Hindawi
Security and Communication Networks
Volume 2021, Article ID 9924684, 13 pages
https://doi.org/10.1155/2021/9924684

Research Article

A Defense Framework for Privacy Risks in Remote Machine Learning Service

Yang Bai,[1,2] **Yu Li**,[1] **Mingchuang Xie**,[1] **and Mingyu Fan**[1]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China*
[2]*No. 30, Institute of CETC, Chengdu, China*

Correspondence should be addressed to Mingyu Fan; ff98@163.com

Received 5 March 2021; Revised 22 May 2021; Accepted 5 June 2021; Published 18 June 2021

Academic Editor: Jiang Ming

In recent years, machine learning approaches have been widely adopted for many applications, including classification. Machine learning models deal with collective sensitive data usually trained in a remote public cloud server, for instance, machine learning as a service (MLaaS) system. In this scene, users upload their local data and utilize the computation capability to train models, or users directly access models trained by MLaaS. Unfortunately, recent works reveal that the curious server (that trains the model with users' sensitive local data and is curious to know the information about individuals) and the malicious MLaaS user (who abused to query from the MLaaS system) will cause privacy risks. The adversarial method as one of typical mitigation has been studied by several recent works. However, most of them focus on the privacy-preserving against the malicious user; in other words, they commonly consider the data owner and the model provider as one role. Under this assumption, the privacy leakage risks from the curious server are neglected. Differential privacy methods can defend against privacy threats from both the curious sever and the malicious MLaaS user by directly adding noise to the training data. Nonetheless, the differential privacy method will decrease the classification accuracy of the target model heavily. In this work, we propose a generic privacy-preserving framework based on the adversarial method to defend both the curious server and the malicious MLaaS user. The framework can adapt with several adversarial algorithms to generate adversarial examples directly with data owners' original data. By doing so, sensitive information about the original data is hidden. Then, we explore the constraint conditions of this framework which help us to find the balance between privacy protection and the model utility. The experiments' results show that our defense framework with the AdvGAN method is effective against MIA and our defense framework with the FGSM method can protect the sensitive data from direct content exposed attacks. In addition, our method can achieve better privacy and utility balance compared to the existing method.

1. Introduction

In recent years, machine learning technology has rapidly gained popularity, as its model can be improved automatically through learning from the training dataset. Benefit from the development enables more efficient storage, processing, and computation, and more and more machine learning models are widely applied in the real world for classification or regression tasks for many domains, such as image classification [1], speech recognition [2], healthcare data management [3], and financial analysis [4]. Samples are used to train the machine learning model to represent many individuals' sensitive information, such as patients' healthcare information, personal preference, and personal photos. For instance, image classification, especially facial recognition technologies, is applied in many scenes including activity monitoring and identification. Healthcare record management requires clinical features and medical data from patients to learn a model for diagnosis and prognosis. In finance, an individual's trade record, current price records, and so on are used to study the financial prediction machine learning and financial risk analysis systems.

Machine learning models deal with collective sensitive data usually trained in a remote public cloud server, for instance, machine learning as a service (MLaaS) system. In

this scene, users upload their local data and utilize the computation capability to train models, or users directly access models trained by MLaaS. Unfortunately, recent works reveal that the curious server (that trains the model with users' sensitive local data and is curious to know the information about individuals) and the malicious MLaaS user (who abused to query from the MLaaS system) will cause privacy risks. For example, Shrokri et al. [5] came up with a membership inference attack (MIA) against machine learning models. The MIA adversary uses machine learning to train an attack model to speculate if a given data record was a member of the target model's training dataset. These privacy risks not only can directly violate the privacy of the training set but also can be a gateway to further attacks [6]. That is, some adversaries use the inference information to construct other security threats. For example, if the individual's biometric features were inferred out, the attacker can utilize this information to make a personating attack or obtain illegal authorities. These further attacks might pose additional severe information leakage or other heavy security risks [5].

The adversarial method is one of the typical mitigations to address the machine learning privacy risks. Nasr et al. [7] formalized the privacy risk preserving as a min-max game optimization problem that minimizes the classification error of the target model and minimizes the inference adversary's maximum gain. They use the stochastic gradient descent algorithm to optimize the min-max problem. Jia et al. [8] proposed a MemGuard to defend against black-box MIAs by adding a carefully crafted noise vector to the target model's output confidence score vector. Wang et al. [9] jointly formulate model compression and MIA as MCMIA, utilizing model compression against MIAs in deep neural networks. However, most of them focus on the privacy-preserving against the malicious user; in other words, they commonly considered the data owner and the model provider as one role and the adversarial perturbation objective always focuses on the target model or the confidence score of the target model. Under this assumption, the privacy leakage risks from the curious server are neglected. It is not suitable for the data user to upload their local data to the MLaaS system for training a machine learning model remotely. That is, the privacy leakage risks from the model provider are commonly neglected, and the owner's data privacy cannot be preserved entirely. Differential privacy methods can defend against privacy threats from both the curious server and the malicious MLaaS user by directly adding noise to the training data. Nonetheless, the differential privacy method will decrease the classification accuracy of the target model heavily.

In this paper, we investigate existing adversarial perturbation-based defenses to categorize them. Then, we propose a generic privacy-preserving framework based on an adversarial method to defend both the curious server and the malicious MLaaS user. The framework can adapt with several adversarial algorithms to generate adversarial examples directly with data owners' original data. By doing so, sensitive information about the original data is hidden. In addition, we explore the constraint conditions of this framework which help us to find the balance between privacy protection and the model utility. The framework consists of one adversarial perturbation generator, shadow models, and privacy evaluator. The generator learns to perturb the original data of the local user. We leverage the adversarial method to diminish the effect of the membership inferences by generating adversarial examples of the original training data. By doing so, sensitive information about the training data is hidden. The privacy evaluator then detects whether there exist privacy risks. The feedback scheme and the constraint are used to find out the balance between privacy-preserving and the performance of the target model. The main contributions of this paper are summarized as follows:

(i) Taxonomy of existing adversarial method-based privacy risk mitigation. We propose a pioneering study to categorize adversarial method-based defenses by different perturbation objects; and we analyze the limitation of this existing mitigation. These works help us to build general recognition in this field.

(ii) The generic defense framework for adversarial method against privacy risk. We utilize the feedback mechanism and some constraint conditions to develop a common adversarial defense framework for machine learning privacy-preserving. We also explore the constraint conditions that make our method more effective. This framework can protect the original training data's privacy under data sharing and MLaaS scenarios; and we implement three different kinds of adversarial example algorithms based on privacy-preserving mitigation with our generic framework. We exploit the MIA as the classical privacy leakage evaluator to verify the effectiveness of these three defenses.

