary.

Gradient norm. It has been shown that the gradient of loss with respect to model parameters, $\frac{\partial L}{\partial w}$, is often smaller for training samples than non-training samples [**89**] and it can be used for membership inference attack in federated learning scenario. In this chapter, we study the gradient of loss with respect to model parameters, $\frac{\partial L}{\partial w}$, and also the gradient of loss with respect to model input, $\frac{\partial L}{\partial x}$, in a non-federated learning setting. The large value for the former indicates that major re-tuning of model parameters is needed for that sample, and hence, it can be an indication of a non-member sample. The large value of the latter indicates that there are input samples with more confident output in the vicinity of that sample, and hence, it can be an indication of a non-training sample. Both $\frac{\partial L}{\partial w}$ and $\frac{\partial L}{\partial x}$ are extremely high dimensional. Thus, we adopt the seven norms used

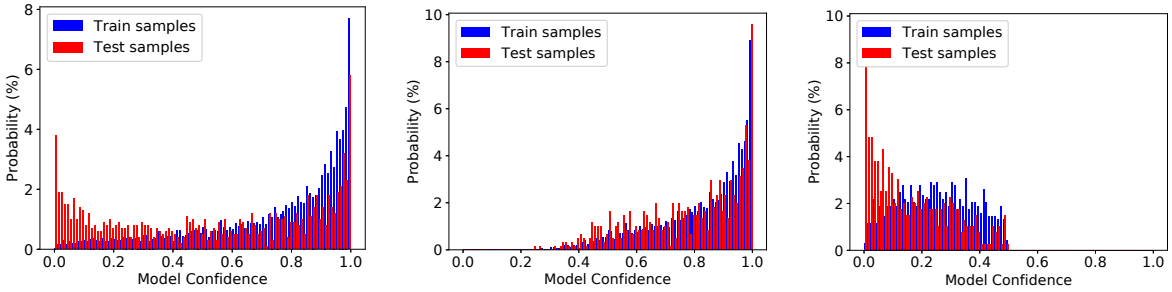
TABLE 3.3. Accuracy of various datasets, target models, and MI attack models

Dataset(Model)	Train	Test	Attack Acc	Naive Attack
MNIST (LeNet)	99.74%	99.05%	50.04%	50.07%
C-10 (AlexNet)	91.80%	77.22%	57.43%	57.87%
C-10 (ResNet)	99.43%	93.89%	54.11%	52.56%
C-10 (DenseNet)	100.00%	95.46%	56.05%	52.45%
C-100 (AlexNet)	92.48%	43.87%	74.02%	70.23%
C-100 (ResNet)	95.80%	74.14%	61.70%	65.68%
C-100 (DenseNet)	99.98%	82.83%	66.65%	71.97%
I (InceptionV3)	87.91%	79.98%	50.03%	54.12%
I (Xception)	87.77%	80.70%	51.18%	53.37%

TABLE 3.4. Membership attack results based on confidence values

Dataset(Model)	-	Attack Accuracy	Attack FAR	Train Confidence	Test Confidence
MNIST	All data	50.04% ± 0.11	50.01% ± 47.81	99.61 ± 4.57	98.90 ± 9.14
	Correctly classified	49.98% ± 0.01	50.21% ± 48.38	99.81 ± 2.04	99.75 ± 2.51
	Misclassified	62.30% ± 17.93	40.83% ± 35.44	77.61 ± 16.05	87.09 ± 15.93
CIFAR-10 (AlexNet)	All data	57.43% ± 2.59	71.4% ± 9.29	85.26 ± 23.08	72.56 ± 34.75
	Correctly classified	50.54% ± 0.82	91.89% ± 5.43	90.77 ± 13.96	89.57 ± 15.52
	Misclassified	52.16% ± 2.57	5.45% ± 5.62	60.17 ± 16.68	66.89 ± 19.68
CIFAR-10 (ResNet)	All data	54.11% ± 1.92	86.60% ± 6.49	98.66 ± 7.03	92.74 ± 22.02
	Correctly classified	51.81% ± 0.84	91.63% ± 4.05	99.08 ± 4.33	97.98 ± 7.33
	Misclassified	60.83% ± 19.74	0.0% ± 0.0	66.23 ± 15.66	79.71 ± 18.61
CIFAR-10 (DenseNet)	All data	56.05% ± 3.88	77.05% ± 26.5	99.97 ± 0.49	94.78 ± 19.33
	Correctly classified	54.0% ± 2.64	81.15% ± 27.45	99.97 ± 0.29	98.77 ± 5.64
	Misclassified	100.0% ± 0.0	0.0% ± 0.0	-	82.83 ± 17.63
CIFAR-100 (AlexNet)	All data	74.02% ± 8.27	38.89% ± 16.69	84.59 ± 24.31	40.58 ± 42.09
	Correctly classified	55.13% ± 7.39	83.12% ± 14.94	89.9 ± 15.82	85.05 ± 19.95
	Misclassified	55.11% ± 11.28	2.01% ± 4.33	50.29 ± 19.45	60.96 ± 23.81
CIFAR-100 (ResNet)	All data	61.7% ± 6.55	64.45% ± 14.81	91.14 ± 18.44	70.19 ± 38.10
	Correctly classified	53.96% ± 5.25	83.7% ± 11.01	94.15 ± 11.43	90.88 ± 15.74
	Misclassified	54.37% ± 16.0	2.66% ± 8.08	57.82 ± 17.69	64.3 ± 21.71
CIFAR-100 (DenseNet)	All data	66.65% ± 8.36	65.0% ± 18.16	99.95 ± 1.07	79.99 ± 34.70
	Correctly classified	60.58% ± 5.98	77.17% ± 15.28	99.96 ± 0.5	94.67 ± 13.06
	Misclassified	98.25% ± 9.2	0.0% ± 0.0	65.01% ± 11.52	67.25% ± 24.61
ImageNet (InceptionV3)	All data	50.03% ± 0.28	45.96% ± 44.27	76.03 ± 25.52	68.62 ± 29.43
	Correctly classified	50.03% ± 0.31	45.46% ± 47.86	83.99 ± 15.55	81.85 ± 16.3
	Misclassified	51.5% ± 8.57	51.69% ± 44.69	13.57 ± 11.44	10.8 ± 9.38
ImageNet (Xception)	All data	51.18% ± 3.56	50.81% ± 44.33	73.56 ± 25.56	66.92 ± 28.58
	Correctly classified	50.72% ± 3.57	50.42% ± 48.80	81.24 ± 16.81	79.08 ± 17.18
	Misclassified	51.95% ± 19.29	52.15% ± 44.08	13.48 ± 10.94	11.27 ± 8.94

in [**91**], originally used for analysis of deep model's uncertainty, namely L_1, L_2, absolute minimum, L_∞, mean, Skewness, and Kurtosis.



(a) All samples (b) Correctly classified samples (c) Misclassified samples

FIGURE 3.2. Distribution of the confidence of the true class for CIFAR-10 (AlexNet) class #3. Although the distribution of all samples seems to be distinguishable, it is only the manifestation of accuracy gap between train and test which is exploited by naive and other attack. When correctly classified and misclassified samples are depicted separately, sample difference of train and test sets is vividly less distinguishable.

3.4. Experimental Evidence

Target models and datasets. We launch MI attack on various CNN-based models on four image classification tasks: MNIST, CIFAR-10 (C-10), CIFAR-100 (C-100), ImageNet (I). For MNIST, we train a LeNet model. For CIFAR-10 and CIFAR-100, we use a set of trained models used in [**89**][5], including AlexNet, ResNet [**33**], and DenseNet [**46**]. For ImageNet, we use pre-trained InceptionV3 [**116**] and Xception [**14**] models without any re-training from Keras package[6]. We deliberately choose to launch MI attacks on models trained by others for two reasons: 1) to make it easier for readers to compare the attack with papers that use the same trained models, such as [**89**], and 2) to avoid any intentional/unintentional model training practices that bias our results. To the best of our knowledge, the models trained on CIFAR-10 and CIFAR-100, used in [**89**], has not adopted the early stopping technique during the training phase. In supplementary material, we train all models on CIFAR-10 and CIFAR-100 from scratch and we show that using early stopping techniques can make MI attacks even harder.

MI attack models. In most cases, we fit three types of attack models: FC neural network (NN), random forest (RF), and XGBoost. For the NN model, we train a model with 2 hidden layers of size 128 and 64. For RF and XGBoost, we perform a random search over a large set of hyper-parameters.

[5]https://github.com/bearpaw/pytorch-classification
[6]https://keras.io/api/applications/

TABLE 3.5. MI attack performance based on the output of intermediate layers. C and I represents CIFAR and ImageNet, respectively

Dataset (Model)	Layer	All Data		Correctly Classified		Misclassified	
-	-	Accuracy	FAR	Accuracy	FAR	Accuracy	FAR
MNIST	-1	47.65% ± 2.6	59.55% ± 17.95	47.58% ± 2.6	59.69% ± 17.98	55.45% ± 20.49	43.33% ± 40.1
MNIST	-2	47.96% ± 3.01	60.77% ± 11.33	47.99% ± 2.93	60.65% ± 11.28	47.42% ± 25.3	69.17% ± 41.51
C-10 (AlexNet)	-1	55.38% ± 2.18	47.45% ± 9.55	51.34% ± 2.46	58.47% ± 12.79	55.47% ± 3.69	14.49% ± 9.95
C-10 (ResNet)	-1	53.89% ± 2.62	45.1% ± 16.25	52.74% ± 2.2	47.63% ± 17.43	60.75% ± 20.8	5.51% ± 7.21
C-10 (DenseNet)	-1	54.4% ± 3.63	48.7% ± 6.99	53.12% ± 3.03	51.28% ± 7.54	100.0% ± 0.0	0.0% ± 0.0
C-100 (AlexNet)	-1	61.34% ± 7.91	38.95% ± 12.48	55.39% ± 10.45	52.46% ± 19.99	57.65% ± 12.98	29.32% ± 15.65
C-100 (ResNet)	-1	53.81% ± 7.24	47.2% ± 16.65	51.46% ± 7.72	52.85% ± 18.99	52.39% ± 23.72	31.2% ± 27.8
C-100 (DenseNet)	-1	64.76% ± 9.99	37.35% ± 14.67	61.7% ± 9.4	43.48% ± 14.99	92.02% ± 20.78	38.73% ± 31.23
I (InceptionV3)	-1	58.37% ± 7.8	40.2% ± 15.68	57.83% ± 9.4	42.58% ± 19.0	55.86% ± 17.26	36.25% ± 35.4
I (Xception)	-1	57.44% ± 8.59	42.3% ± 17.08	57.35% ± 9.36	43.92% ± 18.61	55.52% ± 18.65	40.38% ± 39.49

For ImageNet, we conduct experiments with only 100 classes out of 1000 classes due to the limited computational and time budget. Moreover, we perform random under-sampling of member class and oversampling of non-member class to balance the training dataset on separate experiments. In this chapter, we only report the seemingly best MI attack accuracy we achieve over all attack models and hyper-parameters. It is worth mentioning that by under-sampling or over-sampling of training data, or by chainging the decision threshold of the MI attack, we could decrease FAR at the cost of accuracy. In any case, we could not find a good MI attack model with relatively high accuracy and low FAR. The input of attack models varies which is described in each following subsection. The accuracy of target and MI attack models are shown in Table 3.3. Note that even the best MI attack models can barely outperform the naive attack. In the following sections, we show that separating correctly classified and misclassified samples, and reporting accuracy and FAR gives more insight on the performance of MI attacks.

3.4.1. Confidence Values. Confidence values have been extensively used for MI attacks. As shown in Table 3.4, the MI attacks are more successful on inferring membership of misclassified samples, which often consist a small portion of training samples. Interestingly, the state-of-the-art target models on ImageNet does not even leak membership status of misclassified samples. The best attack performance on correctly classified samples is observed on DenseNet model trained on CIFAR-100, which is 60.58% that still suffers from very high FAR (77.17%). Note that the MI attacks on misclassified samples of DenseNet model may not be meaningful because there are no

TABLE 3.6. Performance of attack models based on the distance to the decision boundary.

Dataset (Model)	Correctly Classified				Misclassified			
-	MI Attack		Average Distance		MI Attack		Average Distance	
-	Accuracy	FAR	Train	Test	Accuracy	FAR	Train	Test
MNIST	49.86% ± 6.0	52.97% ± 8.2	1.372 ± 0.40	1.371 ± 0.45	49.81% ± 9.0	48.3% ± 24.0	.0103 ± .0016	.0108 ± .0024
C-10 (AlexNet)	52.36% ± 4.1	53.19% ± 17	51.59 ± 15.2	50.31 ± 14.7	52.42% ± 4.9	49.53% ± 7.1	48.84 ± 15.8	50.5 ± 16.0
C-10 (ResNet)	51.05% ± 5.5	49.57% ± 8.2	51.3 ± 15.2	50.3 ± 14.7	46.05% ± 4.8	60.49% ± 8.6	51.2 ± 15.2	50.5 ± 16.0
C-10 (DenseNet)	50.14% ± 5.9	50.17% ± 7.8	51.4 ± 15.3	50.2 ± 14.6	100.0% ± 0.0	0.0% ± 0.0	-	51.2 ± 16.4
C-100 (AlexNet)	50.45% ± 9.1	47.77% ± 6.3	53.6 ± 16.3	54.3 ± 15.1	53.78% ± 11.6	45.94% ± 17.7	48.2 ± 16.5	52.6 ± 16.6
C-100 (ResNet)	49.08% ± 6.4	49.28% ± 6.8	52.6 ± 16.4	53.3 ± 15.9	47.24% ± 12.4	52.05% ± 21.4	51.9 ± 16.6	51.6 ± 16.8
C-100 (DenseNet)	49.74% ± 6.41	48.57% ± 7.04	52.8 ± 16.3	53.3 ± 16.0	98.25% ± 9.2	0.0% ± 0.0	51.5 ± 10.4	56.8 ± 16.5
I (InceptionV3)	51.75% ± 8.2	50.43% ± 26.3	212.4 ± 64.8	218.6 ± 63.0	44.95% ± 11.9	46.31% ± 18.8	215.8 ± 67.0	215.1 ± 61.5
I (Xception)	53.01% ± 9.5	46.1% ± 23.7	214.5 ± 64.6	216.2 ± 61.9	49.29% ± 13.5	44.88% ± 17.1	212.4 ± 67.0	214.7 ± 64.0

misclassified samples in the training set of CIFAR-10 and there are only 10 misclassified samples in the training set of CIFAR-100.

To better understand why MI attacks fail, it is better to investigate the average confidence value of target models, shown in the fifth and sixth columns of Table 3.4. As shown, the average confidence values of train samples (members) are often close to the test samples (non-members). MI attacks are only partially successful when average confidence values between train and test samples are far apart and the standard deviation is low. As shown in Figure 3.2, by separating correctly classified samples and misclassified sample, we can observe that sample distribution is very close, particularly for correctly classifies samples.

3.4.2. Output of Intermediate Layers. The attack accuracy based on the output of intermediate layers are shown in Table 3.5. We only launch an attack on the output of FC or flattened layers. The layer column shows the number of layers we go back from the Softmax layer. Only the MI attack on ImageNet is more successful in terms of both accuracy and FAR. Nevertheless, the FAR is still high and accuracy is not considerably better than a random guess.

3.4.3. Distance to the Boundary. Since the distance to the decision boundary is one-dimensional, we only fit a logistic regression to the samples. As shown in Table 3.6, all MI attacks fail. By looking at the average distance and their standard deviation, it is evident that the distance to boundary is not a distinguishable feature for membership inference. Note that finding the exact distance to the boundary is a computationally heavy task for high dimensional data. What is clear

TABLE 3.7. Performance of attack models based on gradient norms with respect to input (x) and weights (w)

Dataset (Model)	Correctly Classified				Misclassified			
-	Grad w.r.t w		Grad w.r.t x		Grad w.r.t w		Grad w.r.t x	
-	Accuracy	FAR	Accuracy	FAR	Accuracy	FAR	Accuracy	FAR
MNIST	52.06% ± 3.7	42.75% ± 28.5	53.19% ± 3.5	34.5% ± 23.7	57.84% ± 26.8	38.48% ± 20.0	52.02% ± 22.2	41.79% ± 16.8
C-10 (AlexNet)	51.94% ± 4.1	41.38% ± 7.9	51.81% ± 4.4	39.38% ± 10.3	61.36% ± 6.6	39.27% ± 8.2	58.69% ± 5.4	35.92% ± 7.2
C-10 (ResNet)	52.75% ± 3.0	21.75% ± 12.2	49.88% ± 3.8	36.25% ± 13.9	50.85% ± 10.6	66.49% ± 26.8	50.26% ± 16.1	50.36% ± 31.3
C-10 (DenseNet)	54.88% ± 2.6	16.38% ± 5.4	54.37% ± 3.1	13.25% ± 8.3	100.0% ± 0.0	0.0% ± 0.0	100.0% ± 0.0	0.0% ± 0.0
C-100 (AlexNet)	57.71% ± 9.6	31.45% ± 14.6	56.65% ± 11.5	31.46% ± 9.9	67.32% ± 11.3	34.51% ± 19.9	59.12% ± 13.8	38.52% ± 23.5
C-100 (ResNet)	52.98% ± 6.2	37.72% ± 21.1	54.31% ± 6.9	29.71% ± 11.3	55.4% ± 17.9	54.87% ± 34.6	61.38% ± 17.3	36.58% ± 27.1
C-100 (DenseNet)	69.7% ± 7.8	8.03% ± 6.2	70.22% ± 7.4	6.12% ± 3.6	98.25% ± 9.2	0.0% ± 0.0	98.25% ± 9.2	0.0 ± 0.0
I (InceptionV3)	49.25% ± 11.7	42.5% ± 19.0	52.1% ± 9.6	48.91% ± 14.7	58.48% ± 15.3	30.8% ± 20.3	50.55% ± 16.5	42.75% ± 17.3
I (Xception)	53.48% ± 9.0	36.53% ± 19.3	53.26% ± 11.4	48.05% ± 14.2	47.89% ± 18.2	43.69% ± 17.7	49.65% ± 15.6	41.25% ± 17.4

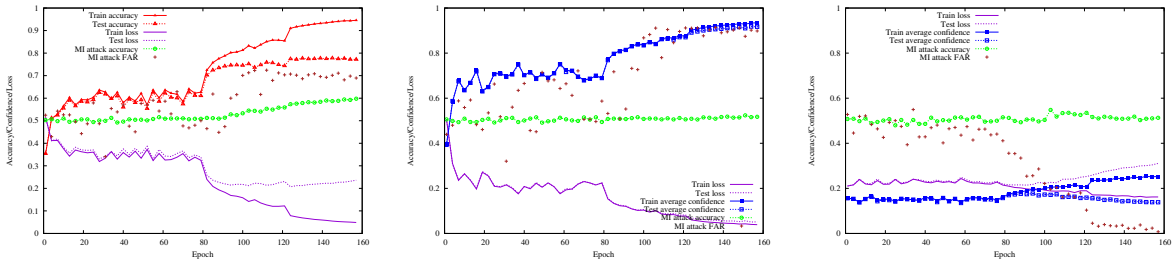
in Table 3.6 is that the FGM-based approximation of distance to boundary does not provide a distinguishable feature. A more accurate approach may reveal more membership information.

3.4.4. Gradient Norm. Table 3.7 shows the performance of MI attack models based on gradient norms. For each case, we fit a logistic regression to the 7 norms introduced in Section 3.3. Except for ImageNet, all MI attacks on correctly classified samples achieve almost the same accuracy while having a lower FAR, in comparison with MI attacks based on confidence values (Table 3.4). Gradient information of misclassified samples leak less membership information than confidence values.

3.4.5. Effect of Overfitting. In this section, we analyze the effect of overfitting on membership inference. Note that extremely overfitted models have no practical use in reality. The goal of this section is to show that the overfitted models may behave differently than well-trained models. As a result, researchers should avoid using overfitted models for MI attack and generalize them to well-trained practical models. To show the effect of overfitting, we train AlexNet, ResNet, and DenseNet models for a fixed amount of epochs on CIFAR-10 and CIFAR-100. We use the same training parameters as used by Wei Yang[7]. We launch MI attack based on confidence values on various epochs during the training. The results are shown in Figure 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8.

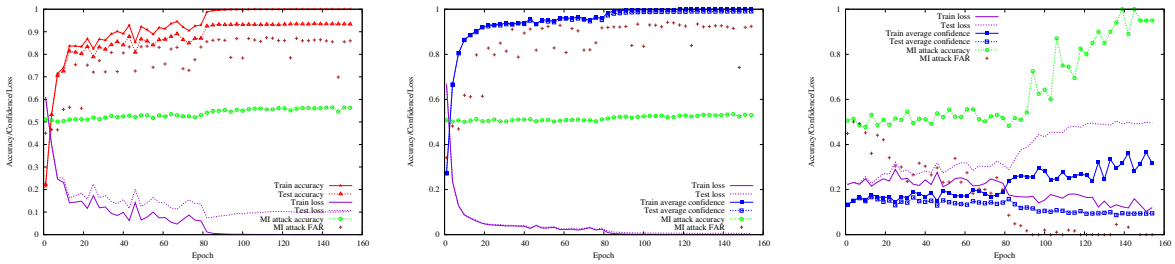
As shown in Figure 3.3(a), the model starts overfitting around epoch 80, when the loss function for the test set stops improving. It is clear that all MI attacks before the epoch 80 suffers from low accuracy (almost similar to random guess) and high FAR, on both correctly classified samples

[7]https://github.com/bearpaw/pytorch-classification



(a) All dataset (b) Correctly classified samples (c) Mis-classified samples

FIGURE 3.3. Training progress and MI attack on CIFAR-10 for AlexNet model



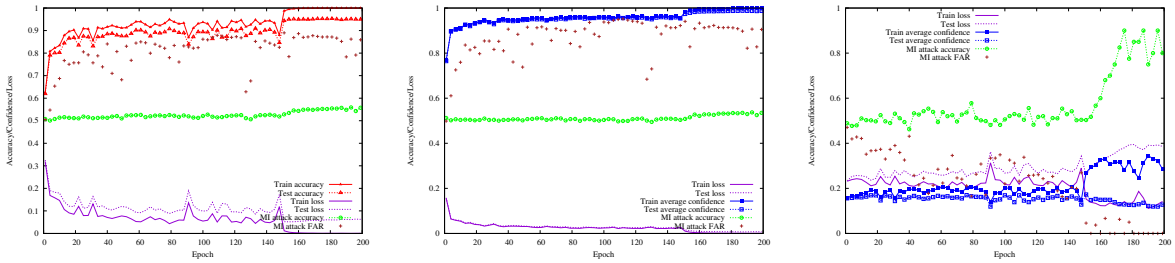
(a) All dataset (b) Correctly classified samples (c) Mis-classified samples

FIGURE 3.4. Training progress and MI attack on CIFAR-10 for ResNet model

(Figure 3.3(b)) and misclassified samples (Figure 3.3(c)). On the other hand, as the target model start overfitting, the performance of MI attacks increases over misclassified samples (Figure 3.3(c)). This phenomenon is more evident on other models, such as ResNet (Figure 3.4(c)). However, overfitting does not significantly improve MI attacks on correctly classified samples. Note than one should consider the number of misclassified training (member) samples to evaluate if the high performance MI attacks on misclassified samples have any real impact. The reason is that as target models overfit, the number of misclassified training samples approaches zero. In most cases, after epoch 160, there are only a handful of misclassified training samples. In other words, even a successful MI attack on an overfitted model only reveals the membership status of a handful of training samples. In any case, adopting a simple technique, such as early stopping, can even eliminate such as possibility.

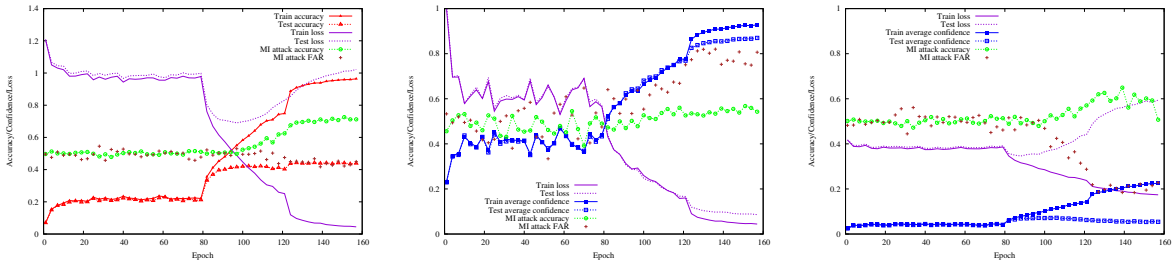
3.5. Conclusion

In this chapter, we show that commonly-used MI attacks based on confidence values of deep models are not as reliable as it has been reported before. By reporting accuracy and FAR together,



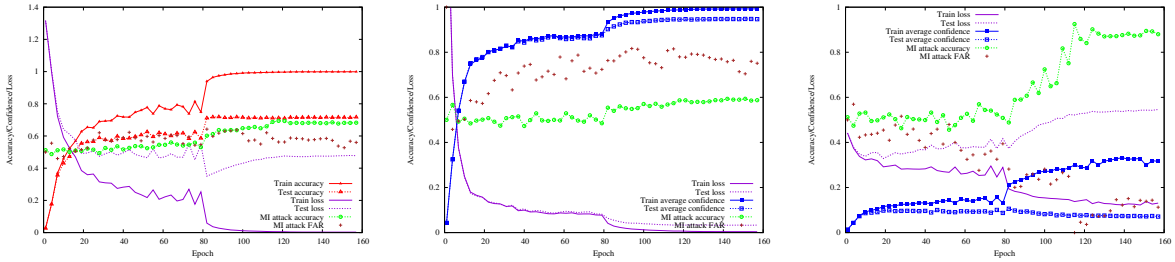
(a) All dataset (b) Correctly classified samples (c) Mis-classified samples

FIGURE 3.5. Training progress and MI attack on CIFAR-10 for DenseNet model



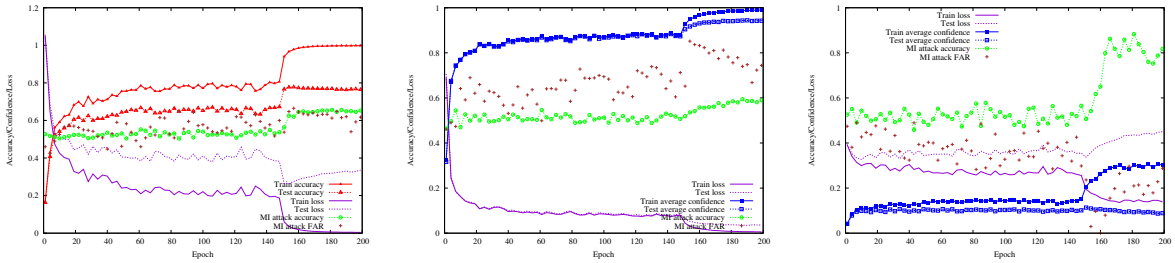
(a) All dataset (b) Correctly classified samples (c) Mis-classified samples

FIGURE 3.6. Training progress and MI attack on CIFAR-100 for AlexNet model



(a) All dataset (b) Correctly classified samples (c) Mis-classified samples

FIGURE 3.7. Training progress and MI attack on CIFAR-100 for ResNet model



(a) All dataset (b) Correctly classified samples (c) Mis-classified samples

FIGURE 3.8. Training progress and MI attack on CIFAR-100 for DenseNet model

we show that MI attacks that achieve higher accuracy suffers from higher FAR. Previous MI attacks extensively rely on confidence values of the target model for membership inference. We show that such a membership inference in general is a difficult task because the distribution of confidence values are similar for member and non-member samples. We report the attack accuracy on correctly classified samples and misclassified samples separately to show that misclassified samples slightly leak more information for membership inference. Additionally, we analyze several other features of input samples, including the distance to the decision boundary and gradient norms, to further illustrate the difficult nature of reliable membership inference attack on deep models. In summary, we find that naturally trained deep models often behave similarly across training and test samples and, hence, an accurate membership inference attack on all training samples in practice is a difficult and inaccurate task under current attack models unless a new revolutionary approach is introduced.

CHAPTER 4

An Efficient Subpopulation-based Membership Inference Attack

State-of-the-art membership inference attacks have shown to achieve good accuracy which poses a great privacy threat. However, majority of SOTA attacks require training dozens to hundreds of shadow models to accurately infer membership. This huge computation cost raises questions about practicality of these attacks on deep models.

In this chapter, we propose a fundamentally different MI attack that achieves the same accuracy as SOTA while significantly reducing the shadow training computational overhead. Here, instead of comparing victim model's confidence on the target sample versus the average confidence of typical models, which requires training numerous shadow models, we compare the victim model's confidence on the target sample versus the victim model's confidence on similar samples from the same subpopulation as the target sample. Hence, we obviate the need to train multiple shadow models. In other words, we calculate the expected value of subpopulation confidence rather than the expected value of shadow models' confidence. However, in practice, the attacker may not have access to samples from the same subpopulation. To tackle this issue, we develop a BiGAN-like architecture to train a generator that craft samples from the subpopulation of a given image. In other words, our attack only needs training a single generator model once and then it can be used for even unseen samples.

4.1. Background

Membership inference attacks aim to distinguish training samples from non-training samples. Training samples are often called *member* samples and non-training samples are called *nonmember* samples. Let x, y, $Y_v(.)$, and $Y_s(.)$ be the target sample, target label, victim model's output, and shadow model's output, respectively. The list of symbols are presented in Table 4.1. Now, let $s(Y, (x, y))$ denote the membership score, where a higher score indicates the higher probability of a sample being member. MI attacks aim to introduce an accurate membership score function.

TABLE 4.1. List of symbols

Symbol	Explanation
x	Target (input) sample
X_{sub}	Samples from the same subpopulation as x
y	Target label
$Y_v(.)$	Victim model's output
$Y_s(.)$	Shadow model's output
$Y_v(.)$	Encoder part of the victim model (output of the layer before the softmax)
$s(Y,(x,y))$	A function returning membership score
D	Shadow model training dataset
$A(c)$	A Randomized training algorithm that samples from the distribution of trained shadow models with a condition c
$G(x)$	A function sampling from the distribution of samples that belong to the same subpopulation as x

This can be the confidence output of the victim model, loss values, or output of an MI attack model [**126**].

In the literature, there are two types of MI attacks: 1) with sample calibration, and 2) without sample calibration. The calibration process modifies the membership score of an MI attack such that it takes the target sample's difficulty into account [**126**]. The former attacks do not take the difficulty of the target sample into consideration. They essentially compare the victim model's response on the target sample versus an average response to infer membership (e.g. via shadow models). The intuition is that models are often more confident on their training samples. For example, [**134**] uses loss function and [**110**] uses confidence value as a score function. However, it is shown that this leads to poor attack performance and high false positive mainly because well-generalized models output high confidence on majority of nonmember samples as well [**100**].

The second category of attacks, achieving state-of-the-art MI performance, use some form of sample calibration to distinguish between hard-to-predict member samples from easy-to-predict nonmember samples [**126**]. For instance, [**103**] calibrates the score using the average score both when the target sample is in training data and when it is not, as follows:

$$s_{Sab}(Y_v,(x,y)) = s(Y_v,(x,y)) - \frac{\mathbb{E}_{Y_s \leftarrow A(x \in D)}[s(Y_s,(x,y))] + \mathbb{E}_{Y_s \leftarrow A(x \notin D)}[s(Y_s,(x,y))]}{2}, \tag{4.1}$$

where $A(c)$ is a randomized training algorithm that samples from the distribution of trained shadow models following a specified condition c. D is the shadow model training dataset. The intuition is that if a sample is easy-to-predict, then the calibration term is also large. So, the total membership score is small. However, training overhead of this attack is large particularly if all target samples are not known during shadow training. In that case, each time a new sample is targeted, $\mathbb{E}_{Y \leftarrow A(x \in D)}$ should be calculated from scratch by training new shadow models. Watson attack [**126**] tackles this issue by estimating the calibration only on shadow models trained without the target sample. Simply put, their membership score is

$$s_{Watson}(Y_v, (x, y)) = s(Y_v, (x, y)) - \mathbb{E}_{Y_s \leftarrow A(x \notin D)}[s(Y_s, (x, y))]. \tag{4.2}$$

For the base membership score, $s(.)$, they [**126**] explored confidence, loss, and gradient norm and showed that loss is slightly outperform others.

Threat model: In this chapter, we assume that the attacker has a dataset with similar distribution as of victim's dataset. This assumption is needed for shadows-based attacks as well. Moreover, our original attack needs white-box access to the victim model. However, we show that our attack still performs well even when the attacker does not have a white-box access to the victim model.

