MemDefense: Defending against Membership Inference Attacks in IoT-based Federated Learning via Pruning Perturbations

Meng Shen, *Member, IEEE*, Jin Meng, Ke Xu, *Senior Member, IEEE*, Shui Yu, *Fellow, IEEE*, and Liehuang Zhu, *Senior Member, IEEE*

***Abstract*—Depending on large-scale devices, the Internet of Things (IoT) provides massive data support for resource sharing and intelligent decision, but privacy risks also increase. As a popular distributed learning framework, Federated Learning (FL) is widely used because it does not need to share raw data while only parameters to collaboratively train models. However, Federated Learning is not spared by some emerging attacks, e.g., membership inference attack. Therefore, for IoT devices with limited resources, it is challenging to design a defense scheme against the membership inference attack ensuring high model utility, strong membership privacy and acceptable time efficiency. In this paper, we propose MemDefense, a lightweight defense mechanism to prevent membership inference attack from local models and global models in IoT-based FL, while maintaining high model utility. MemDefense adds crafted pruning perturbations to local models at each round of FL by deploying two key components, i.e., parameter filter and noise generator. Specifically, the parameter filter selects the apposite model parameters which have little impact on the model test accuracy and contribute more to member inference attacks. Then, the noise generator is used to find the pruning noise that can reduce the attack accuracy while keeping high model accuracy, protecting each participant's membership privacy. We comprehensively evaluate MemDefense with different deep learning models and multiple benchmark datasets. The experimental results show that low-cost MemDefense drastically reduces the attack accuracy within limited drop of classification accuracy, meeting the requirements for model utility, membership privacy and time efficiency.**

***Index Terms*—IoT, Federated Learning, membership inference attack, defense, pruning perturbations.**

I. Introduction

With an immense proliferation of IoT devices, it is important to capitalize on the accelerated Internet speed and the unprecedented potential for an exponentially larger number of endpoints facilitated by the advent of 5G/6G technology [1], [2]. However, as a growing multitude of IoT devices support numerous applications and services, significant challenges emerge in the form of high communication and storage costs, together with privacy concerns of user data [3], [4]. Federated Learning (FL) has emerged as the foremost and promising alternative approach to address the predicament, which can protect the privacy of participants by sharing local models rather than original data [5]. Although FL has been widely used in many IoT application scenarios, such as smart medical treatments [6], smart home [7] and smart city [8], recent studies [9]–[13] have shown that FL still suffers from privacy attacks, e.g., membership inference attacks [14]. Additionally, the limited computing resources of IoT devices also place certain requirements on defense methods against privacy leakage. Therefore, in the IoT-based FL scenario, it is still challenging to implement a lightweight defense scheme that achieves high model utility and strong membership privacy, considering the limited computing resources of IoT devices.

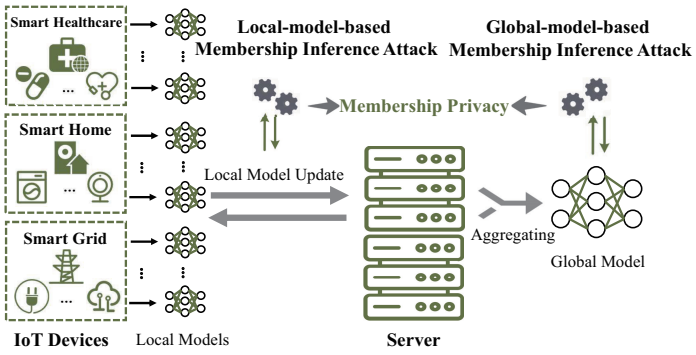


Fig. 1. The membership inference attacks in IoT-based Federated Learning.

Membership inference attacks mean an adversary can infer whether a specific data record is the member of the target models' training dataset or not. According to the different phases in which it is launched, membership inference attacks can be classified into the local-model-based attack and the global-model-based attack [14], [15], as illustrated in Fig. 1. In the training phase of FL, with the local-model-based attack, the adversary (e.g., the curious server or participants) can obtain membership privacy from the local models based on data collected from IoT devices [14], [16], which results in severe privacy leakage and dampens individuals' enthusiasm to participate in FL. In the prediction phase, with the global-model-based attack, the adversary attempts to extract membership privacy from the well-trained global model, which is the aggregation of local models from many IoT devices.

M. Shen, J. Meng and L. Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenmeng@bit.edu.cn, mengjin_1008@163.com, liehuangz@bit.edu.cn).

K. Xu is with the Department of Computer Science, Tsinghua University, Beijing, China, Beijing 100084, China (e-mail: xuke@tsinghua.edu.cn).

S. Yu is with the School of Computer Science, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: shui.yu@uts.edu.au).

The existing defenses [10], [11], [17]–[19] in the centralized machine learning can resist the attacks to the global model by preventing the model overfitting, but some of which are not well transferred to protect the local models. Additionally, the defenses of protecting local models using the secure aggregation [20]–[23] bring heavy computation and communication overhead, which conflicts with the limited resources of IoT devices. Moreover, the defenses using differential privacy [24]–[26] bring poor trade-off between the model utility and membership privacy. Recently, some schemes [27], [28] use model pruning techniques to resist membership inference attacks, but they still focus on the centralized neural network training or incur the huge overhead due to retraining. Therefore, in this paper, we focus on achieving high model utility, *defending membership privacy against local-model-based and global-model-based attacks* and ensuring acceptable overhead for IoT devices with limited computation resources.