(iii) Comprehensive analysis of the influence factors for the proposed method. We investigate defense factors, including adversarial algorithms, perturbation rates, adversarial distance, and data type, showing that, under appropriate factors strategies, our defense framework with the AdvGAN method is effective against MIA and our defense framework with the FGSM method can protect the sensitive original data from direct content exposed attacks. In addition, our method can achieve more privacy and utility compared to the existing method, for instance, the min-max method and the differential privacy. Furthermore, the experiment results show that we can control the defense performance with parameters such as distortion rate and distance threshold.

2. Background and Related Work

In this section, we present background and related work on privacy threats in data sharing and MLaaS scenes and briefly review the membership inference attacks and machine learning privacy-preserving proposed in previous works.

2.1. Privacy Threats in Data Sharing and MLaaS Scenes. Fatemehsadat et al. [10] divided existing threats to privacy in machine learning systems into two main categories of direct and indirect information exposure hazards. They define in the direct threats that the adversary can gain direct access to sensitive datasets. Private data can be exposed through data sharing scenes or the cloud service that receives it to run a process on it [10]. For example, in the MLaaS platform, training data must be revealed to the service. If the MLaaS service operators are malicious, they can directly access the sensitive training data. However, users may not want to reveal their private information. The indirect information exposure means the attacker attempts to infer or guess the information and does not have access to the actual information [10]. A number of works have focused on the indirect information exposure, such as model extraction attacks [11], model inversion [12], and membership inference attacks [5]. Tramer et al. [11] explore model extraction attacks that exploit the tension between query access and confidentiality in machine learning models and aim to copy functions from victim models in the MLaaS setting. Fredrikson et al. [13] devise the model inversion attack in which the adversary access to a machine learning model abused to learn sensitive genomic information about individuals. Fredrikson made a further exploration to develop a new class of model inversion attacks based on prediction results from the MLaaS APIs. Shokri et al. [5] proposed the membership inference attack performed entirely through the MLaaS services. The model extraction attacks refer to the confidentiality of the victim model. The model inversion and the membership inference attack threaten the privacy of training data. The privacy risks in data sharing and MLaaS are not only from adversaries but may also come from the curious server and other malicious users.

2.2. Membership Inference Attacks (MIAs). In an MIA, the adversary queries the victim model to determine whether a particular sample is included in the training set of the victim model. Previous research demonstrates that overfitting is one of the reasons to cause the MIAs [14]. Shokri et al. [5] introduced the first MIA against machine learning system, and they turned MIA into a binary classification problem. In Shokri et al.'s method, the process of MIA is as follows. In the first stage, the attacker prepares a set of candidate data records. Next, they query access to the victim model and gain a classification probability vector from the victim model. After that, the adversary trains shadow models to craft training data for the attack model, followed by training the attack model and using the attack model to determine whether the candidate samples are members of the target model's training dataset. For each candidate record, there are two possible classes: class "member" means that the candidate data is a member of the target model's training dataset, and the class "nonmember" means that the candidate data is not in the training dataset. Hayes et al. [6] present the first MIAs against generative adversarial networks (GANs). ML-leaks [15] relaxes some assumptions of Shokri et al.'s work [5], such as the number of shadow models, the knowledge of the target model structure, and the target model's dataset information.

Despite the fact that the previous research has focused on MIA as a method of attack, recent works [7, 8, 16, 17] have used MIA as a privacy leak assessment tool. Jayaraman et al. [16] used membership inference (MI) to quantify the impact of differential privacy with the privacy loss of different ML models. The MI-based evaluator helps to find the range of differential privacy parameters to achieve a balance between utility and privacy and also to evaluate the privacy leakage of training data at risk of exposure. Song et al. [17] exploited MI to evaluate the defense approach of adversarial regularization [7] and MemGuard [8] with a new metric called the privacy risk score. Jia et al. [8] and Nasr et al. [7] used neural network classifiers to evaluate the mitigation performance. Thus, we choose MI as the classical privacy leakage evaluator to verify the effectiveness of our defense framework.

2.3. Machine Learning Privacy-Preserving. Existing ML privacy-preserving can be classified into three categories, that is, generalization method [5, 15], differential privacy [16, 18, 19], and adversarial methods [7–9].

2.3.1. Generalization Method. Previous works show that the overfitting model can memorize the information about the training data; furthermore, it can cause the privacy leakage risks, such as MIA. Thus, generalization is one of the most popular mitigations against ML privacy risks. Salimans et al. [20] presented the weight normalization, a simple reparameterization of the weight vectors in a neural network, by speeding up the convergence of stochastic gradient descent. Srivastava et al. [21] used dropout to fit the overfitting problem in deep neural network with a large number of parameters. Shokri et al. [22] and Salem et al. [15] found that dropout only was effective to degrade overfitting and strengthen privacy-preserving in neural networks. Salem et al. [15] exploited model stack as the generalization method which is suitable for all ML models. Shokri et al. [5] utilized standard regularization to mitigate privacy risk caused by overfitting. However, Long et al. [14] discovered that overfitting is a sufficient but not necessary condition for MIA to succeed. They found that existing generalization techniques are less effective in MIA protecting.

2.3.2. Differential Privacy. Differential privacy has been regarded as a strong privacy standard [23–27]. Differential privacy is one of the classical defenses against privacy risks. According to the ML processes, the differential privacy mechanism can be applied as the input perturbation, the gradient perturbation, the objective perturbation, and the output perturbation [28]. Reference [29] presented a differentially private GANs model which includes a Gaussian noise layer in the discriminator in case of a generative adversarial network to make the output and the gradients differential privacy with respect to the training data. Papernot et al. [30] used Private Aggregation of Teacher Ensembles (PATE) to construct an output perturbation.

Reference [31] used the differentially private stochastic gradient descent algorithm (DP-SGD) to prevent memorization. Although differential privacy has a significant effect on privacy protection, the introduction of a differential privacy mechanisms will greatly reduce the classification performance of the target model. This disadvantage hampers the application of differential privacy in real-world scenarios [16].

2.3.3. Adversarial Methods. Previous works showed that the adversarial method-related defenses contain two branches of research, that is, adversarial training and adversarial examples. Nasr et al. [7] put forward a min-max game which designs an adversarial training algorithm that minimizes the prediction loss of the model as well as the maximum gain of the inference attacks. This strategy can guarantee that membership privacy acts also helped to generalize the target model. Jia et al. [8] proposed a method based on adversarial examples, which adds noise to each confidence score vector by victim classifier. Its aim is to mislead the classification ability of attack models through carefully crafted adversarial samples. MemGuard [8] can effectively defend against MIAs and achieve better privacy-utility tradeoffs compared to previous works. However, Nasr et al. and Jia et al. considered the data owner and the model provider as one role. Therefore, these mitigations are not suitable for data sharing or MLaaS scenes in which the data owner should reveal their sensitive data to the services. In addition, there are many methods and algorithms for adversarial training and adversarial examples [33–39]. Therefore, it is necessary to construct a general defense framework to adapt the different sample adversarial algorithms, to find a balance between the defense of privacy and the performance of the target model, and to cover a wider range of privacy risk scenarios.

