

# Membership Inference Attacks on Machine Learning: A Survey

HONGSHENG HU, The University of Auckland, New Zealand  
 ZORAN SALCIC, The University of Auckland, New Zealand  
 GILLIAN, DOBBIE, The University of Auckland, New Zealand  
 XUYUN ZHANG, Macquarie University, Australia

Membership inference attack aims to identify whether a data sample was used to train a machine learning model or not. It can raise severe privacy risks as the membership can reveal an individual's sensitive information. For example, identifying an individual's participation in a hospital's health analytics training set reveals that this individual was once a patient in that hospital. Membership inference attacks have been shown to be effective on various machine learning models, such as classification models, generative models, and sequence-to-sequence models. Meanwhile, many methods are proposed to defend such a privacy attack. Although membership inference attack is an emerging and rapidly growing research area, there is no comprehensive survey on this topic yet. In this paper, we bridge this important gap in membership inference attack literature. We present the first comprehensive survey of membership inference attacks. We summarize and categorize existing membership inference attacks and defenses and explicitly present how to implement attacks in various settings. Besides, we discuss why membership inference attacks work and summarize the benchmark datasets to facilitate comparison and ensure fairness of future work. Finally, we propose several possible directions for future research and possible applications relying on reviewed works.

CCS Concepts: • **Security and privacy** → **Privacy protections**.

Additional Key Words and Phrases: Membership inference attack, deep learning, privacy risk, differential privacy.

## 1 INTRODUCTION

Machine learning has achieved tremendous progress for various learning tasks, from image classification and speech recognition to generating realistic-looking data. Besides the powerful computational resources, the availability of large datasets is a key factor contributing to machine learning success. As datasets may contain individuals' private information such as user speech, images, and medical records, it is essential that machine learning models do not leak too much information about their training data. However, recent studies [9, 92, 122] have shown that machine learning models are prone to memorize sensitive information of training data, making them vulnerable to several privacy attacks such as model extraction attack [102], attribute inference attack [23], property inference attack [25], and membership inference attack [90]. In this survey, we focus on membership inference attack.

Membership inference attacks on machine learning models aim to identify whether a data sample was used to train the target machine learning model or not. It can raise severe privacy risks to individuals. For example, identifying that a certain patient's clinical record was used to train a model associated with a disease reveals that the patient has this disease. Moreover, such privacy risk might lead commercial companies who wish to leverage machine learning as a service to violate privacy regulations. Veale et al. [107] argue that membership inference attacks on machine learning models increase their risks to be classified as personal data under the General Data Protection

---

Authors' addresses: Hongsheng Hu, The University of Auckland, Auckland, New Zealand, hhu603@aucklanduni.ac.nz; Zoran Salcic, The University of Auckland, Auckland, New Zealand, z.salcic@auckland.ac.nz; Gillian, Dobbie, The University of Auckland, Auckland, New Zealand, g.dobbie@auckland.ac.nz; Xuyun Zhang, Macquarie University, Sydney, NSW, Australia, xuyun.zhang@mq.edu.au.

Regulation (GDPR) [111]. Homer et al. [36] propose the first membership inference attack that infers the presence of a particular genome in a genomics dataset. Shokri et al. [90] present the first membership inference attack against machine learning models. Specifically, they demonstrate that an adversary can tell whether a data instance has been used to train a classifier or not, solely based on the prediction vector of the data instance (which is also known as black-box access to a trained machine learning model). Since then, a growing body of work further investigates and extends the topic of membership inference attacks on machine learning models. For example, membership inference attacks are initially investigated on classification models and the adversary only has black-box access to the target models. They are further explored on generative models [29] and investigated when the attacker has white-box access [72]. Meanwhile, there is a large body of work that proposes different defense methods that try to mitigate the effectiveness of membership inference attacks.

There are several surveys that summarise the privacy and security issues of machine learning models [18, 59, 81]. However, they include membership inference attacks as a subtopic and only present certain literature to introduce the basic concept. This paper attempts to provide the first comprehensive survey specifically on both membership inference attacks and defenses. Through this comprehensive overview, we aim to prepare a solid foundation for future research in this realm. The main contributions of the paper are as follows:

- We conduct the first comprehensive survey of membership inference attacks and defenses on machine learning models.
- We explicitly present how to implement different membership inference attacks when the attacker has various adversarial knowledge.
- We introduce novel designs of membership inference attacks on both classification and generative models and summarize them. We discuss why membership inference attacks work and categorize and summarize the defense mechanisms. We also summarize benchmark datasets to facilitate comparison and ensure fairness of future work.
- The study concludes with a discussion on future research directions in membership inference attacks and defenses based on reviewed works.

The rest of the paper is organized as follows. Section 2 introduces machine learning preliminaries, including the notations and basic concepts. In Section 3, we divide membership inference attacks into two types based on what knowledge is available to the adversary. In Section 4, we explicitly present how to implement membership inference attacks. Section 5 reviews novel designs of membership inference attacks. We point out why membership inference attacks work in Section 6 and categorize the defense mechanisms in Section 7. We present our outlook and propose some future directions for this promising research topic in Section 8 and conclude the paper in Section 9.

## 2 MACHINE LEARNING PRELIMINARIES

Machine learning (ML) is the study of computer algorithms that learn patterns from massive data. This section introduces machine learning preliminaries at a high level to facilitate the description and discussion of membership inference attacks in the subsequent sections. We present the notations and abbreviations used throughout the paper in Table 1. We introduce the types of machine learning models and how to train them.

### 2.1 Types of machine learning

Generally, we divide ML algorithms into three categories, i.e., *supervised learning*, *unsupervised learning*, and *reinforcement learning* depending on the information provided by the training data and the different learning tasks. We ignore the introduction of reinforcement learning since membership

Table 1. Summary of notations and acronyms used in the paper

Notations&Abbreviation	Explanation
$\mathbf{x}$	a multi-dimensional data instance
$y$	the label of $\mathbf{x}$
$D_{train}$	training dataset
$N$	number of data instances
$\mathcal{L}(\cdot)$	loss function
$f(\mathbf{x}; \theta)$	a machine learning model
$\hat{p}(y   \mathbf{x})$	prediction vector
$p_i$	confidence score
$I(\cdot)$	indicator function
ML	machine learning
SGD	stochastic gradient descent
GANs	generative adversarial networks
VAEs	variational autoencoder
MIA	membership inference attacks
NN	neural network
CNN	convolutional neural network
FL	federated learning
DP	differential privacy

inference attacks have not been involved in this category. *Deep learning* is a part of ML that focuses on deep neural networks (DNN). Membership inference attacks against ML models mainly focus on deep learning models.

**2.1.1 Supervised learning.** A supervised ML model aims to learn a general rule that maps inputs to outputs from a labeled dataset. Let  $D_{train} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$  be a training dataset, which contains  $N$  data instances where each of them consists of a feature vector  $\mathbf{x}$  and label  $y$ . An ML model is a function  $f(\mathbf{x}; \theta)$  that takes as input  $\mathbf{x}$  and outputs  $y = f(\mathbf{x}; \theta)$ , where  $\theta$  are parameters that are learned from  $D_{train}$ . When  $y$  is discrete, the learning task of  $f(\mathbf{x}; \theta)$  is called *classification*. Most of the membership inference attacks on supervised learning focus on classification problems.

**2.1.2 Unsupervised learning.** An unsupervised ML model aims to extract features and patterns from unlabeled data or labeled data without access to the labels. Recently, generative tasks, which aim to learn how to generate samples from the underlying data distribution have gained increasing attention. There are two typical generative models, i.e., Generative Adversarial Networks (GANs) [26] and Variational Autoencoders (VAEs) [49]. Membership inference attacks on unsupervised learning mainly focus on generative models of GANs and VAEs.

**GANs:** GANs aim to generate new samples that approximate the training data distribution. GANs consist of a generator  $\mathcal{G}$  and a discriminator  $\mathcal{D}$  and both of them are neural networks. The generator  $\mathcal{G}$  takes latent variable  $z$  as input and generates new samples  $\mathcal{G}_{\theta_{\mathcal{G}}}(z)$  that approximate the distribution of  $D_{train}$ . The discriminator  $\mathcal{D}$  receives samples of  $D_{train}$  and  $\mathcal{G}_{\theta_{\mathcal{G}}}(z)$  and is trained to learn the difference between them.  $\mathcal{G}$  and  $\mathcal{D}$  are trained simultaneously to compete so that  $\mathcal{G}$  learns to generate more and more realistic samples to fool  $\mathcal{D}$ .

**VAEs:** The aim of VAEs is the same as that of GANs. It is another popular generative model and consists of an encoder and a decoder that are both neural networks. The encoder takes  $\mathbf{x}$  as input and maps it into a latent space, while the decoder maps a latent variable  $z$  sampled from the latent space back to the data space and tries to reconstruct  $\mathbf{x}$  with small reconstruction error. After training, the decoder can take random variables  $z$  and generates new samples that approximate the data distribution of  $D_{train}$ .

## 2.2 Training of machine learning models

We first introduce how to train supervised ML models and unsupervised ML models of GANs and VAEs. Then, based on whether the training data is distributed over multi parties or not, we introduce two training approaches, i.e., *centralized training* and *distributed training*.

**Training of supervised models:** A well-trained supervised ML model should have a small expectation loss on the data it works on. However, as we do not know the true distribution of data, we can not calculate the model's expected risk. A realistic approach to train supervised ML models is Empirical Risk Minimization (ERM). The core idea is to measure the model's performance on a known training dataset. For a given dataset  $D_{train}$ , ERM tries to find the parameters  $\theta^*$  that minimize the following objective function:

$$\mathcal{R}_{D_{train}}(\theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y^{(n)}, f(\mathbf{x}^{(n)}; \theta)) \quad (1)$$

where  $\mathcal{L}(\cdot)$  is a loss function. We usually use an iterative optimization algorithm called *stochastic gradient descent* (SGD) to find the best parameters  $\theta^*$ . The SGD algorithm follows:

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial \mathcal{R}_{\mathcal{D}}(\theta)}{\partial \theta} \quad (2)$$

$$\frac{\partial \mathcal{R}_{\mathcal{D}}(\theta)}{\partial \theta} = \frac{1}{K} \sum_{n=1}^K \frac{\partial \mathcal{L}(y^{(n)}, f(\mathbf{x}^{(n)}; \theta))}{\partial \theta} \quad (3)$$

where  $K$  is the size of a small batch.  $\theta_t$  are iterative parameters in the  $t_{th}$  time and  $\alpha$  is the learning rate. Training is finished when the model converges to a local minimum where the gradient is close to zero.

**Training of GANs and VAEs:** As both the modules in GANs and VAEs are neural networks, SGD is usually used for training GANs and VAEs. SGD tries to find the parameters  $\theta_{\mathcal{G}}^*$  of GANs following the objective function:

$$\min_{\theta_{\mathcal{G}}} \max_{\theta_{\mathcal{D}}} \mathbb{E}_{\mathbf{x} \sim P_{data}} [\log(\mathcal{D}_{\theta_{\mathcal{D}}}(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim P_z} [\log(1 - \mathcal{D}_{\theta_{\mathcal{D}}}(\mathcal{G}_{\theta_{\mathcal{G}}}(\mathbf{z})))] \quad (4)$$

where  $\theta_{\mathcal{G}}$  and  $\theta_{\mathcal{D}}$  are parameters of the generator and the discriminator.  $P_{data}$  is the distribution of  $D_{train}$ , while  $P_z$  is the distribution of  $\mathbf{z}$ . For the training of VAEs, SGD tries to find the parameters  $\theta_{de}^*$  following the objective function:

$$\min_{\theta_{en}, \theta_{de}} -\mathbb{E}_{q_{\theta_{en}}(\mathbf{z}|\mathbf{x})} [p_{\theta_{de}}(\mathbf{x}|\mathbf{z})] + KL(q_{\theta_{en}}(\mathbf{z}|\mathbf{x})||P_z) \quad (5)$$

where  $q_{\theta_{en}}(\mathbf{z}|\mathbf{x})$  and  $p_{\theta_{de}}(\mathbf{x}|\mathbf{z})$  are the encoder and the decoder and  $\theta_{en}$  and  $\theta_{de}$  are their parameters.  $KL(\cdot||\cdot)$  is the KL divergence and  $P_z$  is the distribution of  $\mathbf{z}$ .

**Centralized and distributed training:** Depending on whether the training data is distributed over multi parties or not, there are two types of training approaches. In centralized training, we assume there is one centralized dataset  $D_{train}$  that contains all training data instances. For a predefined model  $f(\mathbf{x}; \theta)$ , the SGD algorithm is only interactive with  $D_{train}$  to update the parameters  $\theta$ . While in distributed training, we assume there are  $m$  distributed datasets  $D_1, D_2, \dots, D_m$  that each contain part of the training data instances. For a predefined model  $f(\mathbf{x}; \theta)$ , the SGD algorithm is interactive with  $D_1, D_2, \dots, D_m$  to update the parameters  $\theta$ .

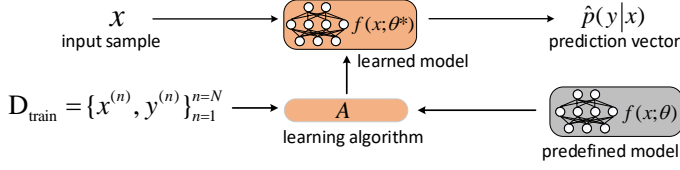


Fig. 1. A typical deep learning process for classification models.

