Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5534270, 17 pages
https://doi.org/10.1155/2021/5534270

Research Article

User-Level Membership Inference for Federated Learning in Wireless Network Environment

Yanchao Zhao,[1] **Jiale Chen**,[1] **Jiale Zhang**,[2] **Zilu Yang**,[1] **Huawei Tu**,[3] **Hao Han**,[1] **Kun Zhu**,[1] **and Bing Chen**[1]

[1]*Collage of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China*
[2]*School of Information Engineering, Yangzhou University, China*
[3]*Computer Science & Information Technology, La Trobe University, Australia*

Correspondence should be addressed to Yanchao Zhao; yczhao@nuaa.edu.cn

Received 13 January 2021; Accepted 2 October 2021; Published 20 October 2021

Academic Editor: Jun Cai

With the rise of privacy concerns in traditional centralized machine learning services, federated learning, which incorporates multiple participants to train a global model across their localized training data, has lately received significant attention in both industry and academia. Bringing federated learning into a wireless network scenario is a great move. The combination of them inspires tremendous power and spawns a number of promising applications. Recent researches reveal the inherent vulnerabilities of the various learning modes for the membership inference attacks that the adversary could infer whether a given data record belongs to the model's training set. Although the state-of-the-art techniques could successfully deduce the membership information from the centralized machine learning models, it is still challenging to infer the member data at a more confined level, the *user level*. It is exciting that the common wireless monitor technique in the wireless network environment just provides a good ground for fine-grained membership inference. In this paper, we novelly propose and define a concept of user-level inference attack in federated learning. Specifically, we first give a comprehensive analysis of active and targeted membership inference attacks in the context of federated learning. Then, by considering a more complicated scenario that the adversary can only passively observe the updating models from different iterations, we incorporate the generative adversarial networks into our method, which can enrich the training set for the final membership inference model. In the end, we comprehensively research and implement inferences launched by adversaries of different roles, which makes the attack scenario complete and realistic. The extensive experimental results demonstrate the effectiveness of our proposed attacking approach in the case of single label and multilabel.

1. Introduction

With the revolution of decentralized machine learning, researches on collaborative learning technologies such as federated learning for resource-constrained devices on mobile edge networks [1] have been increasing and expanding the landscape of use cases. Federated learning [2] enables mobile devices to collaboratively learn a shared prediction model while keeping all the training data locally instead of in the cloud, which may be at risk of privacy leakage. Unlike other collaborative learning frameworks, federated learning updates a global model by aggregating all local parameters from participants, so that the federated model can benefit from a wide range of non-IID [3] and unbalanced data distribution among diverse participants. In keeping with the vigorous development of 5G network technology, as demonstrated in Figure 1, federated learning empowered by wireless networks stimulates the potential of some applications around us, making them smarter and more powerful. Furthermore, edge wireless networks for content caching and data calculation are a promising way to reduce backhaul traffic load. Federated learning, based on a model that uses local training

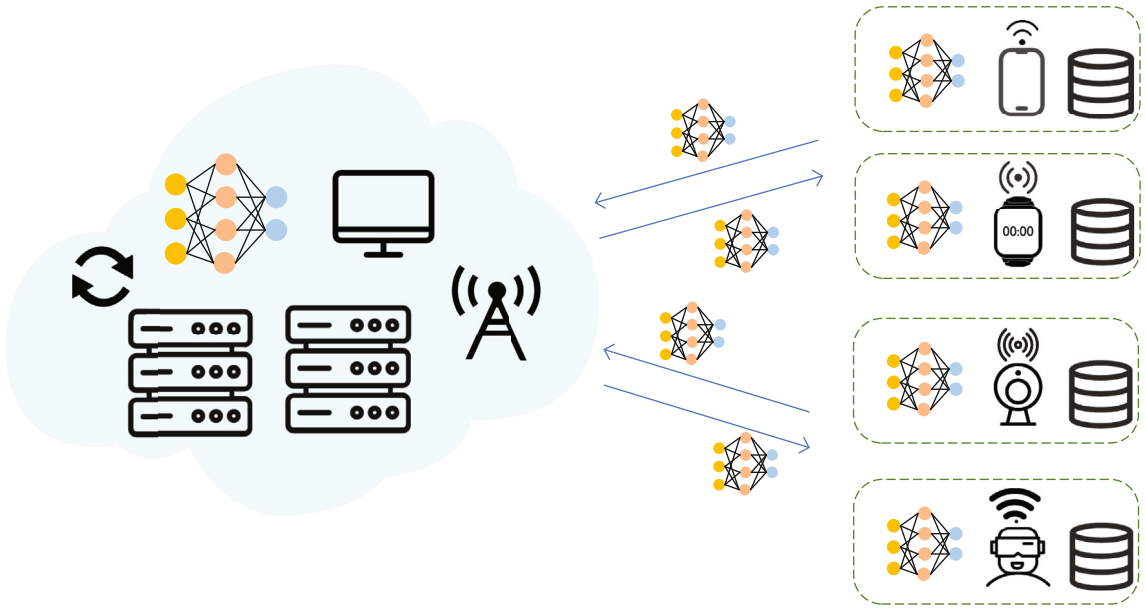

Figure 1: Federated learning for mobile networks.

instead of direct access to participants' data, seems like a perfect fit for content popularity prediction in proactive caching in wireless networks [4].

Although federated learning can provide a basic privacy guarantee with localized training, privacy issues still exist during the aggregation and communication process. Emerging attacking methods, including membership inference, have been undermining the security of federated learning and even the entire artificial intelligence technology. Basically, the membership inference problem is a classification problem that the adversary needs to tell whether the data with unknown ownership is part of a certain collection or not. Although this is an indirect privacy stealing, when membership inference attacks are used as preattacks for other attack scenarios, such as the reconstruction attack [5], the membership information makes these attacks more targeted and disruptive. Shokri et al. [6] first proposed the membership inference against a black-box machine learning API. In this case, the adversary can construct a "shadow model" by obtaining the fluctuation difference in the confidence of the black-box output of similar data obtained from the target model (e.g., "MLaaS" platform), thereby approximating the behavior of the target model to spy the privacy of the training set. In this way, the adversary does not need to get the internal structure and parameters of the target model. However, this attack has many assumptions that the adversary can directly call the API, which is equivalent to the adversary's unauthorized access to the model, and use it in a centralized learning environment. Moreover, the adversary has a dataset from the same distribution as the target model's training data.

For the recent researches on the security issues of artificial intelligence, Salem et al. [7] improved Shokri et al.'s method by containing multiple neural network models in a stack, which is sensitive to the membership information. In this way, the attack model can only focus on the relationship between the membership information and the classification results, even if the data is from different distributions. Nasr et al. [8] proposed a membership inference attack launched from the participant side. The core technique of the scheme was the stochastic gradient ascent (SGA). The adversary extracted the parameters of the target model during the training process, including gradients and loss rates, into fully connected layers to train the neural network. When the gradients of data are forced to increase by SGA every time, the gradients of member data will be forcibly decreased by the stochastic gradient descent (SGD) [9], while the gradients of nonmember data still rise. By detecting this distinction, the membership information is transformed into a score, which is used as a new feature to construct unsupervised learning to distinguish member data from nonmember data.

However, although the above inference attacks can reveal the privacy of training data to varying degrees, there are some limitations when they are transplanted to federated learning. Firstly, in the previous centralized learning, the dataset used to train the attack model had the same distribution as the dataset belonging to the target model, and even these datasets have a certain proportion of intersection. Secondly, there is no research on the possible existence of malicious participants launching the membership inference, which is divorced from the real situation. Usually, the prerequisite for the successful operation of federated learning is that all participants and servers are honest, but as long as there are malicious members in the cluster, such as doing poisoning attacks, this perfect mechanism will be out of balance. After all, no one can guarantee that federated learning is always perfect. Thirdly, even if an attack is to be launched from the client side, the inherent privacy protection mechanism of federated learning, aggregated algorithm, will prevent it from succeeding. Fortunately, in a wireless environment, with a wireless monitor mode, the situation has changed a lot. Motivated by the function of the wireless

monitor and those shortcomings in existing inference approaches, we give a deep analysis of active and targeted membership inference attacks in the federated learning with a white-box access model. We are established on the wireless network environment to implement all the elements from the perspective of a malicious participant and a malicious central server. We name our scheme as the *user-level* membership inference. The reason why we call it "*user-level*" is that we have refined the target of inference from the previous global model to a certain participant (*victim*), caring more about his membership privacy. We try to play two roles of malicious participant and server, respectively, in the federated learning mode to launch inference (see Figure 2). Based on the traditional membership inference in centralized and distributed learning, we take a more practical threat assumption that the adversary does not need to know any prior knowledge about the training datasets. Stuck by the model averaging algorithm and the lack of training data for the membership inference, we make full use of the characteristics of the wireless monitor to further propose a local-deployed data augmentation method relying on the generative adversarial networks (GANs) to generate high-quality fake samples.

