Beyond Model-Level Membership Privacy Leakage: an Adversarial Approach in Federated Learning

Jiale Chen, Jiale Zhang, Yanchao Zhao, Hao Han, Kun Zhu and Bing Chen
Collage of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
Email:yczhao@nuaa.edu.cn

***Abstract*—With the rise of privacy concerns in traditional centralized machine learning services, the 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. However, recent researches reveal the inherent vulnerabilities of the federated learning 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 membership to a more confined level, user-level. In this paper, We propose a novel user-level inference attack mechanism in federated learning. Specifically, we first give a comprehensive analysis of active and targeted membership inference attacks in the context of the 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. The extensive experimental results demonstrate the effectiveness of our proposed attacking approach in the case of single-label and multi-label.**

***Index Terms*—Federated learning; Membership inference; Generative adversarial networks; User-level**

I. Introduction

With the revolution of the decentralized machine learning, researches on collaborative learning technologies such as the federated learning for resource-constrained devices on mobile edge networks [1] have been increasing and expanding the landscape of use cases. The 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, the 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.

Although the federated learning can provide a basic privacy guarantee with localized training, the privacy issues still exist during the aggregation and communication process. Emerging attacking methods, including the membership inference, have been undermining the security of the federated learning. 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

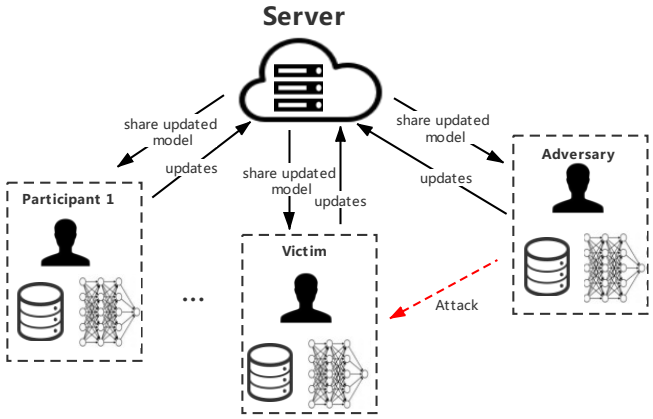

Fig. 1. Membership inference in federated learning.

membership inference attacks are used as pre-attacks for other attack scenarios, such as the reconstruction attack [4], the membership information makes these attacks more targeted and disruptive. Shokri et al. [5] first proposed the membership inference against a black-box machine learning API. In this case, the adversary can simulate the behavior of the target model to compromise the privacy of the training set through many shadow models without knowing the internal structure and parameters of the target model. However, this attack has many assumptions that the adversary has knowledge of the target model structure, and has a dataset from the same distribution as the target models training data.

For the recent researches on the security issues of the artificial intelligence, Salem et al. [6] improved Shokri'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. [7] proposed a membership inference attack launched from the participant side. The core technology of the scheme was the stochastic gradient ascent (SGA). The adversary extracted the parameters of the target model during the training process, including gradients, loss rates, etc. 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) [8], while the gradients of non-member data still rise.

978-1-7281-6607-0/20/$31.00 ©2020 IEEE

By detecting this distinction, the membership information is transformed into a score, which is used as a new feature to construct an unsupervised learning to distinguish member data from non-member data.

Although the above-mentioned inference attacks can reveal the privacy of training data in varying degrees, they presented several limitations. 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 about the possible existence of a malicious participant launching the membership inference, which is close to the real situation. Motivated by those shortcomings in existing inference attacking techniques, we give a deep analysis of active and targeted membership inference attacks in the federated learning with a white-box access model from the perspective of a malicious participant. We name our scheme as the *user-level* membership inference. The reason why we call '*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, and the adversary also plays the role of a certain participant in federated learning, see Figure 1. 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 inherent defense mechanism in federated learning, the model averaging algorithm, and the lack of training data for the membership inference, we further propose a local-deployed data augment method relying on the generative adversarial networks (GANs) to generate high-quality fake samples.