### 3 TYPES OF MEMBERSHIP INFERENCE ATTACKS

Membership inference attacks (MIA) against ML models mainly focus on deep learning models because such models with high complexity are more prone to suffer from the overfitting issues which can be exploited by membership inference attacks. Based on what information is available to the adversary, i.e., *adversarial knowledge*, membership inference attacks can be divided into two types, *black-box inference attacks* and *white-box inference attacks*. In order to better understand these two types of attacks, we introduce a standard learning process of deep neural networks at a high level in Fig. 1.

Fig. 1 shows a typical deep learning process. We use a learning algorithm  $\mathcal{A}$  to train the predefined model  $f(\mathbf{x}; \theta)$  using the dataset  $D_{train} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$ . Once the training is finished, a learned model  $f(\mathbf{x}; \theta^*)$  can be used to make predictions for unseen data. Assume there is a new data instance  $\mathbf{x}$ , the learned model  $f(\mathbf{x}; \theta^*)$  takes it as input and outputs a vector  $\hat{p}(y|\mathbf{x})$ . The vector  $\hat{p}(y|\mathbf{x})$  is the prediction vector of probabilities that are often referred to as *confidence scores*. The class with the highest confidence score is selected as the predicted label for the data instance.

#### 3.1 Adversarial knowledge

MIA aims to determine whether a given data instance is part of the target model's training dataset or not. The target model refers to the model that the adversary wants to attack. For example, in Fig. 1, MIA tries to determine whether  $\mathbf{x} \in D_{train}$  or not. To better understand how we define black-box and white-box inference attacks, we first categorize the adversarial knowledge into four types, i.e., *data knowledge*, *training knowledge*, *model knowledge*, and *output knowledge*.

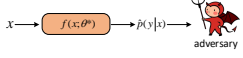
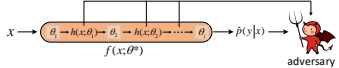
**Data knowledge:** Refers to knowledge of the data distribution of  $D_{train}$  and the dataset  $D_{train}$ . In most membership inference settings, the knowledge of the data distribution of  $D_{train}$  is available to the adversary. Thus, the adversary can obtain a dataset  $D'$  which contains data instances that come from the same data distribution as those in  $D_{train}$ .  $D_{train}$  and  $D'$  can be joint or disjoint, depending on the difficulty of attacks. To conduct a non-trivial membership inference, it is often assumed that  $D_{train}$  and  $D'$  is disjoint.

**Training knowledge:** Refers to knowledge of the learning algorithm  $\mathcal{A}$ . The knowledge includes the type of optimization algorithm, number of training steps, settings of the optimization algorithm, etc. This knowledge reveals how the target model is trained and it is often assumed to be available to the adversary in most MIA setting.

**Model knowledge:** Refers to knowledge of the learned model  $f(\mathbf{x}; \theta^*)$ . This knowledge constitutes two parts, *model architecture* and *model parameters*. The knowledge of model architecture includes the type of neural network, the number of layers, the type of activation function, etc. The knowledge of model parameters essentially is the knowledge of the learned parameters  $\theta^*$  in Fig. 1.

**Output knowledge:** Refers to knowledge of the prediction vector  $\hat{p}(y|\mathbf{x})$ . Depending on the amount of information provided for the adversary, the output knowledge can be divided into *full output knowledge*, *partial output knowledge*, and *label-only knowledge*. Full output knowledge means all the confidence scores of  $\mathbf{x}$  are available to the adversary, while partial output knowledge reveals

Table 2. Black-box and white-box membership inference attacks, based on whether the adversary knows the target model’s parameters or not.

Attacks	Adversarial knowledge		Description
	Common	Particular	
Black-box	Data knowledge Training knowledge Output knowledge	Model architecture	<p>The adversary only knows the architecture of the target model and has black-box access to it. For arbitrary input <math>\mathbf{x}</math>, the adversary can only obtain the prediction vector <math>\hat{p}(y   \mathbf{x})</math> but can not get the intermediate computations of <math>f(\mathbf{x}, \theta^*)</math>.</p> 
White-box		Model architecture Model parameters	<p>The adversary has full target model knowledge, including knowledge of both model architecture and parameters. Thus, the adversary can observe any intermediate computations at hidden layers <math>h(\mathbf{x}; \theta_i)</math>.</p> 

only partial confidence scores, e.g., the maximum three confidence scores. Label-only knowledge is an extreme case that the adversary can only know the predicted label of  $\mathbf{x}$ . It provides the minimum output information for the adversary.

Generally, the adversary is assumed to have data knowledge, training knowledge, and output knowledge of the target model. Based on whether the adversary can have access to the *model knowledge of model parameters*, MIA can be categorized into **black-box** and **white-box** inference attacks. In Table 2, we compare and intuitively demonstrate these two types of inferences.

### 3.2 Black-box inference attacks

In this setting, the adversary has black-box access to the target model. That is, for any data instance  $\mathbf{x}$ , the adversary can only obtain its prediction vector  $\hat{p}(y | \mathbf{x})$  computed by the target model. The internal parameters  $\theta^*$  of the target model are not accessible to the adversary, which means the intermediate computations of  $\mathbf{x}$  are not available. Most MIA focus on the black-box setting, as this setting is more challenge and realistic.

There are three types of black-box MIA based on different amount of output knowledge. Each type corresponds to a difficulty level of the attack. As described in Table 3, the first type of black-box MIA receives the maximum information of the model’s output, and is the easiest attack among the three types. Most black-box MIA [63, 85, 90, 94, 120] belong to this type. The second type receives partial information of the prediction vector. Its difficulty is in the middle among the three types of attacks. It receives less information than the first type of attack, however, Salem et al. [85] show that this type of attack can achieve comparable attack accuracy with that of the first type of attack. The third type receives the minimum information of the model’s output. It is an extreme case that the adversary only receives the predicted label, which means the target model reveals the minimum information to the adversary. Recently, Choquette-Choo et al. [16] and Li and Zhang [57] designed the label-only MIA in the black-box setting. By knowing the minimum output knowledge, they show the label-only attack achieves strong performance on a range of datasets. This indicates that ML models are more vulnerable to privacy attacks than we expect.

### 3.3 White-box inference attacks

In this setting, the adversary obtains all adversarial knowledge and has full access to the target model. This means for any data instance  $\mathbf{x}$ , the adversary not only obtains the prediction vector

Table 3. Three types of the black-box membership inference attack based on different amounts of output knowledge.

Output knowledge	Description	Difficulty
Full confidence scores	The adversary obtains all confidence scores of the prediction vector $\hat{p}(y   \mathbf{x})$ of the input $\mathbf{x}$ . Based on these confidence scores, the adversary knows the predicted label of input $\mathbf{x}$ . The adversary can also calculate loss (e.g., cross-entropy loss) between the prediction vector $\hat{p}(y   \mathbf{x})$ and the true label $y$ of the input $\mathbf{x}$ .	Easy
Partial confidence scores	The adversary obtains partial confidence scores of the prediction vector $\hat{p}(y   \mathbf{x})$ , e.g., the three largest confidence scores. Based on the partial confidence scores, the adversary can know the predicted label of the input $\mathbf{x}$ , but he can not calculate the loss with its true label.	Moderate
Label-only	The adversary only obtains the predicted label of the input $\mathbf{x}$ .	Difficult

$\hat{p}(y | \mathbf{x})$ , but also knows the intermediate computations of  $\mathbf{x}$  on the target model. We assume  $\theta_1, \theta_2, \dots, \theta_i$  are internal parameters of each layer of the target model  $f(\mathbf{x}; \theta^*)$ . The adversary can calculate the output of each layer  $h(\mathbf{x}; \theta_i)$  on the input  $\mathbf{x}$ . As white-box inference attacks have all adversarial knowledge, we do not specify their types as we do in the black-box attack setting.

Generally, white-box MIA perform better than black-box ones, as the former know more information than the latter. Nasr et al. [72] are the first to extend the black-box MIA to the white-box setting. They propose to use the gradient  $\frac{\partial \mathcal{L}}{\partial \theta}$  of prediction loss with regard to target model parameters as additional features to distinguish training samples and the samples that the target model had never seen. They show that their proposed white-box MIA obtains higher attack accuracy than the black-box attacks. However, in Nasr et al.’s [72] proposed white-box attacks, the adversary is assumed to know partial training data of the private training set, which deviates from the most common adversarial knowledge setting. Leino and Fredrikson [55] relax Nasr et al.’s [72] requirement for the adversary and propose an effective white-box MIA that operates without access to any of the target model’s training data.

## 4 MEMBERSHIP INFERENCE ATTACK METHODS

ML models such as deep neural networks are often overparameterized meaning that they have sufficient capacity to memorize information about their training dataset [92, 122]. Moreover, the training datasets are finite in size, and ML models are trained multiple (often tens to hundreds) epochs on the same instances repeatedly. This makes ML models often behave differently on their training data (i.e., members) versus the data that they “see” for the first time (i.e., non-members). Overfitting is a common reason but not the only one (we will discuss more in Section 6). For example, a classifier might classify a data instance of its training set to a class with high confidence (high probability in the entry of corresponding prediction vector) while classifying a new data instance (does not belong to the training set) to a class with relatively smaller confidence. This different behavior enables MIA to exploit such differences to train attack models to distinguish members from non-members. From a taxonomy point of view, there are two types of attack methods. One is **neural network based attacks** and the other is **metric based attacks**.

### 4.1 Neural network based attacks

A neural network (NN) based attack model essentially is an NN binary classifier which maps the target model’s behavior on a data point to its membership in the training dataset. The challenge is how to train such a binary classifier. Based on the adversarial knowledge, an effective technique called **shadow training** proposed by Shokri et al. [90] is often used for training attack models. The main idea is to create multiple *shadow models* to mimic the behavior of the target model, as the adversary is assumed to know the structure and learning algorithm of the target model. Then, for

these shadow models, the adversary knows their training datasets and thus can collect features and the ground truth about membership of data instances in these datasets. Based on the labeled (member vs non-member) dataset collected from the shadow models, the adversary can train the attack models.

We use Fig. 2 to show how to train a NN based attack model for attacking classification models using the shadow training technique.  $D_{train}$  is a private training set that is used for training the target classification model through the learning algorithm  $\mathcal{A}$ .  $D'_1, \dots, D'_k$  are datasets which are disjoint with  $D_{train}$  and they are called *shadow training sets*. Each shadow training set contains data instances from the same distribution as those in  $D_{train}$ , as the adversary is assumed to know the distribution of  $D_{train}$ . At first, the adversary trains  $k$  shadow models using these shadow training sets and the learning algorithm  $\mathcal{A}$ . Each shadow model is trained in such a way to mimic the behavior of the target model.  $T_1, \dots, T_k$  are called *shadow testing sets* which are disjoint from  $D'_1, \dots, D'_k$ . After obtaining the trained shadow models, the adversary queries each of them using its own shadow training set and shadow test set and obtains the output. For each shadow model, these output vectors of instances in the shadow training set and shadow test set are labeled "member" and "non-member" respectively. Thus, the adversary can collect  $k$  "member" prediction sets  $P_1^m, \dots, P_k^m$  and  $k$  "non-member" prediction sets  $P_1^{nm}, \dots, P_k^{nm}$  and they constitute the training sets for the attack model. Finally, based on the collected labeled (member vs non-member) training sets, the adversary can train the NN based attack model, which is a binary classifier.

The shadow training technique can be used for training attack models in both black-box and white-box settings. In both settings, the training procedure of the attack model is the same as we demonstrate in Fig. 2. However, as the input of the attack model in the two settings is different, there are differences between the collected data in the attack model's training sets. In the black-box setting, the adversary only has black-box access to the target model, which means this adversary can only receive target models' prediction vector of arbitrary input instances. Thus, when querying the shadow models using their own shadow training sets and test sets, the adversary only collects the prediction vectors of each data instance. In the white-box setting, the adversary has white-box access to the target model, which means the adversary can observe the intermediate computations at hidden layers of the target model as well as the input instance's prediction vector. Thus, when the adversary queries the shadow model, he collects prediction vectors in addition to the intermediate computations of each data instance.