4.2. Our Subpopulation-based Attack Overview

All MI attacks discussed above require training numerous shadow models to accurately estimate the expectation in the calibration term and thus are computationally heavy. In contrast, we propose a fundamentally different approach by running the expectation over similar samples rather than similar models. In other words, our attack estimates whether the victim model's loss on the target sample is significantly smaller than the victim models' loss on samples from the same subpopulation whose loss should be similar. We define subpopulation-based score as:

$$s_{ours}(Y_v, (x, y)) = s(Y_v, (x, y)) - \mathbb{E}_{x' \leftarrow G(x)}[s(Y_v, (x', y))], \tag{4.3}$$

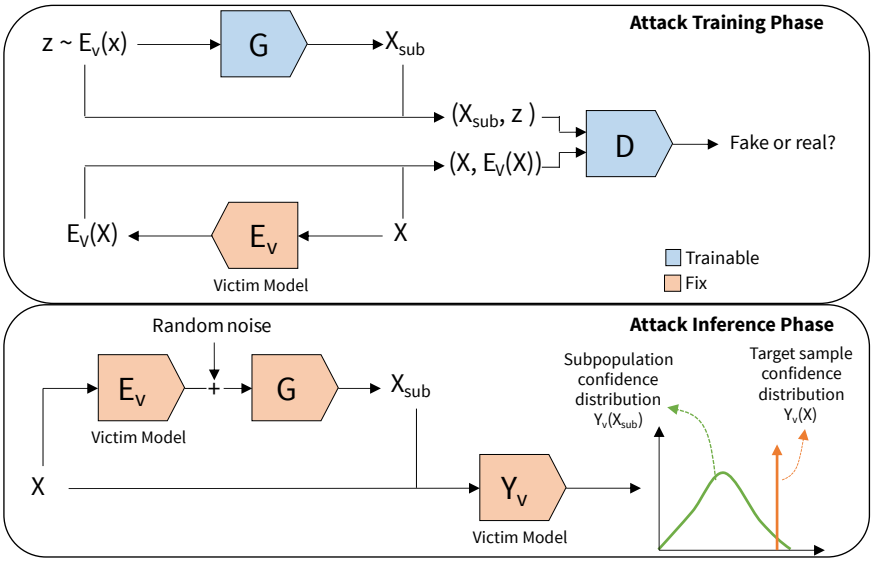


FIGURE 4.1. Subpopulation-based membership inference attack overview.

where $G(x)$ is a function sampling from the distribution of samples that belong to the same subpopulation as x. The benefit of our subpopulation-based approach is that it obviates the need to train numerous shadow models. However, obtaining images from the same subpopulation is not trivial. How to define a subpopulation and to train $G(x)$ is covered in the next section.

4.2.1. Crafting Subpopulations. Latent representation of deep learning models have been extensively used as a means of semantic closeness in various applications [**121**,**131**,**138**]. We define a subpopulation of x, X_{sub}, such that their samples are close to x in a latent space. Formally, a subpopulation is defined as follows

$$X_{sub} = \{x' : Dist(E_v(x'), E_v(x)) < \epsilon\}, \tag{4.4}$$

where E_v is the latent representation of the victim model, and $Dist(.)$ is a distance metric. Here, we consider the output of the last fully connected layer before the softmax as the latent representation. When abundant samples are available to the attacker, she can easily use X_{sub} to launch the subpopulation-based attack as explained in Section 4.2. The downside is that the attacker needs to have multiple extra samples for each target sample.

To solve this challenge, we propose a modified version of the BiGAN architecture [**19**] to train a generator model, G. The generator learns the mapping from the latent space to the input space which can be used to obtain X_{sub} for MI attacks. However, the original BiGAN architecture cannot be used directly for two reasons: 1) the encoder in BiGAN is trainable while the encoder in our case is the fixed victim model which we cannot change. 2) the original BiGAN forces the encoder to map the input to a uniformly distributed latent space. However, here, the victim model is fixed and there is no guarantee that the latent space follows a uniform or any known distribution.

Hence, to address these issues, we make the following changes:

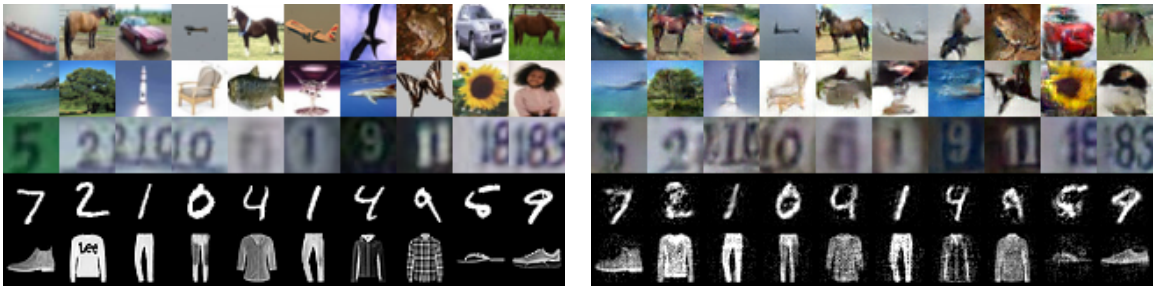
- We replace the encoder with E_v and block the back-propagation from training E_v.
- Instead of sampling from a uniform distribution, we obtain the latent representation of all samples in attacker's dataset ($\neq$ victim training dataset) from which we sample as an input for the generator.

See Figure 4.1 for an overview illustration of our MI attack.

4.3. Experimental Setup

We conduct experiments on multiple image classification benchmarks: MNIST [**64**], FMNIST [**130**], SVHN [**90**], and CIFAR10/CIFAR100 [**61**]. We divide the train set of these datasets into two parts: victim training dataset and attacker training dataset. The test set is only used for attack evaluation. For MNIST and FMNIST, we choose multi-layer perceptron (MLP) with 4 hidden layers as the victim model. For SHVN, we choose LeNet. For CIFAR10/100 we choose both LeNet and ResNet20. Details of the victim models, generators and discriminators are presented in 4.3.1. We train all models using SGD with a learning rate of 0.1. We reduce the learning rate by a factor of 10 at epoch 50 and 75. The performance of the victim models are shown in Table 4.2.

We compare our attack with multiple SOTA MI attacks. Unless specified, we follow the same experimental settings to train attack models as suggested in their original paper. For Shokri attack [**110**], we train 100 shadow models for all datasets. Yeom attack [**134**] requires the knowledge about the average training loss to set the threshold which we assume it is known to the attacker for this particular attack. For attack with calibration, including ours, Watson [**126**], and Sablayrolles [**103**], we use the loss function as the base membership score before calibration. We train 30 shadow models



(a) Original images

(b) Crafted images from the corresponding subpoplation

FIGURE 4.2. Original images (left) and crafted images (right) using BiGAN generator. The datasets/models from top to bottom rows are as follows: CIFAR10/ResNet20, CIFAR100/ResNet20, SVHN/LeNet, MNIST/MLP, and FMNIST/MLP.

for these two attacks as suggested in [**103**]. We also compare our attack with Jayaraman [**51**] because it is similar to our attack in an interesting way: if we define X_{sub} as target samples with random noise, the two attacks would be essentially similar. However, we show that directly adding random noise to input space leads to poor performance. For jayaraman attack [**51**], we first use $T = 100$ and $\sigma = 0.01$ as suggested in the original paper. Due to the poor performance, we perform random hyper-parameter tuning on σ, 10 times and we report the best result.

For our attack, we consider two scenarios: 1) when a large amount of samples is available to find subpopulation, 2) when training a generator is needed. For the first scenario, we only consider SVHN dataset because it is the only dataset with abundant extra data which is often ignored. We use *cosine similarity* in latent space to find subpopulations. We find no significant difference when using L2 distance. For the second scenario, we first train a generator in BiGAN-like architecture the details of which is presented in 4.3.1. We get the latent representation of each sample using E_V. Then, we add small random Gaussian noise, ϵ, with zero mean and standard deviation σ proportional to each activation value. In other words, for the latent representation of x, denoted as $L = E_V(x)$, noisy latent representation is obtained by $L'_i = L_i + |L_i|\epsilon$. The purpose of the scale factor is to make sure that activations with small values remain small, otherwise it may change the subpopulation or even class of the image. We set $\sigma = 0.05$ for mnist and fmnist, and 0.5 for other datasets. Finally, we feed the noisy latent representation to the generator to craft a subpopulation for each sample. We craft 30 images per sample.

Figure 4.2 illustrates some examples of the target images and their corresponding subpopulation images crafted by our BiGAN generator. Generated images are often similar to the crafted versions with small difference in color, orientation, pattern, background, and texture. Clearly, crafted samples belong to the same subpopulation as the original images. Hence, a victim model output confidence should not be significantly greater than images from the same subpopulation, otherwise it is an indication that the target sample was used during training.

4.3.1. Model Architecture Details. The MLP model for MNIST and FMNIST consists of 5 hidden layers of size 1024, 512, 256, 128, and 100 with LeakyReLU activation followed by a softmax layer. The last layer before the softmax is used as a latent representation. As the result, the generator has a reverse architecture, starting from an input of dimension 100 followed by 5 layers of size 128, 256, 512, 1024, and 784 (input image size). We use LeakyReLU in the generator as well. For the discriminator, we use an architecture similar to the encoder with a few changes: 1) the input is the concatenation of the input image of size 784 and latent representation of size 100, and 2) the last softmax layer is replaced with a dense layer of 1 neuron as the task is binary classification.

For SVHN, CIFAR10, and CIFAR100 datasets, we train a well-known LeNet model as the victim/encoder model. Here, the internal representation is a vector of size 84. Hence, our generator has input dimension of 84 followed by a dense layer of size 512. Then, the model reshapes it to (1, 1, 512) followed by four convolutional blocks of size 512, 256, 128, and 64. Each convolutional block consists of a 2D Convolutional Transpose layer (filter size of (2, 2) and strides of (2, 2)), LeakyReLU, and 2D Convolutional (filter size of (3, 3)). The number of filters is specified by the block size. However, the number of filters of 2D Convolutional of the last convolutional block is set to 3 to make sure that the output size is (32, 32, 3).

In our experimental evaluations, a shallow discriminator based on LeNet architecture suffers from mode collapse. So, we use a deeper convolutional model consists of four 2D convolutional layers of size 128, 256, 512, and 1024 (filter size of (3, 3) and strides of (2, 2)) followed by LeakyReLU activation after each 2D convolutional layer. Then, we use a flatten layer and concatenate the latent representation here. We find that concatenating latent representation at the middle of the

TABLE 4.2. Accuracy of victim models

Dataset	MNIST	FMNIST	SVHN	C-10	C-100	C-10	C-100
Model	MLP	MLP	LeNet	LeNet	LeNet	ResNet20	ResNet20
Victim train Acc	100%	100%	99.89%	95.13%	95.27%	98.51%	96.71%
Victim test Acc	97.43%	89.59%	87.82%	57.82%	22.37%	74.47%	33.03%

model achieves better convergence than at the beginning of the model. Finally, there is a dense layer of size 64, followed by a LeakyReLU activation and the last dense layer of size 1.

For CIFAR10 and CIFAR100 datasets, we also use ResNet20 as victim/encoder model. However, we use the same generator and discriminator as we use in LeNet case. In our BiGAN training, we find that when a Cosine similarity loss [1] is around 20% of the first epoch, the generated images are good enough. For CIFAR10 and CIFAR100, we find that training a GAN model for a few epochs and using the pre-trained generator in the BIGAN architecture prevents mode collapse in BiGAN training. The overall training time is similar since the BiGAN converges faster with a pre-trained generator.

4.4. Experimental Results

Table 4.2 illustrates the attack performance of all MI attacks. Sablayrolles attack [**103**] outperforms all existing attacks in literature while our attack achieves similar or better performance. Although Watson attack [**126**] is more efficient, its AUC is lower than Sablayrolles attack [**103**]. All previous MI attacks that do not consider sample difficulty (calibration) are substantially worst. Our subpopulation-based MI attack is on a par with Sablayrolles attack while obviating the need to train a large number of shadow models. A comparison of computational cost of training multiple shadow model versus a generator model is reported in 4.4.2. Moreover, our attack achieves a good performance when abundant data is available in which case no model training is needed. It shows that for the membership inference purpose our proposed BiGAN-like architecture achieves the same performance as natural images.

4.4.1. Black-box Setting. To get the subpopulation of a target sample, our attack needs to know the latent representation to generate or find semantically similar samples. This is done

[1]The loss is defined in TensorFlow as tf.keras.losses.CosineSimilarity().

TABLE 4.3. AUC of various datasets, target models, and MI attack models

Dataset	MNIST	FMNIST	SVHN	C-10	C-100	C-10	C-100
Model	MLP	MLP	LeNet	LeNet	LeNet	ResNet20	ResNet20
Yeom [**134**]	51.58%	54.84%	57.54%	77.56%	91.98%	70.62%	93.03%
Shokri [**110**]	51.98%	57.69%	58.07%	75.52%	84.72%	67.72%	87.75%
Jayaraman [**51**]	52.20%	55.78%	56.45%	75.01%	80.64%	68.58%	85.97%
Watson [**126**]	54.07%	60.52%	62.97%	80.37%	95.47%	72.78%	93.58%
Sablayrolles [**103**]	**55.54%**	62.55%	**63.41%**	81.56%	96.10%	74.84%	**95.21%**
Ours (BiGAN)	54.66%	**62.88%**	62.05%	**81.94%**	**96.23%**	**75.05%**	94.56%
Ours (BiGAN, black-box)	54.12%	62.07%	61.13%	81.24%	95.86%	74.23%	94.24%
Ours (natural samples)	-	-	61.61%	-	-	-	-

by using the last layer before the softmax of the victim model. However, in practice, the victim model's intermediate layers might not be available to the attacker. In this case, the attacker also trains an encoder during the BiGAN training instead of using the victim model. As shown in Table 4.3, the black-box scenario barely changes the attack performance.

4.4.2. Training Cost Comparison. We use Python 3.6.12 and Tensorflow 2.3, and a server with Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz and NVIDIA GeForce RTX 2080 Ti GPU using Ubuntu 18.04. Table 4.4 shows the training time required for each attack on minutes using a single GPU instance. Note that we group experiments that has similar training size and model architecture because they essentially have similar training time. As shown in Table 4.4, our attack significantly reduces the overall training time overhead. Although training a BiGAN is computationally more expensive than a single discriminator (shadow) model, previous MI attacks require training more than one shadow model. Here, both [**126**] and [**103**] is trained using 30 shadow models, as suggested in [**103**]. This leads to the overall larger training time.

Additionally, Sablayrolles attack [**103**] requires the average shadow model output of cases where the shadow model is trained with the target sample. Hence, for each new sample to investigate, the MI attack needs to compute $\mathbb{E}_{Y \leftarrow A(x \in D)}[s(Y, (x, y))]$ from scratch, meaning training 15 new shadow models. Our attack and Watson attack [**126**] do no require training new models for each new target sample. Moreover, our attack is also more effective when the victim model is deeper, such as in ResNet20. In this case, our generator and discriminator architecture is still the same as the LeNet case. Although it might take longer for the generator model to converge and find the

TABLE 4.4. Training time comparison of MI attacks in minutes.

Dataset Model	(F)MNIST MLP	SVHN LeNet	CIFAR LeNet	CIFAR ResNet20
When all target samples are known before the attack				
Watson [**126**]	46.37	86.72	56.42	281.60
Sablayrolles [**103**]	46.37	86.72	56.42	281.60
Ours	4.97	31.26	24.73	46.36
Ours wo. BiGAN	0	0	0	0
Training time per new sample				
Watson [**126**]	0	0	0	0
Sablayrolles [**103**]	23.185	43.36	28.21	140.80
Ours	0	0	0	0
Ours wo. BiGAN	0	0	0	0

mapping from the latent representation to the input space, it is significantly more efficient than training 30 ResNet20 models.

4.5. Conclusion

In this chapter, we propose a fundamentally different approach towards membership inference. Instead of comparing the victims model output versus shadow models' output, we essentially compare the victim model's output on the target sample versus victim model's output on samples from the same subpopulation. This new way of approaching membership inference obviate the need to train dozens to hundreds of shadow models and makes MI attacks more computationally efficient. Moreover, we show that when samples from the same subpopulation is not available, we can train a single generator using BiGAN-like architecture to craft samples of subpopulations. Hence, in the worst case, we only need to train a single generator. Our evaluation results demonstrate that our attack can achieve the state-of-the-art MI attack accuracy with no shadow model training.

CHAPTER 5

User-Level Membership Inference Attack against Metric Embedding Learning

Membership inference (MI) determines if a sample was part of a victim model training set. However, the exact training samples might not be accessible to the attacker. In this chapter, we develop a user-level MI attack where the goal is to find if any sample from the target user has been used during training even when no exact training sample is available to the attacker. We focus on metric embedding learning due to its dominance in person re-identification, where user-level MI attack is more sensible. We conduct an extensive evaluation on several datasets and show that our approach achieves high accuracy on user-level MI task.

5.1. Introduction

Membership inference (MI) attacks aim to identify whether a sample has been used during the training of a victim model or not. The existing research literature has primarily focused on record-level MI attack on classifiers and defense mechanisms against them. Record-level MI attack has a major limitation: it assumes that the exact training samples are available at the inference time to conduct membership inference. For example, a privacy auditor may want to investigate if a user's images have been unlawfully used to train a model connected to a video surveillance camera by using MI attacks. The camera that records people's movements may constantly capture pictures and retrain a vision model. However, if a privacy auditor (using the technique of MI attacks) wants to identify the identity of people whose data is used to train the model (against their will), there is no practical way to retrieve those exact training images. To address this limitation, we focus on user-level membership inference, where the goal is to identify users whose images were used to train a model, given that the exact training images are not available.

Specifically, we investigate a scenario that differs from traditional record-level MI attacks in two key aspects: 1) We focus on a user-level MI attack where the goal is to identify if any image from a target person (user) has been used for training the victim model or not. The primary example of tasks for which the user-level MI attack is more sensible are person re-identification or face recognition. Here, we want to know if any image of a target person was a part of a training dataset, not just one specific image. 2) We focus on metric embedding learning rather than classifiers because they are widely used for person re-identification and face recognition.

These two differences result in two new challenges. First, in most existing work, the user-level setting is either undefined or ignored. For example, in CIFAR dataset, where the task is to classify objects or animals, the notion of a user or an entity beyond a record is not well-defined. Second, in metric embedding learning, the model output does not contain confidence values or labels based on which the majority of existing MI attacks are built. To address these two challenges, we propose a new user-level MI attack against metric embedding based on an **intuitive empirical observation**: users whose data has been used during training form more compact clusters in the latent space. As shown in Figure 5.1, this observation holds both for training samples (green color) and other images of the same person that have not been used during training (yellow color), which solves the first challenge. Moreover, we focus on cluster properties in the latent space rather than on confidence output to address the second challenge.

In this chapter, we introduce a user-level MI attack against metric embedding learning using properties of clusters in latent space. More specifically, we use average distance to the cluster's center and average pair-wise distance as features. We show that our attack achieves high accuracy even when the target model is probed with images of a training user that have not been used in the training, and therefore, we make the user-level MI attack viable.

5.2. Background

5.2.1. User-Level Membership Inference. The goal of record-level membership inference is to identify whether a sample was part of a victim training model or not. Most existing membership inference attacks, such as [**106**, **110**], are *record-level MI attack* on classification tasks. The main

FIGURE 5.1. Green: training members, yellow: non-training members, and red: non-member. The distances are computed based on the latent space embedding of a LuNet model.

intuition behind these MI attacks is that classification models are more confident on training samples than test samples, and hence the confidence values can be used to infer membership [**100**].

In this chapter, we focus on the *user-level MI attack*, where the goal is to identify if any sample (images) from a target user has been used in the training. Here, the attacker might not have access to the exact training samples, but she can obtain other samples from the same user. This attack is more relevant in tasks where a user's identity is in danger of leaking, such as person re-identification. In the literature, there are only a few studies on user-level MI attacks. In [**81**], the authors investigate MI attacks on speech recognition task to infer if any users' data (voice samples) have been used during training. In [**112**], the authors propose a user-level MI attack on text generative models. None of the existing user-level MI attacks can be directly adopted for metric embedding learning scenario as discussed in detail in Sec. 5.4.

5.2.2. Metric Embedding Learning. The goal of metric embedding learning is to learn a mapping from a high-dimensional input space into a lower-dimensional latent space in which semantically similar inputs are closer [**36**]. This includes variations of contrastive loss and triplet loss. In contrastive loss, two samples are taken as the input to a model, and the loss term aims to decrease (increase) the distance of the embeddings of these samples if they belong to similar (different) class(es). Here, samples from similar classes are called *positive samples*, and samples from different classes are called *negative samples*. The triplet loss takes three samples as input: an anchor, a positive sample w.r.t the anchor, and a negative sample w.r.t the anchor. It aims to

push anchor and positive samples together while pulling the anchor and negative samples away. None of the existing MI attacks can be directly adopted for metric embedding learning because the outputs of metric embeddings are not confidence values. To the best of our knowledge, the only MI attack on metric embeddings is EncoderMI [**72**]. Simply put, it computes the closeness of a target image with its augmented versions in latent space as attack feature. However, it is a record-level MI attack, and we show that its extension to a user-level scenario leads to poor performance.

5.3. Attack Overview

5.3.1. Threat Model. Victim Model: In this chapter, we mainly use the LuNet model with soft-margin batch hard loss [**36**], a variant of triplet loss, as a victim model due to its high accuracy and popularity. LuNet loss modifies the original triplet loss to efficiently choose the hardest positive and hardest negative samples for each anchor sample to improve the training. Note that our approach can be trivially extended to any other metric embedding learning because it uses the embedding as a black-box function.

User-level Membership Inference: In contrast to record-level membership inference, where samples are categorized into members and non-members, in user-level membership inference we have three groups of samples: 1) *training members* (D_m^t) are the samples from users that have been used during the training, 2) *non-training members* (D_m^{nt}) are samples that have not been used during the training, but the identity of the corresponding users have been used via training member samples, and 3) *non-members* (D_{nm}) are samples from users whose data has never been used during the training. Here, the goal is to identify non-training members as members without accessing training members, which is in general not available in record-level MI attacks.

Attacker knowledge: We assume that the attacker has access to a set of non-training member samples and a set of non-members. However, the attacker does not know which sample belong to which set. The attacker does not necessarily need training members which is a more realistic assumption in comparison with record-level MI attacks where the exact training samples should be available to the attacker to identify members. Additionally, we assume that the attacker can query the black-box encoder to obtain the latent representation of samples.

5.3.2. Feature Extraction. Key intuition: The key observation that allows an attacker to launch an MI attack against metric embeddings is that the images of the user whose data has been used during the training form a more compact cluster in the latent space of the victim model, as shown in Figure 5.1. This includes both training members (D_m^t) and non-training members (D_m^{nt}).

Attack features: To use the key observation stated above, we need to measure the compactness of user's samples in latent space. To achieve this goal, we define two metrics: 1) average center-based distance (C_u), and 2) average pair-wise distance (P_u). Let's denote $E_v(.)$ as the victim model that outputs the latent representation. We use x_u^i to denote the i^{th} sample of a user, u. Given m_u samples from the user u, average center-based distance is defined as follows:

$$C_u = \frac{1}{m_u} \sum_{i=1}^{m_u} d(x_u^i, \bar{x}_u), \tag{5.1}$$

where $\bar{x}_u = \frac{1}{m_u} \sum_{i=1}^{m_u} x_u^i$, called the center of cluster, and d(.) is a distance measure. We use the L2 norm as the distance measure throughout this chapter. Similarly, we define the average pair-wise distance as follows:

$$P_u = \frac{1}{m_u - 1} \sum_{i=1}^{m_u - 1} \frac{\sum_{j=i+1}^{m_u} d(x_u^i, x_u^j)}{m_u - (i+1)}, \tag{5.2}$$

which obtains the average latent distance across all possible pairs of images of user u. Note that in contrast to existing record-level MI attacks, we cannot infer the membership of a user using only a single sample. To measure the compactness of a cluster, our attack requires multiple samples from the user.

5.3.3. Attack Model Training. Using the two attack features (C_u, P_u) described above as input to the attack model, we train an attack model to output the membership status of a target user. We adopt shadow model training strategy widely used in record-level MI attack proposed in [**110**]. Simply put, we train multiple (shadow) models on the same task as the victim model, but with different data samples. Since the ground truth of members and non-members of the shadow

models are known to the attacker, she can use the ground truth to train the attack model. The details of the shadow models and their dataset is explained in Section 5.4.

5.4. Experimental Setup

Dataset: We use Market-1501 [**143**] and PRID-2011 [**40**]. Market-1501 is a benchmark frequently used to evaluate person re-identification models. After excluding duplicates, distractors and junks, we have 26051 labeled images of 1501 users. PRID-2011 consists of images extracted from multiple person trajectories. After excluding duplicates, we have 71657 labeled images of 934 users.

Victim model: We choose LuNet with soft-margin batch hard loss by [**36**] as our victim model, which is trained on D_m^t. For Market-1501 and PRID-2011, we randomly select D_m^t, D_m^{nt}, and D_{nm} from the dataset. D_m^t and D_m^{nt} includes non-overlapping images from the same 150 memebrs. D_{nm} includes images of 150 non-members, who do not overlap with the members. The remaining images are used as the shadow dataset, D_s.

Shadow models: For each shadow model, we randomly select shadow training members, shadow non-training members, and shadow non-members from the shadow dataset, D_s. We train shadow models on shadow training member set. Here, shadow model architecture is the same as the victim model architecture, both in our attack and [**72**] with which we compare our attack. We train 10 and 100 shadow models for PRID-2011 and Market-1501 datasets, respectively.

Attack model: Our attack model is a shallow neural network with 3 fully connected layers. The input features are the average center-based distance (C_u) and average pair-wise distance (P_u) as described in Section 5.3.2. Throughout our evaluation, we always use the same number of images to obtain these two features. We train the attack model with the shadow dataset. We repeat each experiment 5 times and report the average and standard deviation.

Baselines: To the best of our knowledge, there is no user-level MI attack on metric embedding learning. The two user-level MI attacks in literature [**81**, **112**] require generative models where the victim model's output is a word. Hence, there is no trivial way to adopt them for metric embedding scenario. Moreover, the majority of record-level MI attacks on classifiers rely on confidence values which is not available when using metric embedding. Hence, there is no trivial way to adopt

TABLE 5.1. Performance comparison of user-level MI attacks on metric embeddings.

MIA method	Accuracy		Precision		Recall	
	Market	PRID	Market	PRID	Market	PRID
Our user-level MIA	**66.87** ± 1.87	**74.27** ± 0.83	75.25 ± 0.54	69.80 ± 1.35	50.27 ± 5.51	85.73 ± 2.94
EncoderMI (unknown augmentations)	52.00 ± 1.56	52.67 ± 2.14	54.28 ± 3.32	51.06 ± 3.02	46.67 ± 30.94	63.33 ± 32.49
EncoderMI (full knowledge)	66.00 ± 1.21	69.60 ± 3.12	63.62 ± 3.55	65.20 ± 3.96	77.60 ± 10.55	86.27 ± 6.02

them here. However, we can adopt record-level MI attacks on metric embedding to the user-level scenario with a minor adjustment. There is only one attack that satisfy this condition, called EncoderMI [**72**]. To adopt for the user-level MI scenario, we launch their record-level MI attack on all samples of a user and then we perform majority voting.

5.5. Experimental Results

Table 5.1 shows the performance comparison between our attack and EncoderMI. Here, we only use non-training members and non-members for the evaluation purpose. EncoderMI computes the closeness of the target sample with its augmented variants as features. When the exact data augmentations used by the victim model are not known to the attacker, it chooses a fixed set of augmentations following the original setting of EncoderMI paper. In this case, the EncoderMI performs close to random guess (the second row). However, when all data augmentations during victim model training are known to the attacker, EncoderMI performs better (the third row). Despite such an unrealistic advantage to the EncoderMI, it still cannot outperform our approach.

5.5.1. Access to some training images. In the previous section, we assumed that only the non-training member samples are available to the user-level MIA. In cases where some training member samples are available, we expect to achieve even better performance. As shown in Table 5.2, by increasing the number of training members available to the attacker, we can significantly improve the user-level MI accuracy.

5.5.2. Effect of number of training samples versus MI attack. Intuitively, as the number of training samples for a user increases, we expect the metric embedding process to push those images more towards each other. In other words, as the number of training samples for a user increases, it presents a more compact cluster in the latent space. Table 5.3 shows our user-level MIA recall on different group of users with different number of training samples. Clearly, our MI

TABLE 5.2. User-level MIA performance when some portion of the training samples are available to the attacker.

% of training	Accuracy		Precision		Recall	
	Market	PRID	Market	PRID	Market	PRID
0%	66.87 ± 1.87	74.27 ± 0.83	75.25 ± 0.54	69.80 ± 1.35	50.27 ± 5.51	85.73 ± 2.94
25%	74.60 ± 0.25	76.33 ± 0.30	79.96 ± 1.18	70.78 ± 1.20	65.73 ± 2.33	89.87 ± 3.08
50%	81.87 ± 0.69	78.53 ± 0.54	82.97 ± 1.05	71.76 ± 1.36	80.27 ± 3.00	94.27 ± 2.25
75%	90.00 ± 0.52	78.40 ± 0.65	85.41 ± 1.26	71.70 ± 1.40	96.53 ± 0.98	94.00 ± 2.11
100%	91.73 ± 0.93	78.07 ± 1.00	85.83 ± 1.35	71.56 ± 1.52	100.0 ± 0.00	93.33 ± 1.89

attack is more successful on users with larger number of training samples. This is somehow in contrast with record-level MIA on classifiers where more training data is often construed as less memorization and, hence, less privacy leakage.

TABLE 5.3. User-level MIA's recall on groups with different number of training images per person.

Group	Market		PRID	
	Number of Images	Recall	Number of Images	Recall
1	$22 \leq n \leq 63$	69.33 ± 5.73	$123 \leq n \leq 445$	96.77 ± 2.04
2	$17 \leq n \leq 21$	60.00 ± 7.43	$102 \leq n \leq 112$	85.81 ± 5.62
3	$14 \leq n \leq 16$	40.83 ± 4.86	$88 \leq n \leq 101$	84.14 ± 1.69
4	$11 \leq n \leq 13$	36.47 ± 5.76	$78 \leq n \leq 87$	85.33 ± 1.63
5	$8 \leq n \leq 10$	45.45 ± 5.07	$66 \leq n \leq 77$	75.86 ± 4.88

5.5.3. Ablation Analysis. Table 5.4 illustrates the effect of each attack feature on user-level MIA. Although the highest accuracy is achieved when both features are used, the difference is not significant. Hence, the attacker can also use a single feature to reduce the computation overhead.

5.6. Conclusion

In this chapter, we propose a user-level MI attack on metric embedding learning. Our attack differs from most existing MI attacks in two aspects: First, we focus on the user-level MI attack which is more practical in tasks where the exact training data samples used in training are not available. Second, we focus on metric embedding learning scenario where the existing confidence-based MI attacks do not work. In contrast with existing MI attacks, we use a measure of compactness of clusters in embedding space to identify membership, and consequently, obviate the need to access

TABLE 5.4. User-level MIA performance evaluation using different set of features.

Input Features	**Accuracy**		**Precision**		**Recall**	
	Market	**PRID**	**Market**	**PRID**	**Market**	**PRID**
(C_u)	65.80 ± 3.39	73.13 ± 0.45	75.67 ± 1.29	68.17 ± 0.27	46.93 ± 11.23	86.80 ± 2.12
(P_u)	66.67 ± 2.32	73.53 ± 0.83	74.40 ± 1.23	69.20 ± 1.53	51.07 ± 8.42	85.07 ± 3.34
(C_u,P_u)	66.87 ± 1.87	74.27 ± 0.83	75.25 ± 0.54	69.80 ± 1.35	50.27 ± 5.51	85.73 ± 2.94

confidence values. Our attack achieves the state-of-the-art performance in several datasets, where user-level MI attack is of paramount importance.

CHAPTER 6

Accuracy-Privacy Trade-off in Deep Ensemble: A Membership Inference Perspective

Deep ensemble learning has been shown to improve accuracy by training multiple neural networks and averaging their outputs. Ensemble learning has also been suggested to defend against membership inference attacks that undermine privacy. In this paper, we empirically demonstrate a trade-off between these two goals, namely accuracy and privacy (in terms of membership inference attacks), in deep ensembles. Using a wide range of datasets and model architectures, we show that the effectiveness of membership inference attacks increases when ensembling improves accuracy. We analyze the impact of various factors in deep ensembles and demonstrate the root cause of the trade-off. Then, we evaluate common defenses against membership inference attacks based on regularization and differential privacy. We show that while these defenses can mitigate the effectiveness of membership inference attacks, they simultaneously degrade ensemble accuracy. We illustrate similar trade-off in more advanced and state-of-the-art ensembling techniques, such as snapshot ensembles and diversified ensemble networks. Finally, we propose a simple yet effective defense for deep ensembles to break the trade-off and, consequently, improve the accuracy and privacy, simultaneously.