Our contributions in this paper can be summarized as follows.

- **User-level membership inference:** We further disclose the security hole of the current federated learning by novelly launching fine-grained membership inference attacks and encourage more researches on preventing participants from leaking privacy.
- **Data augment using GANs:** To obtain the data distribution of other participants to perform the membership inference, we innovatively use local-deployed generative adversarial networks to generate samples with all labels.
- **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. Sec. II reviews the related work. Sec. III briefs some background knowledge and introduces the threat model. Sec. IV describes the proposed method framework along with an analysis of the membership inference. The performance evaluation results are presented in Sec. V. Sec. VI discusses the limitations of our method and gives some ideas. Finally, Sec. VII concludes the paper.

II. Related Work

In this section, we will introduce the privacy protection methods for the distributed deep learning and the 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.

A. Privacy-preserving Distributed Learning & 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 the distributed learning, an algorithm named as distributed selective stochastic gradient descent (DSSGD) was proposed by R. Shokri et al. [9]. The results showed that even if only 1% of the parameters are shared, the collaborative learning will bring a higher accuracy than the centralized learning. Moreover, R. Shokri et al. [9] utilized differential privacy [10] to effectively prevent data privacy that may be indirectly leaked. Based on the previous article, Phong et al. [11] 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 [12] 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.

The difference between the collaborative learning and the federated learning is that the central server of the federated learning will average the updates (ie, the weight matrix) after each communication round. Even so, the privacy violation remains a challenge. In the user-level differential privacy algorithm proposed by [13], 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. Stacey Truex et al. [14], 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 the privacy preserving in federated learning.

B. 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 targets training set. As the centralized learning evolves to the 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 [7]. 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 the differential privacy protection still has the risk of leaking membership privacy [15]. Our work focuses on the membership inference in federated learning, but more detailed, we are specific to the privacy of a certain participant, not the privacy of the entire global model. The main reason for the leakage of membership information is the model overfitting [16–18]. This takes the membership inference a step further in the field of study.

C. Attacks Against a Specific Victim in the Collaborative & Federated Learning

In addition to stealing membership information, there are many attacks against participants' privacy in federated learning. These attacks, for example, the poisoning attack [19], the model inversion attack [20], the representative inference [21], the model stealing attack [22], the capturing of extra properties [23], 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 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 alleviates this limitation with the generative adversarial networks (GANs) and does not require a lot of auxiliary information.

III. Preliminaries

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

A. Federated Learning

The federated learning [24] 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 [25]. The updates are averaged and accumulated to the current shared model. Eq. 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 Eq. 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.

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

On the one hand, the federated learning can effectively enrich the diversity of training set and allow more data to participate in calculations. On the other hand, the 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 participants data and output greatly threatens the security of the federated learning.

B. Generative Adversarial Networks

Generative adversarial nets (GANs) were first proposed by Goodfellow [26], 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. 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. Eq. 3 shows the objective function of GANs.

$$\begin{aligned}\min_G \max_D V(D,G) = \mathcal{E}_{x\sim P_{data}(x)}[\log\left(D(x)\right)] \\ +\mathcal{E}_{z\sim P_z(x)}[\log\left(1-D(x)\right)]\end{aligned} \tag{3}$$

The P_{data} and P_z denote the training (real) distribution and prior distribution, respectively. These two models G and D are trained alternately until this minimax game achieves Nash equilibrium, where the generated samples are difficult to be

discriminated from the real ones. Theoretically, the global optimum is achieved at $P_{data} = P_g$ [26], where P_g indicates the distribution of generated samples.

C. Assumptions

As done in previous works, before participants start training, they will declare the labels of the data they hold. In fact, this behavior does not reveal valuable privacy about 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 data set 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. In this case, 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 non-intersecting 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, 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 Sec. VI.

D. Threat Model

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

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.