Our contributions in this paper can be summarized as follows.

(i) *User-Level Membership Inference*. We further disclose the security hole of the current federated learning enabled by 5G wireless networks with novelly launching fine-grained membership inference attacks and encouraging more researches on preventing participants from leaking privacy.

(ii) *Data Augment Using GANs*. To gain insight into the data distribution of other participants to perform the membership inference, we use the information obtained by a wireless monitor and innovatively develop local-deployed generative adversarial networks (GANs) to generate samples with all labels.

(iii) *Systematic Analysis in All Positions*. We progressively launch membership inference from a malicious participant in federated learning. In addition, we investigate and validate the possibility of launching inferences from the server side.

(iv) *Excellent Performances in Experiment*. In experiments, we set two major indexes, the accuracy of the membership inference and the learning task, to measure the effectiveness of our scheme. We also performed multiple sets of comparative experiments to prove the impact of the number of labels on the membership inference attack.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 briefs some background knowledge and introduces the threat model. Section 4 describes the proposed method framework along with an analysis of the membership inference. The performance evaluation results are presented in Section 5. Section 6 discusses the limitations of our method and gives some ideas. Finally, Section 7 concludes this paper.

2. Related Work

In this section, we will introduce the privacy protection methods for distributed deep learning and federated learning. After that, we will refine the issue of privacy leakage to the membership inference attack. Finally, we present the various attacks against a specific victim in the federated learning scenario.

2.1. Privacy-Preserving Distributed Learning and Federated Learning. The traditional centralized machine learning, where the data holder trains the model locally, is limited by the computing resources and data volume. It is difficult to meet the current needs for massive data calculations, data diversity, and storage performance. As a result, the distributed learning framework emerges, providing a collaborative training scenario. But once the third party involves, there will be a problem of privacy leakage. To protect distributed learning, an algorithm named as distributed selective stochastic gradient descent (DSSGD) was proposed by Shokri and Shmatikov [10]. The results showed that even if only 1% of the parameters are shared, collaborative learning will bring a higher accuracy than centralized learning. Moreover, Shokri and Shmatikov [10] utilized differential privacy [11] to effectively prevent data privacy that may be indirectly leaked. Based on the previous article, Phong et al. [12] proposed four cases of indirect privacy leakage and pointed out that even if some gradients are uploaded randomly, there are still significant hidden privacy risks. The author introduced homomorphic encryption technology [13] in the large-scale distributed neural network to ensure that the cloud server cannot steal the privacy of data during the entire process of model training. The only drawback was computationally expensive and time-consuming. Zhang et al. [14] present BatchCrypt, which is a system solution for cross-silo federated learning. In the scheme, they encode a batch of quantized gradients into a long integer and encrypt it in one go, instead of encrypting individual gradients with full precision. BatchCrypt substantially reduces the communication overhead caused by homomorphic encryption.

The difference between collaborative learning and federated learning is that the central server of federated learning will average the updates (i.e., the weight matrix) after each communication round. Even so, the privacy violation remains a challenge. In the user-level differential privacy algorithm proposed by [15], this average is changed and approximated using a random mechanism. This is done to hide the contributions of individual participants in the collection, thereby protecting the entire distributed learning process. Truex et al. [16], in order to compensate for the impact of differential privacy on model accuracy, combining differential privacy with secure multiparty calculations, reduced the noise injection caused by the increase in the number of participants and maintained the accuracy and privacy of the model. Inspired by these efforts, we began to focus on privacy preserving in federated learning.

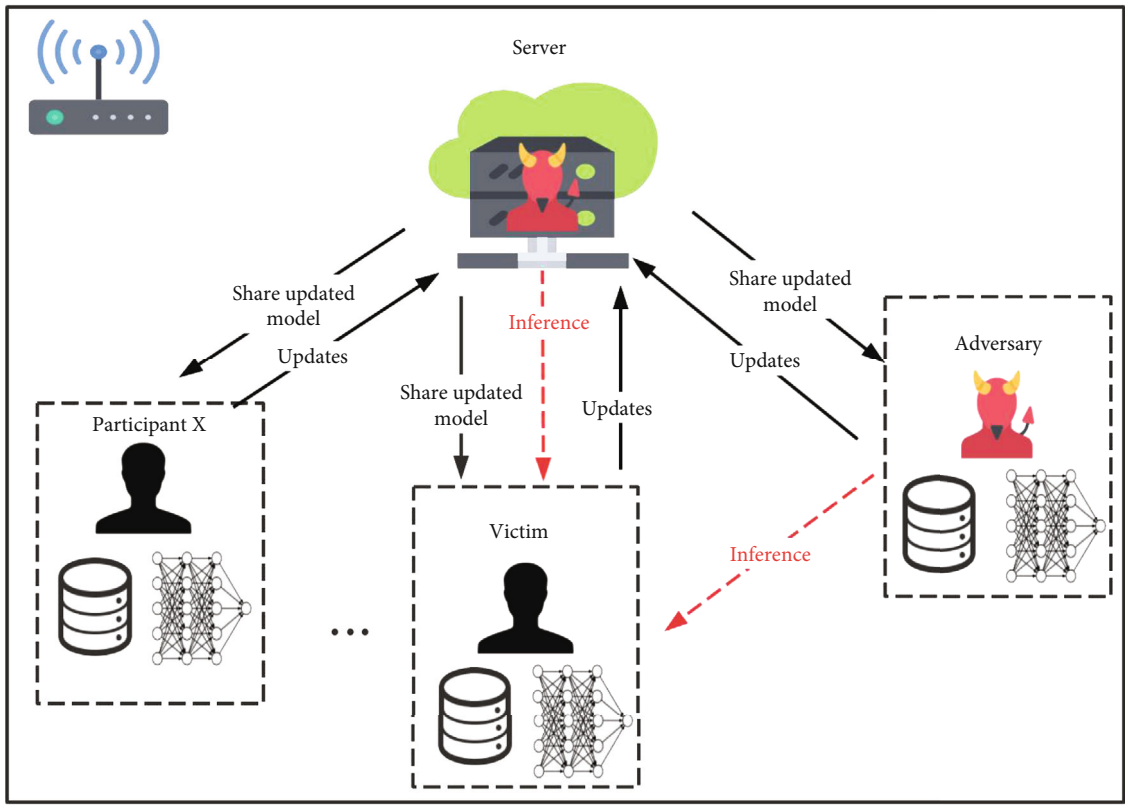

Figure 2: Multiple membership inference methods in federated learning under wireless network.

2.2. Membership Inference Attack. The membership inference attack means that when a record is given to the inference model, the model can tell whether the record belongs to a target's training set. As centralized learning evolves to distributed learning, there are many variants of the membership inference, which can be divided into active attacks and passive attacks, including those launched by a malicious server and by malicious participants [8]. Not surprisingly, the more participants are involved, the less information that adversary can learn from another participant. In other words, the accuracy of the membership inference attack will decrease as the number of participants increases. Taking into the situation of numerous participants, the active local adversaries are facing challenges of lacking training data. Besides, the research found that even a model with differential privacy protection still has the risk of leaking membership privacy [17]. The main reason for the leakage of membership information is model overfitting [18–20]. At present, membership inference has become the third largest attack against AI systems, accounting for 3.5%. Our work focuses on membership inference in federated learning. But what needs to be distinguished from the membership inference mode under centralized learning is that the goal of our proposed method under federated learning is to infringe the privacy of a certain participant, not the privacy of the entire training dataset. We summarize this more fine-grained analysis as the membership inference attack under federated learning. This takes the membership inference a step further in the field of study.

2.3. Attacks against a Specific Victim in the Collaborative and Federated Learning. In addition to stealing membership information, there are many attacks on a certain participant in federated or centralized learning. These attacks, for example, the poisoning attack [21], the model inversion attack [22], the representative inference [23], the model stealing attack [24], and the capturing of extra properties [25], mainly assume that the adversary, whether a malicious server or a malicious participant, actively launches attacks and tries to induce the victim to output more private information to achieve the purpose. However, attacks from the client side in federated learning are limited to recovering class-wised representatives rather than mining user-level privacy because the malicious participant can only access updates aggregated by the server (contributed by all the participants). Therefore, to launch these attacks, more auxiliary information is often required, e.g., class labels or other participant-wise properties. Our method cuts into the crux of this nodus from the wireless monitor technique and the generative adversarial networks (GANs).

3. Preliminaries

In this section, we introduce the background knowledge and the other preliminaries, including assumptions and the threat model of our method.