6.1. Introduction

Ensemble learning has been shown to improve classification accuracy of neural networks in particular, and machine learning classifiers in general [**60**, **62**, **104**]. The most commonly used approach for deep models involves averaging the output of multiple neural networks (NN) that are independently trained on the same dataset with different random initialization, called **deep ensemble** [**74**]. Such a simple approach has been extensively used in practice to improve accuracy [**65**, **123**]. Notably, a majority of the top performers in machine learning benchmarks, such as the

ImageNet Large Scale Visual Recognition Challenge [**102**], have adopted some form of ensemble learning [**33**, **65**, **115**].

Interestingly, a few recent papers argue using ensemble learning to achieve a different goal rather than improving accuracy, that is, to defend against membership inference attack [**47**, **68**, **96**, **133**]. In membership inference attack, the goal of an attacker is to infer whether a sample has been used to train a model–i.e., whether the sample belongs to the train set. In literature, several forms of ensemble learning (different from deep ensembles), such as partitioning, has been used to defend against privacy-harming membership inference (MI) attacks. Membership inference attacks generally use the prediction confidence of NN models to infer membership status of a sample [**105**, **110**, **119**, **134**] by leveraging the insight that trained models may output higher prediction confidence on train samples than non-train samples [**15**]. The intuition behind using ensemble learning approaches, like partitioning, to defend against MI attacks is that training each model on a different subset of data makes the ensemble less prone to overfitting [**105**]. While the idea is discussed in [**47**, **68**, **96**, **133**], none of these papers theoretically or empirically demonstrate the usefulness of *deep ensembles*, in particular, as a defense mechanism.

In this chapter, we show that these two goals of ensemble learning, namely improving accuracy and defending against MI attack, do not trivially sum up in a unified solution in deep ensembles. Figure 6.1 illustrates accuracy and privacy trade-off by plotting accuracy and membership inference attack effectiveness for ensembles comprising of varying number of base models (1, 2, 5, and 10) that are trained for different numbers of epochs (5, 45, 74 and 81). The training epoch is chosen such that the accuracy of a single model best aligns with the accuracy of an ensemble. We make two key observations here. First, there is an increase in both accuracy and MI attack effectiveness as we go from a single model to ensembles comprising of an increasing number of base models. The trade-off is more noticeable for more accurate models trained for a larger number of epochs. Second, we can adapt the design of ensembles to suitably navigate the trade-off between accuracy and privacy. Starting with a single well-trained model (indicated by the the pink circle) achieving around 70% test accuracy as a baseline (for non-ensemble case), ensembling can be adopted to: (1)

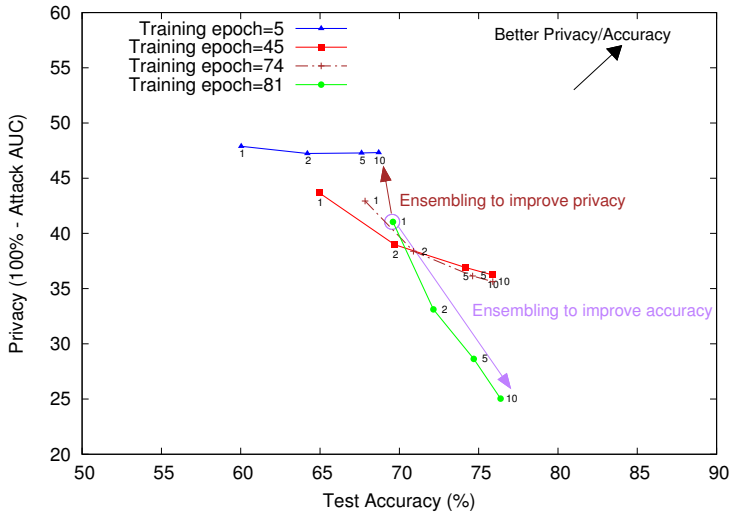


FIGURE 6.1. Trade-off between accuracy and privacy on an AlexNet model trained on CIFAR10. Each curve contains four points corresponding to ensembles comprising of 1, 2, 5, and 10 base models (from left to right). Using the single model trained for 81 epochs as a baseline, there are two choices: (1) making an ensemble of these models to achieve the highest accuracy possible but worse privacy (purple arrow), or (2) making an ensemble of less overfitted models (epoch #5) to achieve slightly lower accuracy of a single model but better privacy (brown arrow).

improve accuracy by using an ensemble of highly accurate models but at the cost of worse privacy[1] (purple arrow); and (2) improve privacy by intentionally using an ensemble of *under-fitted* models instead of a single model but at the cost of accuracy (brown arrow). However, these two objectives are not achieved simultaneously in deep ensembles.

To better study this phenomenon, we start with the most widely-used form of ensembling in deep models, that is, deep ensembles, and the most common type of membership inference attack based on confidence outputs. To understand the root cause of this trade-off, we show that using deep ensembles to improve accuracy exacerbates its susceptibility to membership inference attacks by making train and non-train samples more distinguishable. By analyzing the confidence averaging mechanism of deep ensembles, we investigate potential factors that enable membership inference. We show that the most influential factor is the level of correct agreement among models. Simply put, the number of models that correctly classify a train sample is often greater than the ones that

[1]Note that for complicated tasks, such as image classification, the common practice is to train deep models for a large number of epochs and avoid under-fitted models. That is because memorizing samples from long-tailed subpopulations are shown to be necessary to achieve close-to-optimal generalization error [**24**].

correctly classify a test sample. This results in a wider confidence gap between train and non-train samples, when confidence values are averaged, enabling more effective membership inference attacks.

We further show that the difference in the level of correct agreement between train and non-train samples is correlated with models' generalization gap. Hence, a natural question to ask is "can deep ensembles that use less overfitted models mitigate privacy issues while achieving high accuracy?". To answer this question, we study several regularization techniques, common membership inference defenses, and a few other ensembling approaches. We again observe a privacy-accuracy trade-off pattern similar to that shown in Figure 6.1.

Finally, using the insights obtained in the above analysis, we derive yet effective modification on deep ensembles that not only mitigate the privacy leakage issue in deep ensembles, but also improve privacy significantly. Instead of averaging confidence values, our approach outputs the confidence of the most confident model among the models that predict the same label as the entire ensemble. We show that this approach has several benefits: 1) It mitigates the effectiveness of the membership inference attacks to the point where the attack often performs similar to a random guess. 2) It can still achieve similar accuracy as of deep ensembles (averaging confidence values). 3) It does not require any change to the training process of base models. In other words, this can be easily adopted even for the systems whose base models have already been trained.

Summary of contributions: In this chapter, we perform a systematic empirical study of MI attacks on deep ensemble models. We start with an in-depth analysis of the most common ensembling technique and membership inference attack, and then we extend the results to various ensembling techniques and membership inference attacks. First, we show that when deep ensembles improve accuracy, it also leads to a different distribution shift in the prediction confidence of train and test samples, which in turn enables more effective membership inference. Second, we analyze various factors that potentially cause the prediction confidence of train and non-train samples to diverge. Among potential factors, we show that the most dominant factor is the level of correct agreement among models which indicates that more models in an ensemble agree on their prediction when a sample is a training sample. Hence, the aggregation of their prediction yields higher confidence

output in comparison with non-train samples. We show that common defense mechanisms in membership inference literature, including differential privacy, MemGuard, MMD+Mixup, L1 and L2 regularization, as well as other ensemble training approaches, such as bagging, partitioning, can be used to mitigate effectiveness of MI attacks but at the cost of accuracy. We solve this trade-off issue by changing the fusing mechanism of deep ensembles which improves the accuracy and privacy, simultaneously. Although the main focus of this chapter is on deep ensembles, we also cover bagging, partitioning, weighted averaging (Section 6.5.6), as well as more advanced and state-of-the-art ensembling techniques, such as snapshot ensembles [**45**] and diversified ensemble networks [**139**] (Section 6.5.5). We observe similar trade-off.

6.2. Background

6.2.1. Ensemble Learning. In literature, ensemble learning refers to various approaches that combine multiple models to make a prediction. Models used to construct an ensemble are often called base learners. There are two main factors to construct an ensemble [**104**]: 1) how base learners are trained to ensure diversity, such as random initialization, bagging, partitioning, etc., and 2) how the output of base learners are fused to obtain the final output, including majority voting, confidence averaging, stacking, etc.

The most common forms of ensemble learning in classical machine learning algorithms are bagging, partitioning, boosting, and stacking. In **bagging**, several models are trained with different bootstrap samples of the training dataset. In other words, each model is trained on a randomly selected under-sampled version of the entire dataset. As a results, the diversity is ensured by varying training set of each model. The model outputs are often fused using majority voting or averaging. Random forest is a widely-used example of bagging of decision trees. In **partitioning**, similar to bagging, base models are trained on different subset of the entire dataset and fused with majority voting or averaging. However, unlike bagging, the training datasets are non-overlapping. **Boosting** is an ensemble learning technique that focuses on samples that were misclassified by previous trained models. In other words, the models are trained sequentially such that the second model aim to correct the prediction of the first model, the third model aims to correct the prediction of the second model, and so on. This is often done by changing the weight of each sample during the

training. The most common boosting algorithms are AdaBoost, gradient boosting and XGBoost. **Stacking** is a meta-learning ensembling approach where a meta-learner is trained on top of the base models. Meta-learner is often a simple regression or a shallow neural network. The complexity is often shifted to base models. There are hundreds of variations of these methods in literature that are less common and the study of them is out of the scope of this dissertation.

Unlike classical machine learning domain where several popular ensemble methods exist and are equally used for different scenarios, there is only one heavily-used method for deep learning models, called **deep ensemble** [**60**]. In this method, 1) base models are initialized with random weights and trained on the same training dataset, and 2) their prediction confidence are fused through averaging to construct the final output. Unlike ensemble of traditional machine learning algorithms, in deep ensembles, the main source of diversity often comes only from random initialization of base learners [**25**]. In fact, other sources of diversity, such as bagging, have been shown to considerably degrade the overall accuracy of a deep ensemble [**63**,**65**]. Although some classical ensemble learning approaches, including bagging, partitioning, and stacking, can be easily adopted for deep learning models, they are rarely used due to the low accuracy in comparison with deep ensembles.

Recently, a few promising ensemble methods for deep models have been proposed. In **snapshot ensemble** [**45**], only one deep model is trained. Here, base models are snapshots of that single model at different epoch during the training. Specifically, every time the model is converged, an snapshot is taken and the training process continues by using a large learning rate to find a new local optimum. This process significantly reduces the training time of ensemble which is an important obstacle for training deep models on large dataset. In **diversified ensemble network** [**139**], the output of base models are aggregated in a shared layer (similar to stacking) and are trained jointly. The main novelty is that it uses an additional loss term that ensures each model is optimized in different directions of diversity. Interestingly, unlike deep ensembles, they show that there is an optimum number of base models over which the diversified ensemble network's accuracy starts to degrade. To the best of our knowledge, there is no ensembling approach in literature for deep models that *considerably* outperforms deep ensembles.

6.2.2. Membership Inference Defenses. Defense mechanisms against membership inference attacks can be summarized into two categories [**96**]:

Generalization-based: Shokri [**110**] was the first to correlate membership inference success with overfitting. Since then, many standard regularization techniques have been used to alleviate overfitting, such as L1 regularization [**15**], L2 regularization [**15**, **52**, **89**, **110**, **119**], differential privacy [**15**, **96**], dropout [**52**], and adversarial training [**88**]. Interestingly, ensemble learning has also been proposed as a defense mechanism. In [**105**], they proposed a combination of partitioning and stacking as a defense mechanism. The intuition is that training each model with different subset of data makes the entire ensemble model less prone to overfitting. Note that these defense mechanisms often degrade the accuracy of the target model (see Section 6.5.1) [**15**].

Confidence-masking: These defense mechanisms aim to reduce the amount of information that can be obtained from the output of a target model by perturbing [**52**] or limiting the dimensionality of the output [**15**, **110**, **119**]. Most confidence-masking approaches manipulate confidence values post-training. As a result, the output values of these models do not reliably represent the "confidence" of the model. These approaches are built under the assumption that accurate prediction of confidence is not needed. However, many applications require accurate estimation of confidence. Moreover, if the accurate prediction of confidence is not required, then the trivial MI defense would be to only output the class label and avoid these confidence-masking defenses altogether. In this chapter, we cover MemGuard-random defense as it has already shown to outperforms the other confidence-masking mechanisms [**68**].

6.3. Threat Model

Our threat model works under the scenario of machine-learning-as-a-service (MLaaS) where an ML prediction API is provided by an MLaaS provider. The API is accessible to users who can query the API with an input and obtain the prediction output. In this scenario, a malicious user, referred to as an *attacker*, can query the API to obtain unintended information beyond the prediction output. Specifically, the attacker aims to launch a membership inference attack to identify training samples used to train the MLaaS API's model. In this chapter, we refer to the MLaaS provider as the *defender* or *victim*.

6.3.1. Defender. The assumptions and objectives of the defenders are as follow:

Assumptions: Here, we assume that the defender uses deep ensembles to improve the accuracy of the prediction model. The defender uses the training dataset, D_{tr}, to train multiple base models. The defender may or may not use all available training samples to train each model. As long as a training sample is used to train at least one base model, we label the sample as a member. Moreover, the defender provides an API access and returns prediction confidence values. In a multi-class classification task, the output is a vector of probabilities corresponding to each class, referred to as confidence values. The defender can train base models from scratch and, as a result, apply regularization techniques or membership inference defense mechanisms that requires modification of the training process. The defender can also use any fusing technique rather than confidence averaging which is used in deep ensembles.

Objectives: The main objective of the defender is to mitigate the membership inference attack while benefiting from the ensemble learning's improved accuracy. Preferably, the solution should impose minimal computational cost at both training and inference time. We study several MI defense mechanisms, including MMD+Mixup [**68**], L1 and L2 regularization, and DP-SGD. Moreover, we investigate several ensembling approaches that suggested in literature as a defense mechanism, including bagging, and partitioning. Finally, we propose a simple solution that achieve both objectives of improving accuracy and privacy, simultaneously.

6.3.2. Attacker. The objectives, knowledge, and capabilities of the attackers are as follow:

Objectives: The attacker aims to launch a membership inference attack to identify samples of the defender's training dataset, D_{tr}.

Knowledge: We make the following assumptions about the attacker's knowledge:

1) The attacker has the full knowledge of the classification task. Given that the purpose of the API is to provide a service to users, it is reasonable to assume that the task for which the API is developed is known to all users, including the attacker. This includes the number of classes, class labels, and the input shape.

2) We assume that the attacker has only black-box access to the defender's model. Although it is possible in some scenarios to approximate model parameters through model extraction methods [**118**], it is of the model owner's interest to keep the model proprietary as otherwise it defies the whole point of MLaaS as a business model.

3) The attacker may know the model architecture, training algorithms, the type of ensemble method, and other training information. These information may sometimes be available via the API's documentation. Specifically, when the attacker needs to train shadow models, he uses the same architecture and training parameters as the defender's model. But, as it is shown in [**105**], the lack of this information barely changes the MI attack effectiveness, at least when original shadow-based MI attack of Shokri [**110**] is used.

Capabilities: The attacker has the following capabilities:

1) The attacker can collect a dataset, D_s, that has the similar distribution as the D_{tr} to train shadow models on the same task if needed.

2) The attacker has computational resources to train an attack model, which takes some features from a target sample and outputs the membership status. In the simplest form, it can be a threshold on the output of the defender's confidence output [**134**], or an ML model [**110**].

6.4. How Does Ensembling Increase Membership Inference Effectiveness?

In this section, we thoroughly investigate the most widely-used deep models ensembling technique and membership inference attack, that is, deep ensembles and confidence-based attack. We mainly focus on distributions of confidence values when a deep ensemble is used and how it can lead to more distinguishable distributions. How an actual MI attack can use this feature is studied in Section 6.5. Furthermore, other forms of ensembling techniques and membership inference attacks are shown in Section 6.5.

6.4.1. Confidence Distribution Shift. Ensemble learning is only helpful when base learners disagree on some samples [**62**, **104**]. Otherwise, ensembling does not improve accuracy. Furthermore, when deep ensemble is used, the confidence values of multiple models are averaged to obtain the final prediction. Consequently, when ensembling improves accuracy, it averages the prediction confidence of highly confident predictions (mostly from models which correctly classified the sample) and less confident predictions (mostly from models which misclassified the sample). As a result, confidence distribution shift is inevitable for both train and test samples. This phenomenon can be observed as the distribution of Figure 6.2(a) shifts to that of 6.2(d). This can be better observed by

separating correctly classified samples which have significantly higher prediction confidence (Figure 6.2(e)) and misclassified samples which have lower prediction confidence (Figure 6.2(f)). One can clearly observe that both distributions shift more towards the center from Figure 6.2(b) and (c) to Figure 6.2(e) and (f). However, the change in the distribution of train and test samples does not necessarily cause a more effective membership inference if the change has a similar effect on the confidence distribution of both train and test samples. In the next subsection, we analyze the potential factors that affect the distribution change and how they can change confidence distribution of member and non-member sets differently.

6.4.2. Effect of Ensembling on Individual Samples. We use y_i to denote the confidence value of the i^{th} model in an ensemble of n models. Hence, for a given sample x, the output of the ensemble is:

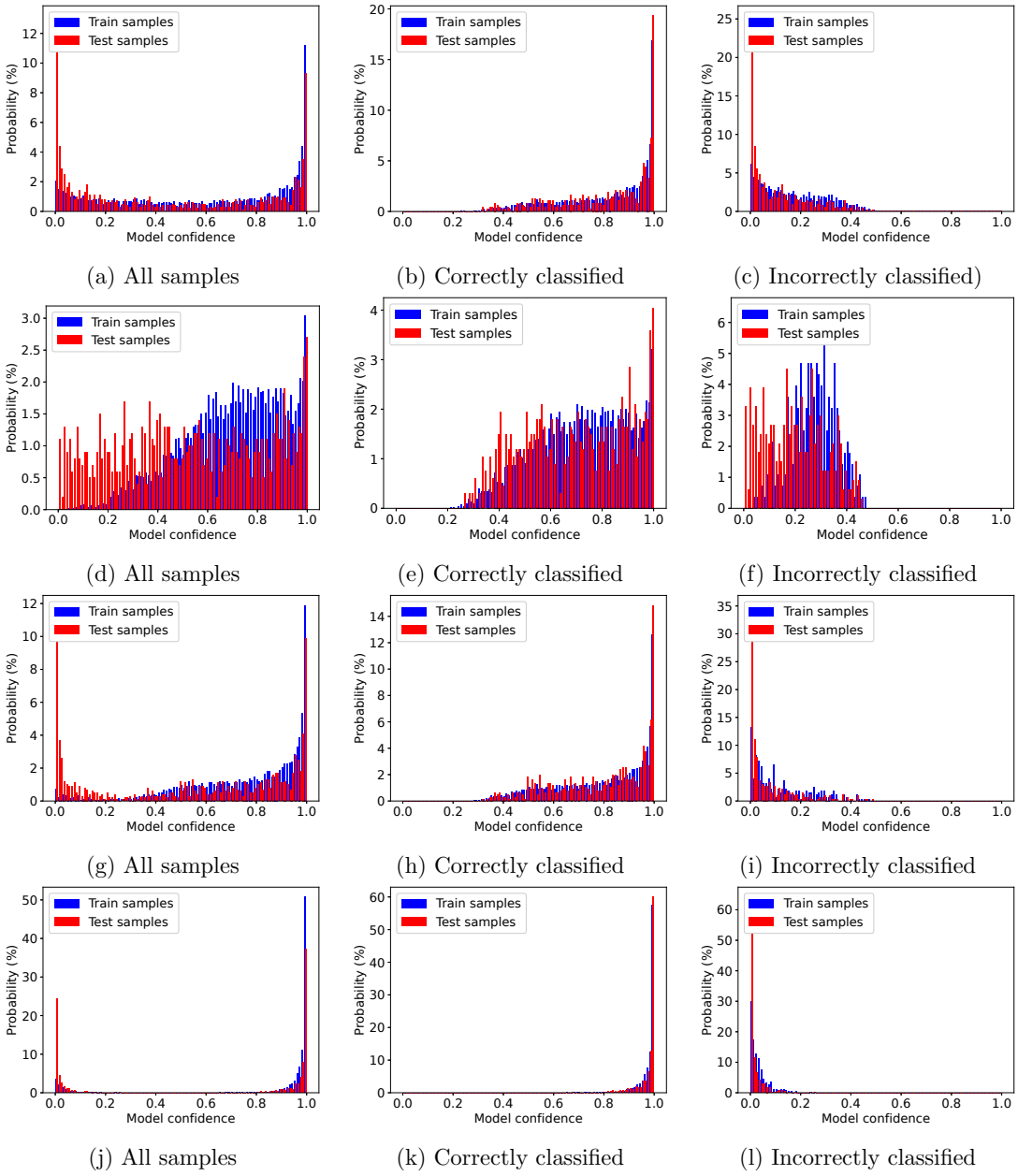
$$y_{el}(x) = \frac{\sum^{n} y_i(x)}{n} = \frac{\sum^{c} y_i^c(x) + \sum^{m} y_i^m(x)}{n}. \tag{6.1}$$

Given a single sample, we can further divide models in the ensemble into two groups: 1) models that correctly classified the sample denoted by y_i^c and 2) models that misclassified the sample by y_i^m. For a given sample x, c models correctly classify it and m models misclassify, where $c+m=n$. Note that the value of c and m depends on the sample[2].

Based on the Eq. (6.1), three major factors affect the final confidence value (y_{el}) of a sample: y^c, y^m, and c. As a result, if these values are different for train and test samples, the ensembling causes different shift in the distributions, and consequently, membership inference attack will be more effective. These factors are as follows:

(1) Confidence value of correctly classifying models (y^c): Since the majority of samples are supposed to be correctly classified by a practical model, any distinguishable confidence difference between train and test samples can lead to a very effective membership inference attack. However, as shown in Figure 6.2(b), we can observe that there is no significant difference between train and test samples.

[2]By an abuse of notation, we use c (m) to refer to (in)correctly classifying models and also as a superscript for the model output of (in)correctly classifying models, that is, y_i^c (y_i^m).



(a) All samples (b) Correctly classified (c) Incorrectly classified)

(d) All samples (e) Correctly classified (f) Incorrectly classified

(g) All samples (h) Correctly classified (i) Incorrectly classified

(j) All samples (k) Correctly classified (l) Incorrectly classified

FIGURE 6.2. ResNet20 on CIFAR10 (Class #3): Confidence (i.e. maximum of the softmax output) distribution of a single model (top row), an ensemble of 10 models using *ensemble averaging* (the second row), an ensemble of 10 models using *first agreed confidence* approach (the third row), and an ensemble of 10 models using *maximum agreed confidence* approach (the fourth row). Jensen–Shannon divergence of the two distributions are as follows: a) .2276, b) .1484, c) .2515, d) .3037, e) .1682, f) .3408, g) .2940, h) .1432, i) .3076, j) .2595, k) .0846, and l) .2753.

(2) Confidence value of misclassifying models (y^m): Unlike correctly classified samples, there is a marginal difference between confidence distribution of train and test samples of misclassified samples (see Figure 6.2(c)). This may be exploited by membership inference attack to partially distinguish between train and test samples.

(3) Level of correct agreement (c) among models: As shown in Figure 6.3(a), the number of models that correctly classify a sample (c) is greater for train samples than test samples. Since prediction confidence of correctly classified samples are higher than misclassified samples on average, i.e., $y^c > y^m$, and c is smaller for test samples, the ensemble confidence of test samples (y_{el}) becomes lower than train samples. As a result, this factor can largely contribute to the effectiveness of membership inference attacks on deep ensembles.

We note that the first two factors are not unique to ensembles and can be exploited by an attacker in a single model scenario as well. As a result, these two factors have been extensively studied in [**100**] in a single model scenario across various image datasets and well-trained models. They have shown that for deep models the first factor (y^c) is almost indistinguishable between train and test set and only the second factor (y^m) is marginally distinguishable. However, this marginal difference does not have a considerable impact on the different distribution shift in train and test sets.

On the other hand, **the level of agreement** has a big impact on different distribution shifts of train and non-train samples. To better demonstrate this, we can analyze the distribution difference in each level of agreement separately. As shown in Figure 6.3(b), the average confidence of train and test samples are very close and indistinguishable when each level of agreement is drawn separately. If the effect of the first two factors were considerable, the two confidence values for each level of agreement would have been more distinguishable. Note that the average confidence between train and test sets is more distinguishable for the first two points in x-axis (where the majority of models misclassify a sample). However, these distributions only constitute a tiny portion of the training dataset, as shown with the first two blue bins in Figure 6.3(a). However, when all samples are combined, we can vividly observe that the average confidence gap between train and test sets considerably widens, as shown in the last point in x-axis in Figure 6.3(b). This clearly

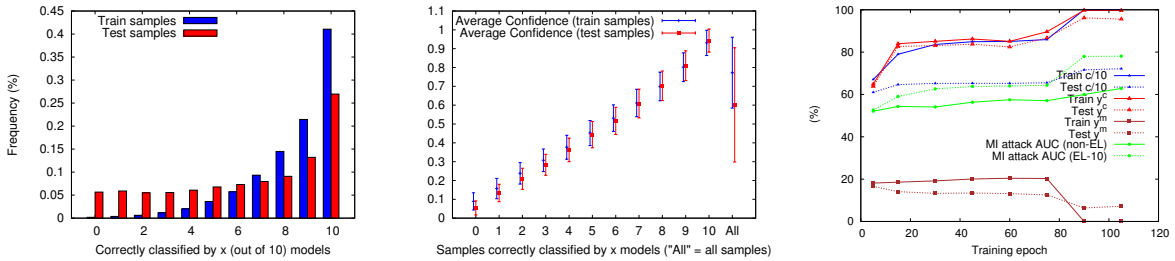
demonstrates that the major factor in different distribution shift between train and test sets is the level of agreement (c).

Note that, unlike the first two factors, the third factor (c) only improves the effectiveness of membership inference attacks in ensemble scenarios because it does not exist in a single model. In other words, if a defence strategy eliminates the effect of the average level of correct agreement (i.e., it ensures that c is close between train and test samples), the membership inference attack is still possible on the ensemble, but only to the degree that it is possible on a single model[3]. As shown in Figure 6.3(c), as the gap between y^c of train and test sets (red lines) and the gap between y^m of train and test sets (brown lines) increases, the attack on both single model (non-ensemble) and also the ensemble (EL-10) increases. However, only when the average level of correct agreement gap between train and test (blue lines) widens, the membership inference attack on ensembles becomes more effective than on non-ensembles.

Another important observation from Figure 6.3(c) is that the minimum level of agreement gap between train and test occurs when models are relatively underfitted (i.e., the blue lines in first few epochs). This phenomena has also been partially observed in [**25**] (Figure 2(c)). The main reason is that underfitted models often only learn the most common and generalizable features and, thus, they often agree on the features and predictions. As they move from underfitted region to overfitted region, their generalization gaps widen (blue lines in Figure 6.3(c)). As a result, they tend to correctly classify train samples more often than test samples. Consequently, they agree on train samples more than test samples and, hence, average gap of correct agreement between train and test set widens. Hence, the wider generalization gap of base learners is, the more effective membership inference attack would be on deep ensembles.

6.4.3. Fusion Approaches to Avoid Diverging Distribution Shifts. As shown in the Section 6.4.2, the main factor for the large diverging distribution shifts of train and test samples in deep ensembles is the level of agreement. This is an inherent consequence of averaging the confidence of multiple base models. There are several ways to avoid outputting the average of base models. Confidence masking approaches can achieve this goal by manipulating the confidence

[3]Although this can be understood by analyzing the Eq. (6.1), it is difficult to demonstrate empirically. The reason is that these three factors are not independent, and, hence, our attempts to significantly change the third factor without changing the other two factors have been unsuccessful.



(a) Correct agreement distribution

(b) Distribution difference at each agreement level

(c) The effect of three factors on MI attack

FIGURE 6.3. AlexNet model trained on CIFAR10. **Left:** The distribution of the number of times a sample is correctly classified by 10 models used in the ensemble. The models often make less classification mistake on train samples than test samples. **Center:** By separating samples based on how many times they have been correctly classified, we can observe that the confidence output of these samples are negligible between train and test sets. Only when all samples are compared the distribution difference is significant and that is the direct effect of the third factor, namely the level of correct agreement. **Right:** The effect of the three factors on the MI attack. The values of y^c and y^m are confidence values in percentage. $c/10$ is the percentage of models that correctly classify a sample. As the gap between the level of correct agreement of train and test widens (the blue lines), the MI attack on ensembles becomes more effective than a single model (green lines).

values. The simplest form is to only output the class label (or the top k classes) [**110**], or to add a random noise to the confidence value [**52**]. The drawback of these approaches is that they make the confidence values unreliable which is critical for applications where confidence estimation is necessary.

To address this issue, we propose three methods to mitigate diverging distribution shift and output a valid confidence value. One simple approach is to output the confidence values of a single model among base models. However, if the model is chosen randomly, this approach does not provide the accuracy improvement of ensembling. To avoid this problem, we first use ensemble averaging over confidence outputs of base models, similar to deep ensembles, to obtain the *ensemble predicted label*. Then we output the confidence values of the first model that predicts the label as the ensemble predicted label. We call this approach *first agreed confidence*. Because the predicted label of the ensemble is essentially the same as deep ensembles, the accuracy of the first agreed label is exactly the same as deep ensembles. As shown in the third row of Figure 6.2, the distribution of the confidence output is similar to a single model (the first row of Figure 6.2), as expected.

Therefore, this approach can easily mitigate the privacy cost of deep ensembles and achieve the same accuracy.

Interestingly, there are simple approaches that not only output a valid confidence values and achieve similar accuracy, but also significantly improve the privacy. Instead of outputting the confidence value of the first model that predicts the ensemble predicted label, we output the confidence values of the most confident model among base models that predicts the ensemble predicted label. We call this approach *maximum agreed confidence*. Similarly, this approach also achieves the same accuracy as deep ensembles because the prediction label is the same. Maximum agreed confidence approach has multiple advantages: 1) it omits the effect of the level of agreement on the output, 2) it outputs a confidence value that reflects one of the base models and hence it is still reliable for the purpose confidence estimation, and 3) it shifts the confidence values of both member and non-member to the extreme ends (either 0 or 1) and, hence, the distributions of member and non-members become even less distinguishable. The last advantage is clearly shown in the last row of Figure 6.2. In Section 6.5.1, we demonstrate that this approach improves both accuracy and privacy at the same time.

Another similar approach is to output the confidence of the most confident model among all models, instead of the model that predicts the ensemble predicted label. We call this approach *maximum confidence*. The accuracy of maximum confidence might be slightly lower than deep ensembles because the most confident model may occasionally be the one that misclassifies the input sample although the majority of base models do not. However, as we show in Section 6.5.1, this approach mitigates membership inference attacks slightly better than maximum agreed confidence because it outputs high confidence for some incorrectly classified samples, which mostly belong to the nonmember samples [**100**].

Each of the three approaches is advantageous in different scenarios. The first agreed confidence approach outputs a confidence of a single model and, as shown in Figure 6.2, it is similar to a single model scenario. Hence, it is beneficial for scenarios where the confidence estimation is needed to be similar to a single model. The maximum confidence and maximum agreed confidence approaches change the overall confidence distribution. Although the confidence values still come from the output of one of the base models, it is not known if it is as useful as a single model

scenario when confidence estimation is concerned. This requires further investigation. Regardless of confidence estimation, the maximum agreed confidence achieves the highest accuracy, as of deep ensembles, while improving privacy. The maximum confidence, however, achieves highest privacy while improving accuracy. Therefore, depending on whether the objective is to maximize accuracy or to maximize privacy, one can use maximum agreed confidence or maximum confidence.

6.4.4. Why Does it Outperform Gap Attack Significantly? Recently, several studies report a simple baseline attack called *gap attack* [**15**], also known as *naive attack* [**66**, **100**] that achieves similar performance as the confidence-based attacks in most scenarios. The gap attack predicts a sample as member if it is correctly classified by the target model, and as non-member otherwise. In other words, gap attack essentially reflects the generalization gap [**100**]. In [**100**], authors extensively analyzed this phenomena in deep models and argued current MI attacks that barely outperform gap attacks are ineffective in practice because they only reflect the generalization gap and cannot infer the membership status of each individual sample accurately.