**NN based attack in the black-box setting:**  $P_1^m, \dots, P_k^m$  are "member" prediction sets which contain prediction vectors of data instances in the shadow training sets.  $P_1^{nm}, \dots, P_k^{nm}$  are "non-member" prediction sets which contain prediction vectors of data instances in the shadow test sets. We denote "member" as 1 and "non-member" as 0. Each "member" prediction set and "non-member" prediction set is represented as follows.

$$P_i^m = \left\{ \hat{p} \left( y \mid \mathbf{x}^{(t)} \right), 1 \right\}_{t=1}^{N_i^m} \quad (6)$$

$$P_i^{nm} = \left\{ \hat{p} \left( y \mid \mathbf{x}^{(t)} \right), 0 \right\}_{t=1}^{N_i^{nm}} \quad (7)$$

For an NN binary classifier  $g(\mathbf{z}; \theta)$ , SGD tries to find the parameters  $\theta^*$  that minimize the following objective function:

$$\mathcal{R}(\theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(I(\mathbf{x} \in P^m), g(\hat{p}(y \mid \mathbf{x}); \theta)) \quad (8)$$



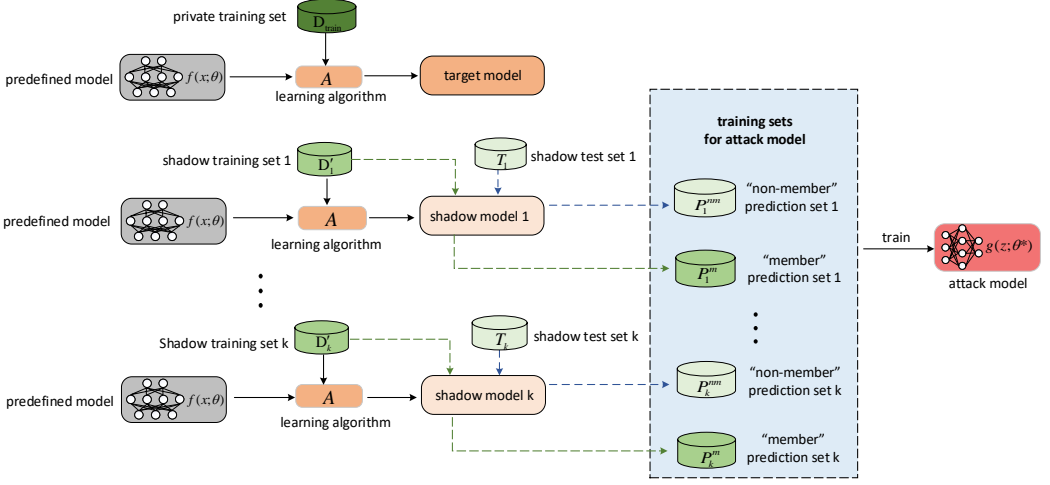


Fig. 2. Training neural network based attack model using the shadow training technique. At first, the adversary trains multiple shadow models using the same learning algorithm as that of the target model and data instances from the same distributions as those in the private training set. These shadow models are trained in such a way to mimic the behavior of the target model. Then, for each trained shadow model, the adversary queries it using its own training set and test set to obtain the output. These output vectors of instances in the training set and test set are labeled “member” and “non-member” respectively. Then, the labeled “member” and “non-member” prediction vectors constitute two sets and they are added to the training sets for the attack model. Based on the collected training sets, the adversary can then train the attack model.

where  $\mathcal{L}(\cdot)$  is the binary cross entropy loss function and  $N$  is the total number of data instances in all shadow training and test sets.  $I(\cdot)$  is the indicator function.

After training, we can use the binary classifier to conduct membership inference on data instances. Fig. 3a demonstrates a membership inference process using a trained NN based attack model in the black-box setting. The binary classifier  $g(z; \theta^*)$  is a trained attack model. It takes  $\hat{p}(y | x)$  as input and outputs whether  $x \in D_{train}$  or not.

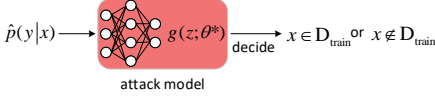
**NN based attack in the white-box setting:** The adversary queries the shadow model on each input instance  $x$  and computes the prediction vector  $\hat{p}(y | x)$ , the intermediate computation  $h(x; \theta_i)$  at each hidden layer, the loss  $\mathcal{L}(y, \hat{p}(y | x))$  in a forward pass, and the gradient of the loss with respect to the parameters of each layer  $\frac{\partial \mathcal{L}}{\partial \theta_i}$  in a backward pass.  $P_1^m, \dots, P_k^m$  are “member” prediction sets which contain the above computations of each data instance in the shadow training sets.  $P_1^{nm}, \dots, P_k^{nm}$  are “non-member” prediction sets which contain the above computations of each data instance in the shadow test sets. We concatenate all the computations of each data instance into a flat vector and the prediction sets are as follows.

$$\mathbf{v} = (\frac{\partial \mathcal{L}}{\partial \theta_1}, h(x; \theta_1), \dots, \frac{\partial \mathcal{L}}{\partial \theta_i}, h(x; \theta_i), \hat{p}(y | x), \mathcal{L}(y; \hat{p}(y | x))) \quad (9)$$

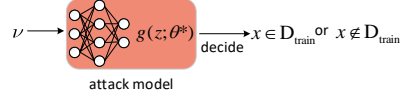
$$P_i^m = \{\mathbf{v}, 1\}_{t=1}^{N_i^m} \quad (10)$$

$$P_i^{nm} = \{\mathbf{v}, 0\}_{t=1}^{N_i^{nm}} \quad (11)$$

The structure of an NN attack model in the white-box setting is usually different from that in the black-box setting, as the input of the attack model in the two settings is very different. Nevertheless,



(a) A neural network based attack model in the black-box setting.



(b) A neural network based attack model in the white-box setting.

Fig. 3. Membership inference process using neural network based attack models in black-box and white-box settings. Both attack models are binary classifiers. In the black-box setting, the attack model takes only the prediction vector  $\hat{p}(y|x)$  as input and outputs its membership status. In the white-box setting, the attack model takes the flat vector  $\nu$  which contains computations related to the data instance as input and outputs its membership status.

the attack model is a NN based binary classifier, for an attack model  $g(\mathbf{z}; \theta)$ , we can use SGD to find the parameters  $\theta^*$  that minimize the following objective function:

$$\mathcal{R}(\theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(I(\mathbf{x} \in P^m), g(\mathbf{v}; \theta)) \quad (12)$$

Fig. 3b demonstrates a membership inference process using a trained neural network based attack model in the white-box setting. The binary classifier  $g(\mathbf{z}; \theta^*)$  is a trained attack model. It takes  $\nu$  as input and outputs whether  $\mathbf{x} \in D_{train}$  or not.

## 4.2 Metric based attacks

Unlike NN based attacks that rely on training binary classifiers, metric based attacks make membership inference decisions by calculating metrics on the prediction vectors and comparing them with a preset threshold. Based on different metric options, there are mainly 4 types of metric based membership inference attacks. They are *prediction correctness*, *prediction loss*, *prediction confidence*, and *prediction entropy* based attacks. Currently, all metric based attacks focus on the black-box setting. Thus, we introduce them in this setting. We denote the metric based attack as  $\mathcal{M}(\cdot)$ , which codes members as 1, and non-members as 0.

**4.2.1 Prediction correctness based attack.** The adversary infers an input data instance  $\mathbf{x}$  as being a member if it is correctly predicted by the target model, otherwise the adversary infers it as a non-member. The intuition is that the target model is trained to predict correctly on its training data, which may not generalize well on the test data. The attack  $\mathcal{M}_{\text{corr}}(\cdot)$  is defined as follows.

$$\mathcal{M}_{\text{corr}}(\hat{p}(y|x); y) = I(\arg\max \hat{p}(y|x) = y) \quad (13)$$

where  $I(\cdot)$  is the indicator function.

Yeom et al. [120] first propose the prediction correctness based attack and demonstrate its performance is comparable with NN based attacks in the black-box setting. It is often considered as a simple baseline attack. Leino and Fredrikson [55], Song and Mittal [94], and Choquette-Choo et al. [16] use it as a baseline to compare the performance of their proposed attacks.

**4.2.2 Prediction loss based attack.** The adversary determines  $\mathbf{x}$  as being a member if its prediction loss is smaller than the average loss of all training samples, otherwise the adversary infers it as a non-member. The intuition is that the target model is trained on its training samples by minimizing

their prediction loss. Thus, the prediction loss of a training sample should be smaller than the loss of an input that was not used in training. The attack  $\mathcal{M}_{\text{loss}}(\cdot)$  is defined as follows.

$$\mathcal{M}_{\text{loss}}(\hat{p}(y|\mathbf{x}); y) = I(\mathcal{L}(\hat{p}(y|\mathbf{x}); y) \leq \tau) \quad (14)$$

where  $\mathcal{L}(\cdot)$  is the cross-entropy loss function.

Yeom et al. [120] first propose the prediction loss based attack. They show that by requiring less computational resources and background knowledge, this attack achieves comparable performance with Shokri et al.'s NN based attack [90].

**4.2.3 Prediction confidence based attack.** The adversary determines  $\mathbf{x}$  as being a member if its maximum prediction confidence is larger than a preset threshold, otherwise the adversary infers it as a non-member. The intuition is that the target model is trained by minimizing prediction loss over its training data, which means the maximum confidence score of a training sample in the prediction vector  $\hat{p}(y|\mathbf{x})$  should be close to 1. The attack  $\mathcal{M}_{\text{conf}}(\cdot)$  is defined as follows.

$$\mathcal{M}_{\text{conf}}(\hat{p}(y|\mathbf{x})) = I(\max \hat{p}(y|\mathbf{x}) \geq \tau) \quad (15)$$

Salem et al. [85] first propose the prediction confidence based attack. The threshold value  $\tau$  is learned through querying the target model with non-members. They experimentally demonstrate that using the maximal confidence achieves very high attack performance. In their work, they choose to use a single threshold for all class labels. Song and Mittal [94] improve this attack by setting different threshold values for different class labels.

**4.2.4 Prediction entropy based attack.** The adversary classifies  $\mathbf{x}$  as being a member if its prediction entropy is smaller than a preset threshold, otherwise the adversary infers it as a non-member. The intuition is that the prediction entropy distributions between training and test data are very different. The target model usually has a larger prediction entropy on its test data than the training data. The entropy of the prediction vector  $\hat{p}(y|\mathbf{x})$  is defined as follows.

$$H(\hat{p}(y|\mathbf{x})) = - \sum_i p_i \log(p_i) \quad (16)$$

where  $p_i$  is the confidence score in  $\hat{p}(y|\mathbf{x})$ . The attack  $\mathcal{M}_{\text{entr}}(\cdot)$  is then defined as follows.

$$\mathcal{M}_{\text{entr}}(\hat{p}(y|\mathbf{x})) = I(H(\hat{p}(y|\mathbf{x})) \leq \tau) \quad (17)$$

Shokri et al. [90] first introduce MIA in machine learning settings. In their original paper [90], Shokri et al. show the difference of prediction entropy distributions between training and test data to explain why membership privacy risks exist. Salem et al. [85] demonstrate the effectiveness of using prediction entropy for attacks. Instead of using a threshold for all classes, Song and Mittal [94] propose to set different threshold values that are dependent on the class labels. They experimentally show that this strategy can achieve higher attack accuracy.

Song and Mittal [94] also propose a modified prediction entropy based attack. They argue that the prediction entropy does not contain any information about the ground truth label, which might misclassify members and non-members. For example, a totally wrong classification with probability 1 leads to zero prediction entropy value and the prediction entropy based attack will classify the data instance as a member. However, the data instance with a totally wrong classification is highly likely a test data instance (i.e., non-member). Thus, they propose a modified prediction entropy metric which is defined as follows.

$$MH(\hat{p}(y|\mathbf{x}); y) = -(1 - p_y) \log(p_y) - \sum_{i \neq y} p_i \log(1 - p_i) \quad (18)$$

where  $p_y$  is the confidence score of the correct label. The attack  $\mathcal{M}_{\text{Mentr}}(\cdot)$  is then defined as follows.

$$\mathcal{M}_{\text{Mentr}}(\hat{p}(y|\mathbf{x}); y) = I(MH(\hat{p}(y|\mathbf{x}); y) \leq \tau) \quad (19)$$

Song and Mittal [94] also experimentally show that the modified prediction entropy based attack is strictly superior to the prediction entropy based attack.

## 5 DESIGN OF MEMBERSHIP INFERENCE ATTACKS

In previous sections, we have introduced the NN based and the metric based membership inference attacks in both black-box and white-box settings. In this section, we present in detail how these attacks are proposed against specific machine learning models. Based on whether the target model is trained centralized or distributed, we separate these attacks into two parts. One is attacks against centralized learning models and the other is attacks against distributed learning models. A comprehensive summary of all attacks is at the end of the section.

### 5.1 Attacks against centralized learning

In centralized learning, there is only one private dataset  $D_{\text{train}}$  which contains all training data instances. The target model is trained on  $D_{\text{train}}$ , and once the training process is finished, the model's parameters are fixed. Based on the learning type of supervised learning and unsupervised learning, in this subsection we divide the membership inference attacks into *attacks against centralized supervised learning* and *attacks against centralized unsupervised learning*.

**5.1.1 Attacks against centralized supervised learning.** Currently, MIA against centralized supervised learning focus on classification models. The adversary aims to identify whether a data instance was used for training a classifier. Shokri et al. [90] conducted pioneering work that proposes the first MIA against ML models. They propose a shadow training technique to train the NN based attack models in the black-box setting. They evaluate their inference attacks on four types of classification target models: two constructed by cloud-based platforms (Google Prediction API and Amazon ML) and one standard CNN classifier and one fully connected NN classifier. They evaluate their attacks on 7 datasets and demonstrate these target models are vulnerable to MIA.

Salem et al. [85] relax two main assumptions in Shokri et al.'s attack [90]. The first assumption in Shokri et al.'s [90] attack is that the adversary needs to establish multiple shadow models to gather training data for the attack model. Salem et al. [85] show that even with one single shadow model the adversary can achieve comparable attack performance. The second assumption in Shokri et al.'s [90] attack is the dataset used to train shadow models must come from the same distribution as the target model's training data. Salem et al. [85] propose a data transferring attack where the dataset used to train the shadow model is not required to come from the same distribution as the target model's private training data. Also, the shadow model is not required to have the same structure as the target model. Besides extending existing NN based attacks in [90], Salem et al. [85] propose a metric based attack that leverages the highest confidence score. They experimentally show the data transferring attack and the metric based attack achieve a strong performance.