3.1. Federated Learning. Federated learning [26] is a distributed deep learning solution first proposed by Google in

2016. In the selection phase of the federated learning, the server will randomly and partly select participants to participate in this round of training. In the reporting phase, the server will wait for each participant to return the trained gradient parameters. After the server receiving parameters, it will use an algorithm to aggregate them and notify participants of the next request time. If there are enough participants returning gradients before the timeout, this round of training is successful; otherwise, it fails. In the entire system, there is a pace control module (Pace Steering), which can manage the connection of all the participants. For the small-scale federated learning training, Pace Steering guarantees that sufficient participants are involved in each round of training. For the large-scale federated learning training, Pace Steering will randomize the request time of the participants to avoid a large number of simultaneous requests, which may cause problems. By the way, the models trained by each participant do not interfere with each other during the training process.

In 2017, Google's Mcmahan et al. proposed the FedAvg algorithm, which is a synchronous protocol [27]. The updates are averaged and accumulated to the current shared model. Equation (1) demonstrates the process. M_t denotes the shared model at the tth iteration, M_{t+1} means the newest model, and u_t^k indicates the update from the kth client at iteration t:

$$M_{t+1} = M_t + \frac{1}{N}\sum_{k=1}^{N} u_t^k. \tag{1}$$

All participants execute Equation (2) in each epoch, where η is the learning rate and b means the batch. Finally, every participant returns his w, weights, to the server:

$$W = W - \eta \nabla L(W; b). \tag{2}$$

On the one hand, federated learning can effectively enrich the diversity of training sets and allow more data to participate in calculations. On the other hand, federated learning allows data to be stored locally, which meets some data-sensitive requirements, such as medical and military scenarios. But this does not mean that privacy will not be a problem in federated learning. Inference against a certain participant's data and output greatly threatens the security of the federated learning.

3.2. Wireless Monitor Mode. By default, the wireless network card and the wireless access point (WAP) [28] are in a managed mode after establishing a connection. In this mode, the wireless network card only serves to receive data sent to itself from the WAP. If you want the wireless network card to monitor all communication information in the wireless environment, you can set the wireless network card to monitor mode (also called RFMON mode [29]) and then use software such as Wireshark to capture data packets for analysis, as shown in Figure 3. Under the normal setting, the network card will only accept data packets sent to itself and discard all other packets. Of course, the network card can accept all the messages unconditionally; this is the so-called promiscuous mode. However, unlike the promiscuous mode, the monitor mode does not require a connection to a WAP or ad hoc network [30]. The monitor mode is special unique to the wireless network card, and the promiscuous mode is applied to the wired and wireless network cards. With the help of the wireless monitor mode, it is convenient for us to collect auxiliary information for the client-side's membership inference.

3.3. Generative Adversarial Networks. Generative adversarial nets (GANs) were first proposed by Mirza and Osindero [31], which is a neural network trained in an adversarial manner. GANs contain two competing neural network models. One is a generator G that draws random samples z from a prior distribution (e.g., Gaussian or uniform distribution) as the inputs, and then, G generates samples from z that approximate the input distribution. Another model is a discriminator D. Given a training set, the discriminator D is trained to distinguish the generated samples from the training (real) samples. Equation (3) shows the objective function of GANs. Briefly speaking, GANs are trained to minimize the divergence between the generated and real data distribution.

$$\min_G \max_D V(D, G) = \mathscr{E}_{x\sim P_{\text{data}}(x)}[\log(D(x))] + \mathscr{E}_{z\sim P_z(z)}[\log(1 - D(z))], \tag{3}$$

$$V(D, G) = \int_x P_{\text{data}}(x) \log(D(x)) + P_g(x) \log(1 - D(x)) dx, \tag{4}$$

$$D(x) = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_g(x)}. \tag{5}$$

The P_{data} and P_z denote the training (real) distribution and prior distribution, respectively. P_g indicates the distribution of generated samples. These two models G and D are trained alternately until this minimax game achieves Nash equilibrium [32], where the generated samples are difficult to be discriminated from the real ones. The proof of Equation (4) shows that there is an optimal discrimination model. According to the nature of JS divergence [33], if and only if the generation distribution P_g is equal to the real data distribution P_{data}, $JSD(P_{\text{data}} \| P_g) = 0$, we can obtain the optimal discriminator, as shown in Equation (5). It should be noted that the GANs originally set by Mirza and Osindero [31] were defined only in the real number (continuous data, such as images), because the output gradient of the discriminator D would give feedback on how to make it more realistic by slightly changing the generated data. However, when it comes to discrete data (e.g., text), the variation among tokens is much greater than that of images. In addition, the GANs can only evaluate the entire sequence, not the quality and trend of the partial sequence.

Committed to overcoming this problem, the new idea is to introduce reinforcement learning (e.g., SeqGAN [34]). The generator is an agent, the state is the generated token,

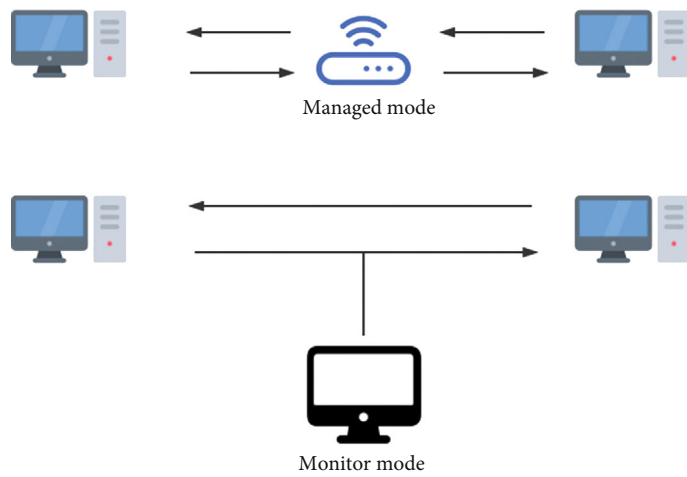

Figure 3: Two wireless card modes.

and the action is the next generated token. Monte Carlo Tree Search (MCTS [35]) is used to estimate the state behavior value and complete the various possibilities of each action. D generates a reward for these complete sequences, as shown in Equation (6), passes it back to G, and updates G through reinforcement learning to train generative networks that can produce the next optimal action. The completed Q function is shown in Equation (7):

$$Q_{D\phi}^{G_\theta} = \left(a = y_T, s = Y_{1:T-1} = D_\phi\left(Y_{1:T}\right)\right), \tag{6}$$

$$Q_{D\phi}^{G_\theta} = (s = Y_{1:t-1}, a = yt) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n), Y_{1:T}^n \in MC^{G_\beta}(Y_{1:t}; N), & t < T, \\ D_\phi(Y_{1:t}), & t = T. \end{cases} \tag{7}$$

Unfortunately, the experimental results prove that this idea does work, and the loss of synthetic data does decrease, but the quality of the generated text is very poor on the real Chinese poetry, which directly affects how this batch of data should be labeled and then becomes the training data of the membership inference attack model. This is currently a bottleneck. The work of Fedus et al., MaskGAN [36], may point out a direction for improving the quality of the generated text.

3.4. Assumptions. As done in previous works, participants will declare the labels of the local data before they start training, which can be verified through a wireless monitor. In fact, this behavior does not reveal valuable privacy about the training set. Because the label cannot reflect the attributes of the data, our scheme is based on a preliminary assumption that the sample labels owned by participants do not overlap. Taking the MNIST dataset as an example, we assume that participant P_1 has data samples with labels "0," "1," and "5"; participant P_2 has data samples with labels "2" and "7"; and so on. Under these circumstances, the declared label "1" cannot reflect the attribute of digit "1" in the picture, such as whether the font is inclined to the right or the left. The purpose of this nonintersecting setting is to facilitate the attack model to compare the results of the attack with the previously declared information to implement the membership inference attack. For example, in training medical data, in order to enrich the training set, different hospitals label the data according to their different pathological information. In this way, the federated model can obtain more pathological classes. Of course, there should be samples with the same label between different hospitals. The membership inference in this case will be discussed in Section 6.

3.5. Threat Model. Here, we will elaborate on the conditions that the adversary has.

3.5.1. Adversary's Objectives. In our settings, the ultimate objective of the adversary is to obtain indirect information about the target victim's dataset. So, we set two indexes in the context of classification tasks to evaluate our attack model: (1) *membership inference accuracy*: means the classification confidence of the target dataset; (2) *main task accuracy*: denotes that the global model should maintain a high prediction accuracy without overfitting.