Figure 6.4 shows that the effectiveness of membership inference attacks increases and outperforms the gap attack as deep ensembles are deployed. This raises significant privacy concern since the gap attack is often suggested as a baseline that is also hard to outperform in non-ensemble setting [**15**, **66**, **100**]. Note that gap attack can be viewed as a metric directly reflecting the generalization gap rather than a reliable membership inference. As suggested in [**100**], we can separate correctly classified samples and misclassified samples to understand why membership inference attacks can barely outperform gap attack. As shown in Figure 6.2(b) and (c), the distributions of train and test samples are similar when separated into correctly classified and misclassified sets. The reason why the distribution of all samples (Figure 6.2(a)) looks more distinguishable when correctly classified samples and misclassified samples are combined is that there are often more misclassified samples in the test set than the train set. This is the information that gap attack exploits which essentially reflects the generalization gap. In order for a membership inference attack to considerably outperform the gap attack, the distribution of correctly classified samples and misclassified samples should leak membership status information, which is not often the case as it is shown in Figure 6.2(b) and (c), and [**100**].

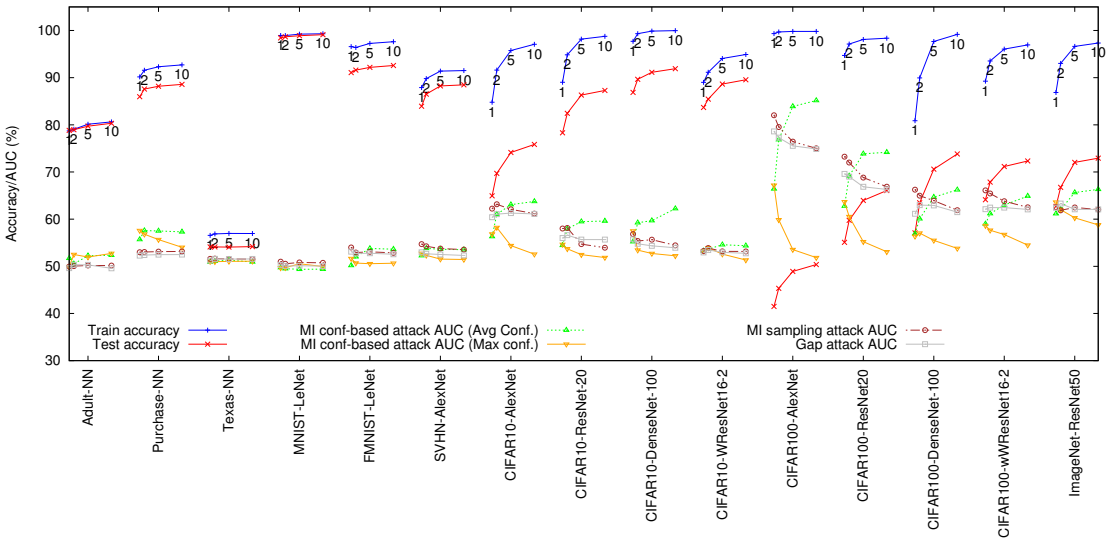


FIGURE 6.4. Membership inference attack results across all datasets/models. Each curve indicates an ensemble of 1 (non-EL), 2, 5, and 10 models from left to right. Green curves indicate original deep ensembles where confidence values are averaged. Yellow curves indicate an ensemble of same models using maximum agreed confidence. Note that train and test accuracy of both approaches are the same. MI sampling and Gap attack is conducted on original deep ensembles.

When ensembling is used, the distribution of confidence values changes dramatically, as explained in Section 6.4.1. By comparing the confidence distribution of correctly classified samples in an ensemble (Figure 6.2(e)) with a non-ensemble scenario (Figure 6.2(b)), the distribution is clearly more distinguishable in ensemble case. This is of significant privacy concern because, as discussed in [**100**], majority of samples in practice belong to the correctly classified set. Similar trend is also observable in misclassified samples (Figure 6.2(f)). Hence, the confidence values, that barely leak more information than generalization gap itself in a single model scenario, now considerably leak more membership information than just the generalization gap. That is the reason why membership inference attacks are significantly more effective in deep ensembles in comparison to the gap attack.

6.5. Experiments Results

6.5.1. Experimental Setup. We explore a wide range of datasets that are often used in deep ensemble literature or membership inference literature: Adult[4], Texas, Purchase [**110**], MNIST [**64**], FMNIST [**130**], SVHN [**90**], CIFAR10 [**61**], CIFAR100 [**61**], and ImageNet [**102**]. For non-image datasets (Adult, Purchase, and Texas), we use a fully connected neural network consisting of a hidden layer of size 128 and a Softmax layer. All other training parameters for these datasets are set as suggested in [**110**]. For image datasets, we use a wide range of convolutional neural networks depending on the input dimension and the difficulty of the task. We use the model implementations adopted in [**89**,**100**][5]. We train 10 models for each dataset with random initialization and construct an ensemble of 2, 5 and 10 models, respectively.

Attack models for Shokri and Watson attacks are NNs with three hidden layers of size 128, 128, and 64, respectively. In this section, we consider a scenario that is most advantageous to the attacker where 80% of the training dataset is given to the attacker and the goal is to infer the membership of the remaining samples, similar to [**100**]. This can be construed as an upper-bound for the confidence-based attacks that does not use difficulty calibration. We explore Shokri and Watson attacks in the next sections. For sampling attack, we perturb each sample 50 times and count the number of time the prediction label has changes, as in [**96**]. For ImageNet, we attack a set of samples including 50 member and nonmember images per class. We explore a random set of ten hyper-parameters, including the one proposed in [**96**], for the noise perturbation and report the highest attack performance. Here, we only report AUC of membership inference attacks. In practice, the attacker needs to train shadow models to estimate the best threshold value which may result in less accurate attack. All other training parameters are set as suggested in [**100**]. See Section 6.5.4 for more results. The results of the weighted averaging, and snapshot ensembles and diversified ensemble networks are shown in Section 6.5.6, and Section 6.5.5, respectively.

Figure 6.4 shows the results on all datasets. For some datasets, such as Adult, Texas, and MNIST, deep ensemble approach barely changes the accuracy or privacy. That is because the disagreement across models is insignificant in these datasets. For all other datasets, deep ensemble

[4]http://archive.ics.uci.edu/ml/datasets/Adult
[5]https://github.com/bearpaw/pytorch-classification

approach improves the accuracy (blue/red curves) as well as the effectiveness of confidence-based membership inference attacks (green curves). As mentioned in Section 6.4.2, the most salient factor in membership inference effectiveness on deep ensembles is the accuracy gap between train and test set. Figure 6.4 clearly shows that whenever this generalization gap is large for non-ensemble case, the attack improvement is significant using ensembling. It is worth noting that the ensembling can often reduce the generalization gap and the effectiveness of the gap attack (e.g., CIFAR10-DenseNet-100, CIFAR100-AlexNet, or CIFAR100-ResNet20). However, due to the reasons explained in Section 6.4.4, the membership inference effectiveness still increases.

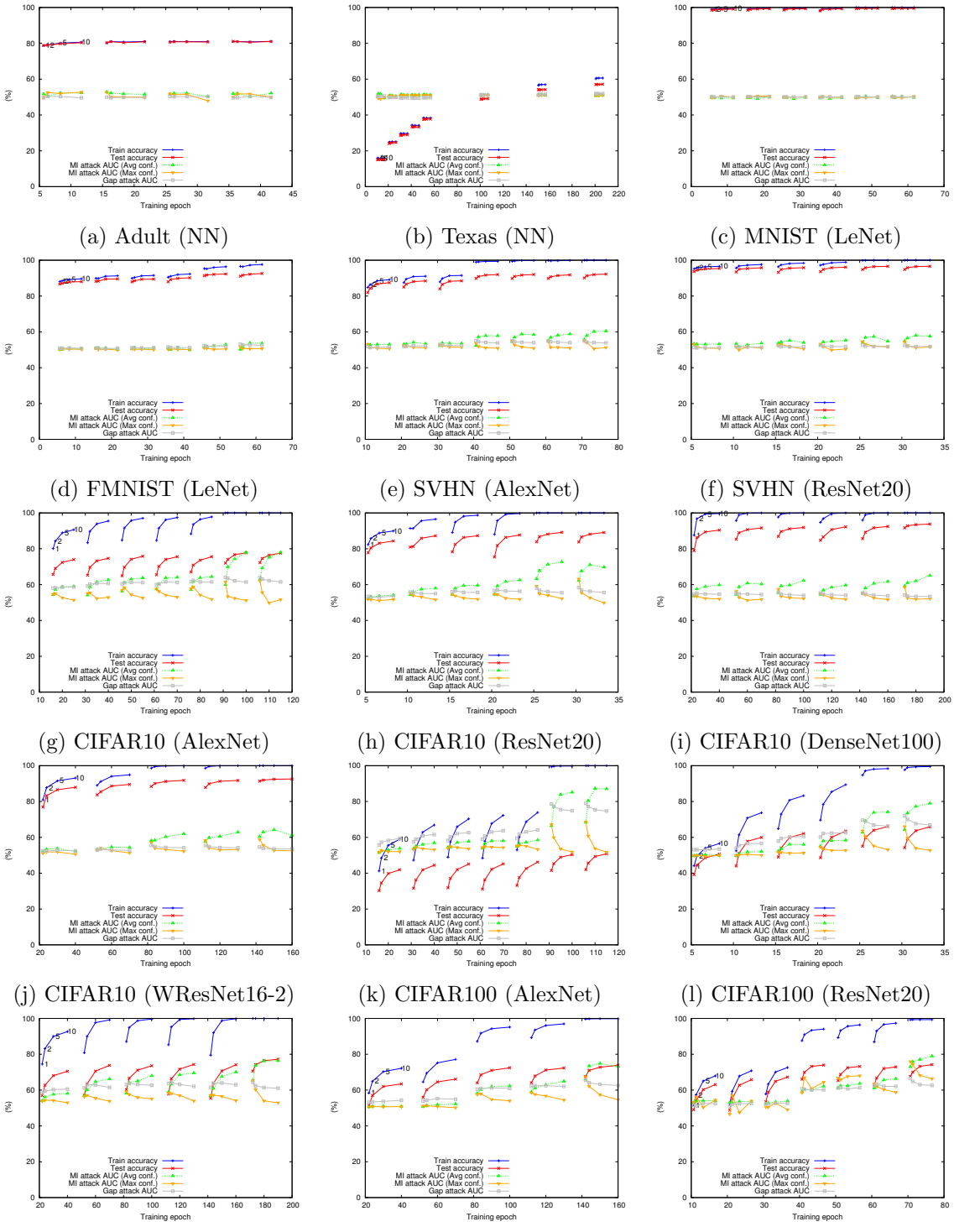
Interestingly, the effectiveness of sampling attack, unlike confidence-based attacks, often decreases on deep ensembles (brown curves). The main reason is that deeps ensembles are more robust than a single model, as shown in [**70**, **132**]. Therefore, perturbing target samples to obtain information about its membership status is less effective in deep ensembles. Another observation is that maximum agreed confidence ensembling (yellow curves) can considerably mitigate the effectiveness of confidence-based membership inference.

Figure 6.5 demonstrates the improvement of accuracy and MI attack over various training epochs. For datasets that ensembling outperforms a single model, using an ensemble of underfitted models is less prone to MI attack. However, it leads to lower accuracy. Due to the computational cost of running sampling attack, we only perform the sampling attack on certain epochs in Figure 6.4.

As discussed in previous sections, the less the generalization gap of base models are, the less effective the membership inference would be on the deep ensemble. Therefore, any standard regularization technique can potentially work as a defense mechanism. In this chapter, we study L1 and L2 regularization, DP-SGD[6] ($\epsilon \approx [3, 5]$ and $\delta = 10^{-5}$), and MMD+Mixup [**68**]. For defense mechanisms, we use L1 and L2 regularization with 0.01 and 0.1 as a weight of the loss function, respectively. With only a single mode, we achieve 42% and 60% accuracy using DP-SGD. The reason model accuracy is slightly lower than the literature [**117**] is due to the fact that we train victim models with half of the training data and we use AlexNet model [7]. The other half is reserved

[6]We use Opacus implementation: https://github.com/pytorch/opacus.

[7]We observe patterns similar to Figure 6.11 using CNN implementation of [**117**] and other models. More results are available in Section 6.5.4.



(a) Adult (NN) (b) Texas (NN) (c) MNIST (LeNet)

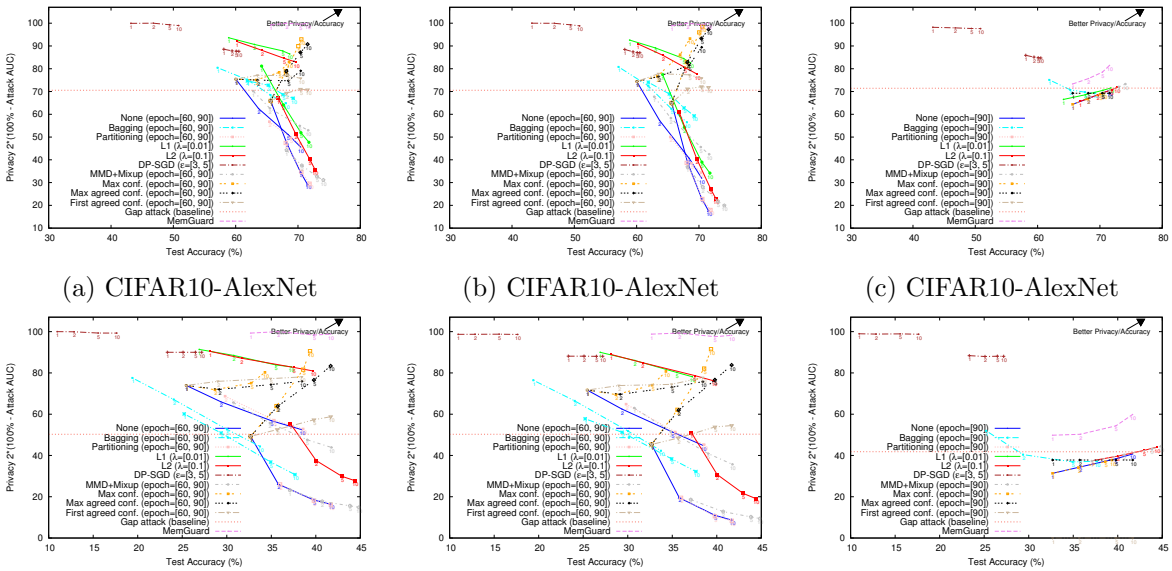
(d) FMNIST (LeNet) (e) SVHN (AlexNet) (f) SVHN (ResNet20)

(g) CIFAR10 (AlexNet) (h) CIFAR10 (ResNet20) (i) CIFAR10 (DenseNet100)

(j) CIFAR10 (WResNet16-2) (k) CIFAR100 (AlexNet) (l) CIFAR100 (ResNet20)

(m) CIFAR100 (DenseNet100) (n) CIFAR100 (WResNet16-2) (o) ImageNet (ResNet50)

FIGURE 6.5. Target models' accuracy and MI attacks' AUC across all datasets and models.



(a) CIFAR10-AlexNet (b) CIFAR10-AlexNet (c) CIFAR10-AlexNet

(d) CIFAR100-AlexNet (e) CIFAR100-AlexNet (f) CIFAR100-AlexNet

FIGURE 6.6. Effect of defense mechanisms on an AlexNet model trained on CIFAR10 and CIFAR100. The size of each point indicates the relative value of its parameter. The horizontal pink line indicates the performance of gap attack on a deep ensemble of 10 models. The first, second, and third columns represent Shokri, Watson, and sampling attacks, respectively.

for MI attacks. Furthermore, one can simply terminate training early to keep the model's weights in less overfitted region. Moreover, some ensembling techniques, such as bagging, and partitioning, limits the access of models to all training samples, which can potentially reduce membership inference effectiveness on deep ensembles. Moreover, see Section 6.5.6 for weighted averaging ensembles. We also evaluate two state-of-the-art deep ensembling approaches, namely snapshot ensemble and diversified ensemble network, in Section 6.5.5.

In the section, we show the effectiveness of Shokri, Watson, and sampling attacks on all defense mechanisms. We divide the training set of each dataset into two disjoint sets: victim's training data, D_{tr}, and attacker's (shadow) training data, D_s. We use D_{tr} to train base models for the victim's ensemble and D_s to train 10 shadow models for Shokri and Watson attacks. For the Shokri attack, we train 10 shadow models with the same architecture and training hyper-parameters as victim models. Note that it has shown that Shokri attack's accuracy barely changes even if the architecture and training hyper-parameters of shadow models do not match the victim model [**105**]. For the Watson attack, we use the Shokri's attack output as the base membership inference score and we

calibrate it using 10 other shadow models trained on D_s, as explained in [**126**]. The sampling attack configuration is similar to Section 6.5.1. Due to the lack of space, we only show the accuracy/privacy trade-off of AlexNet model trained on CIFAR10 and CIFAR100. See Section 6.5.4 for more results.

Confidence-based attacks: Figure 6.6 shows the effect of defense mechanisms on ensemble learning. The pink dashed line indicates the performance of gap attack on a deep ensemble of 10 models with no regularization. Hence, any point below this line means privacy leakage greater than a trivial baseline (Gap attack). Although Watson attack is generally more effective due to its difficulty calibration, both Shokri and Watson attacks change similarly when used against deep ensembles. We can observe a consistent trade-off between ensemble accuracy and privacy that resembles Pareto optimal points. The only exceptions are our proposed approaches, namely maximum agreed confidence, maximum confidence, and first agreed confidence. The difference between maximum agreed confidence and maximum confidence is marginal. The former achieves slightly better accuracy, while the latter achieves slightly better privacy.

Interestingly, none of the approaches that applies modification during the training of the base models could break the trade-off, including bagging, partitioning, L1/L2 regularizations, DP-SGD, and MMD+Mixup. Note that privacy degradation rate for these approaches is clearly not constant. An ensemble of heavily regularized models or under-fitted models barely causes more privacy leakage (e.g., L2 regularization at epoch 60). On the other hand, an ensemble of overfitted models (e.g., non-regularized models trained for 90 epochs) results in large privacy leakage. It is worth mentioning that some approaches are sometimes outperform the original deep ensemble both in terms of accuracy and privacy despite being bounded to the trade-off. For instance, an ensemble of n deep models with L2 regularization (red curves) can often outperform the deep ensemble of n models (blue curves) both in terms of accuracy and privacy. However, it still manifests the trade-off in a sense that increasing the number of L2 regularized models in an ensemble increases the accuracy while decreasing the privacy (in terms of confidence-based MI attack).

Sampling attack: The sampling attack is different from confidence-based membership inference attacks because it does not follow the accuracy-privacy trade-off, as shown in the last row of Figure 6.6. In fact, the effectiveness of the sampling attack decreases in most ensemble learning approaches as more base models are added to the ensemble. As shown in [**96**], the most effective

TABLE 6.1. Comparison of different defence mechanisms with respect to true positive in low false positive regime and average per-sample distortion to the confidence output. This table only includes Watson attack.

Dataset	-	TPR @ 0.001 @ FPR				TPR @ 0.1 @ FPR				Avg per-sample conf. distortion			
-	Ensemble size:	1	2	5	10	1	2	5	10	1	2	5	10
CIFAR10	None (deep ensemble)	0.08%	0.08%	1.16%	2.20%	0.78%	2.42%	4.88%	6.56%	0.0	0.0	0.0	0.0
	Bagging	0.08%	0.19%	0.48%	0.52%	0.78%	0.71%	1.34%	1.72%	0.0	0.18	0.16	0.15
	Partitioning	0.08%	0.09%	1.09%	2.21%	0.78%	1.65%	3.51%	6.77%	0.0	0.07	0.05	0.03
	L1 (0.01)	0.00%	0.00%	0.00%	0.00%	0.21%	0.50%	2.24%	3.02%	0.15	0.13	0.11	0.10
	L2 (0.1)	0.00%	0.21%	0.70%	0.70%	0.55%	1.50%	2.15%	3.10%	0.09	0.07	0.05	0.04
	MMD+Mixup	0.00%	0.56%	0.44%	1.30%	1.90%	3.04%	3.36%	3.15%	0.19	0.18	0.16	0.15
	DP-SGD	0.00%	0.00%	0.00%	0.00%	0.05%	0.04%	0.04%	0.09%	0.49	0.48	0.46	0.46
	MemGuard	0.00%	0.00%	0.00%	0.00%	0.00%	0.0%	0.00%	0.00%	0.37	0.38	0.37	0.37
	Max conf.	0.08%	0.10%	0.0%	0.0%	0.78%	0.32%	0.24%	0.18%	0.0	0.04	0.06	0.07
	Max agreed conf.	0.08%	0.10%	0.00%	0.00%	0.78%	0.32%	0.22%	0.19%	0.0	0.04	0.06	0.06
	First agreed conf.	0.08%	0.08%	0.00%	0.00%	0.78%	0.58%	0.58%	0.29%	0.0	0.04	0.05	0.06
CIFAR100	None (deep ensemble)	0.10%	0.14%	0.21%	0.21%	0.54%	1.52%	3.67%	5.10%	0.0	0.0	0.0	0.0
	Bagging	0.10%	0.12%	0.06%	0.04%	0.54%	1.04%	2.02%	4.52%	0.0	0.38	0.33	0.31
	Partitioning	0.10%	0.08%	0.20%	0.20%	0.54%	1.21%	5.76%	5.84%	0.0	0.17	0.13	0.10
	L1 (0.01)	0.00%	0.00%	0.00%	0.00%	0.08%	0.08%	0.08%	0.08%	0.95	0.95	0.93	0.93
	L2 (0.1)	0.04%	0.04%	0.54%	0.52%	0.43%	1.43%	2.64%	4.47%	0.23	0.21	0.17	0.15
	MMD+Mixup	0.13%	0.10%	0.10%	0.09%	2.21%	4.47%	7.02%	6.97%	0.28	0.26	0.23	0.21
	DP-SGD	0.00%	0.00%	0.00%	0.00%	0.10%	0.08%	0.15%	0.12%	0.87	0.85	0.83	0.82
	MemGuard	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.40	0.39	0.38	0.38
	Max conf.	0.10%	0.02%	0.00%	0.00%	0.54%	0.23%	0.15%	0.09%	0.0	0.10	0.15	0.17
	Max agreed conf.	0.10%	0.02%	0.00%	0.03%	0.54%	0.23%	0.15%	0.13%	0.0	0.10	0.15	0.16
	First agreed conf.	0.10%	0.01%	0.00%	0.01%	0.54%	0.25%	0.11%	0.20%	0.0	0.10	0.13	0.14

defense for sampling attack is DP-SGD. However, as the number of base models increases, the performance of sampling attack degrades to a point where it is often worse than the trivial gap attack. Hence, a deep ensemble with maximum confidence can effectively improve both accuracy and privacy.

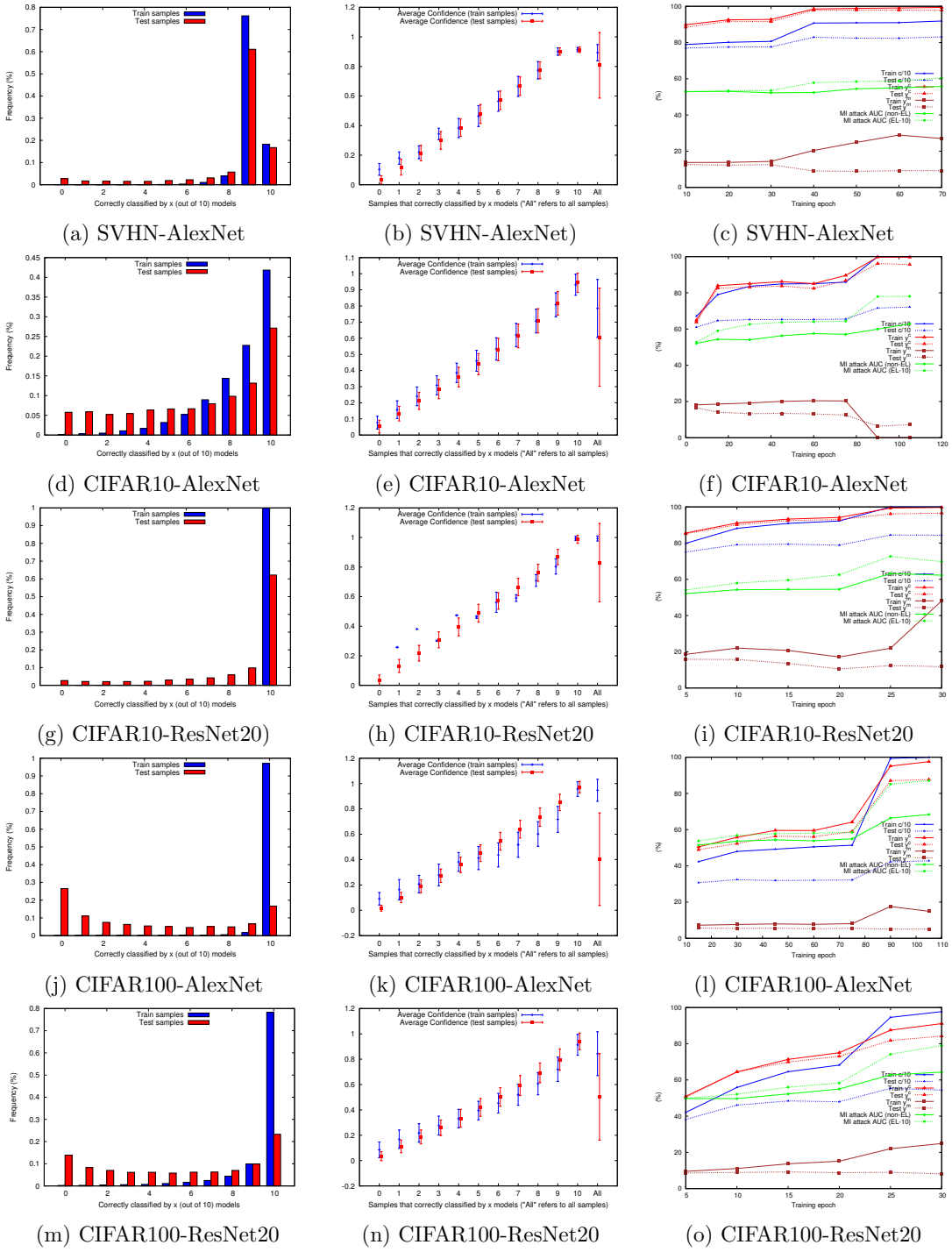
Although AUC has been overwhelmingly used to report the performance of MI attacks, [**7**] first argued that a more reliable metric is true positive at low false positive rate. In Table 6.1 we present this metric for Watson attack. We do not report Shokri and sampling attacks here because their true positive was almost zero for low false positive rate, as also shown in [**7**].

As mentioned earlier, confidence-masking defenses heavily distort the confidence output of models causing an issue for applications that rely on real confidence values, such as uncertainty estimation. In Table 6.1, we present how much distortion each defence mechanism imposes to the original confidence values. Here, for each data sample, we compute the L1 distance between the original deep ensemble confidence values and the defence mechanism's confidence output. Then, we normalize the values to be between 0 and 1 and then take the average. As shown in Table 6.1,

MMD+Mixup, DP-SGD, and MemGuard are heavily distorting the confidence outputs while our proposed approaches impose smaller distortion.

6.5.2. Level of Correct Agreement. As discussed in section 6.4.2, overfitted models tend to disagree more on test samples than train samples. In other word, the distribution of agreement for train and test sets becomes more distinguishable as models overfit. This distinction is more clear for datasets, such as CIFAR10 and CIFAR100, which shows most improvement when ensembling is used, as shown in Figure 6.7. Furthermore, the level of agreement can reveal if an ensemble can actually improve prediction. If all models correctly classify a sample or all models misclassify a sample, ensembling fails to outperform a single model. Due to the lack of space, we omit the results of other datasets/models.

In this section, we average logits (the output of a model before Softmax) of NN models instead of the confidences. We can still observe that ensembling leaks more membership status than non-ensemble scenario. However, the MI attacks with average confidence (Figure 6.5), in general, are slightly more effective than MI attacks with average logits (Figure 6.8). The reason is that that confidence values are normalized and, hence, when aggregated all models have the same contribution to the overall confidence output of the ensemble. However, when logit is used, the confidence output of the ensemble is more influenced by highly activated neurons. These highly activated neurons, which often belong to the correctly classifying models, has significantly more influence on the confidence output of ensemble in comparison with lightly activated neurons of misclassifying models. Hence, the confidence output is heavily influenced by only a portion of models in ensemble that have high activation neurons. In other words, it can be seen as an ensemble of only a portion of models, not all models in the ensemble. Since ensembling with fewer models leaks less membership status, logit averaging of n models leak membership status than confidence averaging of the same number of models. Note that logit averaging is still prone with the same degree to membership leakage in a white-box attack since a white-box attacker has access to all confidence values. Note that the consequence of using logit in certain applications, such as confidence estimation, that requires reliable confidence estimation is out of the scope of this paper. Moreover, one major drawback of using logit is that it can be arbitrary scaled [**123**]. However, in scenarios where only



(a) SVHN-AlexNet (b) SVHN-AlexNet) (c) SVHN-AlexNet

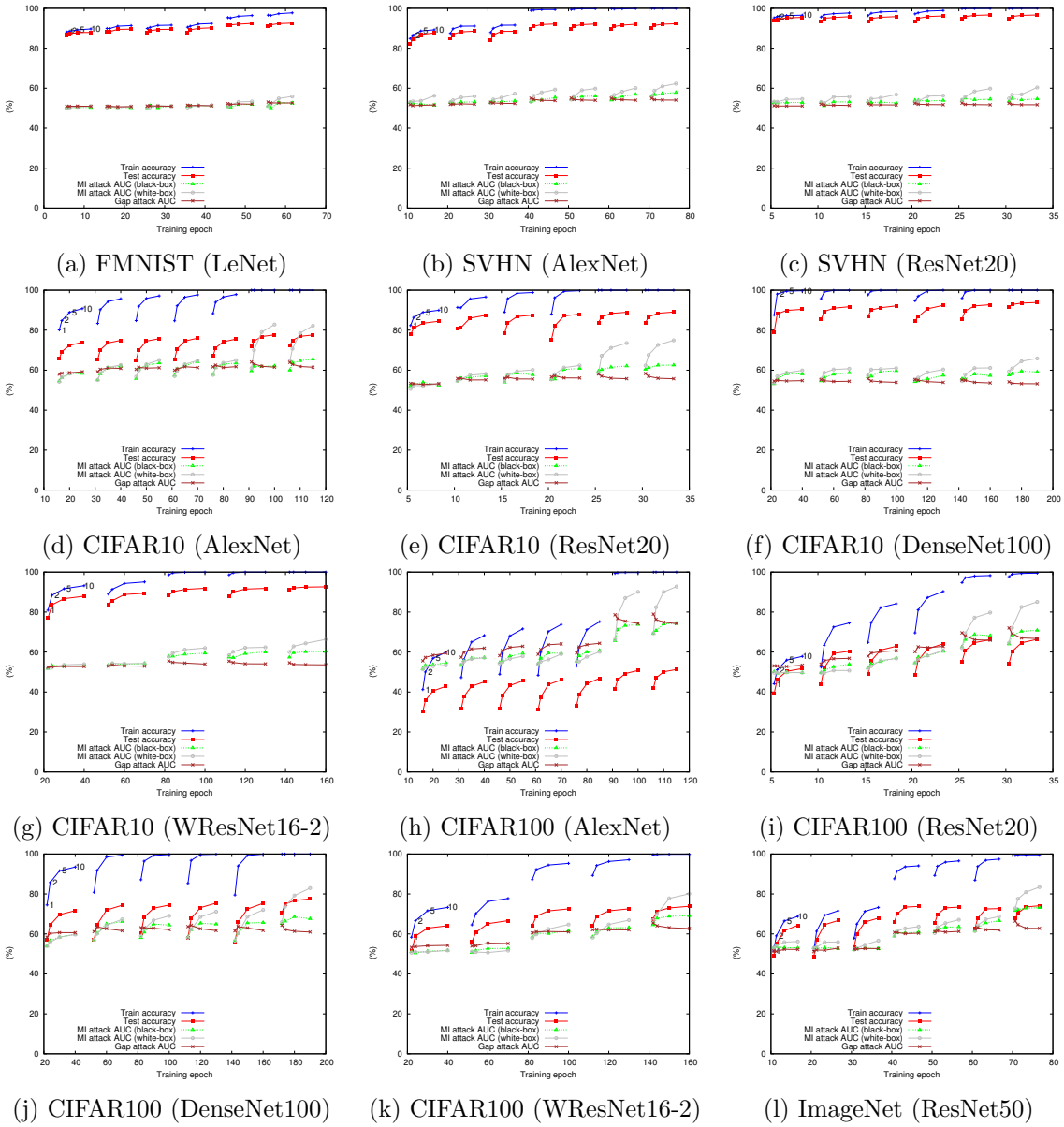
(d) CIFAR10-AlexNet (e) CIFAR10-AlexNet (f) CIFAR10-AlexNet

(g) CIFAR10-ResNet20) (h) CIFAR10-ResNet20 (i) CIFAR10-ResNet20

(j) CIFAR100-AlexNet (k) CIFAR100-AlexNet (l) CIFAR100-AlexNet

(m) CIFAR100-ResNet20 (n) CIFAR100-ResNet20 (o) CIFAR100-ResNet20

FIGURE 6.7. The first column contains correct agreement distribution. The second column shows the average and standard deviation of distribution of samples based on the level of correct agreement. The third column shows the effect of the three factors on the MI attack.