Adversary's observations. Here we will introduce a white-box model to illustrate that what the adversary observes is sufficient to launch the inference attack. Since the 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. The details of our proposed scheme will be introduced in the next section.

Adversary's Capabilities. In this topic we will list what the adversary can do and cannot do to assess his capabilities. 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 hyper-parameters; (4) randomly select local parameters updates over communication rounds. On the other hand, the adversary *cannot* (1) know the gradients uploaded by other participants because of the average algorithm at the server-side; (2) directly access other participants' local data.

IV. 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 a malicious situation in federated learning where a participant is considered as an insider, who will go over the server and directly differentiate the record's ownership.

A. Attack Overview

Figure 2 overviews the attack method we designed. We suppose that 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 average the parameters received from various participants before updating the global model, it is hard for the adversary to directly get clues of the target victim to launch the membership inference. 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'.

B. Reconstruction Data with GANs

The goal of our data augment phase is to make the training set for the attack model complete. The structure of GANs and details of the data augment phase are shown in Figure 3. The generative network $g(z;\theta_G)$ is initialized and generates data records from a 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, x_{gen} are generated images. We apply the optimization algorithm based on the approach proposed by Goodfellow et al. [26] and formulate the problem as:

$$\min_{\theta_G}\max_{\theta_D}\sum_{i=1}^{n_+}\log(f(x_i;\theta_D))+\sum_{i=1}^{n_-}\log(1-f(g(x_{gen};\theta_G);\theta_D)) \tag{4}$$

$$\mathcal{L}_G(\theta_g)=\mathbb{E}_{z\sim p_(z)}[\log(D(G(z)))] \tag{5}$$

The generator G wants to generate samples x_m of class *m*, which belongs to one of the training set. The G yields x_{gen} to discriminator D. If D can classify x_{gen} as class *m*, then