3.5.2. Adversary's Observations. What the adversary can observe depends on which role he plays. When the adversary is a malicious participant, he can obtain the aggregated white-box global model that is fed back from the central server after each iteration. Besides, with the help of the wireless monitor mode, the adversary can see the communication of other participants. And if we set the adversary as a malicious server, then he can clearly know which participant a certain parameter comes from at this round. (1) For the adversary from the client side, a white-box model and monitored information are sufficient to launch the inference attack. Since the honest server distributes updated models to various participants during each iteration, the adversary will keep the latest model snapshot with him. Therefore, everything of the global model is exposed to the adversary, such as the structure of the model, the algorithm L, and

the parameters θ of multiple versions. This is beneficial for us to use GANs to launch the membership inference attack. (2) For the adversary from the server side, the acquisition of target parameters is also divided into active acquisition and passive acquisition. Referring to the work of Wang et al. [23], the adversary can actively obtain the parameters of the victim and then use GAN for data recovery. The details of our proposed scheme will be introduced in the next section.

3.5.3. Adversary's Capabilities. This topic is also divided into two roles to elaborate. We will list what the adversary can do and cannot do to assess his capabilities. As a malicious participant, on the one hand, the adversary *can* (1) have a snapshot of each updated model, (2) fully control his local data and training procedure, (3) arbitrarily modify the hyperparameters, (4) randomly select local parameter updates over communication rounds, and (5) silently monitor the communication content of the entire wireless network environment. On the other hand, the adversary *cannot* (1) directly access other participants' local data. From the perspective of a malicious server, the adversary *can* (1) proactively obtain parameters uploaded by the victim and unearth useful information from them, (2) aggregate all the collected parameters, and (3) optionally feedback special models to the victim. Besides, the adversary *cannot* (1) see the data owned by participants diametrically.

4. Proposed Membership Inference Attack

In this section, we describe the detail of the user-level membership inference attack in federated learning. Specifically, we focus on some malicious situations in federated learning where a participant and the central server are separately considered an insider, who will silently bypass the attention of others in the cluster to complete the task of differentiating the record's ownership.

4.1. Malicious Participant's Perspective

4.1.1. Attack Overview. Figure 4 overviews the first approach we designed: participant's inference attack. This is an active attack by the adversary. We suppose that in a wireless local area network (WLAN) environment, there are N participants, where the victim V is the target participant, and the adversary A is also on the client side. In the kth iteration, both A and V download the same parameters θ_d. V normally uses parameters to update the local training model, then performs training, and finally returns the training update θ_u to the server. Since the server could honestly average the parameters received from various participants before updating the global model, it is hard for the adversary to explicitly get clues of the target victim to launch the membership inference even with the wireless monitor mode. Therefore, we take GANs as a tool for attack. Except using parameters for local training, A will also copy the parameter θ_d to discriminator D in GANs for updating synchronously, so that the generator G can continuously generate samples closer to the real samples. These generated samples will be used to train the ultimate attack model with the corresponding classification algorithm. When the target dataset is obtained, the attack model will predict results. If a sample whose prediction result is consistent with the declaration information, we can judge it as "IN"; otherwise, judge it as "OUT."

4.1.2. Reconstruction Data with GANs. The goal of our data augmentation phase is to make the training set for the attack model complete. The structure of GANs and details of the data augmentation phase are shown in Figure 5. The generative network $g(z;\theta_G)$ is initialized and generates data records from random noise. In the discriminative network $f(x;\theta_D)$, the discriminator D is initialized with the global model. In this way, replacing the network parameters of D with global model parameters is equivalent to training D directly on the overall training data. Let x_i be the original image in the training set and x_{gen} be generated images. We apply the optimization algorithm based on the approach proposed by Mirza and Osindero [31] and formulate the problem as

$$\min_{\theta_G}\max_{\theta_D}\sum_{i=1}^{n_+}\log\left(f(x_i;\theta_D)\right)+\sum_{i=1}^{n_-}\log\left(1-f\left(g\left(x_{\text{gen}};\theta_G\right);\theta_D\right)\right), \tag{8}$$

$$\mathscr{L}_G\left(\theta_g\right)=\mathbb{E}_{z\sim p(z)}[\log\left(D(G(z))\right)]. \tag{9}$$

The generator G wants to generate samples x_m of class m, which belongs to one of the training sets. G yields x_{gen} to discriminator D. If D can classify x_{gen} as class m, then the data augmentation phase sets $x_m \longleftarrow x_{\text{gen}}$ and returns x_m. Otherwise, it will update the generator G to minimize its loss $\mathscr{L}_G(\theta_g)$ as shown in Equation (9). The pseudocode of the data augmentation phase is shown in Algorithm 1. We first initialize the generation network G and use the current federated learning model as the discriminator D to calculate the gradient to distinguish the generated data from the original data. Until that the discriminator is unable to distinguish the generated data, we get the eligible generated samples x_{gen}.

4.1.3. Attack Algorithm. The pseudocode of the attack phase is shown in Algorithm 1. After generating samples with all labels, we begin to train a classification model. The selection of the inference algorithm can be determined after analyzing the specific generated samples as we described in Section 6. In our experimental scenario, we take the MNIST dataset as an example, and we use the CNN model accordingly. After the model training is completed, the adversary launches the membership inference attack against a bunch of data, named target dataset, which contains the training data of the victim and other participants. After the attack is over, we compare the prediction result with the label information declared by the victim. The data with the same comparison result is regarded as the victim's training data, marked as "IN." Other data, which has different comparison results, are marked as "OUT." To calculate the accuracy of our membership inference attack, we divide the number of

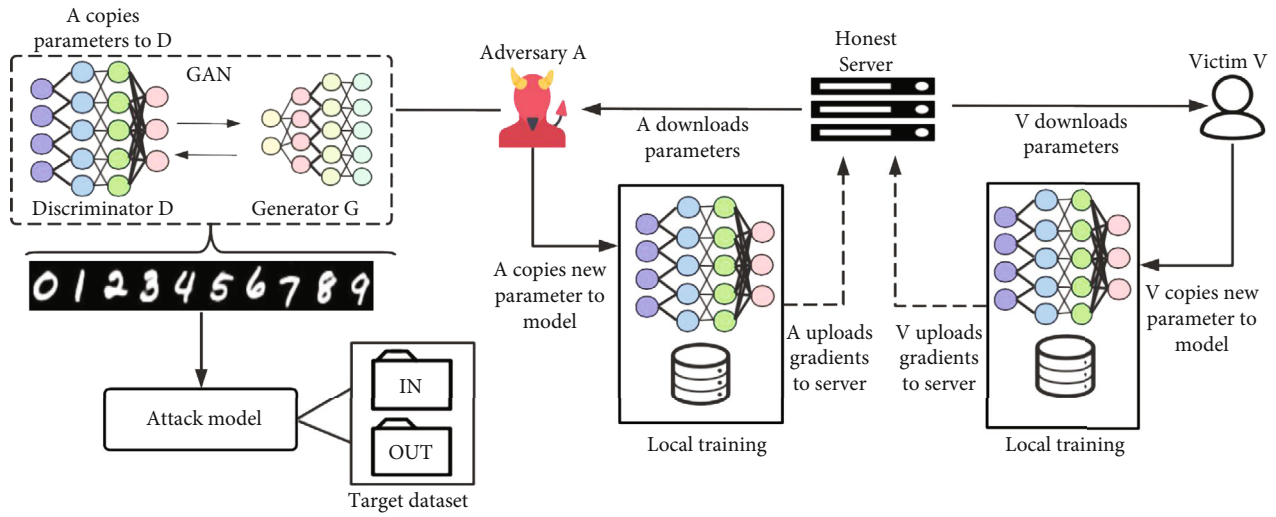

Figure 4: Overview of membership inference from the client side based on GANs.

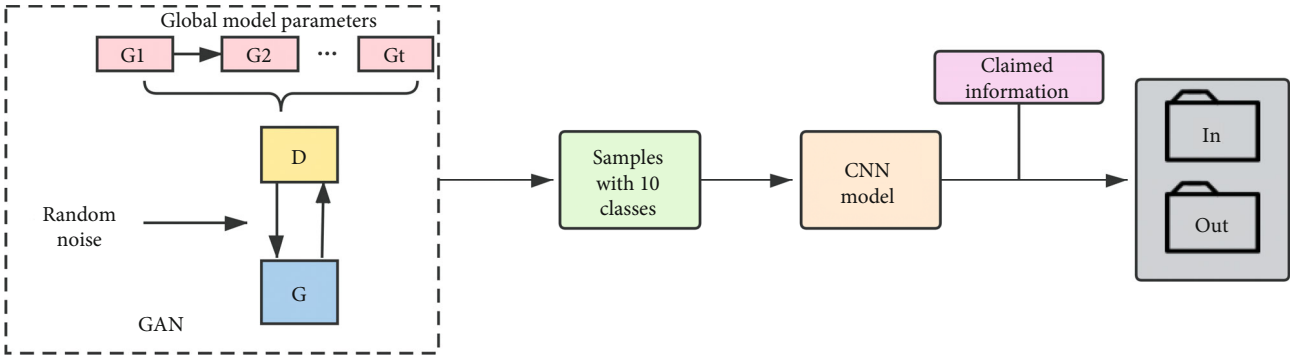

Figure 5: Data augmentation phase.