(a) FMNIST (LeNet) (b) SVHN (AlexNet) (c) SVHN (ResNet20)

(d) CIFAR10 (AlexNet) (e) CIFAR10 (ResNet20) (f) CIFAR10 (DenseNet100)

(g) CIFAR10 (WResNet16-2) (h) CIFAR100 (AlexNet) (i) CIFAR100 (ResNet20)

(j) CIFAR100 (DenseNet100) (k) CIFAR100 (WResNet16-2) (l) ImageNet (ResNet50)

FIGURE 6.8. Target models' accuracy and MI attacks' AUC across all datasets and models. Here, MI attack model uses aggregated logits instead of aggregated confidences.

accuracy is concerned and white-box access is unavailable to the attacker, averaging logits seems to have a better privacy protection of training data than averaging confidences.

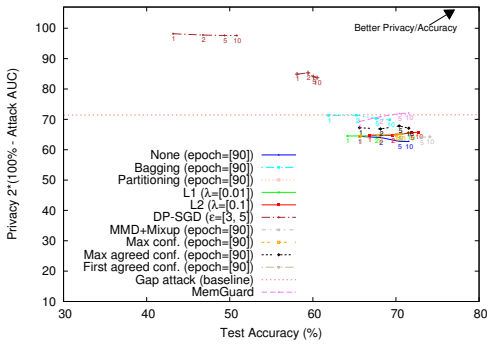


FIGURE 6.9. label only attack on CIFAR10-AlexNet model.

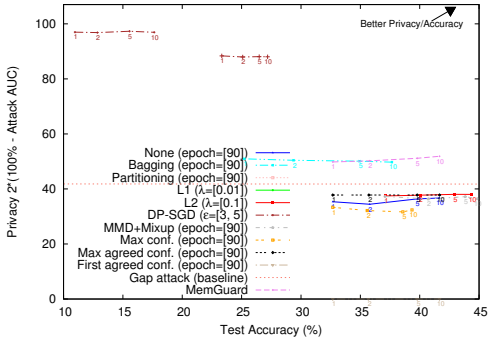
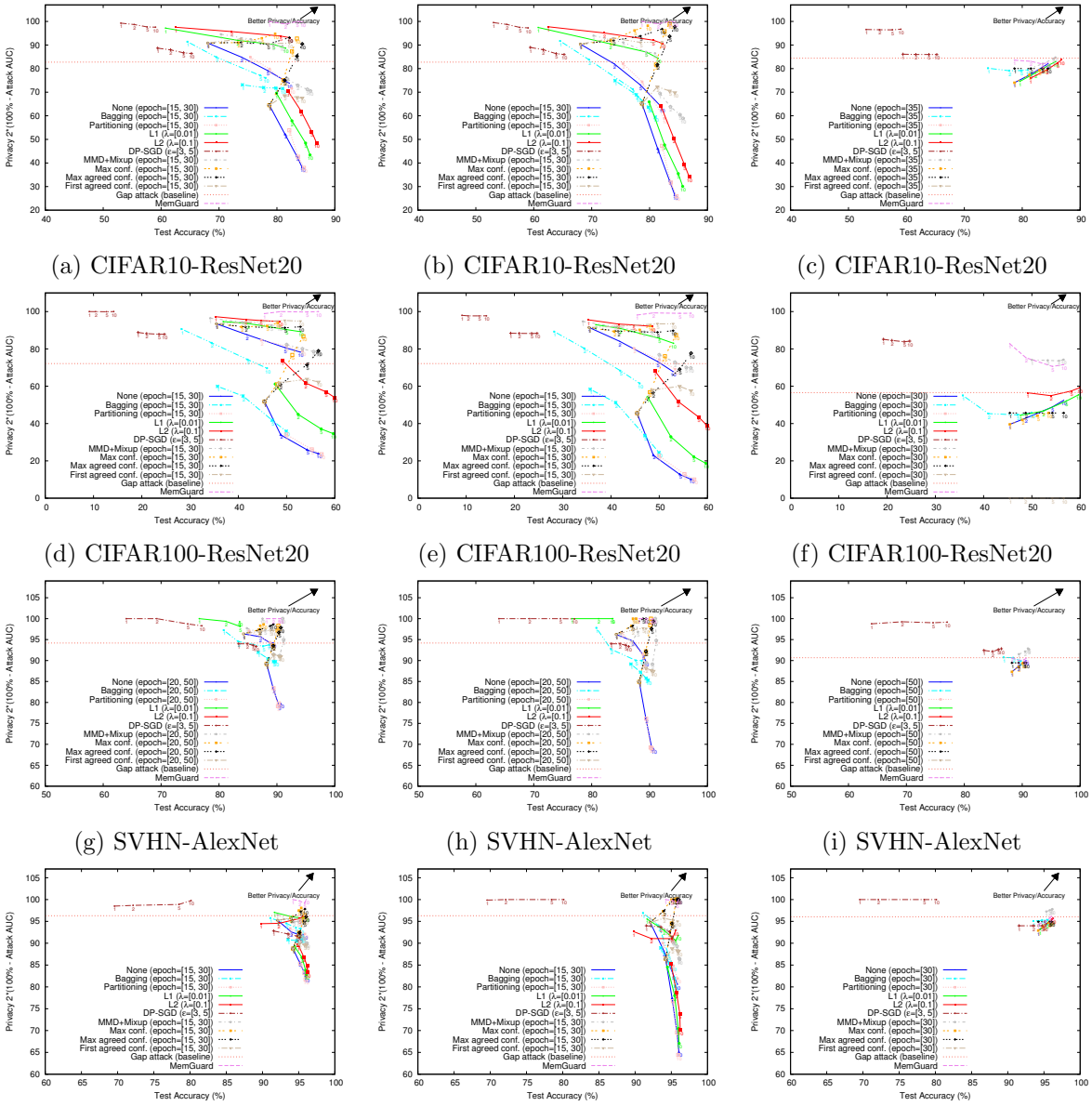


FIGURE 6.10. label only attack on CIFAR100-AlexNet model.

6.5.3. Label Only Attack. In this section, we analyze the effect of label-only MI attack proposed in [**16**]. The idea is to use an adversarial example generator to find the distance to the decision boundary as membership inference metric. They use "HopSkipJump" [**12**] from Cleverhans[8] to craft adversarial examples. We use the same algorithm, implementation and hyper-parameters. The results are shown in Figure 6.9 and 6.10. We have not seen significant difference in terms of MI attack AUC when ensembling is used. However, we observe that by increasing the number of models in an ensemble, the HopSkipJump used in label only attack becomes less effective in finding adversarial samples in the specified number of iterations. Due to the limited time and computational budget, we have not explored stronger approaches. Using more iterations and a stronger adversarial attack may leads to a different result.

[8]https://github.com/cleverhans-lab/cleverhans



(a) CIFAR10-ResNet20 (b) CIFAR10-ResNet20 (c) CIFAR10-ResNet20

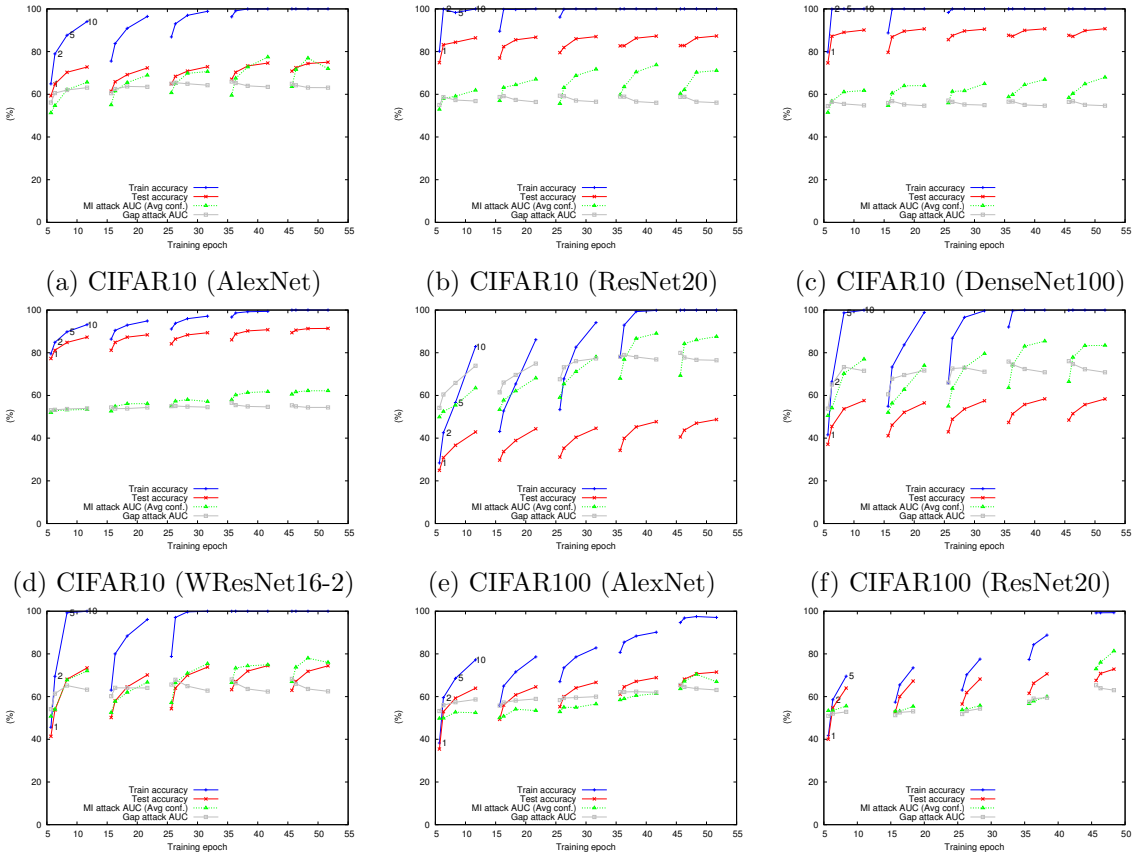
(d) CIFAR100-ResNet20 (e) CIFAR100-ResNet20 (f) CIFAR100-ResNet20

(g) SVHN-AlexNet (h) SVHN-AlexNet (i) SVHN-AlexNet

(j) SVHN-ResNet20 (k) SVHN-ResNet20 (l) SVHN-ResNet20

FIGURE 6.11. Effect of defense mechanisms. The first, second, and third columns represent Shokri, Watson, and sampling attack, respectively.

6.5.4. Defense Mechanism. The effect of all defense mechanisms are shown in Figure 6.11. Most defense mechanisms become less effective when deep ensemble is used. As shown Section 6.5, maximum agreed confidence and maximum confidence achieve the best accuracy and privacy.



(a) CIFAR10 (AlexNet) (b) CIFAR10 (ResNet20) (c) CIFAR10 (DenseNet100)

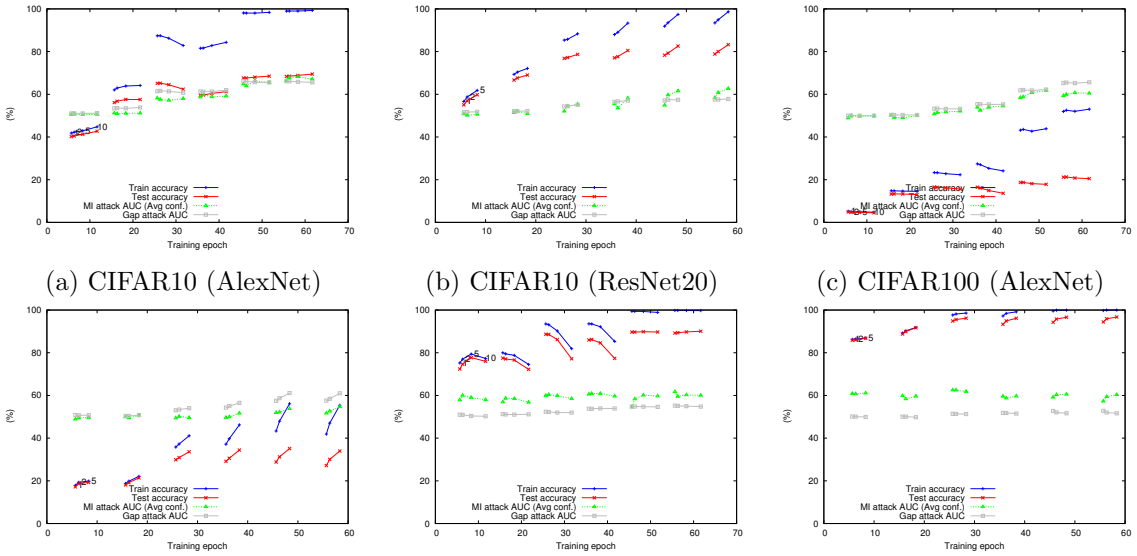
(d) CIFAR10 (WResNet16-2) (e) CIFAR100 (AlexNet) (f) CIFAR100 (ResNet20)

(g) CIFAR100 (DenseNet100) (h) CIFAR100 (WResNet16-2) (i) ImageNet (ResNet50)

FIGURE 6.12. Target models' accuracy and MI attacks' AUC across all datasets and models using snapshot ensemble [**45**].

6.5.5. More Advanced Ensembling Approaches. In this section, we evaluate two state-of-the-art ensembling approaches, namely snapshot ensembles [**45**] and diversified ensemble networks [**139**]. For snapshot ensemble, we train several models on several datasets for 500 epochs and restart the cycle every 50 epochs, similar to the original paper [**45**]. Note that the goal of our evaluation is show the accuracy-privacy trade-off, not to achieve the highest accuracy possible. Due to this reason and limited time we had, we did not perform an exhaustive hyper-parameter tuning. Nevertheless, similar trade-off can be observed in Figure 6.12.

We also conduct the same experiment with diversified ensemble networks [**139**]. The original paper used pre-trained VGG and ResNet models. We did not use pre-trained models for two reasons: 1) It make comparison with other approaches unfair, and 2) It may interfere with the membership

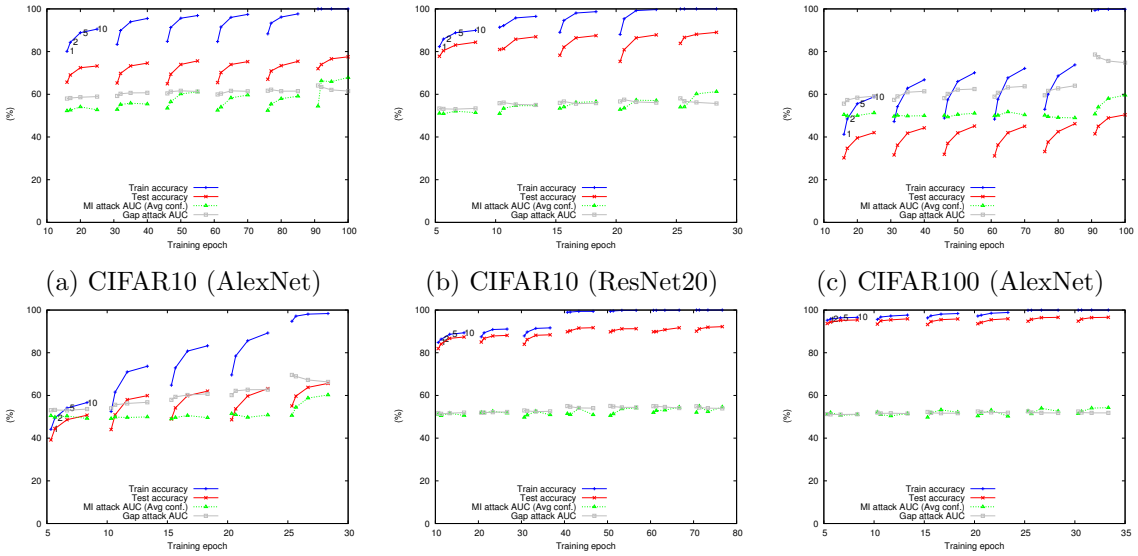


(a) CIFAR10 (AlexNet) (b) CIFAR10 (ResNet20) (c) CIFAR100 (AlexNet)

(d) CIFAR100 (ResNet20) (e) SVHN (AlexNet) (f) SVHN (ResNet20)

FIGURE 6.13. Target models' accuracy and MI attacks' AUC across all datasets and models using diversified ensemble networks [**139**].

inference analysis. We find that by using randomly initialized models to start training, L_d varies significantly and prevents the optimization to converge. Therefore, we add a weight to the L_d term to reduce its effect on the entire loss. We use 0.01 for CIFAR10 and SVHN and 0.001 for CIFAR100. For the shared layer, we use a fully-connected layer of size 128 followed by batch normalization and ReLU activation. We use SGD to train models for 60 epochs while dropping the learning at each 20 epochs by 0.1. We could not achieve the exact same results as reported in the paper for two main reasons: 1) we did not use pre-trained models in the ensemble, and 2) many hyper-parameters and implementation details are not reported in the original paper. We could not find a set of hyper-parameters and conditions to consistently achieve higher accuracy when increasing the number of models. This was also reported in the original paper where they found that more models in the diversified ensemble do not always improve accuracy. One penitential reason is that training base models in diversified neural networks are not independent. This means when the number of models in the diversified neural network is increased, there are significantly more parameters to train, but the number of epochs is constant. So, it is expected that diversified neural network with less models can sometimes outperform diversified neural network with more models if the number



(a) CIFAR10 (AlexNet) (b) CIFAR10 (ResNet20) (c) CIFAR100 (AlexNet)

(d) CIFAR100 (ResNet20) (e) SVHN (AlexNet) (f) SVHN (ResNet20)

FIGURE 6.14. Target models' accuracy and MI attacks' AUC across all datasets and models using weighted averaging ensemble networks.

of training epoch is fixed. Nevertheless, as shown in Figure 6.13, in cases where accuracy increases, the membership inference attack effectiveness also increases.

6.5.6. Weighted Averaging. In this section, we evaluate weighted averaging of deep models. We focused on image CIFAR10, CIFAR100, and SVHN datasets. We trained each model with random initialization and all hyper-parameters similar to Section 6.5.1. Here, we use stochastic gradient descent (SGD) using the entire training set to learn the weight associated with each model. As shown in Figure 6.14, we observed similar accuracy-privacy trade-off.

6.6. Discussion

We note some limitations of our empirical analysis and opportunities for future work. First, privacy is a multi-faceted concept and can be defined or quantified in several ways. In this work, we quantified privacy leakage in terms of the effectiveness of membership inference attacks. One can quantify privacy in terms of other relevant attacks such as model inversion [**26**, **35**], property inference [**2**, **28**], and model stealing attack [**118**] as well as formally-provable measures such as differential privacy [**22**].

Second, ensemble learning is an umbrella term covering a wide variety of methods to combine multiple base learners. Although the ensemble learning is the most widely-used approach in deep learning, an arbitrary method of training and combining base learners can still be construed as ensemble learning while improving privacy. We mainly focused to deep ensembles because of its prevalent use for ensembling deep models. Moreover, we conduct experiment over several other ensembling approaches and the conclusion remain the same. A more exhaustive experimental evaluation may discover new results which is out of scope of this dissertation.

Third, this chapter focuses on black-box attack scenario and our solution to accuracy-privacy trade-off in deep ensembles relies on changing the fusing part of ensemble learning. In other words, base models in the ensemble are intact. Consequently, if all base models are publicly available in a white-box setting, an attacker can still average the outputs and bypass the maximum agree confidence mechanism. One solution can focus on training models sequentially (boosting) and applying some non-trivial criteria during training of each model to force the distribution of correct agreement to be close for train and non-train samples. How to achieve this is not trivial and needs further research.

6.7. Conclusion

In this chapter, we investigate membership inference attacks in deep ensemble learning and demonstrate that there exists a trade-off between accuracy and privacy. We show that the most influential factor that causes more effective membership inferences attack against deep ensembles is the level of agreement between base models. We illustrate the effect of several classical regularization techniques, including L1/L2 regularization and DP-SGD, to mitigate membership inference attacks and conclude that none of them can break the trade-off and improve the accuracy and privacy, simultaneously. Finally, we propose a simple yet highly effective solution that only changes ensemble's fusion post-training.

CHAPTER 7

On the Discredibility of Membership Inference Attacks

Although the first generation of MI attacks (MI attacks without difficulty calibration) has been proven to be ineffective in practice, a few recent studies proposed practical MI attacks that achieve reasonable true positive rate at low false positive rate. The question is whether these attacks can be reliably used in practice. We showcase a practical application of membership inference attacks where it is used by an auditor (investigator) to prove to a judge/jury that an auditee unlawfully used sensitive data during training. Then, we show that the auditee can provide a dataset (with potentially unlimited number of samples) to a judge where MI attacks catastrophically fail. Hence, the auditee challenges the credibility of the auditor and can get the case dismissed. More importantly, we show that the auditee does not need to know anything about the MI attack neither a query access to it. In other words, all currently SOTA MI attacks in literature suffer from the same issue. Through comprehensive experimental evaluation, we show that our algorithms can increase the false positive rate from ten to thousands times larger than what auditor claim to the judge. Lastly, we argue that the implication of our algorithms is beyond discredibility: Current membership inference attacks can identify the memorized subpopulations, but they cannot reliably identify which exact sample in the subpopulation was used during training.

7.1. Introduction

The wide-spread deployment of machine learning in various applications that sometimes deal with sensitive data, such as health records and personal information, has raised concerns about the leakage of sensitive training data post-deployment. Recently, a few studies suggest that machine learning models memorize the training data [**134**] and, consequently, various attacks, called membership inference (MI), have been proposed to identify the training samples [**7**, **51**, **67**, **71**, **75**, **76**, **98**, **101**, **103**, **105**, **110**, **114**, **119**]. Due to its simplicity, membership inference attacks have become a standard way to evaluate the privacy risk of machine learning models [**7**, **87**].

Recent studies have shown that the evaluation of such models using average-case success metrics is misleading [**100**]. Specifically, a trivial random guess adjusted using the generalization gap, called *gap attack* [**15**] or *naive attack* [**66**, **100**], has shown to achieve similar performance as many membership inference attacks. Moreover, as argued in [**7**] and [**77**], privacy is not an average case metric and a pragmatic approach should avoid relying on such metrics. To better demonstrate the privacy risk of a model, true positive rate at low false positive rate is suggested in [**7**] as used in various area of computer security [**42**, **55**, **59**, **80**]. Using the true positive rate at low false positive rate has revealed that many de facto membership inference attacks, such as [**51**, **110**, **134**], catastrophically fail. Only the state-of-the-art MI attacks that use some form of sample difficulty calibration [**126**], such as [**7**, **98**, **103**, **126**], can identify some training samples at low false positive rates.

Contributions. In this chapter, we aim to answer the following question: *Can membership inference attacks be reliably used in practice?* To answer this question, we first introduce a useful potential application of MI attacks for the purpose of auditing. We demonstrate that all contemporary membership inference attacks suffer from *discredibility*. Then, we generalize our findings beyond this auditing application and argue about the inaccuracy of current attacks for *record-level membership inference*. Our findings suggest that current attacks may better suited for *subpopulation-based membership inference*.

Specifically, in the auditing application we propose in this chapter, MI attacks are used as an auditing tool to investigate unlawful use of sensitive training data by a model trainer. Here, an auditor aims to prove to the judge/jury that private data has been unlawfully used by the auditee under investigation. The auditor uses a membership inference attack, and report the performance of the MI attack model along the samples labeled as members (at low false positive rate) to the judge. We show that the auditee can provide an unlimited number of non-member samples to the judge for which the MI attack model constantly fail, without knowing anything about the MI attack or having query access to it. We call this process *discredibility*. Discredibility allows auditee to seriously damage the credibility of MI attack model used by the auditor and, consequently, get the case dismissed.

The intuition for our discredibility approach is that samples semantically close to member samples, with respect to the latent representation, are likely to be identified as members by membership inference attacks. We provide an explanation about how such intuition arises by looking at ReLU neural networks as deterministic functions with locally linear property. Based on the intuition, we propose three algorithms to create a *discrediting dataset*, a dataset for which the false positive of MI attacks are significantly large: 1) searching through a public dataset, 2) crafting semantically similar samples to the target sample using a generative model, and 3) adversarially perturbing a non-member sample to embody the semantic representation of a member sample.

We systematically evaluate our discrediting algorithms over multiple datasets and models. We demonstrate that our approach can increase the false positive rate up to several thousand times more than the claimed low rate for SOTA algorithms. Our algorithms can even increase the false positive rates of older approaches, such as [**110**, **134**]. Nevertheless, the results are less significant because they cannot achieve low false positive rate in the first place to start with.

New Insights. We analyze the two hypotheses implicitly used as building blocks of our discrediting algorithms. The two hypotheses establish a positive correlation between the membership score of a member sample and its neighboring samples, and also a positive correlation between the semantic closeness of two neighbors and their membership scores. Although we start with a potential application of MI attacks in the auditing scenario, these two hypotheses are valid regardless of the application scenario. These two findings suggest that the current MI attacks are more reliable in identifying *memorized subpopulations* than individual samples. To simply put, MI attacks are prone to incorrectly classifying nonmember samples in the neighborhood of member samples as members.

Implications. The new insight, that current MI attacks are identifying memorized subpopulations, undermines the reliability of using MI in real applications. Hence, a new generation of attacks are needed to achieve record-level membership inference. However, this insight implies a new potential direction for MI attacks. It suggests that current *"record-level" MI attacks* are in fact better at *"subpopulation-level" membership inference*. For example, in face recognition where

each subpopulation likely represents a user, current MI attacks may achieve better user-level membership inference than record-level membership inference. This new adoption of MI attacks needs further investigation.

7.2. Threat Model

To better manifest the potential application of membership inference in practice, we showcase a scenario in a trial, where MI is used as an auditing tool to demonstrate the unlawful use of private data. Our threat model consists of three actors: an **auditor**, or attacker in the MI literature, an **auditee**, or MI defender whose model is under MI attack, and a **judge** (or juries), who examines if the auditor's claim is credible enough. Unlike previous defense papers in literature where the goal is to reduce MI effectiveness, either by confidence masking or more private training, we focus on a case where the auditee (defender) can discredit the auditor's (attacker) claim post-attack. This threat model is fundamentally different from the literature and makes known MI attacks ineffective even against already trained or public models.

7.2.1. Auditor (MI Attacker or Investigator). The objectives and assumptions of the auditor are as follow:

Objectives: The goal of the auditor is to use membership inference attacks on the auditee's model to find potential training samples that are private. To do this reliably, we assume that MI attacks are set to perform in the low false positive regime. The auditor then reports the potential members to the judge. We call these samples *claimed member list*. As a proof of low false positive rate, the auditor needs to privately disclose its own training/validation data to the judge such that it can be confirmed. This data is not available to the auditee or any other actor.

Assumptions: The auditor in our threat model has the highest advantage it could have. It has white-box access to the auditee's model with unlimited query. It has the capability to train multiple models if needed. It has access to a dataset coming from the same distribution as the auditee's dataset. It has access to a set of data points some of which have been potentially used as auditee's training data. To identify the member samples, auditor uses MI attacks.

7.2.2. Judge. The objectives of the judge are as follow:

Objectives: The goal of the judge is to examine if auditor's claims are reasonable, i.e. high true positive at low false positive on the auditor dataset. If so, the judge will give the auditee a chance to challenge the auditor's claim. Here, if the auditee can successfully discredit the auditor's method (i.e., the MI attack), the judge will dismiss the case.

7.2.3. Auditee (Defender). The objectives and assumptions of the auditee are as follow:

Objectives: The goal of the auditee is to discredit the MI method used by the auditor. To do so, the auditee aims to find a procedure by which it can craft/find unlimited number of non-member samples which the auditor's MI method likely mislabel as members. We call these samples *discrediting samples* and the corresponding dataset *discrediting dataset*. In other words, the auditee tries to discredit the auditor by showing that his/her low false positive claim was in fact fallacious, and, thereby, every statement using this MI method is unreliable. Note that the non-membership status of discrediting samples should be agreed by all actors beyond reasonable doubt. Otherwise it cannot be used to discredit the auditor's MI attack. To fulfil this criterion, the samples can come from the sources became available only after the model is trained, can be randomly generated on-the-fly in the court, or can be crafted by adding small perturbation to samples that have already been labeled by the auditor as non-member.

Assumptions: The auditee has no information about the MI method deployed by the auditor, the auditor's dataset, or his/her capabilities. In other words, from the perspective of the auditee, the auditor's MI model is a black-box with no online query access to. The only information the auditor has is the claimed member list that the auditor claims to be a part of auditee's training set, which is then given to the auditee by the judge. These are the samples with highest membership score according to the MI attack used by the auditor.

7.2.4. Discredibility Pipeline. Given that the auditee's model is trained and publicly available, the trial's pipeline is as follows:

(1) Using an MI attack, the auditor provides the claimed member list, a list of samples with highest membership score, to the judge stating that they are unlawfully used during training. To demonstrate the reliability of the MI attack, the auditor privately disclose the attack information and the training/validation dataset to prove the low false positive rate.

(2) The judge examines the claim. If the low false positive rate satisfies the low false positive threshold required, the judge gives the claimed member list to the auditee and asks if he/she challenges the claim.

(3) The auditee uses a procedure to find/generate a large number of nonmember samples (discrediting samples), using methods in Section 7.3, that are likely to be mislabeled by the auditor's MI model as members. The auditee, then, gives these discrediting samples to the judge and asks the judge to evaluate the performance of the MI method on.

(4) If the false positive rate of the auditor's MI attack on discrediting samples are significantly larger than what is claimed earlier by the auditor, the judge dismisses the case and consider the auditor's MI attack unreliable.

7.3. Discredibility Mechanisms

7.3.1. Problem Statement. As stated earlier, the goal of the auditee is to find a set of non-member samples which the auditor's MI attack model is likely to mislabel as members. Let $Y(\cdot)$ and $E(\cdot)$ be the auditee's model under investigation, and the encoder part of the auditee's model, respectively. Similar to [**98**], encoder here refers to the output of the last fully connected layer before the softmax, also known as the latent representation. Let's denote the last layer operation of the auditee model by $l(\cdot)$. In other words, $Y(x) = l(E(x))$. We denote the auditor's MI attack model by $M(\cdot)$. Moreover, let D_c, D_p, and D_d be the claimed member list provided by the auditor, public dataset agreed by all parties to be non-member, and the discrediting dataset, respectively.

Formally speaking, the goal is to find a non-trivial mapping from D_c to a subset in D_p to be mislabeled as member by M with high probability. The challenge to find such a mapping is that the MI attack is a complete black-box and it is not even allowed to be queried. Hence, the only information the auditee has about the MI attack is the samples identified as members (D_c) with high membership score. In this section, we will show three algorithms to generate/find nonmember samples on which the MI attack catastrophically fails.

7.3.2. Mapping and Intuition. To simply put, the mapping consists of finding/generating samples that has similar latent representation as the samples in D_c. Auditee uses the encoder, $E(\cdot)$, to find the latent representation to which he/she has white-box access. For a member sample

x marked by auditor with a high membership score, auditee's discredibility algorithm aims to find a non-member samples x', where $E(x) \approx E(x')$. The intuition as of why this causes the current MI attacks to misclassify can be analyzed by considering neural networks as deterministic functions with certain properties.

As a deterministic function, a single layer ReLU network has shown to be locally linear. In fact, the entire multi-layer ReLU network is a piece-wise linear function [**95**]. Because $l(\cdot)$ is a single-layer ReLU function, if $E(x) \approx E(x')$, then $l(E(x)) \approx l(E(x'))$ or $Y(x) \approx Y(x')$. Therefore, any MI attack that only takes the output of $Y(\cdot)$ as a feature fails to distinguish between x and x'. This is particularly an issue for older generation of MI attacks, such as Shokri [**134**] and Yeom [**134**].