Yeom et al. [120] propose two metric based MIA in the black-box setting. One is the prediction correctness based attack and the other is the prediction loss based attack. Compared to NN based attacks proposed by Shokri et al. [90], metric based attacks are much simpler and require far less computational resources. Yeom et al. [120] show that by only using the average training loss of the target model, their attacks can achieve the same recall and only a slightly lower precision than Shokri et al.'s [90] NN based attacks.

Li and Zhang [57] study label-only MIA when the target model only provides the predicted label of an input sample instead of a posterior vector. This is an attack scenario where the adversary

is given minimal information. They propose two label-only MIA, a transfer-based attack and a perturbed-based attack. The transfer-based attack aims to construct a shadow model to mimic the target model. The intuition is that if the shadow model is similar enough to the target model, then the shadow model's confidence scores on an input instance will indicate its membership. The perturbation-based attack aims to add crafted noise to the target instance to turn it into an adversarial example. The intuition is that it is harder to perturb a member instance to a different class than a non-member instance. Thus, the magnitude of the perturbation is used to distinguish members from non-members. The adversary simply considers an instance with perturbation noise larger than a threshold as a member, and vice versa. Choquette-Choo et al. [16] also study label-only MIA and propose two attacks, data augmentation based attack and decision boundary distance based attack. For a target instance, data augmentation based attack creates additional data instances via different data augmentation strategies. These additional data instances are then used to query the target model and the adversary collects all the predicted labels. The attack intuition is that many models use data augmentation techniques during the training process. Thus, a member instance's augmented versions is less likely to change their predicted label. This enables the adversary to distinguish members from non-members through the behavior of augmented data instances. The decision boundary based attack estimates an instance's distance to the model's boundary and decides it as a member if its distance is larger than a threshold. The intuition is similar to Li and Zhang's [57] perturbed-based attack. The success of label-only attacks demonstrate that ML models might be more vulnerable to MIA than we expect.

Long et al. [63, 64] investigate MIA in the scenario where the target model is not overfitted, which deviates from the aforementioned MIA. They propose a new generalized black-box MIA, which can identify the membership of particular vulnerable instances with high precision on well-generalized models. The intuition is that some instances have unique influences on the output of the target model, even when the model is well-generalized. The attack exploits these unique influences of some particular data instances as the indicator of the presence in the training dataset. Their attack first estimates whether a given instance is a vulnerable instance by estimating the number of neighbors in a reference dataset it has. If the number of neighbors is smaller than a threshold, the given instance is more likely to impose unique influence and is then considered as a vulnerable instance. Then, a hypothesis test will decide the membership of the selected instance. They experimentally show these vulnerable instances can be inferred correctly with high precision on well-generalized models which have training and testing accuracy gaps smaller than 1%.

While the above MIA focus on the black-box setting, Nasr et al. [72] first extend MIA in the white-box setting, which means the adversary has additional access to the internal parameters of the target model. Their white-box inference attacks are an extension of Shokri et al.'s [90] NN based black-box attacks, which try to improve the attack performance by leveraging the intermediate computations of an input sample through the target model. However, Nasr et al. [72] find that by simply combining the target model's final predictions and its intermediate computations as features, the white-box attack can not achieve higher attack accuracy than that of the black-box one. Instead, they use the gradient of prediction loss with regard to the target model's parameters as additional features. The intuition is that the gradients of the loss function over the model's parameters of each training data instance is distinguishable through the training of the SGD algorithm. This enables the white-box inference attacks to exploit the gradient differences between members and non-members and achieve better performance than black-box attacks.

Leino and Fredrikson [55] point out that the white-box adversarial knowledge assumption by Nasr et al. [72] is too strong, which deviates from most MIA settings. Nasr et al. [72] assume that the adversary knows a significant portion of the target model's private training set  $D_{train}$ , while the adversary is often assumed to only have a dataset  $D'$  which comes from the same distribution but is

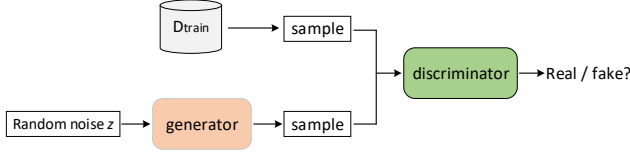


Fig. 4. Architecture of a Generative Adversarial Network (GAN).

disjoint with  $D_{train}$ . Thus, Leino and Fredrikson propose an effective white-box MIA that does not require access to any of the target model’s training data. The attack intuition is that membership information can be leaked through a target model’s idiosyncratic use of features. Features that are distributed differently in the training data than in the true distribution can provide evidence for membership. They first build a Bayes-optimal attack assuming the target model is a simple linear softmax model. When the target model is deep NN models, they approximate each layer as a local linear model, which is then applied to the Bayes-optimal attack. The attacks on different layers are then combined to compute the final membership decision.

**5.1.2 Attacks against centralized unsupervised learning.** Currently, MIA against centralized unsupervised learning focus on GANs, which is the most popular generative models. To better understand such attacks on GANs, we first describe the architecture of a GAN and then introduce how to design the MIA.

Fig. 4 depicts the architecture of a GAN. It consists of two competing NN modules, a generator  $\mathcal{G}$  and a discriminator  $\mathcal{D}$ , which are trained to compete against each other. The generator takes random noise variable  $z$  and generates samples  $\mathcal{G}_{\theta_{\mathcal{G}}}(z)$  that approximate the data distribution of  $D_{train}$ . The discriminator  $\mathcal{D}_{\theta_{\mathcal{D}}}(\mathbf{x})$  receives samples from  $D_{train}$  and generated samples  $\mathcal{G}_{\theta_{\mathcal{G}}}(z)$  and is trained to learn the difference between them.  $\mathcal{D}_{\theta_{\mathcal{D}}}(\mathbf{x})$  essentially is a binary classifier which outputs the probability that  $\mathbf{x}$  was taken from  $D_{train}$  rather than from  $\mathcal{G}$ . After training,  $\mathcal{G}$  receives different  $z$  and generates synthetic samples.

MIA on generative models aim to identify whether a data instance was used for training the generator  $\mathcal{G}$ . They are more challenging than the MIA on classification models. Unlike classification models, the adversary does not obtain confidence scores or prediction labels that is related with the target instance from the victim generative models. This means the adversary has little clues for conducting membership inference. Moreover, current GANs models often encounter model dropping and mode collapse, leading to the problem of underrepresenting certain data samples. This poses additional attack difficulty to the adversary.

Based on whether the adversary has access to the generative models’ internals or not, MIA on generative models can be categorized into black-box and white-box attacks. In the black-box setting, the adversary only makes queries to the generator and receives generated synthetic samples. In the white-box setting, the adversary has full access to the generative model, including knowledge of the generator’s internals and the discriminator. This is the most knowledgeable setting for the adversary.

Hayes et al. [29] introduce the first MIA against generative models in both black-box and white-box settings. The attack intuition is that the discriminator of GANs is more confident to output a higher confidence value on training samples, as it is trained to learn statistical differences in distribution. Assume there are  $M$  member samples and  $M$  non-member samples that the attacker wants to distinguish. In the white-box setting, the adversary just puts all data samples into the discriminator and it will output  $2M$  probability values with each of them corresponding to the probability of being a member. The adversary sorts these probability values in descending order

and picks the samples with largest  $M$  values as members. In the black-box setting, the attacker collects samples from the generator and trains a local GAN to learn information about the target GAN from the collected samples. After the local GAN has been trained, the adversary proceeds the membership inference attack using the discriminator of the local GAN as in the white-box setting. Hayes et al. [29] evaluate their attacks on DCGAN [77], BEGAN [6], and VAEGAN [53]. They experimentally show that the white-box MIA achieve perfect accuracy when DCGAN and VAEGAN are the victim models and the performance of the black-box MIA can be improved by a small amount of auxiliary adversarial knowledge.

Hilprecht et al. [32] propose two MIA, one is Monte Carlo (MC) attack designed for GANs in the black-box setting and the other is reconstruction attack designed for VAEs in the white-box setting. The MC attack exploits the generated synthetic samples that are within a small distance of the target sample to approximate the probability that the target sample belongs to the training set via Monte Carlo integration [82]. The intuition behind the MC attack is that the generator of GANs should be able to produce synthetic samples that are close to the training samples if GANs overfit. The reconstruction attack directly make use of the loss function of VAEs to calculate the reconstruction error of the target sample, as the adversary has access to the internals of VAEs. The attack intuition is that training data should have smaller reconstruction errors compared to that of non-member data. In addition to membership inference attacks for a single sample, Hilpreche et al. [32] introduce the concept of set membership inference that the adversary tries to identify whether a set of individuals belong to the training set. They evaluate their attacks on DCGAN [77] and VAEs [49]. They experimentally show that their attacks work especially well on set membership inference and the MIA's outperform Hayes et al.'s [29] proposed MIA. They also report that VAEs are more vulnerable to MIA than GANs, which suggests VAEs are more prone to overfitting than GANs.

Liu et al. [60] propose co-membership inference, which essentially is the same with the concept of set membership inference proposed by Hilpreche et al. [32]. They propose a new attack method for generative models, which begins with attacking a single target sample and then extends to the co-membership inference. For a given sample  $\mathbf{x}$  and the generator  $\mathcal{G}$  of the victim model, the adversary optimizes a neural network to reproduce the latent variable  $\mathbf{z}$  (the input) of  $\mathcal{G}$  such that  $\mathcal{G}$  generates synthetic output  $\mathcal{G}(\mathbf{z})$  that nearly matches  $\mathbf{x}$ . The attack intuition is that if  $\mathbf{x}$  belongs to the training set, the adversary is able to reproduce similar synthetic samples close to it. The adversary then measure the L2-distance between the synthetic sample and  $\mathbf{x}$  and decides  $\mathbf{x}$  is a member if the distance is smaller than a threshold. The attack method extends to co-membership inference by training the neural network to reproduces each of the  $n$  targets in a co-attack. This attack method is different from Hilpreche et al.'s [32]. It requires retraining new neural networks for different input data, while the MC attack and reconstruction attack only need fixed synthetic samples of the generator. Similar to Hilpreche et al.'s [32] observation, Liu et al. [60] report that VAEs are more susceptible to MIA compared to GANs in general.

Chen et al. [12] present the first taxonomy of MIA against generative models. They propose a generic attack model which is applicable to all adversarial knowledge settings, from full black-box to full white-box settings. For a target sample  $\mathbf{x}$ , the adversary tries to reconstruct a synthetic sample that is closest to  $\mathbf{x}$ . The adversary simply finds the synthetic sample from the collected samples from the generator in the full black-box setting, otherwise it makes use of optimization algorithms to reconstruct it. The distance between the reconstructed sample and  $\mathbf{x}$  is then used for calculating the probability that  $\mathbf{x}$  belongs to the training set. The attack intuition is that the generator should be able to generate more similar samples for members than that of non-members and thus the distance between the reconstructed sample and  $\mathbf{x}$  represents the probability that  $\mathbf{x}$  belongs to the training set. To make a more accurate probability estimation, they also propose to

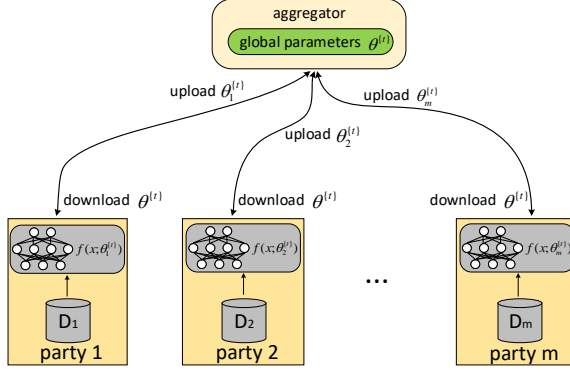


Fig. 5. A typical federated learning training process.

train a reference GAN with a relevant but disjoint dataset to calibrate the reconstruction error (the distance). The attacker decides  $x$  is a member when the calibrated reconstruction error of it is smaller than a threshold. They evaluate their attack on DCGAN [77], VAEGAN [53], PGGAN [46], WGANP [27], and MEDGAN [15]. They report that the attacker’s knowledge about the victim model highly reflects the vulnerability of models. Moreover, a smaller training dataset of the victim model leads to a higher risk of revealing membership information of individual samples. Similar to previous observations in [32] and [60], they also report the high vulnerability of VAEs compared to GANs.

## 5.2 Attacks against distributed learning

Currently, MIA against distributed learning focus on supervised classification tasks in the federated learning (FL) setting. FL [50, 65] is a kind of distributed learning which aims to jointly learn a model without sharing data. As it becomes more popular with promises of model accuracy and data privacy, the successful design of MIA on FL sheds light on how it discloses information. To better understand the membership privacy risk, we first introduce what FL is.

Fig. 5 depicts the FL process of the *FederatedAveraging* algorithm [65]. There are  $m$  parties and each of them owns a different training set  $D_i$ . They agree on model structure to train a global deep learning model, without sharing their private data. To achieve this goal, each party first trains a local model  $f(x; \theta_i)$  with its own data and hence obtains a local set of parameters  $\theta_i$ . Then, in each epoch of training, each party sends its model’s parameters  $\theta_i^{(t)}$  to the central aggregator. The central aggregator will calculate the average value for each parameter and thus obtains the global parameters  $\theta^{(t)}$ . Each party then downloads the global parameters  $\theta^{(t)}$  and updates the local model using the SGD algorithm on its local private data. The federated training continues until the global model converges.