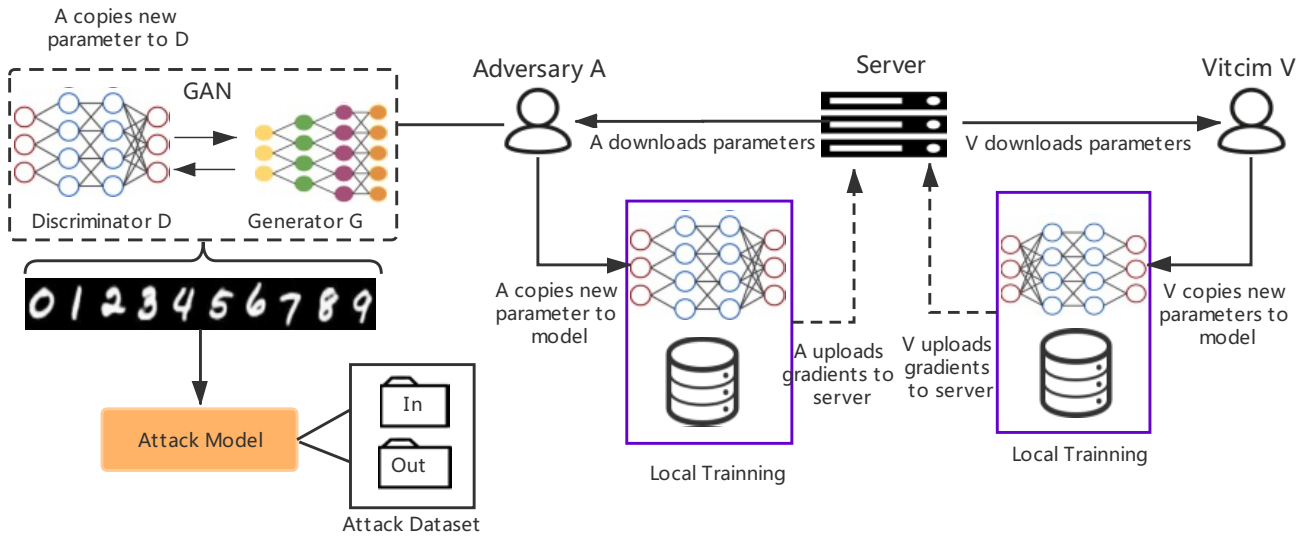

Fig. 2. Overview of membership inference in federated learning based on GANs

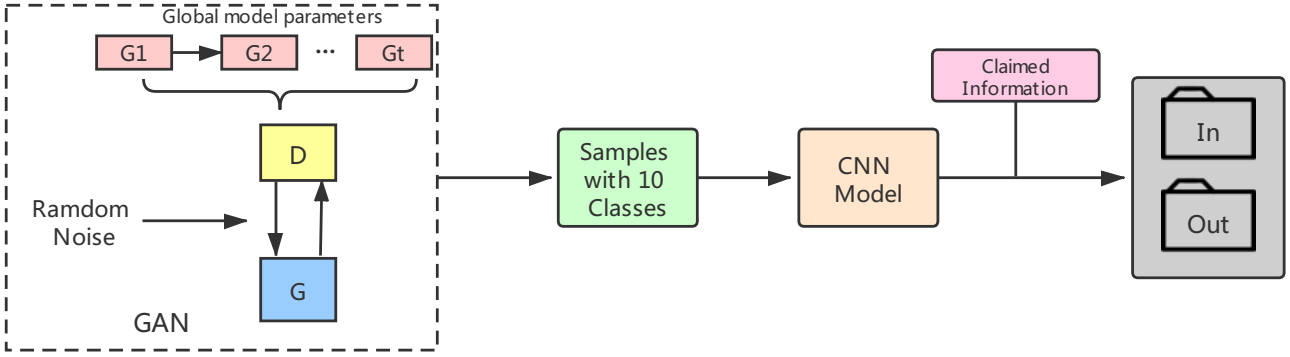

Fig. 3. Data augment phase.

the data augment phase sets $x_m \leftarrow x_{gen}$ and returns x_m. Otherwise, it will update the generator G to minimize its loss $\mathcal{L}_G(\theta_g)$ as shown in Eq. 5. The pseudocode of the data augment 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}.

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

V. Performance Evaluation

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

A. Datasets and Evaluation Goals

We construct GANs and a classification task with two datasets, which are MNIST and CIFAR-10. Details of these datasets are described in the Table I.

- **MNIST**: This dataset includes ten classes of handwritten digits from '0' to '9', which is widely used for training

Algorithm 1 Attack Procedure.

Input: The GANs iteration round i_{max}, the federated learning model $f()$, generator G, discriminator D.

Output: The generated dataset D^{gen}*:(x,y)* and the inference result '*IN*' or '*OUT*'.

Procedure Adversary Execution.
Initialize G
Set $D \leftarrow f()$
for $(i = 1; i <= i_{max}; i++)$ **do**
  Run G to generate sample x_{gen}
  Update G based on Eq 4
end for
$y = f(x_{gen})$
Output: $D^{gen} : (x, y)$
$D^{train}_{attack} = D^{gen} : (x, y)$
Attack Phase:
Train CNN model using D^{train}_{attack} dataset.
Perform membership inference attack against D^{target}_{attack} dataset.
Compare the inference results with the claimed information.
Output: Mark every record as '*IN*' or '*OUT*', where '*IN*'represents the **Victim's** training sample.

TABLE I
SUMMARY OF DATASETS USED IN OUR EXPERIMENTS

Dataset	Labels	Input Size	Training Samples	Testing Samples
MNIST	10	28*28*1	60000	10000
CIFAR	10	32*32*3	50000	10000

and testing in the field of machine learning. It is commonly used in training various image processing models. Total of 70,000 images are divided as the training set (60,000 images) and the testing set (10,000 images). The grayscale image is normalized into 28×28, total of 724 pixels [27].

- **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 [28].