The contemporary MI attacks often use the output of a set of extra models on a target sample, such as [**7**, **103**, **126**]. There are two challenges when it comes to applying the same argument here. First, the extra models the MI attackers use may be different when probing x versus x'. For example, in [**7**], half of the extra models include the target sample in the training set and the other half excludes the target sample from the training set. As a result, when probing x and x', the extra models are not necessarily the same. However, we argue that since both samples belong to the same subpopulation, including or excluding either of them results in a similar behaviour from the final model's perspective on that subpopulation.

The second challenge is that even if we assume the extra models are the same when probing two different samples, the encoder part of them are not the same as the encoder of the auditee's model, $E(\cdot)$. Let's denote the encoder part of an extra model by $E_e(\cdot)$. It has been empirically shown in [**98**] that the $dist(E(x), E(x')) \approx dist(E_e(x), E_e(x'))$ despite $E(x)$ not necessarily being close to $E_e(x)$. This suggests that membership inference score of x and x' is likely to be similar even with respect to the new generation of MI attacks. We show the correlation between the closeness of two samples and their membership scores in Section 7.6 to provide an empirical evidence.

7.3.3. Discredibility Methods. As discussed in Section 7.3.2, discredibility is performed by sampling from D_p provided that the samples belong to the same subpopulation as samples in D_c. Here, we use the latent representation of the auditee's model, $E(\cdot)$, to find subpopulations. Formally, we consider two samples, x and x', from the same subpopulation if $dist(E(x), E(x')) \leq \epsilon$. For the distance measure, the two prominent choices are Cosine distance (Cosine loss) and $L2$ norm

Algorithm 2 Finding Discrediting Samples by Search

Require: Encoder $E(\cdot)$; Claimed member list D_c; Non-member public dataset D_p; the number of neighbors to pull from D_p per sample in D_c, denoted by n_n; a sample and the associated label (x, y); the number of samples from D_c with highest membership score to find neighbors for, denoted by n_c

Initialize discrediting dataset $D_d = \{\}$
$D_c \leftarrow Sort\_by\_membership\_score(D_c)$
$D_c \leftarrow D_c[-n_c :]$
for each $(x, y) \in \mathcal{D}_c$ **do**
    for each $(x', y') \in \mathcal{D}_p$ **do**
        if $y = y'$ **then**
            $d[x, x'] \leftarrow dist(E(x), E(x'))$
        end if
    end for
    $d_s \leftarrow argsort(d[x, :])$
    $D_d \leftarrow D_d \cup d_s[: n_n]$
end for
return D_d

(MSE loss). As shown in [**98**], there is not much difference between these to metrics when it comes to measuring sample similarities. Hence, we mainly use Cosine loss in this chapter.

Algorithm 3 Finding Discrediting Samples by Sample Generation

Require: Auditee's model $Y(\cdot)$; Encoder $E(\cdot)$; Claimed member list D_c; Generator of the Rezaei's BiGAN architecture $G(\cdot)$; the number of samples to pull from D_p per sample in D_c, denoted by n_n; a Gaussian random noise generator $\mathcal{N}(\mu, \sigma^2)$; the number of samples from D_c with highest membership score to craft samples from, denoted by n_c

Initialize discrediting dataset $D_d = \{\}$
$D_c \leftarrow Sort\_by\_membership\_score(D_c)$
$D_c \leftarrow D_c[-n_c :]$
for each $(x, y) \in \mathcal{D}_c$ **do**
    for i = 0 to n_n **do**
        $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$
        $x' \leftarrow G(E(x) + \epsilon)$
        if $y = Y(x')$ **then**
            $D_d \leftarrow D_d \cup \{x'\}$
        end if
    end for
end for
return D_d

In this chapter, we propose three methods to find/generate samples from the same subpopulation:

1. Using a Large Public Dataset: If a large dataset, disjoint from the train set, is available to sample from, auditee can use it to create discrediting dataset, D_d. The procedure is straightforward, as shown in Algorithm 2. Note that there is no guarantee that a sample with subpopulation constrain, i.e. $dist(E(x), E(x')) \leq \epsilon$, exists in D_p. For simplicity, we discard this condition and we add the closest n_n samples to discrediting dataset although they may not necessarily be from the same subpopulation. The only criterion is that their class labels should match (line 6). Otherwise, they obviously do not belong to the same subpopulation. The empirical results in Section 7.5.1 shows that the discrediting samples are good enough for the purpose of discrediting the auditor. Hence, the challenge of defining ϵ is not crucial for the discrediting purpose and, hence, it is ignored in this chapter.

2. Using Generative Model: We can use generative models to craft new samples. However, unconditional sample generation is an extremely inefficient exercise as it may take millions of queries for the model to generate a sample from the same subpopulation. In this chapter, we use the BiGAN architecture proposed in [**98**] to craft new samples. The generator in their architecture take the latent representation as an input and generate a sample accordingly. As shown in Algorithm 3, we add a small random noise to the latent representation of a target sample and use it to generate a new sample from the BiGAN.

3. Using Adversarial Perturbation: In this method, we take a non-member sample that belongs to the same class as the target sample does, and we add a small adversarial perturbation to such that the latent representation of the two samples approaches the same value. The algorithm is shown in Algorithm 4. Here, x and x' should belong to the same class, otherwise the auditor can easily tell the adversarial nature of it because it will be misclassified by the model. Although we can start the adversarial perturbation on any non-member sample (x'), we use a function ($S(\cdot)$) to find the closest neighbor with the same class label to increase the chance of reaching the same latent representation.

7.4. Experimental Setup

7.4.1. Evaluation Metrics. As suggested in [**7**], it is more practical to use membership inference attack at a low false positive rate. Hence, in this chapter, we mainly focus on true

Algorithm 4 Finding Discrediting Samples by Adversarial Perturbation

Require: Encoder $E(\cdot)$; Claimed member list D_c; the number of adversarial samples per sample in D_c, denoted by n_n; a targeted adversarial attack $adv\_attack(x, x', F(\cdot), dist(\cdot))$ that perturbs x' such that $dist(F(x), F(x')) \approx 0$; the number of steps to run the adversarial attack, denoted by n_{adv}; the number of samples from D_c with highest membership score to find adversarially perturbed neighbors for, denoted by n_c; a function returning a non-member neighbor sample with the same class label as the input $S(\cdot)$

Initialize discrediting dataset $D_d = \{\}$
$D_c \leftarrow Sort\_by\_membership\_score(D_c)$
$D_c \leftarrow D_c[-n_c :]$
for each $(x, y) \in \mathcal{D}_c$ **do**
    for i = 0 to n_n **do**
        $(x', y) \leftarrow S(x, y)$
        for j = 0 to n_{adv} **do**
            $x' \leftarrow adv\_attack(x, x', E(\cdot), MSE(\cdot))$
        end for
        $D_d \leftarrow D_d \cup \{x'\}$
    end for
end for
return D_d

positive at a low false positive rate. For the sake of completeness, we also report the AUC of all MI attacks.

The second evaluation metric that we use in this chapter is *false positive to false positive* plot or ratio. This measures the false positive of an MI attack on an auditor's dataset in comparison with the discrediting dataset, proposed by the auditee. Here, we disregard the true positive rate because positive samples include all training member samples on both cases. Thus, this set is assumed to be fixed in both auditor dataset and the discrediting dataset. The goal of the auditee is to come up with a samples generation/look-up scheme that outputs negative (non-member) samples that are labeled as positive with a much larger false positive rate than acceptable. As a result, we only measure the false positive difference between these two datasets. In other words, the true positive is the same regardless of the evaluating dataset.

7.4.2. Experimental Setup. We conduct experiments on a number of image classification benchmarks traditionally used for membership inference attack evaluation, including MNIST [**64**], FMNIST [**130**], SVHN [**90**], and CIFAR-10/CIFAR-100 [**61**]. For Algorithm 2 to work, we need a large public dataset to search. For CIFAR-10, we use the CINIC dataset [**18**] as a public dataset,

TABLE 7.1. Accuracy of the auditee's models

Dataset	Model	Train accuracy	Test accuracy
MNIST	MLP	100%	97.71%
FMNIST	MLP	100%	88.62%
SVHN	LeNet	99.99%	87.72%
Cifar10	LeNet	97.13%	58.22%
Cifar10	ResNet20	98.48%	74.58%
Cifar100	LeNet	98.27%	22.61%
Cifar100	ResNet20	100.00%	33.30%

TABLE 7.2. Comparison of AUC of prior membership inference attacks. S, Y, W, C, and R stands for Shokri, Yeom, Watson, Carlini, and Rezaei attacks, respectively.

Dataset/Model	S [**110**]	Y [**134**]	W [**126**]	C [**7**]	R [**98**]
MNIST/MLP	52.43%	50.95%	53.53%	56.17%	51.11%
FMNIST/MLP	59.62%	56.38%	57.54%	58.55%	54.87%
SVHN/LeNet	57.60%	57.88%	60.24%	69.94%	58.64%
C-10/LeNet	72.62%	78.15%	73.77%	79.55%	76.12%
C-10/ResNet20	74.52%	70.75%	65.06%	72.19%	68.74%
C-100/LeNet	82.03%	91.96%	90.19%	94.30%	93.83%
C-100/ResNet20	91.17%	92.81%	81.27%	93.39%	91.98%

TABLE 7.3. Comparison of prior membership inference attacks at low false positive rate. S, Y, W, C, and R stands for Shokri, Yeom, Watson, Carlini, and Rezaei attacks, respectively. Since the exact false positive is not always possible to achieve, we choose the lowest false positive between the stated ranges of $(0.01\%, 0.03\%)$ and $(1.0\%, 3.0\%)$.

Dataset	Model	TPR @ (0.01%, 0.03%) FPR					TPR @ (1.0%, 3.0%) FPR				
-	-	S [**110**]	Y [**134**]	W [**126**]	C [**7**]	R [**98**]	S [**110**]	Y [**134**]	W [**126**]	C [**7**]	R [**98**]
MNIST	MLP	0.00%	0.00%	0.10%	0.00%	0.01%	0.00%	0.00%	2.40%	2.47%	0.47%
FMNIST	MLP	0.00%	0.00%	0.39%	3.26%	0.08%	2.67%	0.00%	3.25%	5.39%	1.06%
SVHN	LeNet	0.00%	0.00%	0.63%	0.00%	0.11%	0.00%	0.00%	5.20%	6.52%	2.61%
Cifar10	LeNet	0.00%	0.00%	0.52%	0.00%	0.28%	2.57%	0.00%	7.71%	10.37%	4.76%
Cifar10	ResNet20	0.00%	0.00%	0.54%	0.57%	0.06%	3.55%	0.00%	5.80%	10.97%	5.10%
Cifar100	LeNet	0.06%	0.00%	1.68%	0.01%	0.17%	3.44%	0.00%	18.66%	18.71%	19.73%
Cifar100	ResNet20	0.00%	0.00%	1.76%	16.26%	1.46%	4.64%	4.56%	14.20%	37.12%	26.71%

and for SVHN, we use the *extra* portion of the dataset as a public dataset. For other datasets, we could not find a large public dataset to search through, and, hence, we only perform the second and third algorithms on them.

We divide the train set of these datasets into two parts: auditor training dataset and auditee training dataset. Similar to [**98**], we choose multi-layer perceptron (MLP) with 4 hidden layers for

TABLE 7.4. Lowest false positive value on the auditor dataset. The numbers in parenthesis show the ratio of the false positive on discrediting dataset over the false positive on auditor dataset when Algorithm 2 is used for discrediting.

Dataset	Model	Shokri [**110**]	Yeom [**134**]	Watson [**126**]	Carlini [**7**]	Rezeai [**98**]
SVHN	LeNet	3.449% (×25.6 ↑)	67.730% (×1.3 ↑)	0.142% (×88.0 ↑)	1.283% (×3.1 ↑)	0.013% (×326.4 ↑)
CIFAR-10	LeNet	0.791% (×56.5 ↑)	28.631% (×2.1 ↑)	0.003% (×1842.1 ↑)	0.034% (×32.4 ↑)	0.020% (×384.6 ↑)
CIFAR-10	ResNet20	0.049% (×363.3 ↑)	17.469% (×4.3 ↑)	0.029% (×116.7 ↑)	0.011% (×102.9 ↑)	0.009% (×364.6 ↑)

MNIST and FMNIST classification. For SHVN, we choose LeNet. For CIFAR-10 and CIFAR-100 we use both LeNet and ResNet20. We use SGD with a learning rate of 0.1 to train all models. We decrease the learning rate by a factor of 10 at epoch 50 and 75. The performance of the auditee models is shown in Table 7.1.

We evaluate our discredibility methods on five state-of-the-art membership inference attacks, namely Shokri [**110**], Yeom [**134**], Watson [**126**], Carlini [**7**], and Rezaei [**98**]. Unless specified, we follow the same hyper-parameters to train MI attack models as suggested in their original paper. For Shokri attack [**110**], we train 50 shadow models for all datasets. For Watson [**126**] and Rezaei [**98**] attacks, we use the loss function as the base membership score before calibration. We use the same BiGAN architecture proposed in [**98**] for both Rezaei's attack and Algorithm 3.

Table 7.2 shows the AUC of membership inference attacks. In Table 7.3, the true positive rates of the SOTA MI attacks at 0.01% and 1.0% false positive is presented. As also previously shown in [**7**], Shokri [**110**] and Yeom [**134**] attacks does not work well in the low false positive regime. We omit other membership inference attacks in this study, such as Jayaraman [**51**] and Song [**113**], because it has been shown to perform poorly at low false positive rate [**7**].

7.5. Experimental Results

7.5.1. Natural Subpopulation. Our first method to produce discrediting samples rely on searching samples in a large public dataset. The details of the algorithm is shown in Algorithm 2. The only datasets for which we can find a large public dataset with similar classes are CIFAR-10 and SVHN. Figure 7.1, 7.3 and 7.4 show a few examples of member samples and their closest neighbors using Algorithm 2. It is worth mentioning that not all neighboring samples belong to the same class and, interestingly, the membership score of the neighboring samples with different class label are often significantly lower and should be discarded. It is clear from this figure that

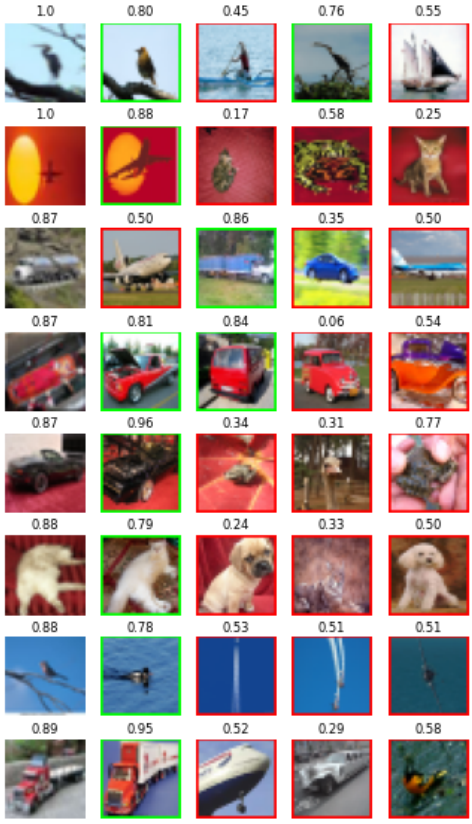
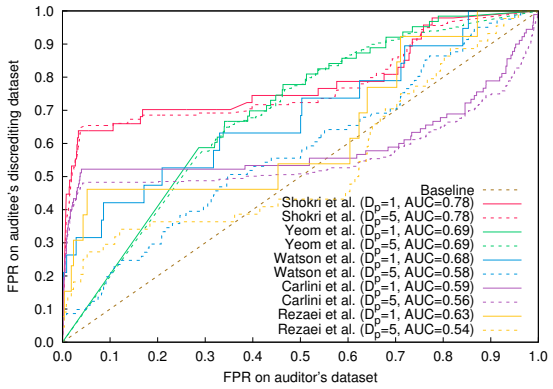


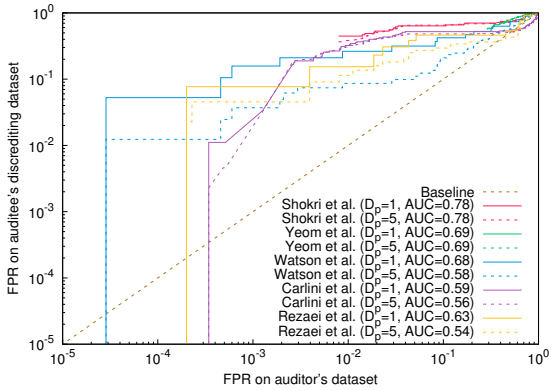
FIGURE 7.1. The first columns shows member samples from CIFAR-10 dataset. The next four columns show the closest samples from the CINIC dataset to the sample in the first column. The value on top of each image shows the normalized Watson attack membership score. The neighboring samples that have the same label as the original sample is indicated by the green boarder. The boarder is red otherwise. Membership score of non-member neighboring samples of the member samples that belong to the same class often have high membership score.

the neighboring samples are not close pixel-wise to their corresponding member sample. Hence, a credible membership inference attack should differentiate the membership status of them.

The false positive to false positive plot of MI attacks for a LeNet model trained on CIFAR-10 is shown in Figure 7.2. Here the x-axis shows the false positive on auditor's dataset for a given threshold and the y-axis shows the false positive on the discrediting dataset. Any region over the baseline indicates that the auditee successfully presents a dataset with larger false positive. For a practical membership inference attack, the false positive should be small. Hence, we mostly focus on the log plot (Figure 7.2 (b)) where we can better study the behavior on low false positive rates.

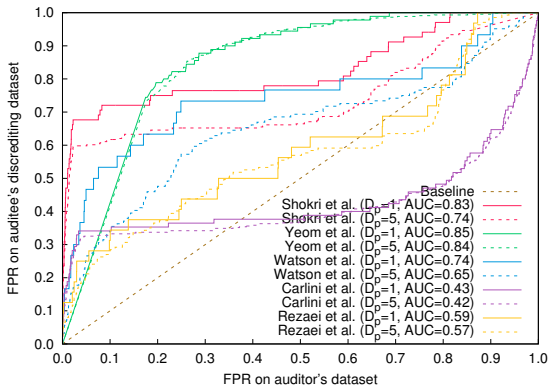


(a) FPR/FPR plot

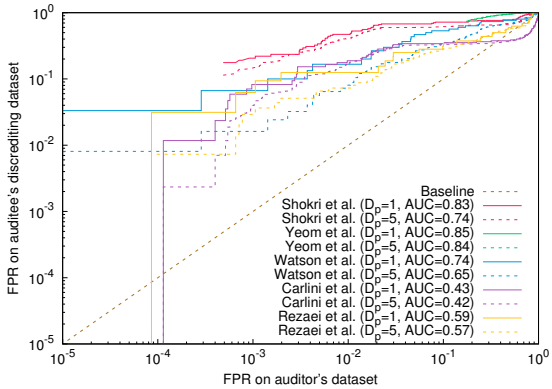


(b) FPR/FPR logscale plot

FIGURE 7.2. CIFAR-10/LeNet model. Discrediting algorithm 2 using CINIC dataset.



(a) FPR/FPR plot

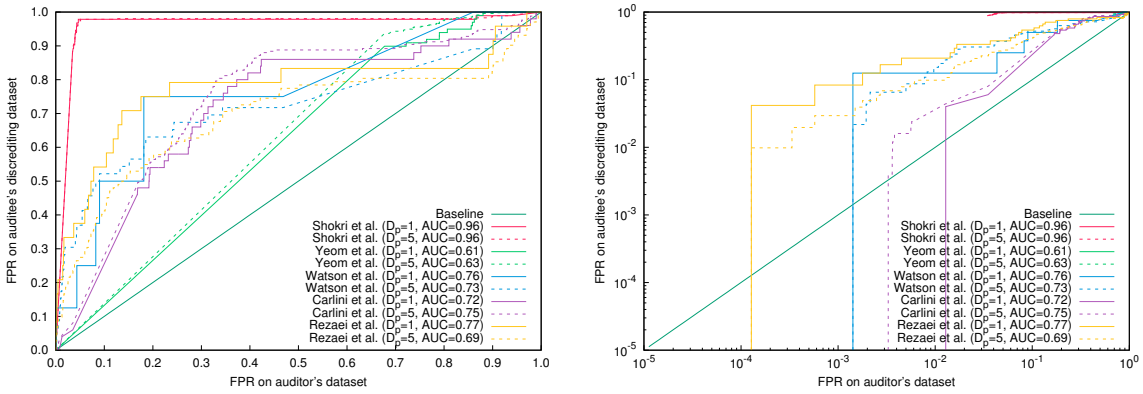


(b) FPR/FPR logscale plot

FIGURE 7.3. Cifar10/ResNet20 model. Discrediting algorithm 2 using CINIC dataset.

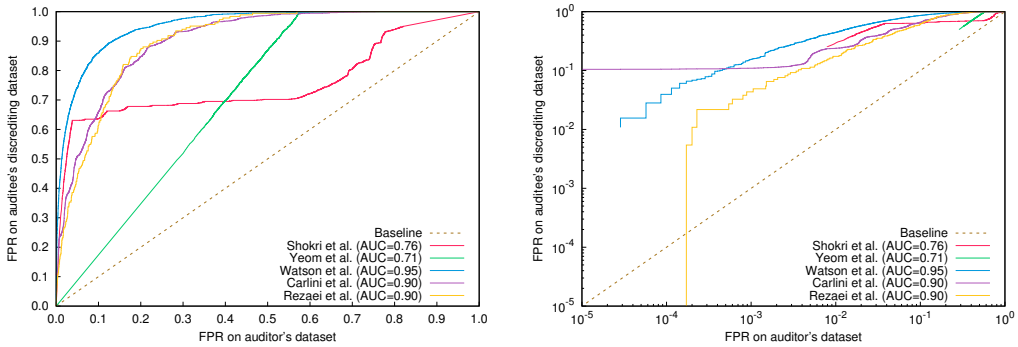
It is clear that the false positive is over hundreds to thousands times larger on discrediting dataset for Watson, Carlini, and Rezaei attacks. In addition, the lowest false positive for Shokri and Yoem attacks are too large for any practical usage. Nevertheless, our discredibility method still increases the false positive even further.

In Figure 7.2, D_p represents the number of neighboring samples from the same class we used to construct the discrediting dataset. As expected, $D_p = 1$ slightly outperforms $D_p = 5$ case potentially because the further away the samples is from the target sample, the less likely it is



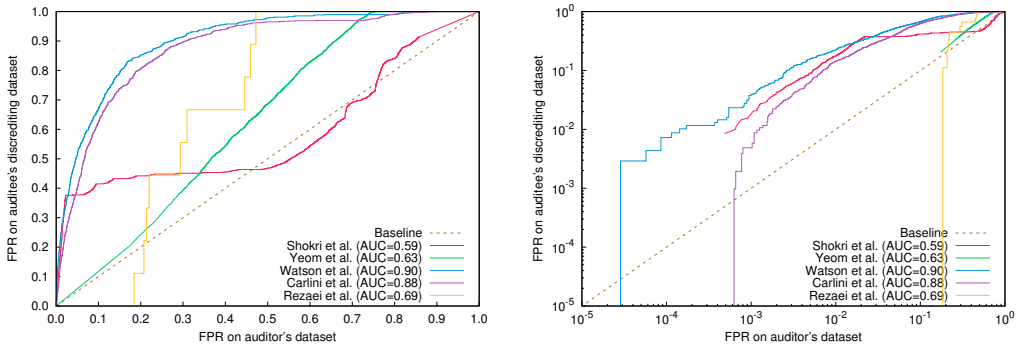
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.4. SVHN/LeNet model. Discrediting algorithm 2 using SVHN (extra).



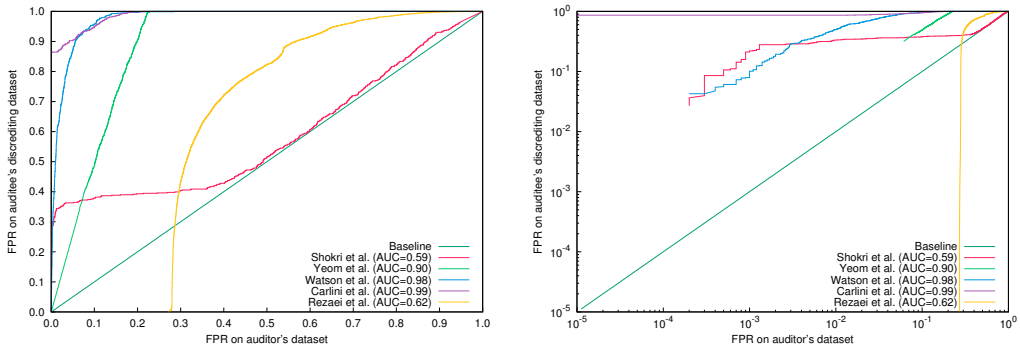
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.5. Cifar10/LeNet model. Discrediting algorithm 3.



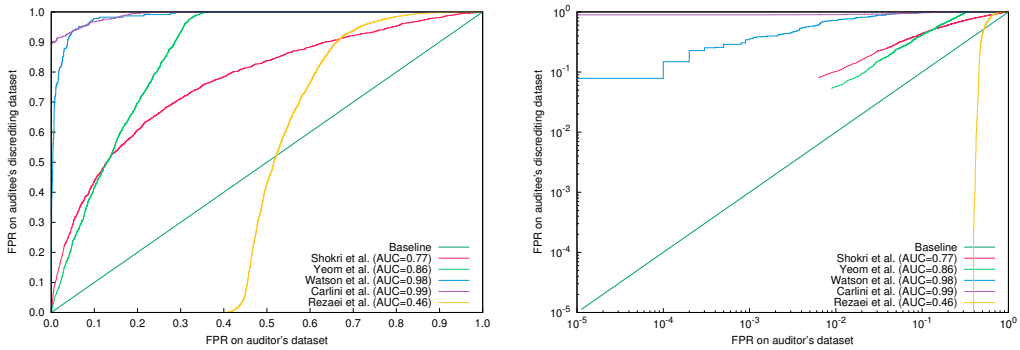
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.6. Cifar10/ResNet20 model. On crafted samples using BiGAN



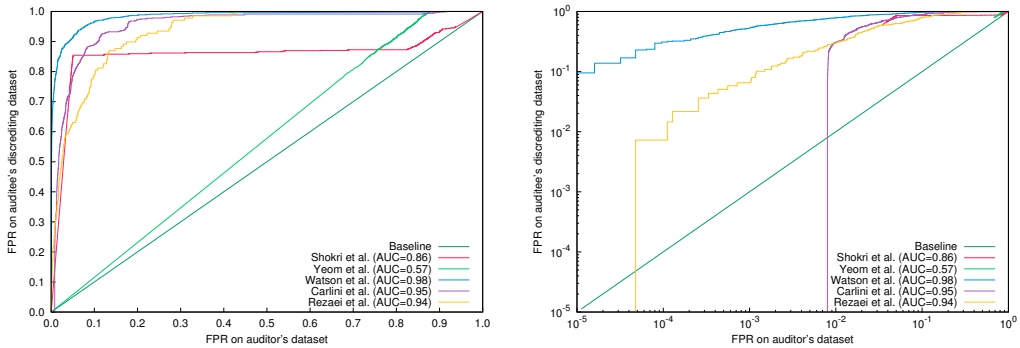
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.7. Cifar100/LeNet model. On crafted samples using BiGAN



(a) FPR/FPR plot (b) FPR/FPR logscale plot

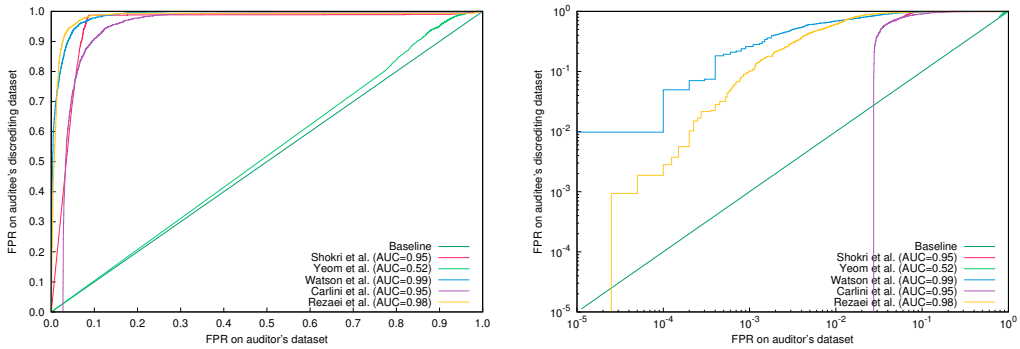
FIGURE 7.8. Cifar100/ResNet20 model. On crafted samples using BiGAN



(a) FPR/FPR plot (b) FPR/FPR logscale plot

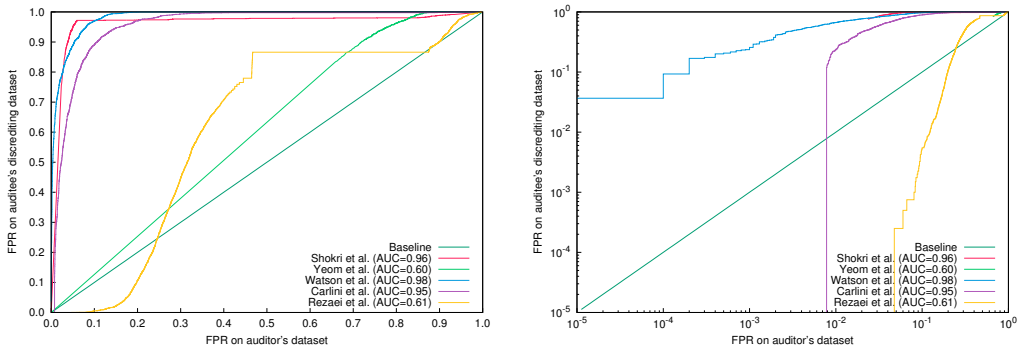
FIGURE 7.9. SVHN/LeNet model. On crafted samples using BiGAN

labeled the same way as the target sample, with respect to membership inference. In Section 7.6, we analyze the correlation between distance and membership score in more depth.



(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.10. MNIST/MLP model. On crafted samples using BiGAN



(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.11. FMNIST/MLP model. On crafted samples using BiGAN

TABLE 7.5. Lowest false positive value on the auditor dataset. The numbers in parenthesis show the ratio of the false positive on discrediting dataset over the false positive on auditor dataset when Algorithm 3 is used for discrediting.

Dataset	Model	Shokri [**110**]	Yeom [**134**]	Watson [**126**]	Carlini [**7**]	Rezeai [**98**]
MNIST	MLP	4.750% (×14.3 ↑)	77.400% (×1.0 ↑)	0.010% (×97.9 ↑)	2.740% (×2.9 ↑)	0.003% (×37.5 ↑)
FMNIST	MLP	2.350% (×32.2 ↑)	65.910% (×1.3 ↑)	0.010% (×366.4 ↑)	0.780% (×15.3 ↑)	4.755% (×0.0 ↑)
SVHN	LeNet	3.449% (×17.5 ↑)	67.730% (×1.2 ↑)	0.002% (×5952.8 ↑)	0.798% (×1.0 ↑)	0.005% (×151.4 ↑)
Cifar10	LeNet	0.791% (×31.9 ↑)	28.631% (×1.7 ↑)	0.003% (×379.1 ↑)	0.003% (×3668.3 ↑)	0.017% (×31.7 ↑)
Cifar10	ResNet20	0.049% (×17.5 ↑)	17.469% (×1.2 ↑)	0.003% (×101.7 ↑)	0.063% (×1.5 ↑)	0.002% (-)
Cifar100	LeNet	0.020% (×8.5 ↑)	6.110% (×1.1 ↑)	0.020% (×17.5 ↑)	0.010% (×1.0 ↑)	0.002% (-)
Cifar100	ResNet20	0.630% (×6.7 ↑)	0.890% (×2.9 ↑)	0.010% (×45.0 ↑)	0.020% (×2.5 ↑)	0.002% (-)

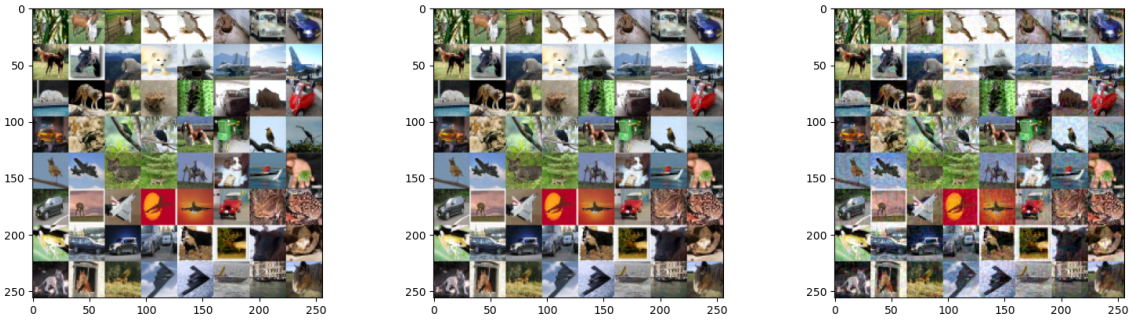
Table 7.4 shows the lowest possible false positive a membership inference can achieve on the auditor's dataset and the ratio of the false positive on discrediting dataset over the false positive on auditor dataset. This significant increase clearly shows that the discrediting Algorithm 2 can be used when a large non-member dataset is available to search samples from.