MIA against FL occur during the training process, which is significantly different from the MIA against centralized training. In the centralized training setting, the adversary is an “outsider” and attacks occur after the training process is finished and the target models are fixed. The “outsider” attacker is a user and is given black-box or white-box access to the trained target model and then constructs his MIA. In the FL setting, there are three possible positions of the attacker, including two positions of “insider” attackers and one position of “outsider” attacker. The “insider” attacker could be either the central aggregator or one of the participant parties and the “outsider” attacker is a user of the final global model. The MIA constructed by an “outsider” attacker in the FL setting essentially



is the same as that in the centralized setting. This is because if the attacker is an “outsider” user, he can only access the final global model after the federated training is finished, thus the MIA against the fixed global model in this case resembles the MIA on the fixed target model in the centralized learning setting. The studies of MIA against FL focus on the “insider” attackers during the federated training phase. That is, either the central aggregator or one of the participant parties, try to identify whether a data instance is being used for training the global model during the training process. The attacks should happen during the training phase because when the federated training process is finished, the “insider” attackers own the fixed global model. Thus the MIA of the “insider” adversary is essentially the same as the white-box MIA in the centralized learning setting.

The adversary in FL is either the central aggregator or one of the participant parties, and aims to identify whether a data instance is being used for training the global model. Note that MIA in the FL setting only require that the adversary identifies whether a data instance is being used for training the global model, but does not need to identify which set the data instance belongs to. This is because MIA on ML models focus on distinguishing target models’ training data from the data that they have never seen. In FL, each participant party contributes its local training data to train the global model, and thus all the local training samples are members of the global model. The adversary in FL saves the snapshot of the global model at each iteration  $t$  of training and thus obtains multiple versions of the target model over time. If the adversary is the aggregator, it obtains multiple version of  $\{\theta^{\{1\}}, \theta^{\{2\}}, \dots, \theta^{\{t\}}\}$ . If the adversary is one of the participant parties (e.g., party 2), he obtains multiple version of  $\{m\theta^{\{1\}} - \theta_2^{\{1\}}, m\theta^{\{2\}} - \theta_2^{\{2\}}, \dots, m\theta^{\{t\}} - \theta_2^{\{t\}}\}$ . There are two ways to construct the attack model. The first is the adversary runs an independent membership inference attack on each version of the target models and then combines their results. The second is the adversary runs a single attack model on all saved version of the target models. That is, the attack component of the single attack model processes all of its corresponding inputs over the observed target models at once. For example, assume the adversary collects  $T$  versions of the target model, the attack component of the loss  $\mathcal{L}$  is a concatenation of  $(\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_T)$  and it is then processed at once by the attack model. Compared to the first way of constructing an attack model, the second one might capture the dependencies between parameters of the target model over time and thus provide better attack performance [72].

The adversary in federated learning can *actively* or *passively* construct membership inference attacks during the training process. When the adversary is the central aggregator, the active attacker adversarially modifies the aggregate parameters  $\theta^{\{t\}}$  and sends it back to each participant party to construct its MIA, while the passive attacker honestly calculates the global parameters  $\theta^{\{t\}}$ . When the adversary is one of the participant parties, the active attacker adversarially modifies his local parameters  $\theta_i^{\{t\}}$  and uploads it to the central aggregator to construct his MIA, while the passive attacker honestly uploads his parameters  $\theta_i^{\{t\}}$  to the central aggregator. We introduce the current research of MIA against FL from two directions, depending on the adversary actively or passively constructing his attacks.

**5.2.1 Active attacks.** Nasr et al. [72] investigate how an adversary can actively extract the membership information about a data instance in the FL setting. The active attacker exploits the SGD algorithm to construct his active attack. For a target data instance  $\mathbf{x}$ , the attacker (assume he is one of the participant parties) runs a gradient ascent on  $\mathbf{x}$  to update the local parameters in the direction of increasing the loss on  $\mathbf{x}$  as follows.

$$\theta_i = \theta_i + \gamma \frac{\partial \mathcal{L}}{\partial \theta_i} \quad (20)$$

where  $\gamma$  is the update rate. The adversary then uploads  $\theta_i$  to the central aggregator. The active attack intuition is that if the target instance  $\mathbf{x}$  is a member of the other parties data, applying the gradient ascent on it by the adversary will trigger the target model to try to minimize its loss by descending in the direction of the target model's gradient by the other parties. Therefore, the gradient ascent by the attacker will be nullified. However, if  $\mathbf{x}$  is a non-member of the other parties data, the model will not explicitly change its gradient because it was not involved in the training process. The active gradient ascent increases the target model's different behavior between the members and non-members and thus makes them more distinguishable. As a result, Nasr et al. [72] report a higher attack accuracy of the active adversary than that of the passive adversary.

**5.2.2 Passive attacks.** Melis et al. [66] first propose MIA in the FL setting. They focus on deep learning models that operate on non-numeric data where the input space is discrete and sparse (e.g., natural-language text). These models first use an embedding layer to transform inputs into a lower-dimensional vector representation via an embedding matrix and the embedding matrix is treated as a parameter of the global model and optimized collaboratively. The attacker is assumed to be one of the participants that exploits the behavior of the update gradient of the embedding layer. During training, the embedding layer's gradient is sparse with respect to the input words: given a batch of text, the embedding is updated only with the words that appear in the batch. The gradients of the other words are zeros. The adversary thus exploits this difference that directly reveals which words occur in the training batches used by the honest participants during collaborative learning to decide whether a text record was a member or not.

Truex et al. [105] introduce the threat of MIA when the adversary is a participant in the FL system. Their FL setting is different from the previously mentioned setting. Instead of sharing the parameters to build a global model, each participant trains its local model and only shares the prediction probability when inferring a new instance. For example in the three party case, for an input  $\mathbf{x}$ , party 2 computes a probability vector  $\hat{p}_2(y | \mathbf{x})$  and share it with the other two parties. They assume different parties of the federated system have very different datasets, which leads to sufficiently different decision boundaries for different parties. The decision boundary differences reveal the underlying training data and thus reveals the membership information. They evaluate the insider attacks on the decision tree models and demonstrate that the insider attacker achieves better attack performance than that of the outsider attacker.

### 5.3 Summary of membership inference attacks

Fig. 6 contains a taxonomy of membership inference attacks according to different attack objectives and threat models. Moreover, to help the reader better understand the MIA introduced above, we summarize them from a high-level in Table 4. Each paper proposes an original attack method. The reference papers are listed in ascending order of the published year. We first examine whether the proposed attack is black-box or white-box in terms of the adversarial knowledge to the attacker. We then check whether it is an NN based or a metric based attack. Note that an attack can be both NN based and metric based. For example, Liu et al. [60] proposed MIA first trains a neural network to reconstruct a synthetic sample and then the reconstruction error between it and the target real sample is compared with a threshold to decide the membership of the real sample. The victim model is the target model that the adversary wants to attack. It is either a classifier of classification tasks or GANs/VAEs of generation tasks. The last two columns indicate how the victim model is trained, either centralized or federated.

Generally, MIA has gained increasing attention and developed rapidly during recent years. There are around 19 papers that proposed different attack methods against various ML models. These attacks develop from the black-box setting to the white-box setting, although most of the attacks

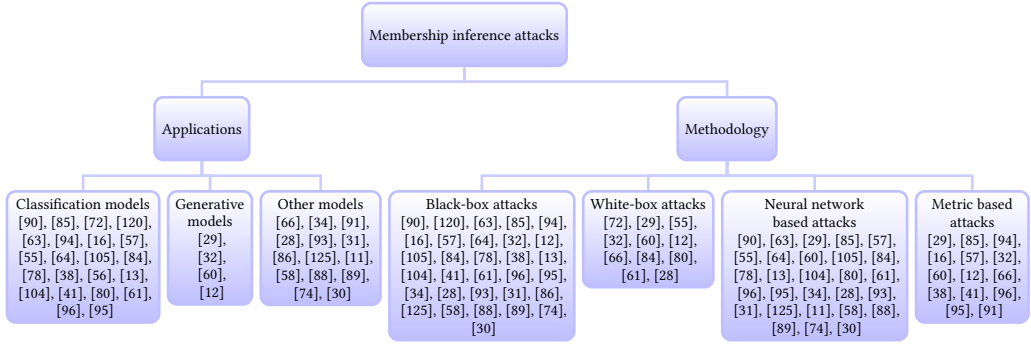


Fig. 6. A taxonomy of membership inference attacks

are black-box ones. In terms of the attack model type, most of them are based on NN to learn a binary classifier which distinguishes the pattern of members from that of non-members. Note that some papers propose more than one type of attack method. For example, Salem et al. [85] not only introduce how to train a NN based attack model, but also suggest to use prediction entropy for attacks. While most of the victim models are classifiers, generative models of GANs and VAEs are attracting more and more attention. While most researchers assume the victim models are trained in a centralized way, Truex et al. [105], Nasr et al. [72], and Melis et al. [66] introduce the privacy risks of membership inference in a federated setting. This indicates the membership inference attacks are general and widely applicable.

## 6 WHY MEMBERSHIP INFERENCE ATTACKS WORK

MIA exploit the different behavior that ML models are more confident on its training data versus the unseen data. Many papers [12, 55, 85, 90, 120] have pointed out that overfitting is a common reason that enables the MIA on ML models. A ML model is said to overfit to its training data when it performs much better on its training data than the unseen data, i.e., the generalization gap is large. Yeom et al. [120] theoretically analyze the connection of overfitting to membership inference risks and suggest a simple attack that will always achieve attack accuracy better than random guessing (i.e., 50% attack accuracy) as long as the generalization gap of the classification model exists. That is, the attacker predicts an instance as a member if it is predicted correctly, otherwise versus. Shokri et al. [90] empirically show the member and non-member output distributions of the overfitted classification model are very different. Salem et al. [85] emphasize that a classification model is more vulnerable to membership inference attack if it is more overfitted. Leino et al. [55] show new insights about how overfitting leads to membership evidence. They argue that the training data memorized by the overfitted classification model manifested in not only the model's output behavior but also the internal layers of the model. Thus, they leverage the idiosyncratic features learned in the internal layers of the target model and propose a white-box MIA that improves upon existing black-box MIA.

Different ML model types display different vulnerability to MIA. Shokri et al. [90] show that even though the machine learning as a service platform of the Google model is less overfitted than the Amazon model and has a better predictive power on the same dataset, the Google platform leaks more membership privacy than the latter. Truex et al. [105] experimentally evaluate membership inference on NN model, logistic regression model, Naive Bayes model, k-nearest neighbor model, and decision tree model on seven datasets and the results shows that the decision tree model has the highest attack precision for six datasets and Naive Bayes models consistently show the lowest

Table 4. A summary of membership inference attacks against machine learning models

Reference	Year	Adversarial Knowledge		Attack Model		Victim Model		Training Approach	
		Black-box	White-box	NN	Metric	Classifier	GANs/VAEs	Centralized	Federated
Shokri et al. [90]	2017	✓	×	✓	×	✓	×	✓	×
Yeom et al. [120]	2018	✓	×	×	✓	✓	×	✓	×
Long et al. [63]	2018	✓	×	✓	×	✓	×	✓	×
Salem et al. [85]	2019	✓	×	✓	✓	✓	×	✓	×
Truex et al. [105]	2019	✓	×	✓	×	✓	×	✓	✓
Sablayrolles et al. [84]	2019	✓	✓	✓	×	✓	×	✓	×
Nasr et al. [72]	2019	✓	✓	✓	×	✓	×	✓	✓
Hayes et al. [29]	2019	✓	✓	✓	×	×	✓	✓	×
Hilprecht et al. [32]	2019	✓	✓	×	✓	×	✓	✓	×
Liu et al. [60]	2019	×	✓	✓	✓	×	✓	✓	×
Melis et al. [66]	2019	×	✓	✓	×	✓	×	×	✓
Li & Zhang [57]	2020	✓	×	✓	×	✓	×	✓	×
Choo et al. [16]	2020	✓	×	✓	×	✓	×	✓	×
Rahimina et al. [78]	2020	✓	×	✓	×	✓	×	✓	×
Hishamoto et al. [34]	2020	✓	×	✓	×	✓	×	✓	×
Leino & Fredrikson [55]	2020	×	✓	✓	×	✓	×	✓	×
Song & Mittal [94]	2020	✓	×	×	✓	✓	×	✓	×
Chen et al. [12]	2020	✓	✓	×	✓	×	✓	✓	×
Hui et al. [38]	2021	✓	×	×	✓	✓	×	✓	×

precision across all datasets. Hayes et al. [29] suggest that VAEs are more prone to overfitting than GANs trained on the same training data, and this leads to VAEs being more vulnerable to MIA. Chen et al. [12] comprehensively conduct MIA on five state-of-the-art GANs and observe that the vulnerability of different generative models varies. For example, WGANGP [27] is consistently more vulnerable than MEDGAN [15] on the MIMIC-III dataset [43].