3. Taxonomy of Adversarial Method-Based Defense

In general, adversarial method defense can be categorized into three types by different perturbation objects: the input training data, the model, and the output prediction vector. We categorize the existing and the proposed adversarial method-based defenses in Table 1. We describe all the possible types in the following paragraphs.

3.1. Output Perturbation-Based Defenses. This kind of defense aims at adding crafted noise to the confidence score vector or labels to synthesize adversarial examples to mislead the attack classifiers. For example, MemGuard [8] adds noise vector to the confidence score with a certain probability to defend against black-box membership inference attacks. Yang et al. [32] proposed a framework to purify the confidence score vectors by reducing their dispersion. Yang's purification framework can defend against the model inversion attack and the membership inference attack. Both Jia et al. [8] and Yang et al. [32] considered the data owner and the model provider as one role. This assumption is not suitable for the scenario where users upload their local data to train a model remotely. This defense is the effective mitigation for privacy threats that the adversary infers the reconstruction and the membership of a training data from the probability predicted by the victim classifier.

Table 1: Taxonomy of adversarial method-based defenses.

Defenses	Perturbation object		
	Input	Model	Output
MemGuard [8]			✓
Yang et al. [32]			✓
MCMIA [9]		✓	
Min-max [7]		✓	
Our method	✓		

3.2. Model Perturbation-Based Defenses. In this defense type, defenders exploit model compression or model adversarial training to reduce the privacy risks. For instance, Wang et al. [9] jointly formulated model compression and MIA as MCMIA to reduce the information leakage from MIA. Min-max method [7] designed an adversarial training algorithm that minimizes the classification loss of victim model and maximum gain of attack model. These defenses can mitigate the privacy risk caused by the model overfitting, whereas these defenses are hard to be deployed to a public MLaaS platform.

3.3. Input Perturbation-Based Defenses. A number of proposed adversarial method-based defenses almost focus on the output perturbation and the model regularization to reduce the privacy leakage. As Figure 1 shows, both output perturbation and model perturbation schemes cannot defend against the privacy risks from the curious model provider. Input perturbation-based defenses which add noise directly to the training data can solve this problem. However, it is a challenge to find the balance between privacy-preserving and the utility of the classification model.

4. Generic Framework for Adversarial Method-Based Defenses

In this section, we begin by describing the privacy leakage in remote model training scenario (MLaaS) to formulate the problem of defending against these threats. Then, the generic defense framework is designed for adversarial method-based defense. In addition, we analyze the constraint conditions of the framework. Finally, we introduce three adversarial algorithms to implement the framework-based mitigation.

4.1. Problem Formulation. In a typical MLaaS application, there are four parties: user, model provider, attacker, and defender. We discuss each party as follows.

4.1.1. User. There are two kinds of users in the MLaaS scenario. The first user has some sensitive training data, such as facial images, healthcare records, financial information, and individual performance. As the user does not have

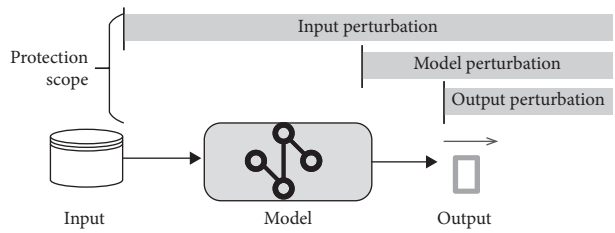

Figure 1: The protection scope of different kinds of adversarial method defenses.

enough computation resources, they want to input these local data to train a machine learning classifier by a remote MLaaS platform. Then the user can gain a trained classifier for their machine learning applications. The second user can directly exploit the classifier deployed in the MLaaS platform to query for some examples and obtains the classification result from the remote services. The local datasets X are uploaded to the remote service platform.

4.1.2. Model Provider. The model provider commonly is the public MLaaS platform. The model provider has two abilities: First of all, it can supply the initial model and the computation resource to users for training a model suitable for their characters. The second capability is that the provider can offer a trained model to users for querying and return the classification result (the confidence score vector of classification) for users. We call the machine model as target model (victim model).

4.1.3. Attacker. An attacker aims to obtain the information about user's sensitive data. The adversary probably is malicious users; they aim to infer the information about the training data. For instance, the adversary utilizes the black-box MIA [5, 6] to infer the members of the training dataset by querying from the target model. These attackers construct a binary classifier that can speculate out whether a query data record is one of the target model's training datasets or not by its confidence vector of the target model. Besides, the attacker may be a curious server of MLaaS that wants to know the details about user's uploading data. During the training and querying process, the user's data record is directly exposed to the server. So, attackers can easily achieve their goals.

4.1.4. Defender. The defender could be the user (data owner) or a trusted third party that has access to the user's uploading data X. For any training tasks or query tasks from the user to the model provider, the defender directly perturbs the uploading data to hide sensitive information to prevent the adversaries from obtaining exposed data or launching inference attacks. The defender aims to achieve two goals: The first goal is that the defender protects user's sensitive data from directly exposed risks or data inference attacks, such as MIAs. The second goal is that the defender tries to find the balance between privacy-preserving and target model's classification utility.

4.2. Framework Architecture. To defend against privacy threats in remote ML service scenarios, we propose a defense framework as depicted in Figure 2. Our defense framework consists of three components: (1) an adversarial perturbation generator that crafts adversarial examples to hide the sensitive information of uploading data, (2) a simulator to mimic the classification probability of the target model, and (3) a privacy leakage evaluator that detects the privacy leakage extent. The evaluator gives feedback to the generator to optimize the performance and the balance between defending against privacy threats and the target model's performance.

4.2.1. Adversarial Perturbation Generator. Although the adversarial examples are treated as harmful in many poisoning scenarios, they can be used to achieve our privacy protection goal. The fundamental idea is to generate an adversarial dataset, in which the sensitive privacy of original data is masked. As shown in Figure 2, the adversarial perturbation generator's goal is to find a set of adversarial examples X^{adv} which can conceal the sensitive information of the original uploading dataset X. In consideration of the target model's classification performance, we do the adversarial distortion as slightly as possible. That is, the Euclidean distance $L_2(x^{\text{adv}}, x)$ between each adversarial record and the original data should be small. We introduce the distance threshold ϵ, and we aim to achieve $L_2(x^{\text{adv}}, x) \leq \epsilon$ to help the generator to find out the appropriate adversarial examples which will be uploaded to train the simulator. There are many adversarial algorithms, for instance, the AdvGAN [40, 41], the Fast Gradient Sign Method (FGSM) [34], and the OPTMARGIN [42]. The adversarial algorithm is regarded as the basic support of the adversarial perturbation generator. The details of adversarial algorithms exploited in this paper are introduced in Section 5.1. We bring a parameter α named as the perturbation rate, which can control the ratio of adversarial examples in uploading training data.