To comprehensively illustrate our proposed attack model, we set the following two goals: (1) mimic data generation: means the effectiveness of our proposed data augment algorithm using GANs; (2) attack success rate: indicates the accuracy of our membership inference in federated learning settings as we described in Sec. IV. Especially, the main task accuracy is the ratio of the correct classification of all samples through global model.

B. Experimental Settings

We implemented the data augment 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 32GB RAM, and Ubuntu 16.04LTS OS. The Python version is 3.6. We set up five participants, one of whom is assumed as the adversary, while the remaining participants are benign. They're 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.

TABLE II
NEURAL NETWORKS STRUCTURE

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

Model and Training Configurations: Considering the dataset used in our experiments, we applied a CNN-based model architecture to construct our membership inference classifier. Table II 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 second convolution layer is 32. The model for the CIFAR-10 dataset is set up as shown in the table II. 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.

The training configurations for two datasets are: participants train MNIST dataset for epoch $E = 30$ with the initial learning rate $\eta = 0.01$ and participants train CIFAR-10 dataset for epoch $E = 60$ with the initial learning rate $\eta = 0.0001$. Besides, we run all the experiments for 400 communication rounds of the federated learning.

C. Performance of Data Agumentation

To illustrate the effectiveness of the data augment phase using generative adversarial networks (GANs) in federated learning protocol, we visualize the process of sample reconstruction. 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. In addition, we set the adversary to start generating samples after the global model accuracy reaching 93%.

Figure 4 shows the visualization images of sample reconstruction as the number of iterations (communication rounds) increases. We show the reconstruction results of 400 iterations of the MNIST dataset, as well as extracted real samples. As shown on the left, 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