ML models trained on different datasets display different vulnerability to MIA. Generally, a model working on a simpler dataset is less vulnerable to membership inference attacks. For example, Salem et al. [85] experimentally show that a CNN model trained on the MNIST dataset [119] has smaller attack precision and recall compared to the model trained on the CIFAR10 dataset [51]. This is because the colored images in CIFAR10 are more sophisticated than the gray digit images in MNIST, which makes the model more difficult to generalize well. The number of classes of the dataset is another factor that influences the model’s vulnerability to membership inference attacks. Generally, a model working on a dataset with more classes are more vulnerable to membership inference attacks. Shokri et al. [90] show that the attack precision of a fully connected NN model trained on the Purchase dataset [44] gets higher and higher when the number of classes increases. Salem et al. [85] show that a CNN model trained on the CIFAR100 dataset has higher attack precision and recall than that of the same model trained on CIFAR10. Lastly, a model trained with more data is less vulnerable to membership inference attacks. This is because more data can represent the feature distribution better and thus leads the learned model to generalize well. Shokri et al. [90] show that a CNN model trained on the CIFAR10 dataset has smaller and smaller attack precision when providing more training samples. Hayes et al. [29], Hilprecht et al. [32], and Chen et al. [12] report that more training data for generative models would reduce the effectiveness of MIA.

## 7 DEFENSES AGAINST MEMBERSHIP INFERENCE ATTACKS

The current defenses against MIA fall into four categories, i.e., *confidence score masking*, *regularization*, *knowledge distillation*, and *differential privacy*. In this section, we comprehensively review the related papers of each defense category and summarize them.

### 7.1 Confidence score masking

Confidence score masking aims to hide the true confidence score returned by the target classification model and thus mitigates the effectiveness of MIA. The defense methods belonging to this category

include restricting the prediction vector to top  $k$  classes [90] or adding crafted noise to the prediction vector [42]. These methods do not need to retrain the target classifiers and are only implemented on the prediction vectors, thus they will not influence the target model's accuracy. They are proposed to mitigate the black-box MIA since the black-box attacks leverage the differences between members' and non-members' prediction vectors. Shokri et al. [90] evaluate the mitigation strategy of restricting the prediction vector to top-3 classes or only returning the prediction label on a fully connected NN classifier on Purchase-100 and Texas-100 datasets. They find that restricting the prediction vector to the top-3 classes does not reduce the attack accuracy of their proposed black-box attack. This finding is not surprising because the latter paper [85] relaxes the assumptions in Shokri et al.'s [90] attack and demonstrates that the black-box attack leveraging partial prediction vectors can achieve similar attack performance compared to using the complete prediction vectors. Shokri et al. [90] indeed show that only returning the classifier's predicted label will reduce the attack accuracy. However, as long as the generalization gap exists, a simple *prediction correctness based attack* will always achieve better attack accuracy than random guessing. Moreover, the label-only attacks proposed by Li and Zhang [57] and Choo et al. [16] further investigate the privacy risks when the adversary only gets access to the classifiers' predicted label and demonstrates that the adversary can still achieve strong performance. Jia et al. [42] observe that when the attack model is an NN based binary classifier, it is vulnerable to adversarial examples. Thus, they leverage adversarial machine learning technique [52] and propose a defense method called MemGuard that adds a carefully crafted noise vector to a prediction vector and turns it into an adversarial example. MemGuard does not influence the classification models' prediction accuracy while effectively mitigating the black-box NN based attack to a random guess level. However, Song and Mittal [94] re-evaluate the effectiveness of Memguard using metric based attacks and find that the defended models still have high membership inference accuracy.

The confidence score masking mechanism defends MIA when the adversary is in the black-box setting and has the advantage of implementation simplicity. It directly works on the trained models' prediction vector and thus does not need to retrain the model. It is a natural mitigation mechanism against the adversaries who make use of the complete prediction vector of the target classifier. However, as we previously discussed, confidence score masking might not be effective as the label-only attacks work well when the model only provides a predicted label and the metric based attacks achieve high attack accuracy on Memguard.

## 7.2 Regularization

As we previously discussed, overfitting is the main factor that contributes to MIA. Therefore, regularization techniques that can reduce the overfitting of ML models can be leveraged to defend against MIA. Regularization techniques including L2-norm regularization, dropout [97], data argumentation, model stacking, early stopping, label smoothing [99], adversarial regularization [71], Mixup + MMD [56] have been proposed and investigated as defense methods in many papers [38, 48, 56, 71, 85, 87, 90, 94]. Among them, L2-norm regularization, dropout, data argumentation, model stacking, early stopping, label smoothing are classical regularization methods proposed to improve the generalizability of a learned model. They are initially proposed to reduce the overfitting of ML models, but they are shown to be quite effective in mitigating MIA. This is because they help the learned model generalize better to new unseen data and reduce the difference of the model's behavior on its training data and unseen data. The adversarial regularization [71] and Mixup + MMD [56] are specially designed regularization techniques that aim to mitigate MIA. They add new regularization terms to the classifier's objective function during the training phase and force it to generate similar output distributions for members and non-members. Nasr et al. [71] propose the adversarial regularization defense method. They add membership inference gain of the attack

model as a new regularization term to the loss function of the target model during the training process. The target ML model needs to simultaneously minimize its classification loss and the attack model's accuracy. The target model is trained in such a way as to preserve its prediction accuracy while mitigating the attacker's accuracy. Li et al. [56] propose Mixup + MMD defense method. They add the distance between members and non-members output distribution, computed by Maximum Mean Discrepancy (MMD) [22], as a new regularizer to the training loss function of the classifier. The new regularizer term forces the classifier to generate similar output distributions for its members and non-members. As MMD tends to reduce the prediction accuracy of the classifier, they propose to combine MMD with mix-up training [123] to preserve the prediction utility. Note that regularization techniques not only work as defense methods for classification models, with Hayes et al. [29] and Hilprecht et al. [32] also demonstrating that dropout can be leveraged as an effective defense method against MIA on generative models.

Unlike the confidence score masking mechanism, regularization techniques defend against MIA no matter who the adversary is in the black-box or white-box settings. This is because regularization techniques change not only the target models' output distribution but also their internal parameters, while the confidence score masking mechanism only modifies models' prediction vectors. Although regularization techniques are effective and widely applicable, one drawback of them is that they might not be able to provide satisfactory membership privacy-utility tradeoffs. For example, Shokri et al. [90] show that L2-norm regularization can mitigate the accuracy of MIA to random guess level when setting the regularization factor to relatively large values, however, this results in a significant reduction of the target model's prediction accuracy. How to design new regularization methods to improve the tradeoffs between model utility and membership privacy remains an open question.

### 7.3 Knowledge distillation

Knowledge distillation [4, 33] uses the outputs of a large teacher model to train a smaller one, in order to transfer knowledge from the large model to the small one. It allows the smaller student model to have similar accuracy to their teacher models [17]. Based on knowledge distillation, Shejwalkar and Houmansadr [87] propose Distillation For Membership Privacy (DMP) defense method. DMP requires a private training dataset and an unlabeled reference dataset. DMP first trains an unprotected teacher model and uses it to label data instances in the unlabeled reference dataset. Then, DMP selects data instances in the labeled reference dataset that have low prediction entropy to train the target model. The intuition of the selection is such samples are easy to classify and will not be significantly affected by the members of the private training dataset. DMP finally trains a protected model based on the selected labeled samples. The intuition of DMP is to restrict the protected classifier's direct access to the private training dataset, thus significantly reduces the membership information leakage. DMP enables defense against both black-box and white-box inference attacks. Shejwalkar and Houmansadr [87] evaluate the performance of DMP and compare it with regularization techniques and differential privacy, demonstrating that DMP achieves the state-of-the-art tradeoffs between membership privacy and classification model's prediction accuracy.

### 7.4 Differential privacy

Dwork et al. [20] propose a probabilistic privacy mechanism called *differential privacy* (DP) that provides an information-theoretical privacy guarantee. Many paper [12, 14, 16, 29, 38–42, 55, 56, 70, 79, 87, 104, 114, 120, 121] have investigated DP as an effective defense method against MIA on ML models. When an ML model is trained in a differentially private manner, the learned model does not learn or remember any specific user's details. Thus, DP naturally counteracts MIA. Shokri

et al. [90] first point out that differentially private models should limit the success probability of MIA. Yeom et al. [120] theoretically connect DP to MIA and prove that the MIA's advantage of an adversary is limited by a function of the privacy budget  $\epsilon$ . Rahman et al. [79] first empirically evaluate MIA against differentially private deep classification models. They find that differentially private models provide privacy protection against strong adversaries by only offering poor model utility. Jayaraman and Evans [40] further demonstrate that current mechanisms for differentially private machine learning rarely provide acceptable membership privacy-utility tradeoffs. They comprehensively evaluate MIA on different variants of the DP mechanism including differential privacy with advanced composition [19], zero concentrated DP [8], and Rényi DP [68] for ML models. They find that membership privacy leakage is high when setting DP with limited classifiers' accuracy loss, and setting DP providing strong privacy guarantees results in useless models. Truex et al. [104] evaluate how MIA differ across classes and how DP affects models when they are trained on skewness data. They report that the minority groups are more vulnerable to MIA. Moreover, as a mitigation technique, DP tends to decrease more model's utility on the minority groups. DP of ML models is usually achieved by DP-SGD [1] that adds noise to the gradients of the model during training. Rahimian et al. [78] argue that DP-SGD might significantly hinder the model's prediction performance when the adversary is in the black-box setting. They propose DP-Logits that uses a Gaussian mechanism to only add noise to the logits of the input instance at prediction time and restrict the number of queries. They report that the privacy budget for the DP-Logits is generally lower than the DP-SGD method.

DP can also defend against MIA on generative models. Many paper [5, 14, 73, 103, 114–116, 124] have proposed various differentially private generative models to ensure the privacy of their training samples. Hayes et al. [29] first evaluate how MIA perform on a differentially private GANs proposed by Triastcyn and Faltings [103]. They find that their proposed white-box attack achieves high accuracy when  $\epsilon$  is relatively high and does no better than random guessing when  $\epsilon$  is small. However, the acceptable levels of privacy (when  $\epsilon$  is small) leads GANs to generate bad quality samples. Chen et al. [14] report similar findings. They report that DP indeed reduces the effectiveness of MIA even when  $\epsilon$  exceeds practical values (i.e.,  $\epsilon > 10^{10}$ ). However, they also mention that DP deteriorates the generation quality of GANs and applying it into training leads to a much higher computation cost (10× slower as reported). Wu et al. [114] theoretically prove that the generalization gap of the GAN trained with a differentially private learning algorithm can be bounded. This indicates DP limits the GAN overfitting to a certain extent and explains why it helps to mitigate MIA.

DP provides a theoretical guarantee to protect the membership privacy of individual samples. It can be leveraged as a defense mechanism against MIA on both classification models and generative models, no matter whether the adversary is in a black-box or white-box setting. Although it is widely applicable and effective, one drawback is that it rarely offers acceptable utility-privacy tradeoffs with guarantees for complex learning tasks. That is, it provides meaningless membership privacy guarantees at settings with limited model utility loss, and it results in useless models at settings with strong privacy guarantees. As evaluated in [87], Shejwalkar and Houmansadr report that the knowledge distillation based defense DMP provides better utility-privacy tradeoffs than two state-of-the-art DP techniques for training deep learning models, i.e., the DP-SGD [1] and PATE [75]. However, one must be aware that differential privacy defends not only membership inference attacks, but also other forms of privacy attacks such as attribute inference attacks [23, 24] and property inference attacks [3, 25]. More interestingly, DP has been indicated to have a connection to model robustness against adversarial examples [54].

Table 5. A summary of defenses against membership inference attacks on machine learning models

Defense	Applicable Model		Adversarial Knowledge		Reference
	Classifier	GANs & VAEs	Black-box	White-box	
Confidence Score Masking	✓	×	✓	×	[13, 16, 42, 57, 78, 85, 90]
Regularization	✓	✓	✓	✓	[29, 32, 48, 56, 71, 85, 87, 90, 94]
Knowledge Distillation	✓	×	✓	✓	[87]
Differential Privacy	✓	✓	✓	✓	[12, 14, 16, 29, 38, 40–42, 55, 56, 70, 79, 87, 104, 114, 120, 121]

### 7.5 Summary of defenses

We summarize the aforementioned defense mechanisms in Table 5. The applicable model column indicates which type of ML model the defense mechanisms can be leveraged to defend against MIA. The adversarial knowledge column indicates which type of attack the defense mechanisms can defend, i.e., the black-box or white-box attacks. As we can see, DP is the most applicable membership inference defense mechanism that works for both classification and generative models and does not care what knowledge the adversary has. Note that as a theoretical privacy guarantee, DP has been investigated to defend other forms of privacy attacks such as attribute inference and property inference attacks [40, 70]. Regularization mechanisms are also available to defend against both black-box and white-box attacks for both classification and generative models. However, for MIA on generative models, only dropout is investigated as a defense method among all regularization techniques. Whether and how to use other regularization techniques to mitigate MIA on generative models remains an open question. For knowledge distillation based methods, i.e., DMP, [87] is the state-of-the-art defense for classification models that offer better privacy-utility tradeoffs than regularization techniques and DP. It can defend against both black-box and white-box MIA.

Moreover, we summarize all related datasets that are used for evaluating the membership inference risks on ML models in Table 6. We hope Table 6 can help future researchers working on this topic to select appropriate datasets for evaluating their proposed attacks or defenses. We select 16 datasets from related papers of membership inference attacks and defenses. We introduce them along with their data type of images or binary features, the learning tasks they are used for, the size, dimension of each instance, how many categories, and the corresponding reference papers. Generally, various datasets are used to evaluate the privacy risks of membership inference on ML models. Most of those datasets are image datasets that are used for classification or generation tasks. The samples in them range from simple handwritten digits with size  $28 \times 28 \times 1$  in MNIST [119] to celebrity images with size  $218 \times 178 \times 3$  in CelebA [62]. Among the 16 datasets, MNIST [119], CIFAR-10 [51], and CIFAR-100 [51] are the top 3 datasets that are frequently used for evaluating MIA on both classification and generation tasks. Purchase-100 [44] and Texas-100 [100] are another 2 popular binary datasets.