4.2.2. Simulator. The simulator is constituted by one or several ML models that mimic the probability distribution of the target model. Shokri et al. [5] and Salem et al. [15] explored how to construct shadow models to mimic the target model effectively. We use the same method to construct the simulator. In our framework, we aim to generate adversarial perturbed data which has two attributes: (1) the perturbed data X^{adv} has the same classification label as the original record's label. (2) The generated data can be used to train and minimize the quadratic loss function $L(Y, f(X^{\text{adv}}))$ of the simulator.

4.2.3. Privacy Leakage Evaluator. We exploit the privacy leakage evaluator to detect the privacy risk level in order to evaluate the privacy-preserving effectiveness of the adversarial example generated module. The privacy leakage evaluator can adapt to multiple evaluation plugins. In this paper, we utilize the membership inference method as the

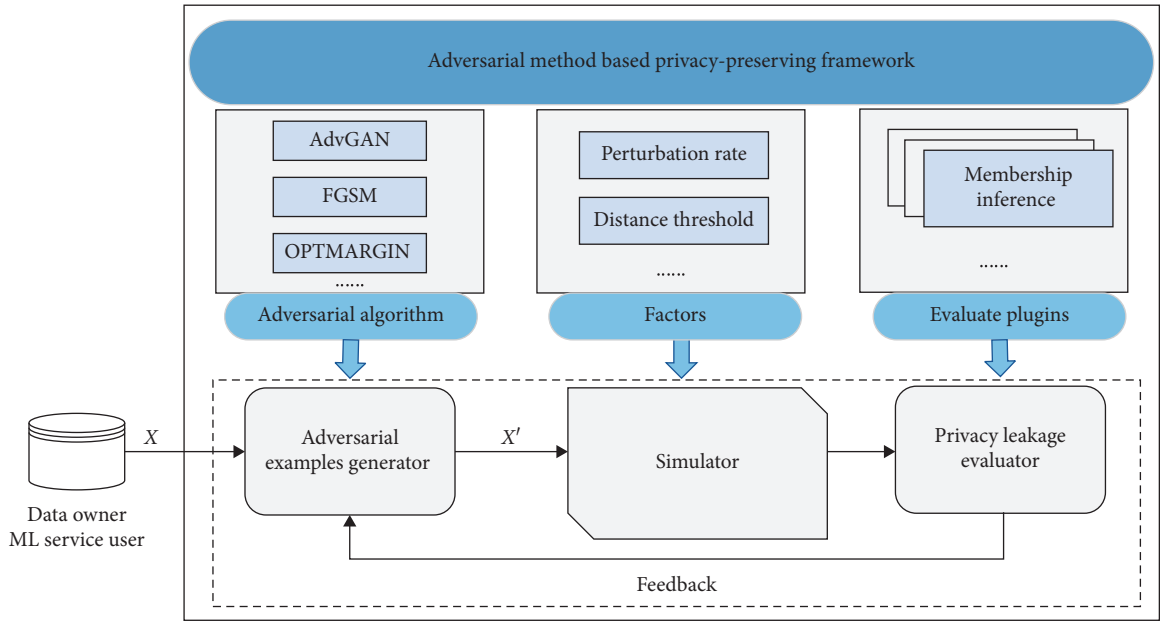

Figure 2: Adversarial method-based privacy-preserving framework.

typical evaluation plugin to verify the performance of our defense framework. The evaluator is an MI classifier that speculates whether an example is one of the simulator's members. In addition, this module sends the evaluation result to adversarial perturbation generator for helping the generated module to find out the proper adversarial examples. The assessment function considers several constraint conditions, for example, the generalization gap, the decreasing level of MI accuracy, and the loss function of the MI classifier. The loss function of the MI classifier is formulated as $R_{\text{emp}}(f) = (1/N)\sum_{N}^{i=1} L(y_i, f(x_{x_i}))$. In this module, the constraint conditions will be sent to the adversarial perturbation generator module for noise production parameter adjustment direction. If the MI inference accuracy is too large, we need to adjust the direction of greater perturbation disturbance. On the contrary, we need to adjust in the direction of less perturbation disturbance.

4.2.4. Defense for MIA. The defense framework can be detailedly specialized in the specific scenario of mitigating the MIA when the user uploads their local data to the remote machine learning service. In this paper, we explore the MIA proposed by Shokri et al. [5] in which the user supplies training data as inputs to the MLaaS service (model provider) to construct a model from the uploading data. Then, the user can use the model in other applications. In this scene, the adversary can access querying the model and obtain the classification result.

Algorithm. 1 shows the work method in this case. The adversarial perturbation generator is trained to synthesize a set of training data, which can mask their data information to the MI adversary (Goal I). At the same time, the performance of the simulator trained by these syntheses data does not degrade heavily (Goal II). The simulator and the evaluator are the assistant tools to help the generator to produce adversarial examples which satisfy the needs of Goal I and Goal II.

We formally present this MIA defense problem by the following optimization problem.

Let x be the user's original data. x^{adv} is the adversarial example based on x. The Euclidean distance $d(x^{\text{adv}}, x)$ is used to measure the adversarial distortion between the adversarial example and the original data. The distance threshold ϵ is used as the upper bound of the adversarial distortion as inequation (1). We chose the adversarial example, whose adversarial distortion is smaller than the threshold ϵ, to construct the uploading training dataset. We introduce the perturbation rate α to control the degree of adversarial perturbation. N denotes the size of the uploading dataset. The size of adversarial examples which the generator needs to synthesize is $\alpha \times N$. When $\alpha = 1$, it means that all the upload training data is the generated adversarial examples. The value of ϵ can be set by experiences.

$$d\left(x^{\text{adv}}, x\right) = L_2\left(x^{\text{adv}}, x\right) \leq \epsilon. \tag{1}$$

We aim to maintain the performance of the simulator. Thus, we exploit the method to maintain the same classification label between the original data x and the perturbed data x^{adv} proposed by Jia et al. [8]. The formulation can be described as in equation (2). Let $f: X \longrightarrow Y$ be the classification function of the simulator. Each output $f(x)$ is a confidence vector. The classification label of original data and adversarial data can be represented as $\arg\max f(x)$ and $\arg\max f(x^{\text{adv}})$, respectively.

$$\arg\max f\left(x^{\text{adv}}\right) = \arg\max f(x). \tag{2}$$

Let X^{adv} be perturbed uploading dataset. Y is the label set of original examples X. Let $x\{\hat{\_}i\text{adv}\}$ indicate the ith training

examples. The cross-entropy loss function of simulator is given in equation (3). We aim to minimize the loss function of the simulator, as equation (4) shows, to guarantee that the mitigation causes the minimum influence to the simulator.