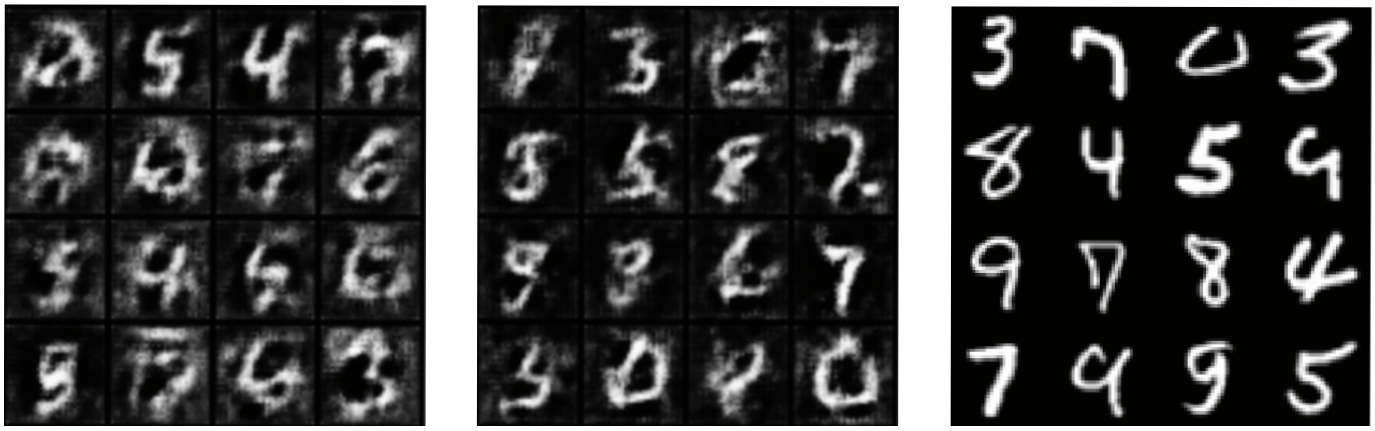

Fig. 4. Reconstruction of MNIST based on GAN

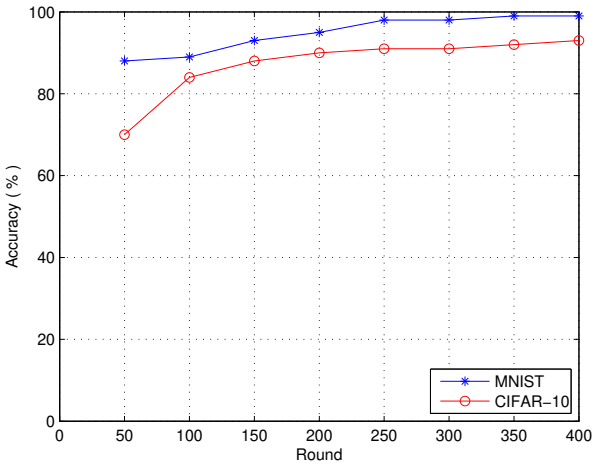

Fig. 5. Effectiveness of main task

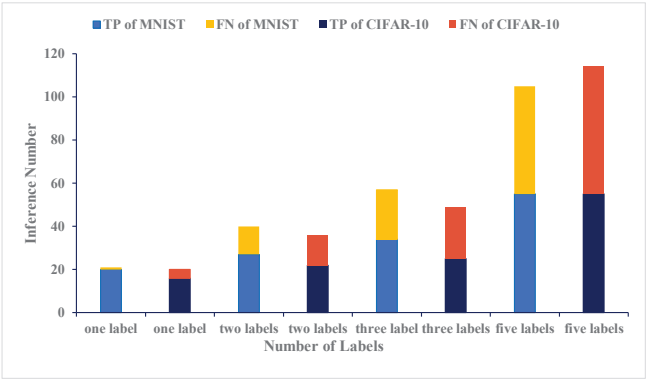

Fig. 6. Effectiveness of inference task

discriminator D, the performance of the generator G becomes better. Therefore, by deploying GANs, the adversary can successfully simulate real samples of all participants like the image on the right.

D. Performance of Membership Inference

In the membership inference evaluation, the indexes are the accuracy of the membership inference and the main task.

The accuracy of models based on MNIST and CIFAR-10 reaches 99.45% and 93.71% as shown in Figure 5, respectively, which is accurate enough to complete the main task of correctly predicting all testing data.

Simultaneously, as mentioned before, the adversary has gotten enough fake samples through a locally deployed GANs and trained the attack model. After the membership inference, we evaluate the attack from the perspective of the label. Figure 6 illustrates our attack effectiveness on the two datasets, where TP means true positive and FN means false negative. 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 can be seen that when participants hold more data classes, the effectiveness of the membership inference is worse. We also draw an ROC curve based on the membership inference attack performance against the two datasets, where the variable is still the number of labels. Figure 7 and Figure 8 show that when the target victim owns one or two labels around, we can accurately mark the member data as 'IN' and the non-member data as 'OUT'. Next, we will try to solve the problem of how to improve the accuracy of membership inference when the victim or participants hold data with multiple classes.

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. [7]. 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. [7] 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