## 8 FUTURE DIRECTIONS

In this section, we discuss the future directions of MIA from three perspectives, i.e., *membership inference attacks on different domains*, *novel defenses against membership inference attacks*, and *disparate membership inference risk across individuals and classes*.

### 8.1 Membership inference attacks on different domains

In addition to classification and generative models, MIA have been investigated on various domains, including embedding models [91], regression models [28], sequence-to-sequence models [34, 93], image segmentation [31, 86], transfer learning models [11, 58, 125], algorithmic fairness [10], model explanations [88, 89], adversarial machine learning [95, 96], and graph neural network [30, 74]. Song



Table 6. A summary of benchmark datasets used in evaluating membership inference attacks and defenses on machine learning models

Dataset	Data Type		Learning Task		Size	Dimension	Category	Reference
	Image	Binary	Classification	Generation				
Adult [83]	×	✓	✓	×	48,842	14	2	[38, 39, 55, 63, 85, 90, 104]
Cancer [112]	×	✓	✓	×	699	10	2	[55, 63]
Diabetic Retinopathy [45]	✓	×	×	✓	88,702	varies	5	[29]
MNIST [119]	✓	×	✓	✓	70,000	28×28×1	10	[16, 32, 55–57, 60, 63, 69, 78–80, 85, 90, 103–105, 120]
Fashion-MNIST [119]	✓	×	✓	✓	70,000	28×28×1	10	[32, 48, 57, 61, 69, 78]
CH-MNIST [47]	✓	×	✓	×	5,000	150×150×1	8	[38, 42, 78]
CIFAR-10 [51]	✓	×	✓	✓	60,000	32×32×3	10	[16, 29, 32, 48, 55–57, 69, 78–80, 84, 85, 87, 90, 104, 105, 120]
Foursquare [118]	×	✓	✓	×	528,878	446	30	[38, 42, 66, 78, 85, 90]
GTSRB [98]	✓	×	✓	✓	50,000	64×64×3	40	[57]
CIFAR-100 [51]	✓	×	✓	✓	60,000	32×32×3	100	[16, 38, 40, 48, 55–57, 71, 72, 78, 80, 84, 85, 87, 90, 94, 120]
Purchase-100 [44]	×	✓	✓	×	197,324	600	100	[38, 40, 41, 56, 71, 72, 78, 85, 87, 90, 94, 104, 105]
Texas-100 [100]	×	✓	✓	×	67,330	6,170	100	[38, 41, 42, 56, 71, 72, 78, 87, 90, 94]
Birds-200 [109]	✓	×	✓	×	11,788	N.A.	200	[38]
LFW [37]	✓	×	✓	✓	13,233	62×47×3	5749	[29, 55, 57, 66, 69, 85, 114]
CelebA [62]	✓	×	×	✓	202,599	218×178×3	10,177	[12, 60, 61, 103]
ChestX-ray8 [108]	✓	×	×	✓	108,948	1024×1024×1	32,717	[60]

and Raghunathan [91] investigate the membership risks on embedding models and demonstrate a simple threshold attack can achieve a 30% improvement in attack accuracy over random guessing on Word2Vec [67], Fast-Text [7], Glove [76], LSTM [35], and Transformer [106] models. Hisamoto et al. [34] present the possibility of MIA on the sequence-to-sequence model of machine translation. Liew and Takahashi [58] present that in the application of transfer learning on face recognition, an adversary can successfully implement MIA on the teacher model even when it only accesses the student models. Shokri et al. [88, 89] show that releasing transparency reports of ML models can result in membership privacy risks. Song et al. [95, 96] find that the adversarially trained model is more vulnerable to MIA. As there is a growing body of work on MIA in various domains, why they succeed and the corresponding defenses need to be explored.

## 8.2 Novel defenses against membership inference attacks

Dwork et al. [21] argue that if a machine learning model is useful, it must reveal information about the data on which it was trained. It seems inevitable for ML models to leak some private information of their training data. In addition to the aforementioned four defense mechanisms, there are a few novel perspectives on how to defend MIA. Tople et al. [101] suggest to leverage causal learning methods [2] that aims to learn stable features across distributions to mitigate the membership privacy risks. However, their method is proposed for causal models but not suitable for deep NN models, which are the focus of this survey. Wu et al. [113] show that Bayes deep learning helps to mitigate MIA. They demonstrate Stochastic Gradient Langevin Dynamics (SGLD) [110], an effective method to enable Bayes deep learning, which can prevent membership leakage to a certain extent

while preserving similar model accuracy compared to SGD. Finding new defense mechanisms and exploring new perspectives of membership privacy to explain behaviours of machine learning models remains an open question.

### 8.3 Disparate membership inference risks

The effectiveness of membership inference attacks and defenses are mainly evaluated on the whole population in the training dataset. However, understanding the membership inference risks on different classes and individuals are also important. Long et al. [63] first present that outlier samples are more vulnerable to MIA. Yaghini et al. [117] show that there are disparate attack success across different subgroups of the data. They further demonstrate that even if MIA achieve nearly random guessing over the whole population, there are still vulnerable subgroups. Truex et al. [104] show that the minority populations are more likely to be identified by the adversary with a higher confidence. Rezaei and Liu [80] investigate how attacks may differ on a target classifier’s correctly classified and misclassified samples. They find that the current MIA are difficult to distinguish members from non-members on these correctly classified samples. Humphries et al. [39] challenge that DP provides a strong defense against MIA. They show that when the member data distribution is sufficiently distinct from that of the rest of the population, the current MIA might break the theoretical attack success upper bound provided by differentially private models. Investigating and understanding MIA under different settings remains an open question.

## 9 CONCLUSIONS

Membership inference, including attacks and defenses, is a critical and booming research area. In this paper, we present the first comprehensive survey on this topic. We summarize and categorize existing membership inference attacks and defenses and explicitly present how to implement MIA in various settings. We explicitly introduce novel designs of MIA on both classification and generative models. We discuss why membership inference works and summarize benchmark datasets to facilitate comparison and ensure fairness of future works. Based on reviewed literature, we propose several possible directions for future research. Through this comprehensive overview, we hope to prepare a solid foundation for future research in this field.

## REFERENCES

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (Vienna, Austria) (CCS ’16)*. Association for Computing Machinery, New York, NY, USA, 308–318. <https://doi.org/10.1145/2976749.2978318>
- [2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893* (2019). <https://arxiv.org/abs/1907.02893>
- [3] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. 2015. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks* 10, 3 (2015), 137–150. <https://doi.org/10.1504/IJSN.2015.071829>
- [4] Lei Jimmy Ba and Rich Caruana. 2014. Do Deep Nets Really Need to Be Deep?. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (Montreal, Canada) (NIPS’14)*. MIT Press, Cambridge, MA, USA, 2654–2662.
- [5] Brett K Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P Bhavnani, James Brian Byrd, and Casey S Greene. 2019. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes* 12, 7 (2019), e005122. <https://doi.org/10.1161/CIRCOUTCOMES.118.005122>
- [6] David Berthelot, Thomas Schumm, and Luke Metz. 2017. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717* (2017). <https://arxiv.org/abs/1703.10717>
- [7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. [https://doi.org/10.1162/tacl\\_a\\_00051](https://doi.org/10.1162/tacl_a_00051)

- [8] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography*, Martin Hirt and Adam Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 635–658.
- [9] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 267–284. <https://www.usenix.org/conference/usenixsecurity19/presentation/carlini>
- [10] Hongyan Chang and Reza Shokri. 2020. On the Privacy Risks of Algorithmic Fairness. *arXiv preprint arXiv:2011.03731* (2020). <https://arxiv.org/abs/2011.03731>
- [11] Cen Chen, Bingzhe Wu, Minghui Qiu, Li Wang, and Jun Zhou. 2020. A Comprehensive Analysis of Information Leakage in Deep Transfer Learning. *arXiv preprint arXiv:2009.01989* (2020). <https://arxiv.org/abs/2009.01989>
- [12] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (Virtual Event, USA) (CCS '20)*. Association for Computing Machinery, New York, NY, USA, 343–362. <https://doi.org/10.1145/3372297.3417238>
- [13] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2020. When Machine Unlearning Jeopardizes Privacy. *arXiv preprint arXiv:2005.02205* (2020). <https://arxiv.org/abs/2005.02205>
- [14] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. 2018. Differentially private data generative models. *arXiv preprint arXiv:1812.02274* (2018). <https://arxiv.org/abs/1812.02274>
- [15] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. 2017. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. In *Proceedings of the 2nd Machine Learning for Healthcare Conference (Proceedings of Machine Learning Research, Vol. 68)*, Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens (Eds.). PMLR, Boston, Massachusetts, 286–305. <http://proceedings.mlr.press/v68/choi17a.html>
- [16] Christopher A Choquette Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2020. Label-Only Membership Inference Attacks. *arXiv preprint arXiv:2007.14321* (2020). <https://arxiv.org/abs/2007.14321>
- [17] Elliot J. Crowley, Gavin Gray, and Amos Storkey. 2018. Moonshine: Distilling with Cheap Convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (Montréal, Canada) (NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 2893–2903.
- [18] Emiliano De Cristofaro. 2020. An overview of privacy in machine learning. *arXiv preprint arXiv:2005.08679* (2020). <https://arxiv.org/abs/2005.08679>
- [19] Cynthia Dwork. 2008. Differential Privacy: A Survey of Results. In *Theory and Applications of Models of Computation*, Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–19.
- [20] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284.
- [21] Cynthia Dwork and Moni Naor. 2010. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality* 2, 1 (2010).
- [22] Robert Fortet and Edith Mourier. 1953. Convergence de la répartition empirique vers la répartition théorique. In *Annales scientifiques de l'École Normale Supérieure*, Vol. 70. 267–285.
- [23] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (Denver, Colorado, USA) (CCS '15)*. Association for Computing Machinery, New York, NY, USA, 1322–1333. <https://doi.org/10.1145/2810103.2813677>
- [24] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association, San Diego, CA, 17–32. [https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/fredrikson\\_matt](https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/fredrikson_matt)
- [25] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. 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 (Toronto, Canada) (CCS '18)*. Association for Computing Machinery, New York, NY, USA, 619–633. <https://doi.org/10.1145/3243734.3243834>
- [26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>

- [27] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved Training of Wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) (*NIPS'17*). Curran Associates Inc., Red Hook, NY, USA, 5769–5779.
- [28] Umang Gupta, Dimitris Stripelis, Pradeep K Lam, Paul Thompson, Jose Luis Ambite, and Greg Ver Steeg. 2021. Membership Inference Attacks on Deep Regression Models for Neuroimaging. (2021).
- [29] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 01 Jan. 2019. LOGAN: Membership Inference Attacks Against Generative Models. *Proceedings on Privacy Enhancing Technologies* 2019, 1 (01 Jan. 2019), 133 – 152. <https://doi.org/10.2478/popets-2019-0008>
- [30] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. 2021. Node-Level Membership Inference Attacks Against Graph Neural Networks. *arXiv preprint arXiv:2102.05429* (2021). <https://arxiv.org/abs/2102.05429>
- [31] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. 2020. Segmentations-Leak: Membership Inference Attacks and Defenses in Semantic Image Segmentation. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 519–535.
- [32] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. 01 Oct. 2019. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. *Proceedings on Privacy Enhancing Technologies* 2019, 4 (01 Oct. 2019), 232 – 249. <https://doi.org/10.2478/popets-2019-0067>
- [33] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015). <https://arxiv.org/abs/1503.02531>
- [34] Sorami Hisamoto, Matt Post, and Kevin Duh. 2020. 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), 49–63. [https://doi.org/10.1162/tacl\\_a\\_00299](https://doi.org/10.1162/tacl_a_00299)
- [35] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735> arXiv:<https://doi.org/10.1162/neco.1997.9.8.1735>
- [36] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet* 4, 8 (2008), e1000167. <https://doi.org/10.1371/journal.pgen.1000167>
- [37] Gary B. Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Marseille, France. <https://hal.inria.fr/inria-00321923>
- [38] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. 2021. Practical Blind Membership Inference Attack via Differential Comparisons. In *Network and Distributed Systems Security Symposium 2021*. Internet Society.
- [39] Thomas Humphries, Matthew Rafuse, Lindsey Tulloch, Simon Oya, Ian Goldberg, and Florian Kerschbaum. 2020. Differentially Private Learning Does Not Bound Membership Inference. *arXiv preprint arXiv:2010.12112* (2020). <https://arxiv.org/abs/2010.12112>
- [40] Bargav Jayaraman and David Evans. 2019. Evaluating Differentially Private Machine Learning in Practice. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 1895–1912. <https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman>
- [41] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. 2020. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881* (2020). <https://arxiv.org/abs/2005.10881>
- [42] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) (*CCS '19*). Association for Computing Machinery, New York, NY, USA, 259–274. <https://doi.org/10.1145/3319535.3363201>
- [43] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9. <https://doi.org/10.1038/sdata.2016.35>
- [44] Kaggle. 2014. *Acquire Valued Shoppers Challenge*. <https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data> (2020, Dec. 16).
- [45] Kaggle.com. [n.d.]. *Diabetic Retinopathy Detection*. <https://www.kaggle.com/c/diabetic-retinopathy-detection#references> (2020, Dec. 16).
- [46] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017). <https://arxiv.org/abs/1710.10196>
- [47] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. 2016. Multi-class texture analysis in colorectal cancer histology. *Scientific*