Input: The GANs iteration round $i_{\max}$, the federated learning model $f()$, generator G, discriminator D.
Output: The generated dataset $Data^{gen}$:(x,y) and the inference result '*IN*' or '*OUT*'.
1: **Procedure** Adversary Execution.
2: Initialize G
3: Set $D \longleftarrow f()$
4: **for** $(i = 1; i <= i_{\max}; i++)$ **do**
5: Run G to generate sample x_{gen}
6: Update G based on Eq (8)
7: **end for**
8: $y = f(x_{\text{gen}})$
9: Output: $D^{gen} : (x, y)$
10: $D^{train}_{attack} = D^{gen} : (x, y)$
11: Attack Phase:
12: Train CNN model using D^{train}_{attack} dataset.
13: Perform membership inference attack against D^{target}_{attack} dataset.
14: Compare the inference results with the claimed information.
15: Output: Mark every record as '*IN*' or '*OUT*', where '*IN*'represents the **Victim's** training sample.

Algorithm 1: Participant's attack procedure.

data marked "IN" by the number of victim's data in the target dataset. Detailed experimental results are presented in the next section.

4.2. Malicious Server's Perspective

4.2.1. Attack Overview. Figure 6 shows the second attack architecture: the server's inference attack. It should be emphasized that this is still an active attack from the server side. The so-called active denotes that the server is highly proactive, e.g., without undermining federated learning, the malicious server can cunningly employ unfair tricks to accomplish the purpose. We also assume that there are a total of N participants, including the victim V and the adversary A at this time on the server side. To achieve a better inference accuracy, from the beginning, the malicious server isolates the victim from other participants [8]. In this situation, the adversary can control the victim individually by sending the victim a quarantined version of the shared model, M_{iso}. Naturally, the adversary intentionally obtains valuable parameter information for reconstructing the victim's data. We adopt a similar approach to that used by Wang et al. [23], using an affiliated server that the victim V will connect to it without his knowledge. So that the shared model of each communication round between them is M_{iso} instead of M. There are two points to note: (1) with this setting, M_{iso} is still sharing weights with D in local-deployed GANs; (2) M_{iso}'s aggregation update is not synchronized with M, which means the target participant's model does not get aggregate with the parameters of other parties. It is aimed to let more membership privacy information be stored in the victim's model.

The next step is to train the membership-oriented GANs on the affiliated server, where the training process is the same as the normal GANs. But since the malicious server simply trains on the real data of the target victim at this time, the generated samples X_{gen} are of higher quality and easier to identify. When participants have multiple labels, but there is no label overlap between them, we can divide the data of these labels into "IN" according to the labels of X_{gen}. After acquiring a new batch of data, we can determine the membership.

In summary, while the server is doing evil, as we described in Section 3.5, the adversary's ability does not exceed the limit of the server's own ability. A lot of previous work has realized this situation.

4.2.2. Reconstruction Data with GANs in Affiliated Server. In the work of Phong et al. [37], the participant's data is restored based on the weight. The malicious server can access the updated parameters under federated learning. But the disadvantage of this scheme is that the shared model has to be a fully connected network. Besides, the update is needed to be obtained by training on a single sample. It is apparent that these constraints obstruct the applicability of the scheme to other algorithm models. Inspired by this solution, we break through the limitations of the shared model by introducing GANs deployed on the affiliated server to secretly access the victim's data privacy. In this attack mode, the operation of GANs is basically the same as that described above. The white-box model of the target victim will serve as the discriminator of GANs, gradually promoting the generator to output high-quality samples with the target labels.

4.2.3. Attack Algorithm. As the pseudocode described in Algorithm 2, at the beginning of federated learning, M_{iso} is initialized consistently with M, but these two models gradually become asynchronous. The affiliated server simply copies the parameters uploaded by the victim for use with GANs, so it will not interfere with the global model's training and will mislead the victim into thinking that he is contributing to global training. After the whole federated learning period, a fully functional global model held by the malicious server can be utilized to identify the generated samples and obtain labels, y_{gen}. The complete generated data with labels, $X_{\text{gen}} : (x_{\text{gen}}, y_{\text{gen}})$, are then marked as "IN," and the rest are marked as "OUT." In this way, they are supplied to build the attack model. The experimental results are also detailed in the next section.

5. Performance Evaluation

In this section, we evaluate our proposed methods, including GANs and the membership inference, in different ways.

5.1. Datasets and Evaluation Goals. In the participant inference experiment, we use two datasets, MNIST [38] and CIFAR-10, to verify the indicators of the program. Then, we put the AT&T dataset [39] into an application to test the performance of GANs in the server inference experiment. Our prepared data are mainly images because we believe that image data has a very special property; that is, the label of the image cannot fully reflect the content of the image. From a privacy perspective, this should be a loophole in the data used for training. Text data has also been considered to be introduced to enrich the experiment, but GANs did not achieve surprising results in natural language processing (NLP) tasks, which hindered the application of text data in our scheme. Many reasons are discussed above. Details of experimental datasets are described in Table 1.

(i) *MNIST*. This dataset includes ten classes of handwritten digits from "0" to "9," which is widely used for training and testing in the field of machine learning. It is commonly used in training various image processing models. A total of 70,000 images is divided into the training set (60,000 images) and the testing set (10,000 images). The grayscale image is normalized into 28×28, a total of 724 pixels.

(ii) *CIFAR-10*. It consists of a training set of 60,000 images and a testing set of 10,000 images with 32×32 pixies in ten classes. These images are mainly cats, dogs, horses, etc.

(iii) *AT&T*. The AT&T dataset (a.k.a. Olivetti dataset of faces) contains 40 topics, each with 10 pictures, and a total of 400 samples. The subject is Olivetti employees or Cambridge University students. The

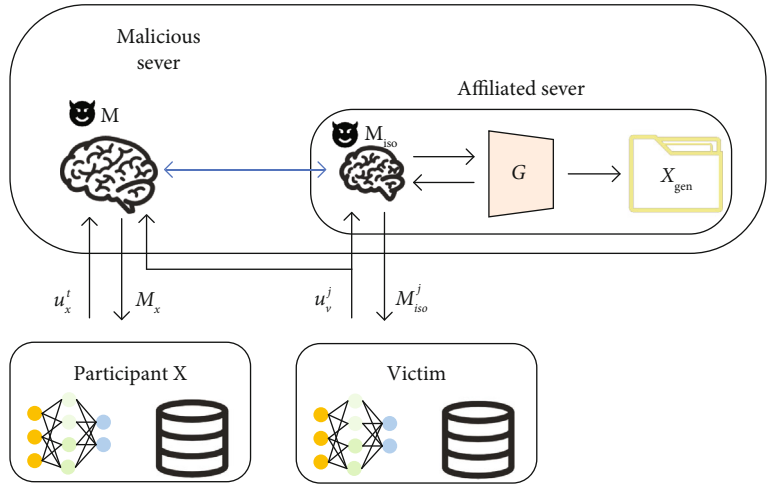

Figure 6: Overview of server-side membership inference using GANs on the affiliated server.

Input: The iteration round of GANs deployed on affiliated server i_{iso}, the federated learning model $f()$, The iteration round of normal training i_{normal}, generator G, discriminator D, isolated model M_{iso} and victim's updated parameters u_v.
Output: The generated dataset X_{gen} :(x,y) and the inference result '*IN*' or '*OUT*'.
Procedure Adversary Execution.
2: Initialize G
Deceptively connect the affiliated server with M_{iso} to the victim
4: Set $D \longleftarrow M_{iso} \longleftarrow M$
for $(i = 1; i <= i_{normal}; i++)$ **do**
6: Train global model $f()$ by collecting all parameters from all participants
Synchronize the following 'for' loop
8: Update $f()$ based on Eq (1)
end for
10: **for** $(i = 1; i <= i_{iso}; i++)$ **do**
Copy u_v to affiliated server
12: Run G to generate sample x_{gen} in a targeted manner
Update G based on Eq (8)
14: **end for**
$y = f(x_{gen})$
16:$y_{other} = f(x_{other})$
Output: $X_{gen} : (x_{gen}, y_{gen}) \longrightarrow$ IN
18: Output: $X_{other} : (x_{other}, y_{other}) \longrightarrow$ OUT
$D_{attack}^{train} = X_{gen} : (x, y) + X_{other} : (x, y)$
20: Attack Phase:
Train CNN model using D_{attack}^{train} dataset.
22: Perform membership inference attack against D_{attack}^{target} dataset.
Compare the inference results with the claimed information.
24: Output: Mark every record as '*IN*' or '*OUT*', where '*IN*'represents the **Victim's** training sample.