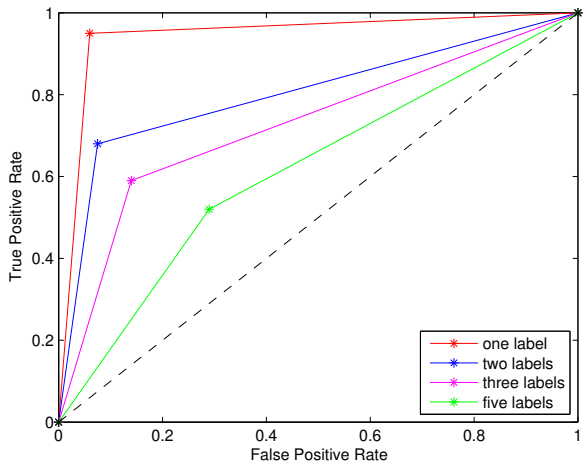

Fig. 7. ROC curve of MNIST

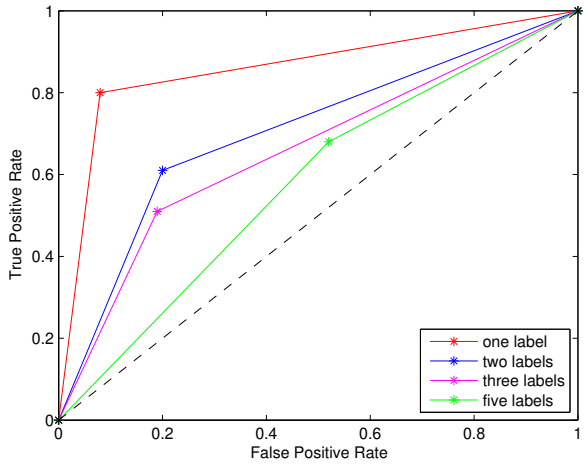

Fig. 8. ROC curve of CIFAR-10

the federated learning, not specific enough. Our method can directly launch an inference attack on an individual participant from the local adversary.

VI. Discussion

This paper focuses on the scenario of data instances where different participants do not have the same class. This is reasonable because, in the federated learning scenario, 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, the 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. Of course, the more general scenario is where many participants have some data with the same label. Such an assumption can be relaxed by solving the key difficulty that when using GANs, how can an adversary distinguish the data that approximates the victim's having from the generated data. One potential solution is to extract other 'non-target features' of the participants as the distinguishing elements [29], which needs to be analyzed based on specific data. As a common example, the federated learning training bases on a globally distributed face dataset and the target representative is '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 solution is the 'conspiracy' that the server is colluding with the adversary or multiple adversaries are colluding. Demonstrating the feasibility of these solutions will be put into our future work.

As for the defense method of the attack, we envisage that the declaration information can be encrypted before the participants start training. Except for the central server, the participants do not know the label information between each other. In this way, it is difficult for the adversary to distinguish the ownership of data labels.

VII. Conclusion

This paper aims to explore an active and targeted membership inference attack model in the federated learning scenario. We proposed a fine-grained membership inference method, called the user-level membership inference. Given the traditional membership inference in the centralized and distributed learning, we release the assumptions of some previous researches and launch membership inference from the client-side against a specific participant. In order to solve the problem of privacy protection of the federated learning, where the server will average the received parameters from all participants, we propose a data augment method using GANs to obtain the high-quality generated samples with all labels. Through the extensive experiments on two classic datasets, MNIST and CIFAR-10, we manage to prove that our proposed membership inference attack model can successfully compromise the victims privacy in 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.

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)*. IEEE, 2018, pp. 1385–1393.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 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." *arXiv: Learning*, 2018.

[4] S. L. Garfinkel, J. M. Abowd, and C. Martindale, "Understanding database reconstruction attacks on public data," *ACM Queue*, vol. 16, no. 5, p. 50, 2018.

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

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

[7] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning," in *2019 ieee symposium on security and privacy*.

[8] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

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

[10] 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*. IEEE, 2014, pp. 464–473.

[11] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[12] C. Gentry, "Fully homomorphic encryption using ideal lattices," pp. 169–178, 2009.

[13] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[14] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 1–11.

[15] 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.

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

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

[18] J. Hayes, L. Melis, G. Danezis, and E. L. De Cristofaro, "Evaluating privacy leakage of generative models using generative adversarial networks. arxiv 2017," *arXiv preprint cs.CR/1705.07663*.

[19] 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)*. IEEE, 2019, pp. 374–380.

[20] 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*, 2017, pp. 603–618.

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

[22] 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*, no. CONF. Usenix Assoc, 2016, pp. 601–618.

[23] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Inference attacks against collaborative learning," *arXiv preprint arXiv:1805.04049*, 2018.

[24] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

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

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[27] 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.

[28] R. C. Çalik and M. F. Demirci, "Cifar-10 image classification with convolutional neural networks for embedded systems," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2018, pp. 1–2.

[29] 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," *arXiv preprint arXiv:1306.4447*, 2013.