$$L\left(Y, f\left(X^{\text{adv}}\right)\right) = \sum_{i=1}^{N} y_i \log f\left(x_i^{\text{adv}}\right) + (1 - y_i)\log\left(1 - f\left(x_i^{\text{adv}}\right)\right), \tag{3}$$

$$\min\left(L\left(Y, f\left(X^{\text{adv}}\right)\right)\right). \tag{4}$$

Let g be the MI evaluator's model. The optimization problem for the MIA is maximizing the loss function of the evaluator, as in the following equation:

$$\max\left(\frac{1}{N}\sum_{i=1}^{N} L\left(y_i^{\text{label}}, g\left(f\left(x^{\text{adv}}\right)\right)\right)\right). \tag{5}$$

5. Implementation and Evaluation

In this section, we present details of implementation, including the adversarial algorithms, models, and datasets. Then, we give out experiment results and analysis of our defense framework.

5.1. Adversarial Algorithms

5.1.1. Generative Adversarial Networks- (GANs-) Based Method. GANs-based method generates adversarial examples with generative adversarial networks (GANs) [41]. As the GANs can learn and approximate the distribution of original records, the GANs-based method can generate perturbation efficiently. G is the generator of AdvGAN. f denotes the target model. Feed the original instance into generator G, and produce perturbation $G(x)$. Then, input $x + G(x)$ to discriminator D. D aims to encourage generator G to synthesize instance which is indistinguishable from the original data's classification result.

$$f(x) = f(x + G(x)). \tag{6}$$

5.1.2. Fast Gradient Sign Method (FGSM). FGSM was first proposed by Goodfellow et al. [34]. θ denotes the model parameters. x is the original data. y is the real label of the original data x. $J(\theta; x; y)$ is the loss function used to train the model with respect to the inputs. ε is the distortion parameter that controls the perturbation level.

$$\eta = \varepsilon \text{sign}\left(\nabla_x J(\theta; x; y)\right). \tag{7}$$

The adversarial example x_{adv} can be expressed as

$$x_{\text{adv}} = \eta + x. \tag{8}$$

FGSM is a fast and reliable method to generate adversarial examples which cause more noticeable perturbation compared to other adversarial algorithms. This attribute can be used to protect the original data from the direct content exposed attack.

5.1.3. OPTMARGIN-Based Defense. He et al. [42] proposed the OPTMARGIN adversarial method, which can generate low-distortion adversarial examples that are robust to small perturbations. The OPTMARGIN method creates a surrogate model of the region classifier, which classifies a smaller number of perturbations. f is the point classifier; v_i is perturbation applied to the original data x. $f_i(x) = f(x + v_i)$; they define a loss term for each model:

$$l_i(x') = l(x' + v_i). \tag{9}$$

The minimization problem of one $l(x')$ is

$$\text{minimize } \|x' - x\|_2^2 + c \cdot (l_1(x') + \cdots + c \cdot (l_n(x'). \tag{10}$$

5.2. Datasets

MNIST (http://yann.lecun.com/exdb/mnist/): we use MNIST [43] for our experiments, which consists of 60000 training images and 10000 test images, all drawn from the same distribution. All these black and white digits are size-normalized and centered in a fixed-size image, where the center of gravity of the intensity lies at the center of the image with 28×28 pixels.

CIFAR10 (https://www.cs.toronto.edu/kriz/cifar.html): the CIFAR10 dataset consists of 60000 32×32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images.

Data separation: in our experiments, we randomly select 5000 records X as the original instances used to generate adversarial examplesX_{adv} and 5000 records X_{test} as test data for training the simulator. We randomly select the other 5000 X_{infer} for the inference evaluator. The evaluated data is expressed as $X \cup X_{\text{infer}}$.

5.3. Performance of Our Defense Framework. Table 2 shows the performance of our defense framework against MIA. We analyze the training accuracy and test accuracy of the target model and the attack accuracy of IM model under two kinds of public datasets (CIFAR10 and MNIST). In this experiment, we exploit the AdvGAN algorithm. We find the following: (1) our defense framework can restrain the inference accuracy. Under the CIFAR10, our mitigation reduces the attack accuracy from 93.71% to 51.4%, while with the MNIST, our defense decreases the attack accuracy from 89% to 52.5%. As these attacks nearly randomly guess probability (50%), the mitigation was proven to be effective. (2) Our defense can improve the test accuracy of the target classifier. It is interesting to see that even though the training accuracy is declined after applying our defense, the test accuracy of the target is promoted. Specifically speaking, the

Input: user's original data X for remote training; the test dataset X_{test} for evaluation; α is the perturbation rate;
X_{adv} is the dataset constructed by the adversarial examples which synthesized by the generator;
N is the size of X;
? is the Euclidean distance threshold between the adversarial example and the original data. m is the maximize number of adversarial perturbation rounds.
Output: h, X_{adv}
init the adversarial model h, the simulator's model f, the evaluation model g;
$S = \alpha \times N$;
random choose $(S, X) \longrightarrow X_{\text{choosed}}$;
$X = X_{\text{choosed}} \cup X_{\text{rest}}$;
$X_{\text{adv}} = \varnothing$
for($i = 0$; $i < m$; i++){//to generate the adversarial examples
  Train(h);
  for($j = 0$; $j < S$; j++){
    $h(x^j_{\text{gen}}) \longrightarrow X^j_{\text{gen}}$;
    for($k = 0$; $k <$ size(X^j_{gen}); k++) {
      $L_k = \infty$;
      $d_k = d(x^{jk}_{\text{gen}}, x^j_{\text{choosed}})$//Euclidean distance
      if($d_k \le$?){
        $l_k = \text{loss}(y^j_{\text{choosed}}, f(x^{jk}_{\text{gen}}))$
        if($l_k < L_k$){
          $L_k = l_k$
          $x_{\text{return}} = x^{jk}_{\text{gen}}$
        }
      }
    }
    $x_{\text{return}} \longrightarrow X_{\text{gen}}$
    if(cross entropy(Y^j_{gen}, fX^j_{gen}) > cross entropy $(Y_{\text{adv}}, f(X_{\text{adv}}))$
    $x_{\text{adv}} \longrightarrow X^j_{\text{gen}}$
  }
Return h, X_{adv}

Algorithm 1: Synthesis of the uploading data.

Table 2: Performance of our defense framework.

Without defense	Train acc. (%)	Test acc. (%)	Attack acc. (%)	With defense	Train acc. (%)	Test acc. (%)	Attack acc. (%)
CIFAR	10 96.1	62.24	93.71	Defense framework (AdvGAN)	78.3	69.7	51.4
MNIST	100	57.9	89	Defense framework (AdvGAN)	91.66	91.12	52.5

test accuracy with CIFAR10 increases from 62.24% to 69.7%, while the test accuracy with MNIST increases from 57.9% to 91.12%.