Algorithm 2: Server's attack procedure.

age of the subjects was concentrated between 20 and 35 years old. Among them, 36 are males and 4 are females. The subject's picture allows only limited lateral movement and limited tilt. So for most objects, the images are taken at different times and under different lighting conditions. The image was manually cropped and rescaled to 92×112 resolution, 8-bit gray levels. Five images of each object are used for training, and five images are used for testing, a total of 200 training images and 200 test images.

To comprehensively illustrate our proposed attack model, we set the following two goals: (1) mimic data generation: means the effectiveness of our proposed data augmentation algorithm and data reconstruction in server side using

Table 1: Summary of datasets used in our experiments.

Dataset	Labels	Input size	Training samples	Testing samples
MNIST	10	28 ∗ 28 ∗ 1	60000	10000
CIFAR-10	10	32 ∗ 32 ∗ 3	50000	10000
AT&T	40	92 ∗ 112	200	200

GANs; (2) attack success rate: indicates the accuracy of our membership inference in federated learning settings as we described in Section 4. In particular, the main task accuracy is the ratio of the correct classification of all samples through the global model.

5.2. Experimental Settings

5.2.1. Experimental Environment. We implemented the data augmentation and the membership inference in federated learning by using the PyTorch1.0, Tensorflow2.0, and Keras framework. All experiments are done on an RHEL7.5 server with NVidia Quadro P4000 GPU with 32 GB RAM and Ubuntu 16.04LTS OS. The Python version is 3.6. The router device is Redmi router AX6 supporting WIFI6. The experimental wireless network parameters are download rate≈32.81 Mbps, upload rate≈7.41 Mbps, delay≈10 ms, and signal strength≈-40 dBm. In the participant attack experiment, we set up five participants, one of whom is assumed as the adversary, while the remaining participants are benign. They are all subordinate to the same central server. In each round of the federated training, participants' local models are trained separately. Then, they synchronously upload their updates into a new global model. There are similar settings in the server attack experiment. One of the five participants is a victim, but their common server is malicious. The training process is still local training, and the victim will also participate in federated training, but its local model update is not synchronized with other participants, and this model is consistent with the affiliated server.

5.2.2. Model and Training Configurations. Considering the dataset used in our first part of the experiment about client-side inference, we applied a CNN-based model architecture to construct our membership inference classifier. Table 2 shows the neural network structure for two datasets. The model of MNIST consists of two convolutional layers and two dense layers. The kernel size of these convolution layers is 5×5. The number of filters for the first convolutional layer is 16 and for the second convolution layer is 32. The model for the CIFAR-10 dataset is set up as shown in Table 2. There are four convolutional layers with the 3×3 kernel size and 32×32 input shape. The number of filters for the first two convolutional layers is 32 and for the other convolution layers is 64. The activation function applied to all the neural network models is ReLU. In the second part of the experiment, the server-side inference, our GAN network structure refers to Table 3. We build a convolutional neural network consisting of three convolutional layers and three maximum pooling layers and set a fully connected layer at the end. The activation function is Tanh. The output of the model has 40 categories, corresponding to 40 categories of human faces. In the more complex situation of reconstructing face data, the generator G queries the discriminator D more times (the size of the adversary's training data divided by the batch size). Tanh in G is applied to set the output to the range of [-1, +1]. Since the AT&T image is large (64×64), G has an additional (5th) convolutional layer. G accepts a 100-dimensional uniform distribution as input and converts it to AT&T's image (64×64).

Table 2: Neural network structure.

Classifier	MNIST	CIFAR-10
Structure	Conv2D(16,5,5)+ReLU	Conv2D(32,3,3)+ReLU
	MaxPooling2D(2,2)	Conv2D(32,3,3)+ReLU
	Conv2D(32,5,5)+ReLU	MaxPooling2D(2,2)
	MaxPooling2D(2,2)	Conv2D(64,3,3)+ReLU
	FCL(1000)+ReLU	Conv2D(64,3,3)+ReLU
	FCL(10)+Softmax	MaxPooling2D(2,2)
		FCL(512)+ReLU
		FCL(10)+Softmax

The training configurations for two datasets are the following:

(i) Participants train MNIST dataset for epoch $E = 30$ with the initial learning rate $\eta = 0.01$

(ii) Participants train CIFAR-10 dataset for epoch $E = 60$ with the initial learning rate $\eta = 0.0001$

(iii) Participants train AT&T dataset for epoch $E = 60$ with the initial learning rate $\eta = 0.0002$

Besides, we run all the experiments for 400 communication rounds of the federated learning.

5.3. Performance of Data Augmentation. To illustrate the effectiveness of the data augmentation phase and data reconstruction using generative adversarial networks (GANs) in the malicious inference period, we visualize the process of sample generation. The total number of participants and the samples are not changed. The generator G is formatted as random noise with 100 lengths, and its output size is reshaped to 28×28 and 64×64 severally. In addition, we set the adversary to start generating samples after the global model accuracy reaches 80%.

Figures 7 and 8 show the visualization images of the sample reconstruction process as the number of iterations (communication rounds) increase and real samples for comparison. In Figure 7, the blurred contours of the reconstructed samples of 100 iterations can be recognized. As shown in the middle, in 400 iterations, the contours of the generator samples become clearer, because, with the update of the discriminator D, the performance of the generator G becomes better. Therefore, by deploying GANs, the adversary can successfully simulate real samples of all participants

Table 3: Neural network structure of GANs.

	Discriminator	Generator
Structure	Conv(1 ⟶ 32, 5 × 5)+Tanh MaxPooling(3 × 3,3,3) Conv(32 ⟶ 64, 5 × 5)+Tanh MaxPooling(2 × 2,2,2) Conv(64 ⟶ 128,5 × 5)+Tanh MaxPooling(2 × 2,2,2) Reshape(512) Linear(512 ⟶ 400) Tanh Linear(400 ⟶ 40) SoftMax	Conv(100 ⟶ 512,4 × 4) BatchNormalization ReLU Conv(512 ⟶ 256, × 4,2,2,1,1) BatchNormalization ReLU Conv(256 ⟶ 128, × 4,2,2,1,1) BatchNormalization ReLU Conv(128 ⟶ 64, × 4,2,2,1,1) BatchNormalization ReLU Conv(64 ⟶ 1, × 4,2,2,1,1) Tanh

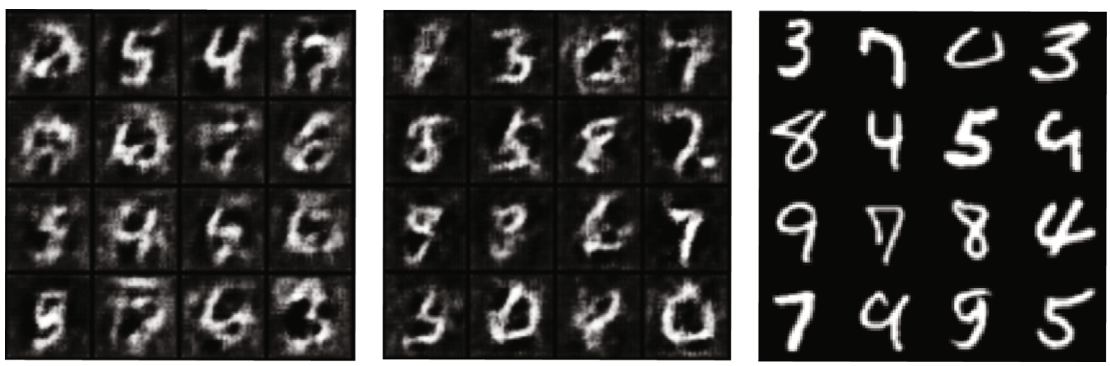

Figure 7: Reconstruction of MNIST based on GAN.

Figure 8: Reconstruction of AT&T based on GAN.

like the image on the right. Figure 8 selects the pictures of the three sequential stages generated by GANs in the malicious server using the isolation method. The four pictures in the first block preliminarily outline the cheek features of these people. The quality of pictures in the second block gradually becomes better. The four images in the third block are very close to the real samples on the far right.

5.4. Performance of Membership Inference. In the membership inference evaluation, the indexes are the main task (federated learning) and the accuracy of the membership inference.