7.5.2. Crafted Subpopulation. In this section, we evaluate the effectiveness of Algorithm 3 to craft discrediting samples. This method assumes that a generator model is available that takes a latent representation and crafts samples corresponding to the representation. We use the same BiGAN architecture proposed in [**98**] to train such a generator. Since the generator is trained by the auditee, we directly use the auditee's model under investigation in the BiGAN architecture. All training hyper-parameters and details are adopted from the [**98**].

The false positive to false positive plot for a LeNet model trained on the CIFAR-10 is shown in Figure 7.5. In comparison with using real samples by Algorithm 2, the effectiveness of this method varies accross different attacks/models/datasets. Nevertheless, it still increases the false positive rate more than 10 times for most MI attacks. Interestingly, Rezaei attack [**98**] seems to be more immune to discrediting based on the BiGAN approach. The reason lays on how this attack works. Rezaei attack uses the same BiGAN architecture to craft similar samples to the target sample. Then, it uses the difference between the target sample's loss and the loss of average samples from the same subpopulation as the membership score. We find that the average loss difference between two crafted samples are often smaller than a natural sample and a crafted sample. Consequently, the membership scores of auditor samples (which are natural) are on average larger than the discrediting samples (which are crafted). In other words, the Rezaei attack [**98**] is immune to this discrediting method because it can distinguish between crafted and natural samples, and not because it can identify member samples versus non-member samples. This occurs mainly because the BiGAN architecture is not good enough to generate indistinguishable natural samples. In Section 7.6, we analyze this method in more depth. Nevertheless, the auditee can still use the other two methods to safely discredit the auditor if he/she uses Rezaei's MI attack.

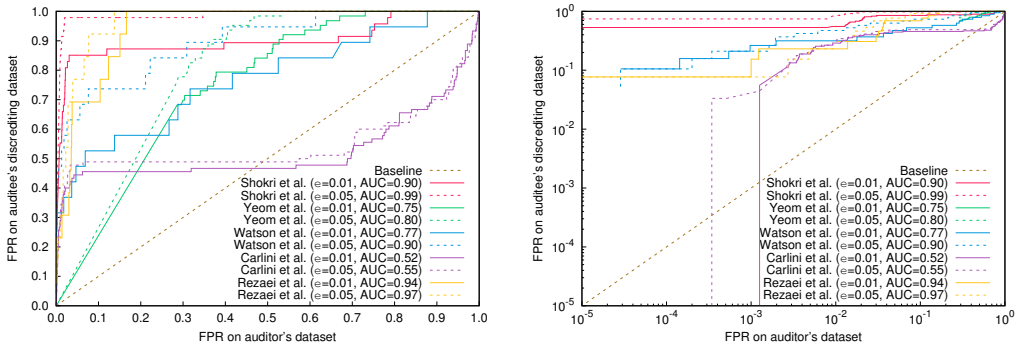
Table 7.5 shows the full results of all membership inference attacks for the lowest false positive. Clearly, the BiGAN approach of crafting discrediting samples does not work as effective on harder classification tasks, such as CIFAR-10 and CIFAR-100. This probably stems from the difficulty in training a high quality BiGAN to craft natural samples for these datasets. Further research is needed to see if this problem can be solved by using stronger generator trained on larger datasets.

7.5.3. Adversarially Tuned Subpopulation. In this section, we evaluate Algorithm 4 effectiveness in producing discrediting samples. This method requires an adversarial attack algorithm



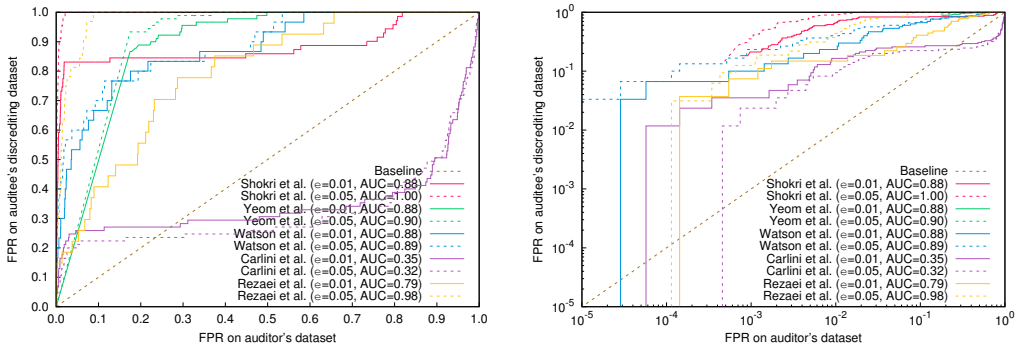
(a) Original images (b) Adversarially tuned images ($\epsilon = 0.01$) (c) Adversarially tuned images ($\epsilon = 0.05$)

FIGURE 7.12. Natural samples versus the corresponding adversarially perturbed versions.



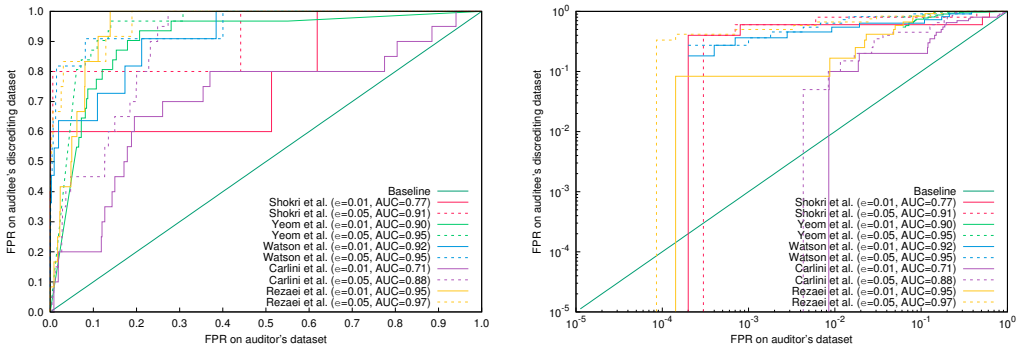
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.13. Cifar10/LeNet model. Discrediting algorithm 4.



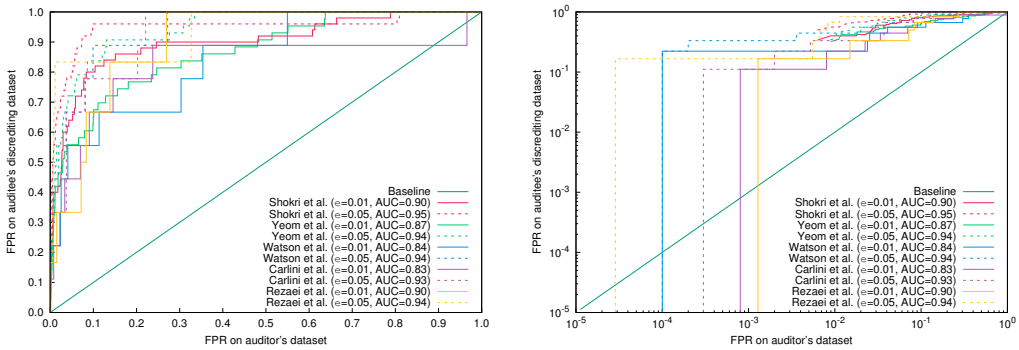
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.14. Cifar10/ResNet20 model. On adversarially tuned samples.



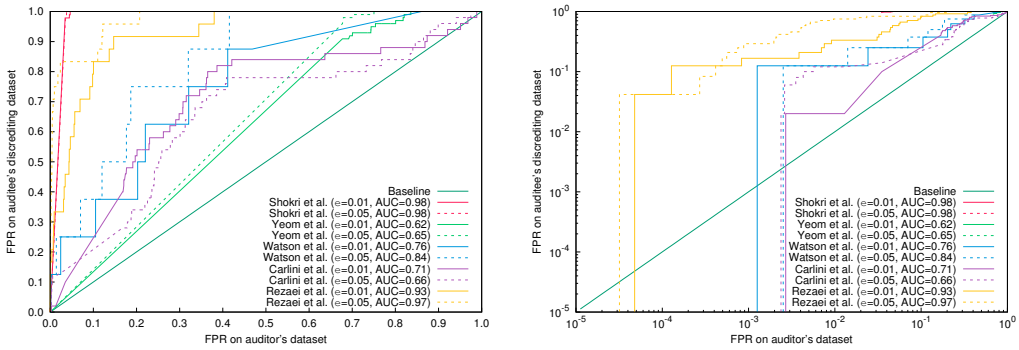
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.15. Cifar100/LeNet model. On adversarially tuned samples.



(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.16. Cifar100/ResNet20 model. On adversarially tuned samples.

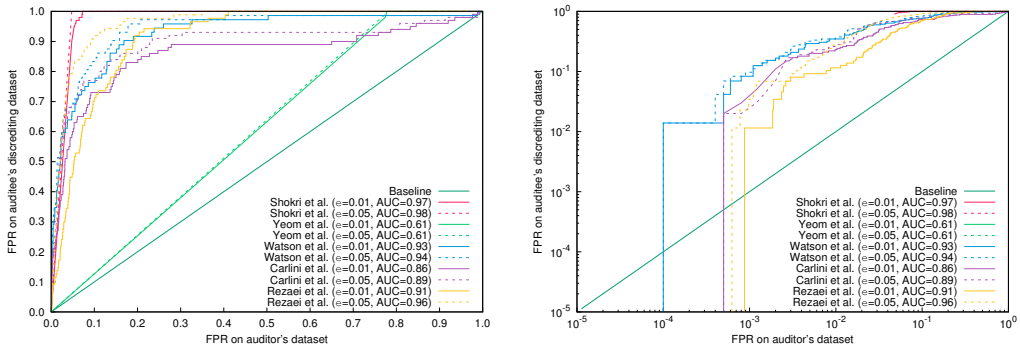


(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.17. SVHN/LeNet model. On adversarially tuned samples.

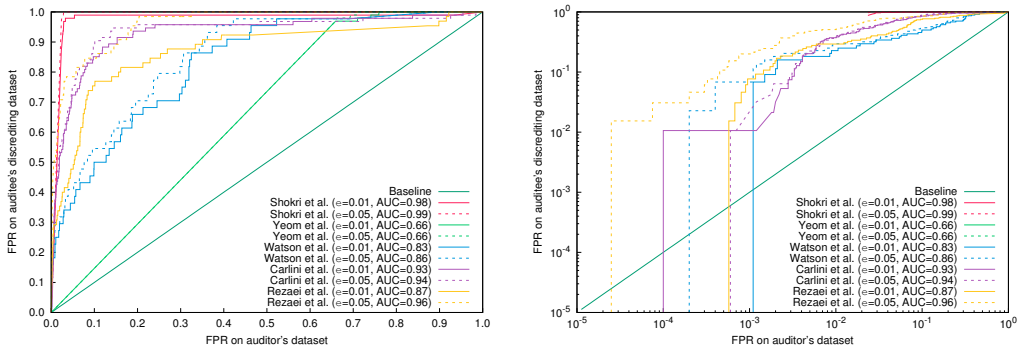
to perturb the input such that its latent representation of the sample converges to the latent representation of the target sample. We use projected gradient descent[1] (PGD) algorithm with step

[1]We use the public implementation by Cleverhans lab at https://github.com/cleverhans-lab/cleverhans



(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.18. MNIST/MLP model. On adversarially tuned samples.



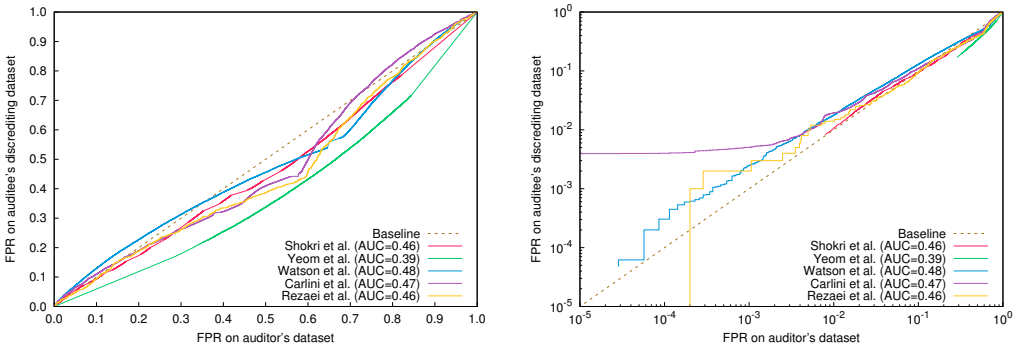
(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.19. fMNIST/MLP model. On adversarially tuned samples.

TABLE 7.6. Lowest false positive value on the auditor dataset. The numbers in parenthesis show the ratio of the false positive on discrediting dataset over the false positive on auditor dataset when Algorithm 4 ($\epsilon = 0.05$) is used for discrediting.

Dataset	Model	Shokri [**110**]	Yeom [**134**]	Watson [**126**]	Carlini [**7**]	Rezeai [**98**]
MNIST	MLP	4.750% (×21.1 ↑)	77.400% (×1.3 ↑)	0.010% (×138.9 ↑)	0.050% (×20.0 ↑)	0.062% (×18.4 ↑)
FMNIST	MLP	2.350% (×41.2 ↑)	65.910% (×1.5 ↑)	0.020% (×113.6 ↑)	0.060% (×17.7 ↑)	0.003% (×615.4 ↑)
SVHN	LeNet	3.449% (×27.7 ↑)	67.730% (×1.4 ↑)	0.252% (×49.6 ↑)	0.238% (×8.4 ↑)	0.003% (×1305.4 ↑)
Cifar10	LeNet	0.791% (×94.1 ↑)	28.631% (×2.7 ↑)	0.003% (×1842.1 ↑)	0.034% (×97.2 ↑)	0.271% (×28.3 ↑)
Cifar10	ResNet20	0.049% (×302.8 ↑)	17.469% (×5.3 ↑)	0.003% (×1166.7 ↑)	0.046% (×25.7 ↑)	0.011% (×273.4 ↑)
Cifar100	LeNet	0.030% (×1333.3 ↑)	6.110% (×13.2 ↑)	0.020% (×1363.6 ↑)	0.430% (×11.6 ↑)	0.009% (×972.2 ↑)
Cifar100	ResNet20	0.630% (×79.4 ↑)	0.890% (×39.2 ↑)	0.010% (×2222.2 ↑)	0.030% (×370.4 ↑)	0.003% (×5833.3 ↑)

size 0.001 for 100 iterations. We try $\epsilon = 0.01$ and $\epsilon = 0.05$ to assess different perturbation budget. Figure 7.12 shows several natural samples from CIFAR-10 and the corresponding adversarially perturbed versions. Samples with perturbation of $\epsilon = 0.01$ are imperceptible to human eyes from the



(a) FPR/FPR plot

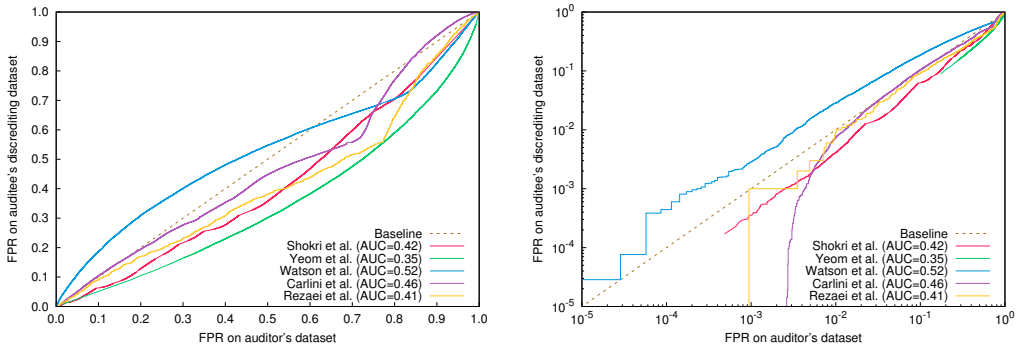
(b) FPR/FPR logscale plot

FIGURE 7.20. CIFAR-10/LeNet model. The effect of distribution shift. Here, instead of using discrediting algorithms, we use entire CINIC dataset as the discrediting dataset without filtering out any sample.

natural samples. Perturbation of $\epsilon = 0.05$, however, leaves visible footprint on otherwise natural samples.

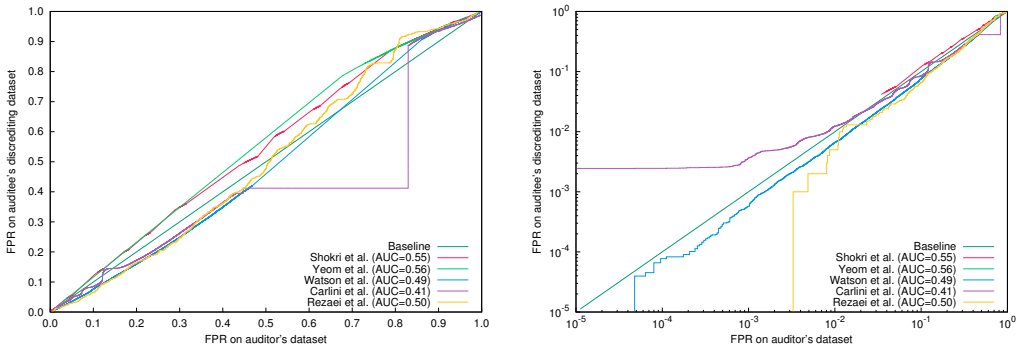
Figure 7.13 demonstrates the false positive to false positive plot for a LeNet model trained CIFAR-10 dataset. In comparison with both Algorithm 2 (Figure 7.2) and Algorithm 3 (Figure 7.5), using adversarial perturbation is a more effective on average. Even the perturbation of $\epsilon = 0.01$ which does not produce perceptible artifacts is highly effective. It is worth emphasizing that the way the adversarial perturbation is used in this context is different from adversarial attack literature. In adversarial attack literature, the attacker has either white-box or black-box access to the model it tries to mislead, which would have been the MI attack in this case. However, in our scenario, the auditee who uses the adversarial attack does not even know the type of membership inference attack, let alone a query access or white-box access to it. The auditee, in this case, tries to perturb a sample so that it mimics the latent representation of another sample to which the MI attack has already assigned a high membership score.

Table 7.6 represents the results of the method on all datasets/models. Interestingly, in a few cases, the false positive is more than thousand times larger on discrediting samples. Given the simplicity of this approach in comparison with Algorithm 3 and the lack of the need for a large public dataset in comparison with 2, the effectiveness of this approach as a discrediting tool is significant.



(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.21. CIFAR-10/ResNet model. Here, the discrediting dataset is the entire CINIC dataset.

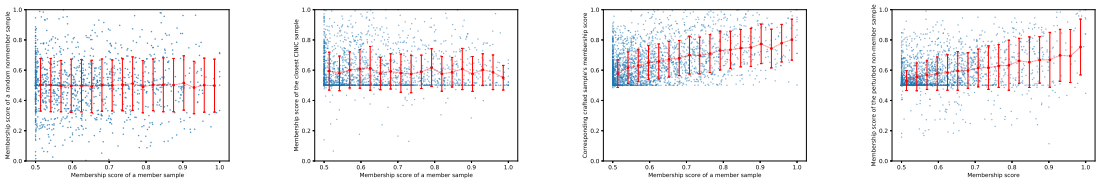


(a) FPR/FPR plot (b) FPR/FPR logscale plot

FIGURE 7.22. SVHN/LeNet model. Here, the discrediting dataset is the entire extra portion of the SVHN dataset.

7.5.4. Could it be a Repercussion of Domain Shift? A natural question upon the success of the three algorithms to significantly increase the false positive rate is if a domain shift across datasets are the real culprit. In other words, one may suspect that using the entire CINIC dataset as a discrediting dataset may achieve the same goal as the proposed algorithms because the MI attacks are vulnerable to domain shift.

To refute the hypothesis, we illustrate the false positive to false positive plot in Figure 7.20 for a LeNet model trained on CIFAR-10. The results for other datasets/models are presented in Figure 7.21 and 7.22. Here, the auditor dataset is the test portion of the CIFAR-10 dataset. The MI attack models that require dataset for training use the unused portion of the training set of the CIFAR-10 dataset. The auditee's discrediting dataset is the entire CINIC dataset. Due to the huge



(a) Random samples from CINIC

(b) Closest sample from CINIC (Algorithm 2)

(c) Crafted samples using Algorithm 3

(d) Adversarial samples using Algorithm 4

FIGURE 7.23. Correlation of (Watson attack) membership score between CIFAR-10 member samples (x-axis) and nonmember samples (y-axis). The criterion to select nonmember samples are specified by the title of each sub-figure.

computational complexity of training individual models containing each sample in CINIC dataset separately, here, we use the offline version of the Carlini attack [**7**].

Interestingly, it is clear that the domain shift works in favor of the auditor by slightly decreasing the false positive. The reason is that the auditee's model trained on CIFAR-10 is naturally less confident on samples from another distribution. Unless carefully picked by an algorithm, such as Algorithm 2, the confidence output of the model is lower on average and, hence, less likely to be incorrectly labeled as member (positive). Therefore, discrediting process cannot be simply reduced to finding a dataset with different distributions.

7.6. Key Hypotheses and Validation

In this section, we investigate two hypotheses implicitly used as a cornerstone of the three discrediting algorithms. Here, the notion of closeness and neighborhood are all in the latent representation space, not the pixel space, unless specified otherwise. For more efficient visualization, we only show a small random set of samples in scatter plots. The average and standard error, however, is computed over all samples.

Hypothesis 1. *There is a correlation between the membership score of a member sample and its neighboring nonmember samples.*

This is the key assumptions used in all three discrediting algorithms. By sorting the D_c dataset with respect to the membership score and finding/crafting samples based on them, we implicitly incorporating this assumption in all algorithms. To investigate this assumption, for each member sample in CIFAR-10 dataset, we use algorithm 2 (using CINIC dataset) and 3 to find/craft

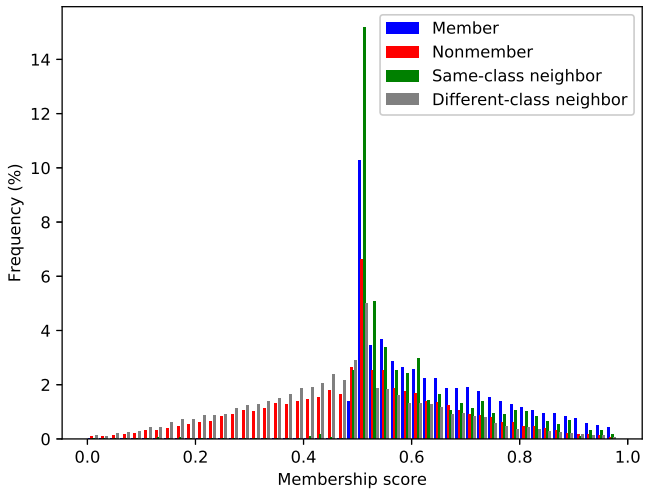
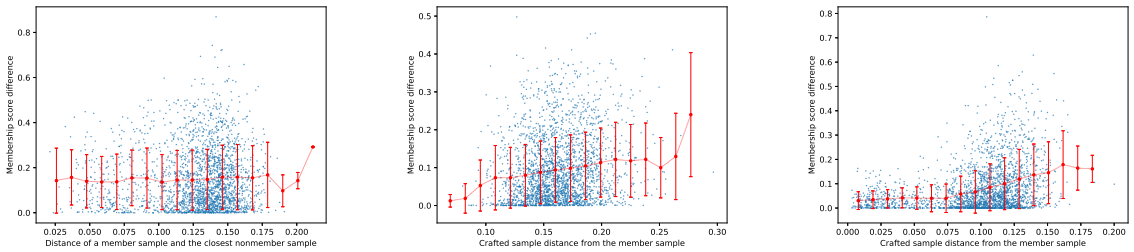


FIGURE 7.24. Distribution of Watson attack's membership score for member samples, non-member samples, and same-class neighbors, and different-class neighbors from CINIC dataset (Algorithm 2)

neighboring samples. Here, we use Watson attack [**126**] to compute the normalized membership score. Additionally, for each member sample, we randomly select a nonmember sample without any particular constraint to illustrate a case where no discrediting algorithm is used.

Figure 7.23 (a) presents a case where no discrediting algorithm is used. X-axis shows the membership score of member samples, and the y-axis shows the score of a random sample from the nonmember set. As shown, the membership score of member samples are between 0.5 and 1. The membership score of nonmember samples, however, can be any value between 0 and 1. Figure 7.23 (b-d) demonstrates the case where our discrediting Algorithms are used. It is clear that what discrediting algorithms do is eliminating majority of samples with low membership score. The output of discrediting algorithms are a set nonmember samples whose membership score is between 0.5 and 1, similar to member samples.

Figure 7.23 (b-d) also illustrates the potential correlation between membership score of a member sample and its neighboring nonmember sample. It seems that there is no correlation when searching neighboring samples in CINIC dataset. The correlation analysis for this case is inconclusive and we speculate that if a much larger public dataset covering the entire portion of input space was available the results would have been different. The correlation can be better investigated



(a) Closest sample from CINIC (Algorithm 2)

(b) Crafted samples using Algorithm 3

(c) Adversarial samples using Algorithm 4

FIGURE 7.25. Correlation between the distance and the membership score difference of a member sample and its neighboring nonmember sample.

with the generator model that allows us to generate arbitrary nonmember samples with different distance to the member samples. In this case, as shows in Figure 7.23 (c), there is a clear positive correlation between membership score of a member sample and its neighboring sample. The positive correlation is also clearly depicted in Figure 7.23 (d) for adversarially perturbed samples.

The effectiveness of using neighboring samples become more clear by looking at the distribution of membership scores of member, nonmember, and same-class neighbors from Algorithm 2, as shown in Figure 7.24. Here, same-class neighbors are closest samples whose class labels are the same as their neighbor member samples (corresponding to the if statement at line 6 in Algorithm 2) but are not members themselves. Different-class neighbors are closest samples whose class labels are different from their member neighbors. We filter out different-class neighbors in Algorithm 2 and 3 for the following reason: The distance in latent space does not have a fixed scale and it is only meaningful locally. In other words, two samples ϵ away from each other in one region of the latent space might be semantically very similar and two other samples ϵ away from each other in another region of the latent space might be semantically very different. To filter out the samples that are likely to be semantically different, we match the class label as a rudimentary criterion. Perhaps, more research is needed to find the proper scale for semantic similarity in each region of latent space. As shown in Figure 7.24, the distribution of same-class neighbors are much closer to the member samples and the distribution of different-class neighbors are closer to nonmember samples. That is the reason why MI attacks cannot avoid large false positive on discrediting samples.

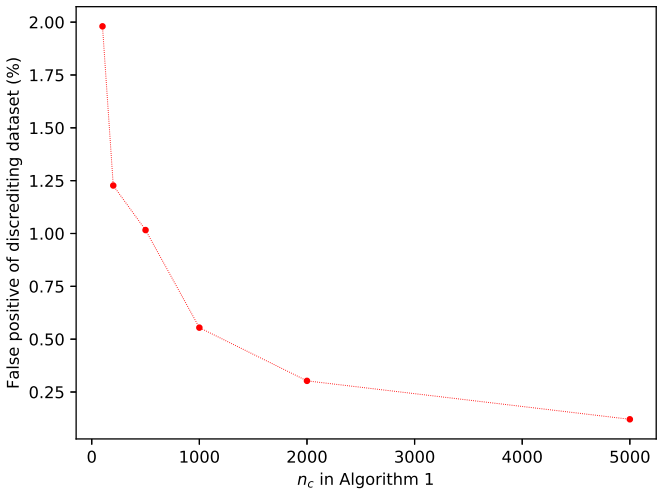


FIGURE 7.26. n_c in Algorithm 2 versus false positive rate of the discrediting dataset. As more samples from D_c set is involved in the process, the discrediting capability of the algorithm diminishes.

Interestingly, the observation from Figure 7.23 (b) that the membership scores of nonmember neighbors do not have clear positive correlation may lead to the perception that any member sample can be used as a part of D_c to create discrediting dataset. There is a fundamental limitation in the experiment related to Figure 7.23 (b): When searching for the closest neighbor for each member sample in CINIC dataset, many duplicate samples are picked. In other words, many member samples share the same closest sample in CINIC dataset. Consequently, although member samples in x-axis of Figure 7.23 (b) are all unique, the corresponding neighbor samples in y-axis are not necessarily unique. This is important because the discrediting dataset provided to the judge should not have duplicate samples, otherwise the discrediting process was trivial. That is why such experiment is inconclusive for algorithm 2 in Figure 7.23 (b).

To investigate the correlation between the membership score of a member sample and the quality of corresponding discrediting dataset, we conduct an extra experiment. Instead of using all member samples, we use algorithm 2 with different n_c. The larger the n_c is, the more samples with lower membership score are involved in the process. Here, we set the threshold such that the false positive rate is 0.01% on the test dataset. Then, we use that threshold to compute the false positive on the discrediting dataset. As shown in Figure 7.26, it is clear that including samples

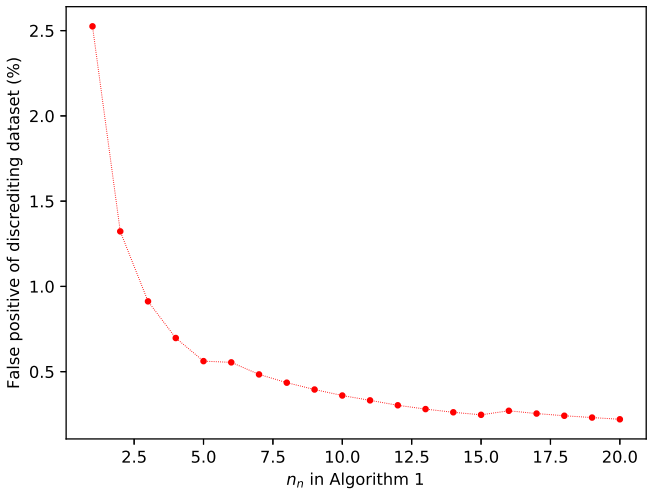


FIGURE 7.27. n_n in Algorithm 2 versus false positive rate of the discrediting dataset. As more samples from D_c set is involved in the process, the discrediting capability of the algorithm diminishes.

with smaller membership score degrades the discredibility quality. Hence, it implies a positive correlation between the membership score of a member sample and the membership score of the corresponding nonmember neighbor.

Hypothesis 2. *The closer the neighboring nonmember sample is to the member sample, the more similar their membership score would be.*

In Algorithm 2 we sort all neighbors with respect to their distance and explicitly prioritize the closest samples. The natural question is if there is a correlation between distance and the membership score. Figure 7.25 demonstrates the correlation between the distance of a member sample to its nonmember neighbor and the absolute membership score difference. Similar to the previous experiment, there is a clear positive correlation in the case of crafted samples using Algorithm 3 and apparent lack of correlation in the case of natural samples. As discussed earlier, an experiment with a larger set of natural samples is needed to investigate the correlation for Algorithm 2 conclusively.

It is also interesting to see the correlation of the index of the neighbors and their membership score. In Figure 7.27, as we include second, third, and n-th closest sample in the discrediting dataset, the false positive rate diminishes. It conveys that the further away from a sample we go, the membership score decreases. Although the previous experiment in Figure 7.25 (a) is inconclusive

about the correlation between the distance and the membership score, this experiment implies otherwise.