5.4. Comparison with Existing Methods. We illustrate the comparison of our method with existing methods including min-max [7] and differential privacy [16]. We use CIFAR 10 as the dataset; and we use the AdvGAN algorithm and set $\epsilon = 50$ in differential privacy mitigation; the distance of our method is 23. Table 3 demonstrates that all the three defenses can restrain the MIA accuracy close to 50%, after 50 epochs. Among these three mitigations, the inference accuracy under differential privacy is the lowest. However, the training accuracy and the test accuracy are just at 1.2% and 1%, respectively. Under the min-max method, the inference accuracy, the training accuracy, and the test accuracy are 52.9%, 68.6%, and 62.7%, respectively. Under our method, the training accuracy and the test accuracy are 51.94%, 78.3%, and 69.7%. Our method achieves MI privacy with the minimum utility cost of the target classifier.

5.5. Different Adversarial Algorithm-Based Privacy-Preserving Capability Evaluation

5.5.1. Protect Content Disclosed Attack. We visualize MNIST outputs with different adversarial algorithms with our defense framework. As demonstrated in Figure 3, the AdvGAN-based method produces the clearest records compared to the FGSM and OPTMARGIN-based ones. Examples synthesized by the FGSM-based method are more noticeable, causing the content of these outputs to be unrecognizable. This attribute can be used to protect the sensitive

Table 3: Comparing our method with existing mitigation.

Defenses	Train acc. (epoch = 50) (%)	Test acc. (epoch = 50) (%)	Inference acc. (epoch = 50) (%)
Min-max (CIFAR 10)	68.6	62.7	52.9
Differential privacy (epsilon = 50 CIFAR 10)	1.2	1	50.00
Framework-AdvGAN (distance = 23 CIFAR 10)	78.3	69.7	51.94

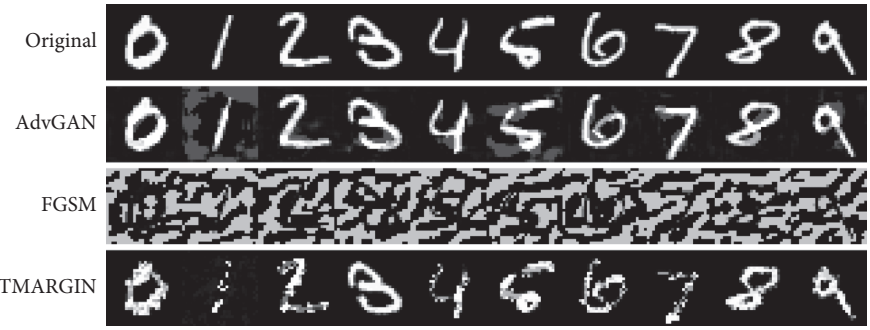

Figure 3: Visualization of adversarial outputs. We visualize outputs with different adversarial algorithms with our defense framework.

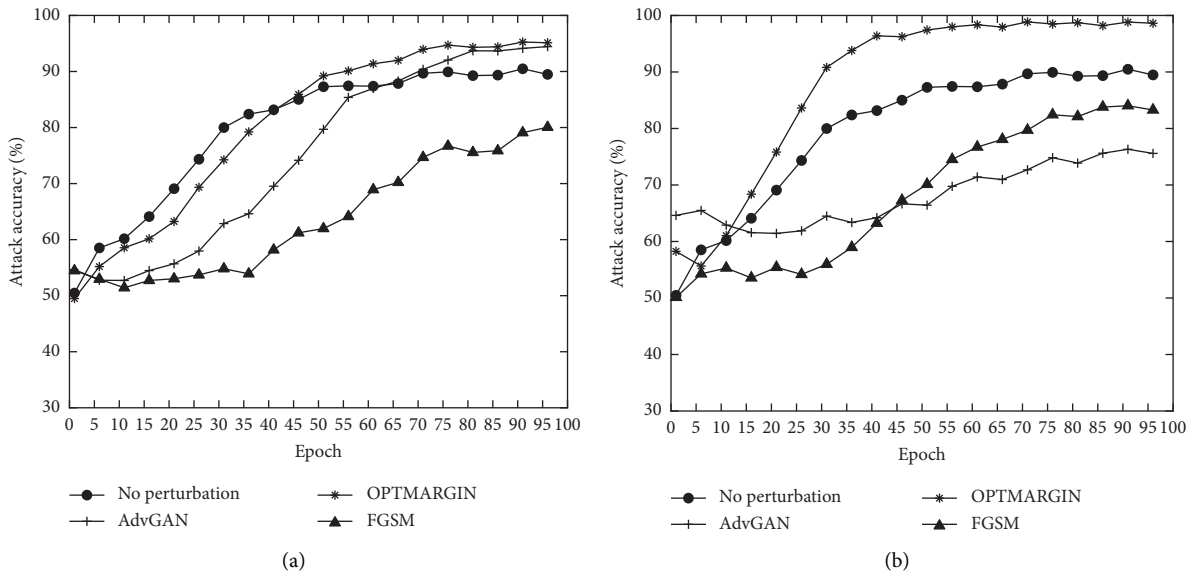

Figure 4: The defense performance with the different adversarial methods. (a) Without constraint conditions. (b) With constraint conditions.

information from the direct content disclosed risks coming from the curious model provider platform.

5.5.2. Privacy-Preserving. Figure 4 shows the MIA accuracy of various adversarial construction methods: AdvGAN, OPTMARGIN, and FGSM. In Figure 4(a), the adversarial instances are generated without constraint conditions introduced in 4.2. We run this experiment with MNIST and set the distance threshold ϵ as 100. We let the distortion rate α be 100%; in other words, we train the target model with 100% adversarial instances. In Figure 4(b), the adversarial examples are synthesized by our defense framework (with constraint conditions introduced in 4.2). The result indicates that our constrained conditions can optimize the defense performance of the AdvGAN- and FGSM-based mitigation. However, the defense effectiveness of OPTMARGIN-based defense is not satisfying. The reason might be that the parameter of our defense framework should be adjusted with a different algorithm. The relationship between parameters and the defense performance will be further discussed in 5.6.