As shown in Figure 9, with the secret membership inference attack, the accuracy of the models corresponding to three datasets, respectively, reaches 94.45%, 92.71%, and 84.48%, which are drawn with solid lines of three colors. In order to prove that the membership inference does not affect the normal training efficiency too much, we use the dotted lines with similar colors to draw the ordinary training process of these models as a baseline. Obviously, all three

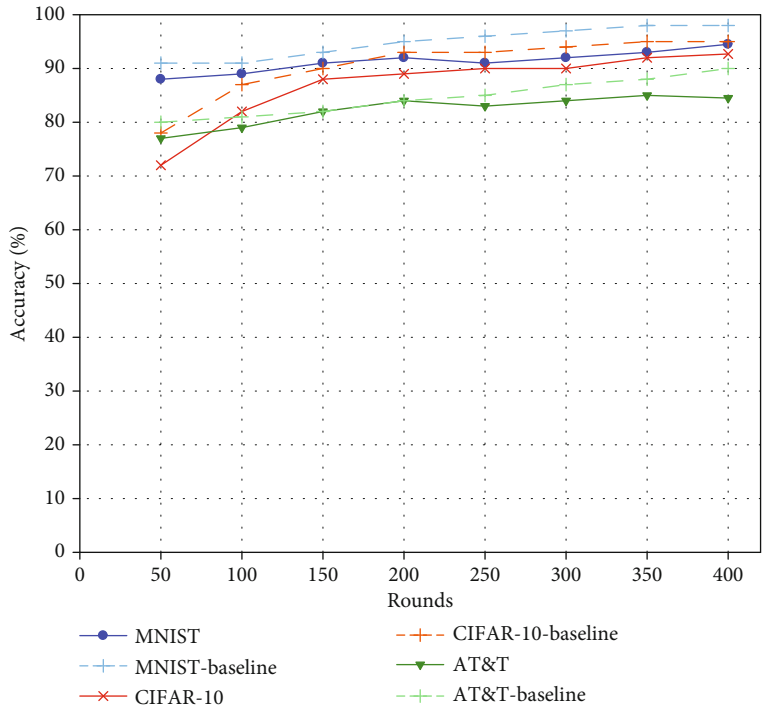

Figure 9: Effectiveness of main task.

models are accurate enough to complete the main task of correctly predicting all test data:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{11}$$

Simultaneously, behind the normal federated learning, the adversary is subtly acquiring enough fake samples through GANs and trained the attack model. After the membership inference, we evaluate the attack from the perspective of the class. Figure 10 illustrates our attack effectiveness on the three datasets. In order to fully demonstrate the inference results, we use the values of precision and recall to visualize the effect. The calculation formulas are shown in formulas (10) and (11), where TP means true positive, FN means false negative, and FP means false positive. We take the number of labels each participant has into account, supposing that the victim holds data with more than one label, which may disrupt the membership inference. We observe the effectiveness of attacks under the conditions of one label, two labels, three labels, and five labels. As is distinctly exhibited in the diagram, when participants hold more labels, the effect of membership inference launched by adversary pretending to be a participant will be worse, just as the values of precision and recall both drop from more than 80% to around 50%. But the adversary as a central server is not influenced by multiple classes, which benefits from restoring a designated participant's data pertinently. Contrast can be seen in these two modes, server-side's directive membership inference breaks through the bottleneck of being a client-side adversary that has no idea about the source of the fake data with multiple classes. We also draw some ROC curves to highlight this difference, where the variable is still the number of labels. Figures 11 and 12 show that when the target victim owns one or two labels around, the inference model on the client side can mark the member data as "IN" and the nonmember data as "OUT" with relative precision. However, with more classes, the false positive rate goes up a lot. Next, we will focus on trying to solve the problem of how to identify member data when there is data overlap between participants.

To highlight the advantages of our scheme, we compare the scheme with the active inference attacks based on the SGA method designed by Nasr et al. [8]. The inference accuracy of experiments based on the SGA method can reach about 76% on the CIFAR-100 dataset, which is close to the case where the participant holds one label in our CIFAR-10 experiment. But the biggest innovation is the attack objective. Nasr et al. [8] stated that the local adversary performs the inference against all other participants. In other words, this is the membership inference for the entire training data of the federated learning, which is not specific enough in our opinion. Our method can invoke membership inference directed to an individual participant under federated learning.

6. Discussion

This paper concentrates on the scenario of data instances where different participants do not have the data with the same class. This is reasonable because, in the federated

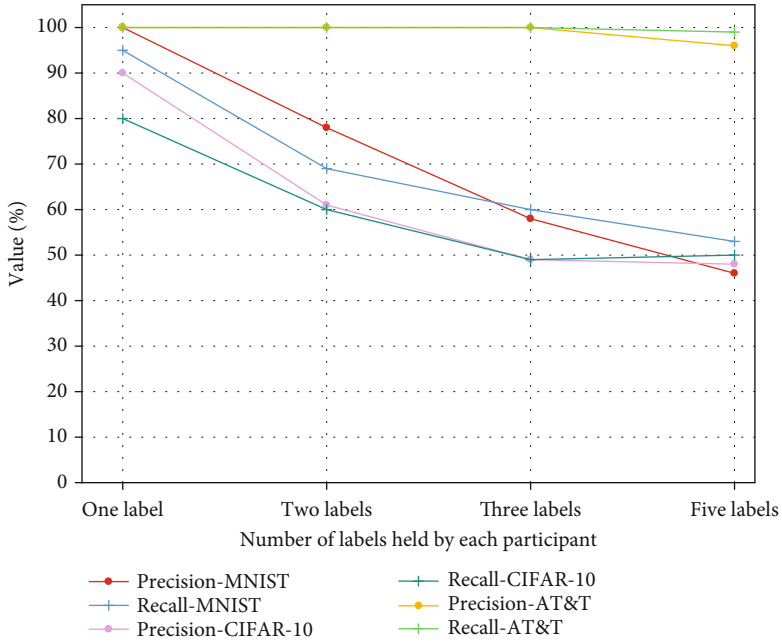

Figure 10: Effectiveness of inference task.

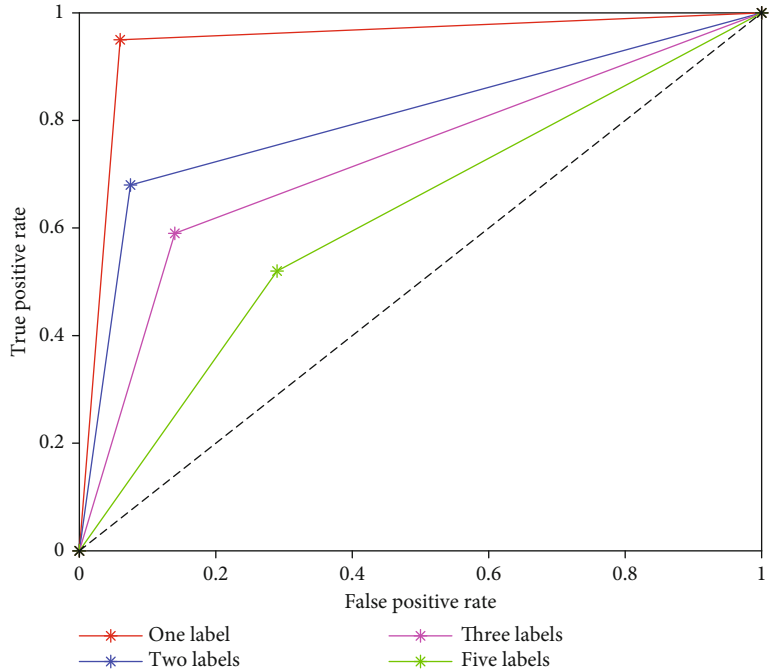

Figure 11: ROC curve of MNIST.

learning field, there are cases where different participants uphold a common training objective but possess very confidential (not shared) data with different labels. For example, if multiple variants of a virus are found in different countries and research institutions are reluctant to share data records with other foreign virologists for analysis, federated learning can play a huge role. At this time, the data with different labels are scattered across countries without any overlap, which is in line with our hypothetical scenario. However, what needs to be soberly aware is that the adversary's preparatory knowledge of the label information is not the end of our solution. Frankly speaking, in this paper, analyzing and classifying the ownership of target data are still at the primary stage of an ideal membership inference for federated learning. It is relatively easy for the adversary to get the answer. In fact, when some different data with the same label are scattered among the participants, the attack model can still give the data attribution, which is the advanced stage. Back to our proposed method, the ultimate goal of the data augmentation architecture we built is to simulate the