7.7. Discussion

Implication of Discredibility: The implications of the discredibility is beyond the example of auditing we discuss in this dissertation. What we have shown is that the membership score distribution of member samples are similar to their nonmember neighbors. Using the loose definition of subpopulation, referring to samples close in the latent space, we argue that current membership inference attacks identify the *memorized subpopulations*, not the *memorized samples*. In other words, MI attacks can identify that a sample from a subpopulation is a member, but they cannot reliably identify which exact sample in that subpopulation is in the train set and which is not. The notion of memorized subpopulations might be interesting by itself in certain applications as discussed next. However, it certainly is not what membership inference attacks promise to deliver.

Experimental vs Practical Setting: As argued in [**7**], MI attack reports should include the true positive rate at low false positive rate like various area of computer security [**42**, **55**, **59**, **80**]. Despite the similarities, there is an inherent difference between MI and other computer security applications. In membership inference, the ratio of positive samples are very small in comparison with all natural samples, similar to other computer security applications. However, the number of positive samples are *fixed*, unlike other applications. Now, let's assume the common practice in MI literature where the entire fixed positive (member) samples are included in the performance evaluation. Now, if we randomly collect billions of samples and add to the evaluation dataset, we only increase the number of negative samples because all positive samples had already been included. This means that the ratio of the number of true positive (TP) to the number of false positive (FP) depends on the size of the evaluating dataset because FP can infinitely grow in practice while TP is fixed. That is why the *low false positive* **ratio** in the evaluation setting does not necessarily indicate *small false positives* in practice. The existence of regions of high false positive rate, as shown in this chapter, means that in practice when a large number of negative samples exists in wild, the false positive samples dramatically outnumber the true positive samples. This limits the application of MI attacks in practice.

Membership Inference Application: The ability to identify memorized subpopulations is useful in certain applications. For example, if a notion of subpopulation in latent space indicates individual users, it can be used for user-level membership inference, similar to [**67**]. A prominent example is face recognition where MI attacker (auditor) aims to know if a person's images have been unlawfully used or not. Interestingly, what we have shown in this chapter suggests that the attacker does not need to know the exact training images to perform user-level membership inference. Hence, the MI attack in this case may be more practical than previously thought.

Auditing as An MI Application: While many MI attacks have been proposed in the literature, not much discussion exist on how MI attacks can be used in real world scenarios. The auditing example we propose is a first attempt to address this limitation, by providing a potential real-world application of MI. While our work demonstrates the limitation of *existing* MI attack techniques, it does not imply that MI attacks cannot be useful/practical. Especially, considering the above-mentioned user-level MI attack, we hope that the potential usage of MI methodologies as a privacy auditing tool can inspire new research directions on MI and its practical applications.

Limitations: Our analysis lacks comprehensiveness in two areas. First, we do not have much larger dataset than CINIC to make sure that the majority of input space is covered. It might not be even remotely possible. Although we indirectly shows the evidence of such a positive correlation in Section 7.6, better experimental setting/dataset is needed for Algorithm 2. Second, the BiGAN architecture proposed in [**98**] to train a generator is far from perfect. Since we use the BiGAN in both Algorithm 3 and Rezaei's MI attack [**98**], it affects the performance of both of them. Hence, a better generator model may dramatically change the results of these two methods. It remains unclear whether a better generator helps the MI attack more or helps the discrediting algorithm more.

7.8. Conclusion

In this chapter, we show that there exist numerous regions of input space where membership inference attacks frequently label non-member samples as members and, thus, exhibit high false positive rate. These regions are of paramount importance because the victim (auditee) can find them without any information about the membership inference attack or query access to the attack.

Then, we showcase a practical scenario where the membership inference attacks are used in a trail by an auditor (investigator or MI attacker) to prove to the judge that the auditee (MI victim) unlawfully used private data. Then, we show that the auditee can provide unlimited samples from the aforementioned regions and seriously challenge the credibility of the auditor (MI attack).

To achieve this goal, we propose three algorithms. The goal of all these algorithms is to search/craft samples whose latent representation is similar to a member sample or a sample to which the MI attack has already assigned high membership score. Using these algorithms allows the auditee to provide a dataset to the judge where MI attacks perform catastrophically poorly. We show that false positive rate of SOTA algorithm can jump from 0.01% to hundreds or thousands time larger when evaluated on auditee's dataset in comparison with the auditor's dataset. Therefore, we demonstrate that the discredibility issue is a serious concern when MI attacks are used in practice. In future, we investigate the possibility of new types of membership inference attacks immune to discredibility.

CHAPTER 8

Future Directions and Conclusion

8.1. Conclusion

In this dissertation, we aim to study membership inference attacks in a more practical settings and also to propose membership inference attacks that are more reliable in real world scenarios. Our first study, dated back to 2019 focused on the first generation of the membership inference attacks which were the state-of-the-art MI attack of the time. Through comprehensive evaluations, we show that the common practice of reporting accuracy/precision/recall is misleading for membership inference attacks. We show that FAR can provide a better picture of the state of a MI attack under investigation. We, along a few other studies [**15**, **66**], simultaneously propose a simple baseline, with which the random guess should be substitute, called Gap/Naive attack. We show that none of the contemporary attacks were able to consistently outperform such baseline and, hence, they are not reliable in practice.

Our observation about the ineffectiveness of the first generation of membership inference attacks have been corroborated multiple times [**7**,**39**,**66**,**126**]. In particular, Carlini at al. [**7**] suggest using true positive at low false positive often used in security domain to reliably report and compare effective membership inference attacks. Our work and the aforementioned studies triggered a new generation of membership inference attacks that adopted difficulty calibration strategy that allows them to achieve moderate true positive in very low false positive ratio.

In this dissertation, we propose a new membership inference attack with calibration that reduces the computation costs of the state-of-the-art MI attacks significantly. SOTA MI attacks with difficulty calibration need a dozens or hundreds of models to be trained per sample [**7**, **126**]. This limitation makes these MI attacks impractical for large complex models which take days to train. The simplified versions of these attacks are proposed to tackle this issue but they cannot achieve the same performance as the original ones and they still need a dozen to hundreds of models to

be trained per target model. We propose a subpopulation-based MI attack which fundamentally change the calibration process such that it does not need shadow models to calibrate the membership score. Instead, it calibrates the membership score based on semantically similar samples. In other words, our attack essentially compares the victim model's output on the target sample versus victim model's output on samples from the same subpopulation, instead of comparing the victim's model output versus shadow models' output.

This new way of approaching membership inference obviates the need to train dozens to hundreds of shadow models and makes MI attacks more computationally efficient. Moreover, we show that when samples from the same subpopulation are not available, we can train a single generator using BiGAN-like architecture to craft samples from subpopulations. Hence, in the worst case, we only need to train a single generator. Our evaluation results demonstrate that our attack can achieve the state-of-the-art MI attack accuracy with no shadow model training.

Furthermore, we propose a user-level MI attack on metric embedding learning. This attack differs from most existing MI attacks in two aspects: First, we focus on the user-level MI attack which is more practical in tasks where the exact training data samples used in training are not available. For instance, if a live video of a user is captured and used during the training, the same video might not be available to the user itself, let alone an MI attacker. However, the user can capture numerous new samples with his/her cell phone's camera and use it to launch user-level MI attack. That is why our user-level MI attack is more practical. Second, the attack focuses on metric embedding learning scenario where the existing confidence-based MI attacks do not work. In contrast with existing MI attacks, we use a measure of compactness of clusters in embedding space to identify membership, and consequently, obviate the need to access confidence values. Our attack achieves the state-of-the-art performance in several datasets, where user-level MI attack is of paramount importance.

Additionally, we study whether there are machine learning methodologies that have negative impact in terms of privacy. In particular, we discover that there is a trade-off between accuracy and privacy (in terms of membership inference) when deep ensembles are used. More precisely, when deep ensembles improve the overall accuracy of the classification, it also allows more effective membership inference on the deep ensemble model in comparison with a single model. Conversely,

one can construct a deep ensemble using a set of under-fitted models and achieve the same accuracy as a single well-trained model, while decreasing the effectiveness of membership inference on the deep ensemble. In other words, the widely-used deep ensemble approach can be used to improve accuracy or privacy, but not both at the same time.

We show that similar accuracy-privacy trade-off appears in most membership inference attacks. We comprehensively evaluate several membership inference defense mechanisms to break the trade-off and allow the deep ensemble to perform better with respect to both accuracy and privacy. However, none of the common defense mechanisms could achieve that. We investigate the root cause of the issue and we find that deep ensembles cause distribution shift of the confidence values in a way that makes the train and test set more distinguishable. After revealing the root cause of the issue, we suggest a simple, yet effective, approach to prevent the distribution shift to favor membership inference attack. We suggest replacing the ensemble averaging mechanism of deep ensembles with maximum confidence or first-correct confidence value. By doing so, we avoid confidence averaging over several models which essentially cause the issue. We show our maximum confidence approach can improve both accuracy and privacy at the same time by distorting the confidence signal from the output. We compare the confidence distortion caused by our approach and other confidence-masking approach and illustrate that our approach imposes the least distortion to the confidence values. This is crucial for applications that need accurate confidence estimation. The main advantage of our approach is that it can be easily adopted for the already-deployed deep ensemble models with minimum disruption/cost/time while it improves its robustness against MI attack with no cost.

Finally, we examine a practical use of membership inference attacks where it may be used by users to identify if their data has been unlawfully used to train a model. We introduce a legal scenario where an auditor uses MI attacks to identify unlawful use of user's data by an auditee. Then, the auditor needs to convince juries/judge that the MI attack is reliable with near zero false positive. In this dissertation, we show that the auditor can produce/find a large number of data samples for which the MI attack catastrophically fails without any knowledge about the MI attack itself. We call this phenomenon *discredibililty* because it can be used by the auditee to challenge the credibility of the auditee and, hence, dismisses the case.

For the auditee to be able to challenge the auditor, we propose three discredibililty algorithms. The goal of these algorithms is to search/craft samples whose latent representation is similar to a member sample or a sample to which the MI attack has already assigned high membership score. Using these algorithms allows the auditee to provide a dataset to the judge where MI attacks perform poorly. We show that false positive rate of SOTA algorithm can jump from 0.01% to hundreds or thousands time larger when evaluated on auditee's dataset in comparison with the auditor's dataset. Therefore, we demonstrate that the discredibility issue is a serious concern when MI attacks are used in practice.

8.2. Future Work

Despite numerous studies on membership inference attacks, we show that they do not perform well in practice. First, the metrics used to report the performance and the experimental settings was not good enough for the purpose of reliable membership inference attack. Second, we show that there are unlimited number of non-member samples that even SOTA MI attacks falsely mislabel as member samples. These issues limit the application of MI attacks in practice.

The future studies should focus on specific use cases or application where MI attacks may actually perform well. Currently, majority of MI attacks are general-purpose attacks which claim to work regardless of the target model/task/dataset. This might be a wrong way to do membership inference. It is possible that general-purpose MI attack might not be possible. It is better to focus on a specific model/task/dataset, similar to [**9**] where they focus on GPT-2 language models and they were able to recover exact training samples. Similar approaches specific to a model/task/data needs to be investigate in computer vision, speech recognition, etc.

One possible scenario where current SOTA membership inference attack may work well without the discredibility issue is probably the user-level or subpopulation-level membership inference, instead of record-level membership inference. The capability of such attack needs further investigation and comprehensive evaluation. Nevertheless, our results in Section 6 indicates that such attack scenarios are more suitable for current SOTA membership inference attacks. More importantly, there are many applications where user-level or subpopulation-level membership inference makes more sense to begin with, such as models trained on faces, voices, medical images, etc.

Bibliography

[1] N. Akhtar and A. Mian, *Threat of adversarial attacks on deep learning in computer vision: A survey*, Ieee Access, 6 (2018), pp. 14410–14430.

[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, 10 (2015), pp. 137–150.

[3] S. Axelsson, *The base-rate fallacy and the difficulty of intrusion detection*, ACM Transactions on Information and System Security (TISSEC), 3 (2000), pp. 186–205.

[4] M. Azadmanesh, B. S. Ghahfarokhi, and M. A. Talouki, *A white-box generator membership inference attack against generative models*, in 2021 18th International ISC Conference on Information Security and Cryptology (ISCISC), IEEE, 2021, pp. 13–17.

[5] L. Batina, S. Bhasin, D. Jap, and S. Picek, *{CSI}{NN}: Reverse engineering of neural network architectures through electromagnetic side channel*, in 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 515–532.

[6] D. Bu, X. Wang, and H. Tang, *Haplotype-based membership inference from summary genomic data*, Bioinformatics, 37 (2021), pp. i161–i168.

[7] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, *Membership inference attacks from first principles*, in 2022 IEEE Symposium on Security and Privacy (SP), IEEE, 2022, pp. 1897–1914.

[8] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, *The secret sharer: Evaluating and testing unintended memorization in neural networks*, in 28th {USENIX} Security Symposium ({USENIX} Security 19), 2019, pp. 267–284.

[9] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al., *Extracting training data from large language models*, in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2633–2650.

[10] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, *A survey on adversarial attacks and defences*, CAAI Transactions on Intelligence Technology, 6 (2021), pp. 25–45.

[11] D. Chen, N. Yu, Y. Zhang, and M. Fritz, *Gan-leaks: A taxonomy of membership inference attacks against generative models*, in Proceedings of the 2020 ACM SIGSAC conference on computer and communications security, 2020, pp. 343–362.

[12] J. Chen, M. I. Jordan, and M. J. Wainwright, *Hopskipjumpattack: A query-efficient decision-based attack*, in 2020 ieee symposium on security and privacy (sp), IEEE, 2020, pp. 1277–1294.

[13] J. Chen, H. Zheng, M. Su, T. Du, C. Lin, and S. Ji, *Invisible poisoning: Highly stealthy targeted poisoning attack*, in Information Security and Cryptology: 15th International Conference, Inscrypt 2019, Nanjing, China, December 6–8, 2019, Revised Selected Papers 15, Springer, 2020, pp. 173–198.

[14] F. Chollet, *Xception: Deep learning with depthwise separable convolutions*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[15] C. A. C. Choo, F. Tramer, N. Carlini, and N. Papernot, *Label-only membership inference attacks*, arXiv preprint arXiv:2007.14321, (2020).

[16] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, *Label-only membership inference attacks*, in International conference on machine learning, PMLR, 2021, pp. 1964–1974.

[17] P. Danhier, C. Massart, and F.-X. Standaert, *Fidelity leakages: Applying membership inference attacks to preference data*, in IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2020, pp. 728–733.

[18] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, *Cinic-10 is not imagenet or cifar-10*, arXiv preprint arXiv:1810.03505, (2018).

[19] J. Donahue, P. Krähenbühl, and T. Darrell, *Adversarial feature learning*, arXiv preprint arXiv:1605.09782, (2016).

[20] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, *Boosting adversarial attacks with momentum*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9185–9193.

[21] V. Duddu, D. Samanta, D. V. Rao, and V. E. Balas, *Stealing neural networks via timing side channels*, arXiv preprint arXiv:1812.11720, (2018).

[22] C. Dwork, *Differential privacy: A survey of results*, in International conference on theory and applications of models of computation, Springer, 2008, pp. 1–19.

[23] A. Fawzi, S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto, *Empirical study of the topology and geometry of deep networks*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3762–3770.

[24] V. Feldman, *Does learning require memorization? a short tale about a long tail*, in Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, 2020, pp. 954–959.

[25] S. Fort, H. Hu, and B. Lakshminarayanan, *Deep ensembles: A loss landscape perspective*, arXiv preprint arXiv:1912.02757, (2019).

[26] 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, 2015, pp. 1322–1333.

[27] 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 23rd USENIX Security Symposium (USENIX Security 14), 2014, pp. 17–32.

[28] 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 Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 619–633.

[29] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, *Loss surfaces, mode connectivity, and fast ensembling of dnns*, in Advances in Neural Information Processing Systems, 2018, pp. 8789–8798.

[30] U. Gupta, D. Stripelis, P. K. Lam, P. Thompson, J. L. Ambite, and G. Ver Steeg, *Membership inference attacks on deep regression models for neuroimaging*, in Medical Imaging with Deep Learning, PMLR, 2021, pp. 228–251.

[31] I. Hagestedt, M. Humbert, P. Berrang, I. Lehmann, R. Eils, M. Backes, and Y. Zhang, *Membership inference against dna methylation databases*, in 2020 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2020, pp. 509–520.

[32] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, *Logan: Membership inference attacks against generative models*, arXiv preprint arXiv:1705.07663, (2017).

[33] 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.

[34] X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, *Node-level membership inference attacks against graph neural networks*, arXiv preprint arXiv:2102.05429, (2021).

[35] Z. He, T. Zhang, and R. B. Lee, *Model inversion attacks against collaborative inference*, in Proceedings of the 35th Annual Computer Security Applications Conference, 2019, pp. 148–162.

[36] A. Hermans, L. Beyer, and B. Leibe, *In defense of the triplet loss for person re-identification*, arXiv preprint arXiv:1703.07737, (2017).

[37] B. Hilprecht, M. Härterich, and D. Bernau, *Monte carlo and reconstruction membership inference attacks against generative models.*, Proc. Priv. Enhancing Technol., 2019 (2019), pp. 232–249.

[38] ——, *Reconstruction and membership inference attacks against generative models*, arXiv preprint arXiv:1906.03006, (2019).

[39] D. Hintersdorf, L. Struppek, and K. Kersting, *Do not trust prediction scores for membership inference attacks*, arXiv preprint arXiv:2111.09076, (2021).

[40] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, *Person Re-Identification by Descriptive and Discriminative Classification*, in Proc. Scandinavian Conference on Image Analysis (SCIA), 2011.

[41] S. Hisamoto, M. Post, and K. Duh, *Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system?*, Transactions of the Association for Computational Linguistics, 8 (2020), pp. 49–63.

[42] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner, *Detecting credential spearphishing in enterprise settings*, in 26th USENIX Security Symposium (USENIX Security 17), 2017, pp. 469–485.

[43] H. Hu and J. Pang, *Membership inference attacks against gans by leveraging over-representation regions*, in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 2387–2389.

[44] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, *Membership inference attacks on machine learning: A survey*, ACM Computing Surveys (CSUR), 54 (2022), pp. 1–37.

[45] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, *Snapshot ensembles: Train 1, get m for free*, arXiv preprint arXiv:1704.00109, (2017).

[46] 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.

[47] H. Huang, W. Luo, G. Zeng, J. Weng, Y. Zhang, and A. Yang, *Damia: Leveraging domain adaptation as a defense against membership inference attacks*, arXiv preprint arXiv:2005.08016, (2020).

[48] H. Huang, J. Mu, N. Z. Gong, Q. Li, B. Liu, and M. Xu, *Data poisoning attacks to deep learning based recommender systems*, arXiv preprint arXiv:2101.02644, (2021).

[49] A. Jagannatha, B. P. S. Rawat, and H. Yu, *Membership inference attack susceptibility of clinical language models*, arXiv preprint arXiv:2104.08305, (2021).

[50] B. Jayaraman and D. Evans, *Evaluating differentially private machine learning in practice*, in 28th {USENIX} Security Symposium ({USENIX} Security 19), 2019, pp. 1895–1912.

[51] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans, *Revisiting membership inference under realistic assumptions*, In Proceedings on Privacy Enhancing Technologies (PoPETs), (2021).

[52] 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.

[53] W. Jiang, H. Li, S. Liu, X. Luo, and R. Lu, *Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles*, IEEE transactions on vehicular technology, 69 (2020), pp. 4439–4449.

[54] A. Jnaini, A. Bettar, and M. A. Koulali, *How powerful are membership inference attacks on graph neural networks?*, in Proceedings of the 34th International Conference on Scientific and Statistical Database Management, 2022, pp. 1–4.

[55] A. Kantchelian, M. C. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A. D. Joseph, and J. D. Tygar, *Better malware ground truth: Techniques for weighting anti-virus vendor labels*, in Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, 2015, pp. 45–56.

[56] H. Karimi, T. Derr, and J. Tang, *Characterizing the decision boundary of deep neural networks*, arXiv preprint arXiv:1912.11460, (2019).

[57] M. Kesarwani, B. Mukhoty, V. Arya, and S. Mehta, *Model extraction warning in mlaas paradigm*, in Proceedings of the 34th Annual Computer Security Applications Conference, 2018, pp. 371–380.

[58] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, *On large-batch training for deep learning: Generalization gap and sharp minima*, arXiv preprint arXiv:1609.04836, (2016).

[59] J. Z. Kolter and M. A. Maloof, *Learning to detect and classify malicious executables in the wild.*, Journal of Machine Learning Research, 7 (2006).

[60] D. Kondratyuk, M. Tan, M. Brown, and B. Gong, *When ensembling smaller models is more efficient than single large models*, arXiv preprint arXiv:2005.00570, (2020).

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

[62] L. I. Kuncheva and C. J. Whitaker, *Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy*, Machine learning, 51 (2003), pp. 181–207.

[63] B. Lakshminarayanan, A. Pritzel, and C. Blundell, *Simple and scalable predictive uncertainty estimation using deep ensembles*, in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, 2017, p. 6405–6416.

[64] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[65] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, *Why m heads are better than one: Training a diverse ensemble of deep networks*, arXiv preprint arXiv:1511.06314, (2015).

[66] K. Leino and M. Fredrikson, *Stolen memories: Leveraging model memorization for calibrated white-box membership inference*, in 29th {USENIX} Security Symposium ({USENIX} Security 20), 2020, pp. 1605–1622.

[67] G. Li, S. Rezaei, and X. Liu, *User-level membership inference attack against metric embedding learning*, in ICLR 2022 Workshop on PAIRˆ2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data, 2022.

[68] J. Li, N. Li, and B. Ribeiro, *Membership inference attacks and defenses in classification models*, in Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, 2021, pp. 5–16.

[69] Z. Li and Y. Zhang, *Label-leaks: Membership inference attack with label*, arXiv preprint arXiv:2007.15528, (2020).

[70] Y. Liang and R. Samavi, *Towards robust deep learning with ensemble networks and noisy layers*, arXiv preprint arXiv:2007.01507, (2020).

[71] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng, *Socinf: Membership inference attacks on social media health data with machine learning*, IEEE Transactions on Computational Social Systems, 6 (2019), pp. 907–921.

[72] H. Liu, J. Jia, W. Qu, and N. Z. Gong, *Encodermi: Membership inference against pre-trained encoders in contrastive learning*, in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 2081–2095.

[73] K. S. Liu, C. Xiao, B. Li, and J. Gao, *Performing co-membership attacks against deep generative models*, in 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 459–467.

[74] E. Lobacheva, N. Chirkova, M. Kodryan, and D. Vetrov, *On power laws in deep ensembles*, arXiv preprint arXiv:2007.08483, (2020).

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

[76] 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).

[77] Y. Long, L. Wang, D. Bu, V. Bindschaedler, X. Wang, H. Tang, C. A. Gunter, and K. Chen, *A pragmatic approach to membership inferences on machine learning models*, in 2020 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2020, pp. 521–534.

[78] S. Mehnaz, N. Li, and E. Bertino, *Black-box model inversion attribute inference attacks on classification models*, arXiv preprint arXiv:2012.03404, (2020).

[79] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, *Exploiting unintended feature leakage in collaborative learning*, in 2019 IEEE symposium on security and privacy (SP), IEEE, 2019, pp. 691–706.

[80] V. Metsis, I. Androutsopoulos, and G. Paliouras, *Spam filtering with naive bayes-which naive bayes?*, in CEAS, vol. 17, Mountain View, CA, 2006, pp. 28–69.

[81] Y. Miao, X. Minhui, C. Chen, L. Pan, J. Zhang, B. Z. H. Zhao, D. Kaafar, and Y. Xiang, *The audio auditor: user-level membership inference in internet of things voice services*, Proceedings on Privacy Enhancing Technologies, 2021 (2021), pp. 209–228.

[82] Y. Miao, M. Xue, C. Chen, L. Pan, J. Zhang, B. Z. H. Zhao, D. Kaafar, and Y. Xiang, *The audio auditor: user-level membership inference in internet of things voice services*, arXiv preprint arXiv:1905.07082, (2019).

[83] D. Mickisch, F. Assion, F. Gressner, W. Günther, and M. Motta, *Understanding the decision boundary of deep neural networks: An empirical study*, arXiv preprint arXiv:2002.01810, (2020).

[84] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri, *Quantifying privacy risks of masked language models using membership inference attacks*, arXiv preprint arXiv:2203.03929, (2022).

[85] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, *Robustness via curvature regularization, and vice versa*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9078–9086.

[86] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, *Towards poisoning of deep learning algorithms with back-gradient optimization*, in Proceedings of the 10th ACM workshop on artificial intelligence and security, 2017, pp. 27–38.

[87] S. K. Murakonda and R. Shokri, *Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning*, arXiv preprint arXiv:2007.09339, (2020).

[88] 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.

[89] ——, *Comprehensive privacy analysis of deep learning*, in 2019 ieee symposium on security and privacy, 2019.

[90] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, *Reading digits in natural images with unsupervised feature learning*, in NIPS Workshop, 2011.

[91] P. Oberdiek, M. Rottmann, and H. Gottschalk, *Classification uncertainty of deep neural networks based on gradient information*, in IAPR Workshop on Artificial Neural Networks in Pattern Recognition, Springer, 2018, pp. 113–125.

[92] I. E. Olatunji, W. Nejdl, and M. Khosla, *Membership inference attack on graph neural networks*, in 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), IEEE, 2021, pp. 11–20.

[93] T. Orekondy, B. Schiele, and M. Fritz, *Knockoff nets: Stealing functionality of black-box models*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4954–4963.

[94] M. P. Parisot, B. Pejo, and D. Spagnuelo, *Property inference attacks on convolutional neural networks: Influence and implications of target model's complexity*, arXiv preprint arXiv:2104.13061, (2021).

[95] R. Pascanu, G. Montufar, and Y. Bengio, *On the number of response regions of deep feed forward networks with piece-wise linear activations*, arXiv preprint arXiv:1312.6098, (2013).

[96] S. Rahimian, T. Orekondy, and M. Fritz, *Sampling attacks: Amplification of membership inference attacks by repeated queries*, arXiv preprint arXiv:2009.00395, (2020).

[97] K. Ren, T. Zheng, Z. Qin, and X. Liu, *Adversarial attacks and defenses in deep learning*, Engineering, 6 (2020), pp. 346–360.

[98] S. Rezaei, , and X. Liu, *An efficient subpopulation-based membership inference attack*, arXiv preprint arXiv:2203.02080, (2022).

[99] S. Rezaei and X. Liu, *A target-agnostic attack on deep models: Exploiting security vulnerabilities of transfer learning*, International conference on learning representations (ICLR), (2019).

[100] ———, *On the difficulty of membership inference attacks*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

[101] S. Rezaei, Z. Shafiq, and X. Liu, *Accuracy-privacy trade-off in deep ensemble: A membership inference perspective*, in 2023 IEEE Symposium on Security and Privacy (SP), 2023.

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

[103] 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, PMLR, 2019, pp. 5558–5567.

[104] O. Sagi and L. Rokach, *Ensemble learning: A survey*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8 (2018), p. e1249.

[105] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, *Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models*, arXiv preprint arXiv:1806.01246, (2018).

[106] 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.

[107] M. A. Shah, J. Szurley, M. Mueller, A. Mouchtaris, and J. Droppo, *Evaluating the vulnerability of end-to-end automatic speech recognition models to membership inference attacks.*, in Interspeech, 2021, pp. 891–895.

[108] V. Shejwalkar, H. A. Inan, A. Houmansadr, and R. Sim, *Membership inference attacks against nlp classification models*, in NeurIPS 2021 Workshop Privacy in Machine Learning, 2021.

[109] J. Shen, X. Zhu, and D. Ma, *Tensorclog: An imperceptible poisoning attack on deep neural network applications*, IEEE Access, 7 (2019), pp. 41498–41506.

[110] 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.

[111] C. Song, T. Ristenpart, and V. Shmatikov, *Machine learning models that remember too much*, in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 587–601.

[112] C. Song and V. Shmatikov, *Auditing data provenance in text-generation models*, in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 196–206.

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

[114] 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.

[115] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[116] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[117] F. Tramer and D. Boneh, *Differentially private learning needs better features (or much more data)*, arXiv preprint arXiv:2011.11660, (2020).

[118] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, *Stealing machine learning models via prediction {APIs}*, in 25th USENIX security symposium (USENIX Security 16), 2016, pp. 601–618.

[119] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, *Demystifying membership inference attacks in machine learning as a service*, IEEE Transactions on Services Computing, (2019).

[120] W.-C. Tseng, W.-T. Kao, and H.-y. Lee, *Membership inference attacks against self-supervised speech models*, arXiv preprint arXiv:2111.05113, (2021).

[121] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, E. Dutkiewicz, et al., *Learning latent representation for iot anomaly detection*, IEEE Transactions on Cybernetics, (2020).

[122] 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.

[123] X. Wang, D. Kondratyuk, K. M. Kitani, Y. Movshovitz-Attias, and E. Eban, *Multiple networks are more efficient than one: Fast and accurate models via ensembles and cascades*, arXiv preprint arXiv:2012.01988, (2020).

[124] X. Wang and W. H. Wang, *Group property inference attacks against graph neural networks*, in Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 2871–2884.

[125] Z. Wang, N. Huang, F. Sun, P. Ren, Z. Chen, H. Luo, M. de Rijke, and Z. Ren, *Debiasing learning for membership inference attacks against recommender systems*, in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 1959–1968.

[126] L. Watson, C. Guo, G. Cormode, and A. Sablayrolles, *On the importance of difficulty calibration in membership inference attacks*, arXiv preprint arXiv:2111.08440, (2021).

[127] B. Wu, X. Yang, S. Pan, and X. Yuan, *Adapting membership inference attacks to gnn for graph classification: Approaches and implications*, in 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 2021, pp. 1421–1426.

[128] K. Wu, Z. Chen, and W. Li, *A novel intrusion detection model for a massive network using convolutional neural networks*, IEEE Access, 6 (2018), pp. 50850–50859.

[129] M. Wu, X. Zhang, J. Ding, H. Nguyen, R. Yu, M. Pan, and S. T. Wong, *Evaluation of inference attack models for deep learning on medical data*, arXiv preprint arXiv:2011.00177, (2020).

[130] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*, 2017.

[131] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, *Deep music analogy via latent representation disentanglement*, arXiv preprint arXiv:1906.03626, (2019).

[132] Z. Yang, L. Li, X. Xu, B. Kailkhura, T. Xie, and B. Li, *On the certified robustness for ensemble models and beyond*, arXiv preprint arXiv:2107.10873, (2021).

[133] Z. Yang, B. Shao, B. Xuan, E.-C. Chang, and F. Zhang, *Defending model inversion and membership inference attacks via prediction purification*, arXiv preprint arXiv:2005.03915, (2020).

[134] 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.

[135] C. Zhang and L. Bonomi, *Mitigating membership inference in deep learning applications with high dimensional genomic data*, in 2022 IEEE 10th International Conference on Healthcare Informatics (ICHI), IEEE, 2022, pp. 01–03.

[136] C. Zhang, X. Costa-Pérez, and P. Patras, *Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms*, IEEE/ACM Transactions on Networking, (2022).

[137] M. Zhang, Z. Ren, Z. Wang, P. Ren, Z. Chen, P. Hu, and Y. Zhang, *Membership inference attacks against recommender systems*, in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 864–879.

[138] Q. Zhang and S.-C. Zhu, *Visual interpretability for deep learning: a survey*, arXiv preprint arXiv:1802.00614, (2018).

[139] S. Zhang, M. Liu, and J. Yan, *The diversified ensemble neural network*, Advances in Neural Information Processing Systems, 33 (2020).

[140] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, *Adversarial attacks on deep-learning models in natural language processing: A survey*, ACM Transactions on Intelligent Systems and Technology (TIST), 11 (2020), pp. 1–41.

[141] Z. Zhang, C. Yan, and B. A. Malin, *Membership inference attacks against synthetic health data*, Journal of biomedical informatics, 125 (2022), p. 103977.

[142] B. Z. H. Zhao, A. Agrawal, C. Coburn, H. J. Asghar, R. Bhaskar, M. A. Kaafar, D. Webb, and P. Dickinson, *On the (in) feasibility of attribute inference attacks on machine learning models*, in 2021 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2021, pp. 232–251.

[143] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, *Scalable person re-identification: A benchmark*, in Computer Vision, IEEE International Conference on, 2015.

[144] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou, and P. S. Yu, *Adversarial attacks and defenses in deep learning: from a perspective of cybersecurity*, ACM Computing Surveys (CSUR), (2023).

[145] X. Zhou, M. Xu, Y. Wu, and N. Zheng, *Deep model poisoning attack on federated learning*, Future Internet, 13 (2021), p. 73.

[146] Y. Zou, Z. Zhang, M. Backes, and Y. Zhang, *Privacy analysis of deep learning in the wild: Membership inference attacks against transfer learning*, arXiv preprint arXiv:2009.04872, (2020).