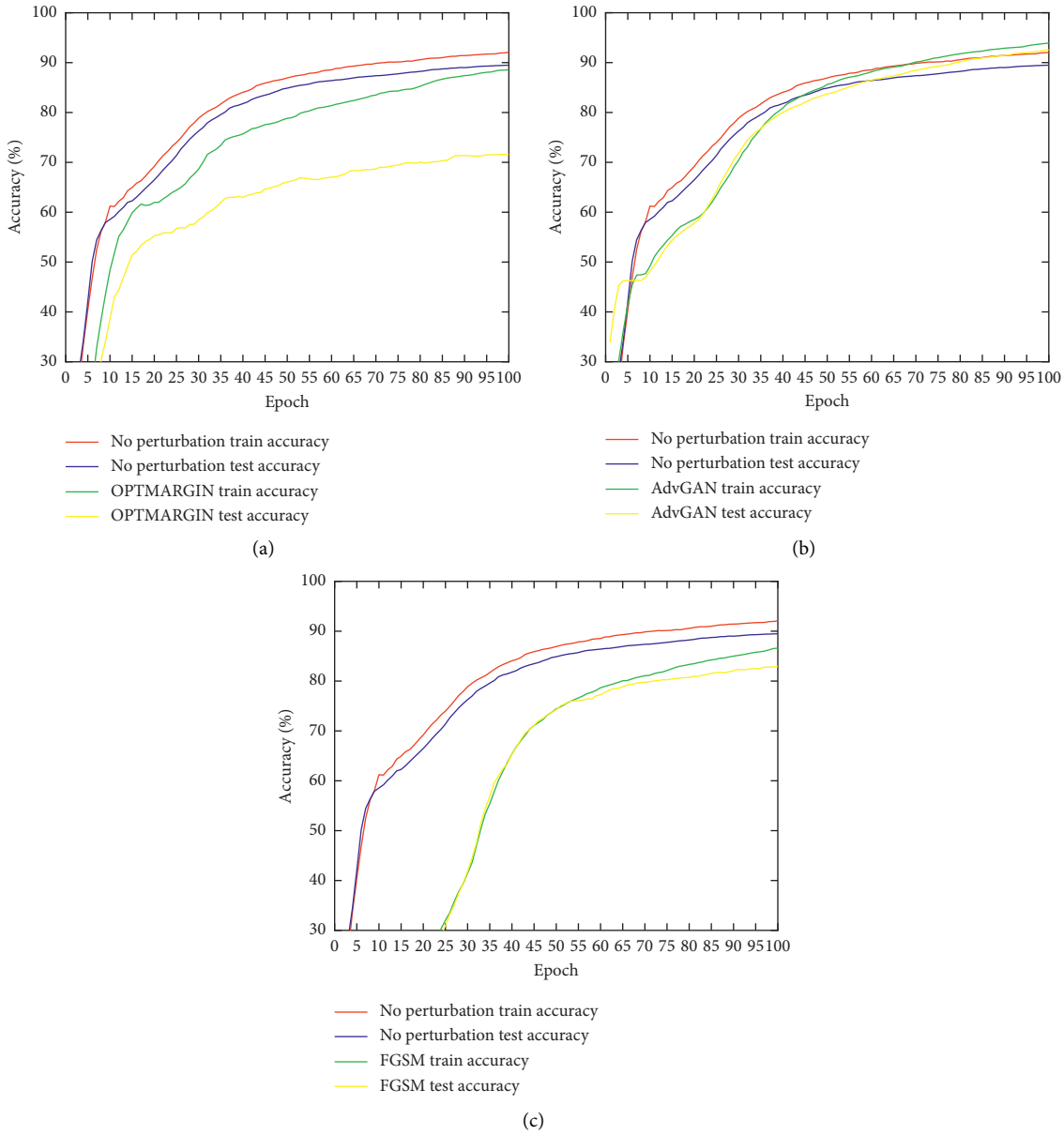

Figure 5: Comparing the training/test accuracy of original training data (no perturbation) with adversarial perturbation training data (OPTMARGIN, AdvGAN, and FGSM). (a) OPTIMARGIN perturbation with constraint condition. (b) AdvGAN perturbation with constraint condition. (c) FGSM perturbation with constraint condition.

5.5.3. Evaluation of Privacy and Utility. In Figure 5, we compare the training accuracy and test accuracy of the target classification model with different adversarial algorithms. We run this experiment with MNIST and set the distance threshold ϵ as 100. We let the distortion rate α be 100%; in other words, we train the target model with 100% adversarial instances. As we can see, the OPTIMARGIN method has the lowest test accuracy, but its training accuracy is higher than that of FGSM. The AdvGAN algorithm has the highest training accuracy and test accuracy after 100 epochs; these values are even bigger than the value without defense. The FGSM algorithm has the coincident training accuracy and test accuracy, and, after 100 epochs, values are close to the without defense ones.

5.6. Analyze the Influence Factors. We analyze the influence factors of our generic defense framework, including Euclidean distance and perturbation rate. We exploit the AdvGAN as the adversarial algorithm, and the original training data is MNIST. We set the training size to 5000 and the shadow model number is 5.

5.6.1. Distance Threshold. Figure 6 shows that, under different Euclidean distance threshold, privacy risks vary. The perturbation rate = 100%. We output one MIA accuracy for every five epochs in the 100 epochs. As shown in Figure 6, when the distance = 50, after 45 epochs, the attack accuracy area converges and is between 52% and 54%. When the

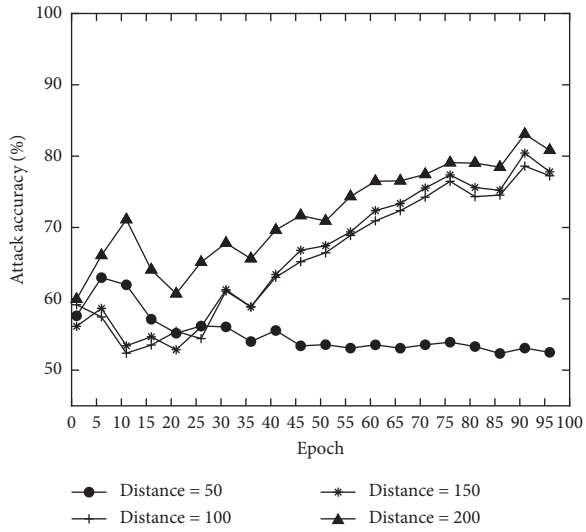

Figure 6: MIA accuracy under different Euclidean distance.