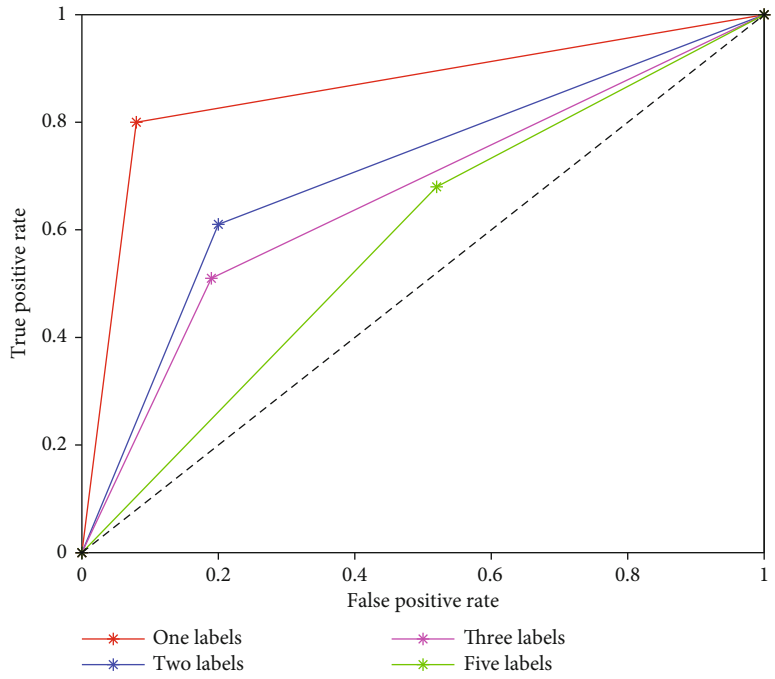

Figure 12: ROC curve of CIFAR-10.

victim's training data as much as possible to achieve advanced federated learning inference. At present, we have not been able to achieve this function. But we firmly believe that the introduced data augmentation architecture is the prototype of the possible technology for solving the advanced inference of federated learning in the future. We designed it as a reference to lay the foundation for future research. Currently, one potential breakthrough is to extract other "nontarget features" of the participants as the distinguishing elements [40], which need to be analyzed based on specific data. For a common example, the federated learning training bases on a globally distributed face dataset and the target representative are "whether wearing glasses or not." At this point, the face samples from the target area can be further filtered according to "complexion." Accordingly, the key feature of the membership inference model is changed to "complexion." Another possible way is the "conspiracy" that the server is colluding with the adversary or multiple adversaries are colluding. Demonstrating the feasibility of these conceptions will be put into our future work.

It is worth adding that, with the continuous training of massive data, the artificial intelligence algorithm model we adopted can better mine the intrinsic correlation of data, which may have better or even unexpected results.

Last but not least, in view of the instability and risk of the federated learning mechanism disclosed in this paper, we would like to propose some defense methods for the federated learning training process as our research contribution. We envision that the declaration information can be encrypted before participants start training. In this way, with the exception of the central server, the label information of all participants is unknown to each other, and it is difficult for the client-side adversary to distinguish the ownership of data. Obviously, asynchronous federated learning might be a more recommended privacy-preserving approach in more scenarios.

7. Conclusion

This paper is aimed at exploring an active and targeted membership inference attack model for federated learning in the wireless network environment. We proposed a fine-grained membership inference mode in the wireless environment, called the user-level membership inference. Given the traditional membership inference in centralized and distributed learning, we release the assumptions of some previous researches and launch membership inference from the client side and server side against a specific participant's data privacy. In order to counteract the influence of the privacy protection mechanism of federated learning posed to membership inference, where the adversary in the client side could only access an aggregated global model, we propose a data augmentation method using GANs with wireless monitor technique to obtain the high-quality generated samples with all labels. Not only that, we comprehensively study the case of a malicious server and successfully complete the membership inference task when participants hold data with multiple labels. Through the extensive experiments on three classic datasets, MNIST, CIFAR-10, and AT&T, we manage to prove that our proposed membership inference attack model can practically compromise the victim's privacy at the user level.

At last, we discuss the hypothetical premises of this paper and come up with some possible ideas. In future work, we will study these promising aspects, especially the

duplicated samples in the training sets, to prove their rationalities through experiments.

Data Availability

The data used in the experiment in this paper are all public datasets. They can be downloaded from the following link: MNIST: http://yann.lecun.com/exdb/mnist/, CIFAR-10: https://www.cs.toronto.edu/~kriz/cifar.html, and AT&T: https://www.kaggle.com/kasikrit/att-database-of-faces.

Disclosure

This paper is an extension of our previous work, which has been presented in the conference ICCCN2020 "Beyond Model-Level Membership Literacy: An Adversarial Approach in Federated Learning," DOI:10.1109/ICCCN49398.2020.9209744.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102000, in part by the National Natural Science Foundation of China under Grant 62172215, and in part by the Natural Science Foundation of Jiangsu Province (No. BK20200067).

References

[1] J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, and X. Zhu, "Deep learning towards mobile applications," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1385–1393, Vienna, Austria, July 2018.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[3] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018, http://arxiv.org/abs/1806.00582.

[4] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[5] S. L. Garfinkel, J. M. Abowd, and C. Martindale, "Understanding database reconstruction attacks on public data," *ACM Queue*, vol. 16, no. 5, pp. 28–53, 2018.

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

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

[8] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning," in *2019 IEEE Symposium on Security and Privacy*, San Francisco.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds., , pp. 177–186, Physica-Verlag HD, 2010.

[10] 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, New York, NY, USA, October 2015.

[11] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: efficient algorithms and tight error bounds," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473, Philadelphia, PA, USA, October 2014.

[12] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[13] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC '09: Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169–178, New York, NY, USA, May 2009.

[14] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: efficient homomorphic encryption for cross-silo federated learning," in *2020 USENIX Annual Technical Conference (USENIX ATC 20). USENIX Association*, pp. 493–506, July 2020, https://www.usenix.org/conference/atc20/presentation/zhang-chengliang.

[15] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: a client level perspective," 2017, http://arxiv.org/abs/1712.07557.

[16] S. Truex, N. Baracaldo, A. Anwar et al., "A hybrid approach to privacypreserving federated learning," in *AISec'19: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 1–11, New York, NY, USA, November 2019.

[17] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model," *Transactions on Data Privacy*, vol. 11, no. 1, pp. 61–79, 2018.

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

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

[20] J. Hayes, L. Melis, G. Danezis, and E. L. De Cristofaro, "Evaluating privacy leakage of generative models using generative adversarial networks," 2017, http://arxiv.org/abs/acs.CR/1705.07663.

[21] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 374–380, Rotorua, New Zealand, August 2019.

[22] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on*

Computer and Communications Security, pp. 603–618, New York, NY, USA, October 2017.

[23] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: userlevel privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520, Paris, France, April-May 2019.

[24] F. Tramer, 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*, pp. 601–618, Austin, TX, USA, August 2016.

[25] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Inference attacks against collaborative learning," 2018, http://arxiv.org/abs/1805.04049.

[26] J. Konečný, M. M. HB, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," 2016, http://arxiv.org/abs/1610.05492.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.

[28] M. Martich, R. Hathaway, and K. Parsa, "Wireless access point," uS Patent App. 11/415,738, 2007.

[29] A. P. Jardosh, K. N. Ramach, K. C. Almeroth, and E. M. Belding-royer, "Understanding link-layer behavior in highly congested IEEE 802.11b wireless networks," in *E-WIND '05: Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, New York, NY, USA, August 2005.

[30] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *The Kluwer International Series in Engineering and Computer Science, vol 353*, T. Imielinski and H. F. Korth, Eds., pp. 153–181, Springer, Boston, MA, USA, 1996.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, http://arxiv.org/abs/1411.1784.

[32] M. Rabin, "Incorporating fairness into game theory and economics," *The American Economic Review*, vol. 83, no. 5, pp. 1281–1302, 1993.

[33] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[34] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: sequence generative adversarial nets with policy gradient," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, ser. AAAI'17*, pp. 2852–2858, San Francisco, 2017.

[35] G. Chaslot, E. Bakkes, I. Szita, and P. Spronck, "Montecarlo tree search: a new framework for game AI," *Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*, M. Mateas and C. Darken, Eds., , pp. 216-217, AAAI Press, Menlo Park, 2008.

[36] W. Fedus, I. J. Goodfellow, and A. M. Dai, "Maskgan: better text generation via filling in the," in *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April-May 2018https://openreview.net/forum?id=ByOExmWAb.

[37] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: revisited and enhanced," in *Applications and Techniques in Information Security*, L. Batten, D. S. Kim, X. Zhang, and G. Li, Eds., pp. 100–110, Springer Singapore, Singapore, 2017.

[38] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, 2012.

[39] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pp. 138–142, Sarasota, FL, USA, December 1994.

[40] G. Ateniese, G. Felici, L. V. Mancini, A. Spognardi, A. Villani, and D. Vitali, "Hacking smart machines with smarter ones: how to extract meaningful data from machine learning classifiers," 2013, http://arxiv.org/abs/1306.4447.