- reports* 6, 1 (2016), 1–11. <https://doi.org/10.1038/srep27988>
- [48] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. 2020. On the Effectiveness of Regularization Against Membership Inference Attacks. *arXiv preprint arXiv:2006.05336* (2020). <https://arxiv.org/abs/2006.05336>
  - [49] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013). <https://arxiv.org/abs/1312.6114>
  - [50] Jakub Konečný, Brendan McMahan, and Daniel Ramage. 2015. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575* (2015). <https://arxiv.org/abs/1511.03575>
  - [51] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
  - [52] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* (2016). <https://arxiv.org/abs/1611.01236>
  - [53] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2016. Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1558–1566. <http://proceedings.mlr.press/v48/larsen16.html>
  - [54] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified Robustness to Adversarial Examples with Differential Privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*. 656–672. <https://doi.org/10.1109/SP.2019.00044>
  - [55] Klas Leino and Matt Fredrikson. 2020. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 1605–1622. <https://www.usenix.org/conference/usenixsecurity20/presentation/leino>
  - [56] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2020. Membership inference attacks and defenses in supervised learning via generalization gap. *arXiv preprint arXiv:2002.12062* (2020). <https://arxiv.org/abs/2002.12062>
  - [57] Zheng Li and Yang Zhang. 2020. Label-Leaks: Membership Inference Attack with Label. *arXiv preprint arXiv:2007.15528* (2020). <https://arxiv.org/abs/2007.15528>
  - [58] Seng Pei Liew and Tsubasa Takahashi. 2020. FaceLeaks: Inference Attacks against Transfer Learning Models via Black-box Queries. *arXiv preprint arXiv:2010.14023* (2020). <https://arxiv.org/abs/2010.14023>
  - [59] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2020. When Machine Learning Meets Privacy: A Survey and Outlook. *ACM Computing survey* (2020).
  - [60] Kin Sum Liu, Chaowei Xiao, Bo Li, and Jie Gao. 2019. Performing Co-membership Attacks Against Deep Generative Models. In *2019 IEEE International Conference on Data Mining (ICDM)*. 459–467. <https://doi.org/10.1109/ICDM.2019.00056>
  - [61] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2021. ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models. *arXiv preprint arXiv:2102.02551* (2021). <https://arxiv.org/abs/2102.02551>
  - [62] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
  - [63] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diye Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. 2018. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889* (2018). <https://arxiv.org/abs/1802.04889>
  - [64] Y. Long, L. Wang, D. Bu, V. Bindschaedler, X. Wang, H. Tang, C. A. Gunter, and K. Chen. 2020. A Pragmatic Approach to Membership Inferences on Machine Learning Models. In *2020 IEEE European Symposium on Security and Privacy (EuroS P)*. 521–534. <https://doi.org/10.1109/EuroSP48549.2020.00040>
  - [65] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, Fort Lauderdale, FL, USA, 1273–1282. <http://proceedings.mlr.press/v54/mcmahan17a.html>
  - [66] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. 691–706. <https://doi.org/10.1109/SP.2019.00029>
  - [67] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (Lake Tahoe, Nevada) (NIPS’13)*. Curran Associates Inc., Red Hook, NY, USA, 3111–3119.
  - [68] I. Mironov. 2017. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. 263–275. <https://doi.org/10.1109/CSF.2017.11>
  - [69] Sumit Mukherjee, Yixi Xu, Anusua Trivedi, and Juan Lavista Ferres. 2019. privGAN: Protecting GANs from membership inference attacks at low cost. *arXiv preprint arXiv:2001.00071* (2019). <https://arxiv.org/abs/2001.00071>

- [70] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. 2020. Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy. *arXiv preprint arXiv:2009.03561* (2020). <https://arxiv.org/abs/2009.03561>
- [71] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine Learning with Membership Privacy Using Adversarial Regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) (CCS '18). Association for Computing Machinery, New York, NY, USA, 634–646. <https://doi.org/10.1145/3243734.3243855>
- [72] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. 739–753. <https://doi.org/10.1109/SP.2019.00065>
- [73] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. 2021. Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning. *arXiv preprint arXiv:2101.04535* (2021). <https://arxiv.org/abs/2101.04535>
- [74] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. 2021. Membership Inference Attack on Graph Neural Networks. *arXiv preprint arXiv:2101.06570* (2021). <https://arxiv.org/abs/2101.06570>
- [75] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the International Conference on Learning Representations*. <https://arxiv.org/abs/1610.05755>
- [76] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543. <https://www.aclweb.org/anthology/D14-1162.pdf>
- [77] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015). <https://arxiv.org/abs/1511.06434>
- [78] Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. 2020. Sampling Attacks: Amplification of Membership Inference Attacks by Repeated Queries. *arXiv preprint arXiv:2009.00395* (2020). <https://arxiv.org/abs/2009.00395>
- [79] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. 2018. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv.* 11, 1 (2018), 61–79.
- [80] Shahbaz Rezaei and Xin Liu. 2020. Towards the Infeasibility of Membership Inference on Deep Models. *arXiv preprint arXiv:2005.13702* (2020). <https://arxiv.org/abs/2005.13702>
- [81] Maria Rigaki and Sebastian Garcia. 2020. A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646* (2020). <https://arxiv.org/abs/2007.07646>
- [82] Christian Robert and George Casella. 2013. *Monte Carlo statistical methods*. Springer Science & Business Media.
- [83] Kohavi Ronny and Becker Barry. 1996. *UCI Machine Learning Repository*. <http://archive.ics.uci.edu/ml/datasets/Adult>
- [84] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Herve Jegou. 2019. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 5558–5567. <http://proceedings.mlr.press/v97/sablayrolles19a.html>
- [85] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. 2019. 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. <http://hdl.handle.net/21.11116/0000-0002-5B4C-4>
- [86] Avital Shafraan, Shmuel Peleg, and Yedid Hoshen. 2021. Reconstruction-Based Membership Inference Attacks are Easier on Difficult Problems. *arXiv preprint arXiv:2102.07762* (2021). <https://arxiv.org/abs/2102.07762>
- [87] Virat Shejwalkar and Amir Houmansadr. 2021. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [88] Reza Shokri, Martin Strobel, and Yair Zick. 2020. Exploiting Transparency Measures for Membership Inference: a Cautionary Tale. In *The AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI)*. AAAI, Vol. 13.
- [89] Reza Shokri, Martin Strobel, and Yair Zick. 2020. On the Privacy Risks of Model Explanations. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*.
- [90] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. [n.d.].
- [91] Congzheng Song and Ananth Raghunathan. 2020. Information Leakage in Embedding Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) (CCS '20). Association for Computing Machinery, New York, NY, USA, 377–390. <https://doi.org/10.1145/3372297.3417270>
- [92] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine Learning Models That Remember Too Much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) (CCS '17). Association for Computing Machinery, New York, NY, USA, 587–601. <https://doi.org/10.1145/3133956.3134077>

- [93] Congzheng Song and Vitaly Shmatikov. 2019. Auditing Data Provenance in Text-Generation Models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 196–206. <https://doi.org/10.1145/3292500.3330885>
- [94] Liwei Song and Prateek Mittal. 2020. Systematic Evaluation of Privacy Risks of Machine Learning Models. *arXiv preprint arXiv:2003.10595* (2020). <https://arxiv.org/abs/2003.10595>
- [95] L. Song, R. Shokri, and P. Mittal. 2019. Membership Inference Attacks Against Adversarially Robust Deep Learning Models. In *2019 IEEE Security and Privacy Workshops (SPW)*. 50–56. <https://doi.org/10.1109/SPW.2019.00021>
- [96] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Privacy Risks of Securing Machine Learning Models against Adversarial Examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) (CCS '19). Association for Computing Machinery, New York, NY, USA, 241–257. <https://doi.org/10.1145/3319535.3354211>
- [97] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
- [98] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks* 32 (2012), 323–332. <https://doi.org/10.1016/j.neunet.2012.02.016> Selected Papers from IJCNN 2011.
- [99] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2818–2826.
- [100] Texas Health Care Information Collection Center. 2006–2009. *Texas Inpatient Public Use Data File (PUDF)*. <https://www.dshs.texas.gov/thcic/hospitals/Inpatientpdf.shtm> (2020, Dec. 16).
- [101] Shruti Tople, Amit Sharma, and Aditya Nori. 2020. Alleviating Privacy Attacks via Causal Learning. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 9537–9547. <http://proceedings.mlr.press/v119/tople20a.html>
- [102] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 601–618. <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer>
- [103] Aleksei Triastcyn and Boi Faltings. 2019. Generating Artificial Data for Private Deep Learning. In *Proceedings of the PAL: Privacy-Enhancing Artificial Intelligence and Language Technologies, AAAI Spring Symposium Series*.
- [104] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Wenqi Wei, and Lei Yu. 2019. Effects of Differential Privacy and Data Skewness on Membership Inference Vulnerability. In *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 82–91. <https://doi.org/10.1109/TPS-ISA48467.2019.00019>
- [105] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying Membership Inference Attacks in Machine Learning as a Service. *IEEE Transactions on Services Computing* (2019), 1–1. <https://doi.org/10.1109/TSC.2019.2897554>
- [106] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.
- [107] Michael Veale, Reuben Binns, and Lilian Edwards. 2018. Algorithms that remember: model inversion attacks and data protection law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376, 2133 (2018), 20180083. <http://doi.org/10.1098/rsta.2018.0083>
- [108] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. 2017. ChestX-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [109] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-UCSD birds 200. (2010).
- [110] Max Welling and Yee W Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer, 681–688.
- [111] Wikipedia contributors. 2021. General Data Protection Regulation — Wikipedia, The Free Encyclopedia. [https://en.wikipedia.org/w/index.php?title=General\\_Data\\_Protection\\_Regulation&oldid=1003412589](https://en.wikipedia.org/w/index.php?title=General_Data_Protection_Regulation&oldid=1003412589) [Online; accessed 3-February-2021].
- [112] Wolberg William, Street Nick, and Mangasarian Olvi. 1995. *UCI Machine Learning Repository*. [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))
- [113] Bingzhe Wu, Chaochao Chen, Shiwang Zhao, Cen Chen, Yuan Yao, Guangyu Sun, Li Wang, Xiaolu Zhang, and Jun Zhou. 2020. Characterizing Membership Privacy in Stochastic Gradient Langevin Dynamics. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (Apr. 2020), 6372–6379. <https://doi.org/10.1609/aaai.v34i04.6107>

- [114] Bingzhe Wu, Shiwan Zhao, Chaochao Chen, Haoyang Xu, Li Wang, Xiaolu Zhang, Guangyu Sun, and Jun Zhou. 2019. Generalization in Generative Adversarial Networks: A Novel Perspective from Privacy Protection. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2019/file/47d1e990583c9c67424d369f3414728e-Paper.pdf>
- [115] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. 2018. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739* (2018). <https://arxiv.org/abs/1802.06739>
- [116] Chugui Xu, Ju Ren, Deyu Zhang, Yaoyue Zhang, Zhan Qin, and Kui Ren. 2019. GANobfuscator: Mitigating Information Leakage Under GAN via Differential Privacy. *IEEE Transactions on Information Forensics and Security* 14, 9 (2019), 2358–2371. <https://doi.org/10.1109/TIFS.2019.2897874>
- [117] Mohammad Yaghini, Bogdan Kulynych, and Carmela Troncoso. 2019. Disparate vulnerability: On the unfairness of privacy attacks against machine learning. *arXiv preprint arXiv:1906.00389* (2019). <https://arxiv.org/abs/1906.00389>
- [118] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory Cultural Mapping Based on Collective Behavior Data in Location-Based Social Networks. *ACM Trans. Intell. Syst. Technol.* 7, 3, Article 30 (Jan. 2016), 23 pages. <https://doi.org/10.1145/2814575>
- [119] LeCun Yann, Cortes Corinna, and Burges Christopher J. C. [n.d.]. *The MNIST database of handwritten digits*. <http://yann.lecun.com/exdb/mnist/> (2020, Dec. 16).
- [120] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. 268–282. <https://doi.org/10.1109/CSF.2018.00027>
- [121] Zuobin Ying, Yun Zhang, and Ximeng Liu. 2020. Privacy-Preserving in Defending against Membership Inference Attacks. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice (Virtual Event, USA) (PPMLP'20)*. Association for Computing Machinery, New York, NY, USA, 61–63. <https://doi.org/10.1145/3411501.3419428>
- [122] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. [n.d.]. Understanding deep learning requires rethinking generalization. ([n. d.]). <https://arxiv.org/abs/1611.03530>
- [123] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=r1Ddp1-Rb>
- [124] Xinyang Zhang, Shouling Ji, and Ting Wang. 2018. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594* (2018). <https://arxiv.org/abs/1801.01594>
- [125] Yang Zou, Zhikun Zhang, Michael Backes, and Yang Zhang. 2020. Privacy Analysis of Deep Learning in the Wild: Membership Inference Attacks against Transfer Learning. *arXiv preprint arXiv:2009.04872* (2020). <https://arxiv.org/abs/2009.04872>