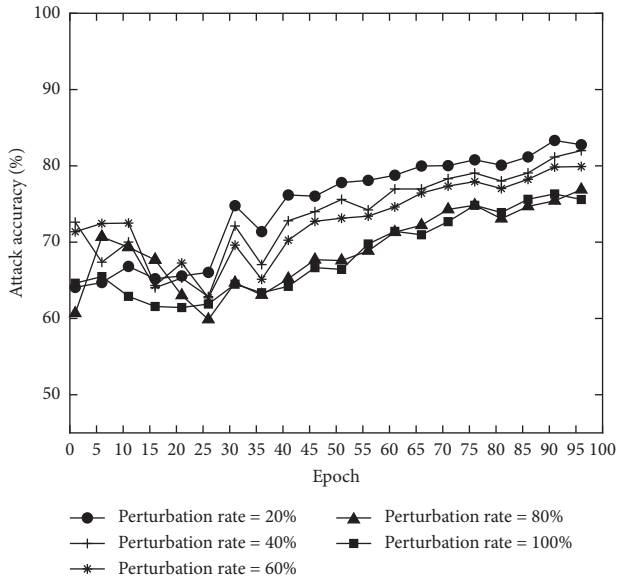

Figure 7: MIA accuracy under different adversarial perturbation rate.

distance = 200, 150, and 100, attack accuracy increases gradually between the 1st and 10th epochs, decreases between the 10th and 20th epochs, and increases slowly thereafter. According to Figure 6, the greater the Euclidean distance, the higher the MIA accuracy, the greater the risk of a privacy breach. In Euclidean distance, the smaller the distance, the better the defense. In particular, a value of 50 is the best defense against Mia, accuracy close to the random selection result (50%).

5.6.2. Distortion Rate. The Euclidean distance = 100. Figure 7 shows that we output one MIA accuracy for every five epochs in the 100 epochs. As shown in Figure 7, MIA accuracy is lower when perturbation rate = 100% compared to when distortion rate = 80%, 60%, 40%, and 20%. That is, the least amount of private information is disclosed. MIA accuracy is the highest when the perturbation rate is 20%. It means that the most private information is leaked. As you can see from Figure 7, the greater the perturbation rate is, the less privacy is exposed. The smaller the distortion rate is, the more privacy is exposed.

6. Conclusion and Future Work

We have designed and implemented a defense framework for privacy-preserving in remote machine learning services. We aim at devising a framework that not only can mitigate the privacy risk, that is, MIAs, but also can protect the training data from direct content exposed attacks, with the different adversarial algorithm. The evaluation result shows that our defense framework with the AdvGAN method is effective against MIA and our defense framework with the FGSM method can protect the sensitive original data from direct content exposed attacks. In addition, our method can achieve more privacy and utility compared to the existing method, for instance, the min-max method and the differential privacy. Furthermore, the experiment results show that we can control the defense performance with parameters such as distortion rate and distance threshold.

In this paper, we just evaluate the performance of our defense framework against MIA. Thus, exploring our defense framework against other privacy threats, such as model inversion, is left as an open question. We should do more comprehensive investigating and refining our countermeasures for future work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[2] A. Hannun, C. Case, J. Casper et al., "Deep speech: scaling up end-to-end speech recognition," 2014, https://arxiv.org/abs/1412.5567.

[3] A. Esteva, A. Robicquet, B. Ramsundar et al., "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, 2019.

[4] B. Krollner, B. J. Vanstone, and G. R. Finnie, "Financial time series forecasting with machine learning techniques: a survey," in *Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN 2010)*, Bruges, Belgium, April 2010.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models,"

in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, IEEE, San Jose, CA, USA, May 2017.

[6] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: evaluating information leakage of generative models using generative adversarial networks," 2018, https://arxiv.org/abs/1705.07663.

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

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

[9] Y. Wang, C. Wang, Z. Wang et al., "Mcmia: model compression against membership inference attack in deep neural networks," 2020, https://ui.adsabs.harvard.edu/link_gateway/2020arXiv200813578W/arxiv:2008.13578.

[10] F. Mirshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh, "Privacy in deep learning: a survey," 2020, https://arxiv.org/abs/2004.12254.

[11] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*, pp. 601–618, Austin, TX, USA, August 2016.

[12] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, Denver, CO, USA, October 2015.

[13] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing," in *Proceedings of the 23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pp. 17–32, San Diego, CA, USA, August 2014.

[14] Y. Long, V. Bindschaedler, L. Wang et al., "Understanding membership inferences on well-generalized learning models," 2018, https://arxiv.org/abs/1802.04889.

[15] A. Salem, Y. Zhang, M. Humbert et al., "Model and data independent membership inference attacks and defenses on machine learning models," 2018, https://arxiv.org/pdf/1806.01246.pdf.

[16] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *Proceedings of the 28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 1895–1912, Santa Clara, CA, USA, August 2019.

[17] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *Proceedings of the 30th {USENIX} Security Symposium ({USENIX} Security 21)*, Vancouver, Canada, August 2021.

[18] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "On the connection between differential privacy and adversarial robustness in machine learning," *Stat*, vol. 1050, p. 9, 2018.

[19] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," 2017, https://arxiv.org/abs/1712.09136.

[20] T. Salimans and D. P. Kingma, "Weight normalization: a simple reparameterization to accelerate training of deep neural networks," 2016, https://arxiv.org/abs/1602.07868.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310–1321, Denver, CO, USA, October 2015.

[23] C. Dwork, "Differential privacy: a survey of results," in *Proceedings of the International Conference on Theory and Applications of Models of Computation*, pp. 1–19, Springer, Changsha, China, May 2008.

[24] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 371–380, Bethesda, MD, USA, May 2009.

[25] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: privacy via distributed noise generation," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology–EUROCRYPT 2006*, pp. 486–503, Springer, St. Petersburg, Russia, May-June 2006.

[26] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60, IEEE, Vancouver, Canada, December 2010.

[27] C. D, work and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[28] S. Rahimian, T. Orekondy, and M. Fritz, "Differential privacy defenses and sampling attacks for membership inference," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), PriML Workshop (PriML)*, vol. 13, Vancouver, Canada, December 2019.

[29] A. Triastcyn and B. Faltings, "Generating differentially private datasets using gans," 2018, https://www.arxiv-vanity.com/papers/1803.03148/.

[30] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," 2016, https://arxiv.org/abs/1610.05755.

[31] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: measuring unintended neural network memorization & extracting secrets," 2018, https://arxiv.org/abs/1802.08232.

[32] Z. Yang, B. Shao, B. Xuan, E.-C. Chang, and F. Zhang, "Defending model inversion and membership inference attacks via prediction purification," 2020, https://arxiv.org/abs/2005.03915.

[33] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE, San Jose, CA, USA, May 2017.

[34] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, https://arxiv.org/abs/1412.6572.

[35] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, https://arxiv.org/abs/1607.02533.

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, https://arxiv.org/abs/1706.06083.

[37] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, Honolulu, HI, USA, July 2017.

[38] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, Las Vegas, NV, USA, June 2016.

[39] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, IEEE, Saarbrucken, Germany, March 2016.

[40] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," 2017, https://arxiv.org/abs/1710.11342.

[41] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," 2018, https://arxiv.org/abs/1801.02610.

[42] W. He, B. Li, and D. Song, "Decision boundary analysis of adversarial examples," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, April-May 2018.

[43] Y. LeCun, "The MNIST database of handwritten digits," 1998, http://yann.lecun.com/exdb/mnist/.