In this paper, we propose MemDefense to defend against the membership inference attack in the both training and prediction phase of FL. Since the pruning technique can advance in the limited storage and bandwidth in the scenario of IoT, our scheme introduces the idea of model pruning [29], [30]. Considering that model overfitting and model parameter exposure are two main factors leading to the membership inference attack, model perturbations can not only introduce randomness and uncertainty to alleviate overfitting, but also hide the original model parameters. From this perspective, model pruning is equivalent to adding noise to some model parameters so that their values become 0 to resist membership inference attacks. Therefore, we design two new components, *parameter filter* and *noise generator* for our pruning perturbation method. Specifically, the parameter filter attempts to find some model parameters that have less impact on the model accuracy and have greater influence on the attack accuracy. The noise generator generates the pruning perturbations, which are added to make selected model parameters zero. The proposed scheme thereby protects membership privacy, while maintaining high model utility and low resource overhead.

We evaluate MemDefense extensively on four real-world datasets and compare it with the state-of-the-art defenses, including dropout [11], L2 regularization [19], adversarial regularization [10], differential privacy [25] and homomorphic encryption [20]. Our experimental results show that MemDefense can effectively defend against the white-box membership inference attacks to local models in FL and also protect the global model, while guaranteeing the model utility. Additionally, MemDefense does not bring the significant overhead to devices, making it suitable for the IoT scenario. Compared with no defense, MemDefense achieves similar classification accuracy, while obtaining the outstanding reduction in attack accuracy among existing defenses.

In summary, our contributions are as follows.

- We propose MemDefense to defend against the local-model-based and global-model-based membership inference attacks via pruning perturbations, while ensuring high model utility and strong membership privacy, as well as supporting for IoT devices with limited resources.
- We design the parameter filter and noise generator to select parameters which have little impact on the model accuracy of the training task and contribute more to membership inference attacks and generate pruning perturbations to make selected parameters zero.
- We evaluate its performance on four real-world datasets CIFAR-10, CIFAR-100, MNIST and FashionMNIST. The experimental results demonstrate that lightweight MemDefense can defend against the local-model-based and global-model-based membership inference attacks effectively, with limited loss of classification accuracy.

TABLE I
THE MAIN NOTATIONS USED IN THIS PAPER

Notations	Descriptions
$\boldsymbol{\theta}$	Target model of FL
h	The attack model
m	The number of selected participants at each iteration
L	The loss function
T	The iterations of FL
D	The dataset in FL
D_i	The local dataset of i-th participant
G	The gain of the membership inference atack model
N	The number of total participants in FL
$\boldsymbol{\Theta}$	The global model of FL
$\boldsymbol{\theta}_i$	The local model of i-th participant
$\boldsymbol{\theta}_i^*$	The noisy local model of i-th participant
$\boldsymbol{\theta}_i^{sele}$	The selected local model parameters of i-th participant
$\boldsymbol{n}$	The pruning noise of MemDefense
$\boldsymbol{n}_i$	The pruning noise added to local model of i-th participant
γ	The pruning threshold of MemDefense
γ_i	The pruning threshold of i-th participant
$\|.\|$	The number of variables

The remainder of this paper is organized as follows. We introduce the background and the related work in Section II. Next, we describe the problem formulation in Section III and present the design of MemDefense in Section IV. Then, we describe the datasets and model architectures for evaluation in Section V, and present the evaluation results in Section VI. Finally, we conclude this paper in Section VII.

II. BACKGROUND AND RELATED WORK

In this section, we present the background of FL and review the existing membership inference attacks and the corresponding defense methods.

A. Federated Learning

FL is a computing paradigm for distributed learning with multiple participants [5]. We focus on typical horizontal Federated Learning [31], where the participants upload local models trained over their private data with the same feature space, and then the server aggregates the local models and updates the global model. Table I shows the main notations used in this

TABLE II
SUMMARY OF DEFENSES AGAINST MEMBER INFERENCE ATTACKS

Defense Ideas	Solutions	Defense methods	Attack modes	Descriptions
Mitigating overfitting	Shokri et al. [19]	L2 regularization	White-box and black-box	Use L2 regularization to generalize models, leading to more than 70% attack accuracy with acceptable accuracy loss.
	Nasr et al. [10]	Min-max game	Black-box	Modify the loss function using the attack model, getting more than 10% accuracy loss with 50±5% attack accuracy.
	Salem et al. [11]	Dropout, Model stacking	Black-box	Leverage model stacking and dropout to train models and provide similar trade-off with L2 regularization.
	Shejwalkar et al. [18]	Knowledge distillation	Black-box and white-box	Use knowledge distillation to train models, which cannot protect membership privacy of shared gradients of FL.
	Wang et al. [27]	Model Pruning	Black-box	Use model pruning technique to train models, which is not considered to defend against local model privacy in FL.
	Jia et al. [17]	Adversarial examples	Black-box	Add well-designed noises to the predictions of well-trained models, which is limited to defend against black-box attacks.
Hiding the original model	Aono et al. [20]	Homomorphic encryption	Black-box and white-box	Use secure model aggregation to protect local models, bringing huge computation and communication overhead.
	Wei et al. [25]	Differential privacy	White-Box	Use NbAFL to train models, providing poor trade-off between the privacy and utility on well-generalized models.
Both	**MemDefense (This paper)**	Pruning perturbations	White-box	Adding pruning noises to local models, providing better trade-off between the membership privacy and model utility.

paper. Assuming the number of participants is N and each participant i has the private dataset D_i. The objective of each participant i is to minimize the loss function in Eq. (1).

$$L_i(\boldsymbol{\theta}_i) = \frac{1}{|D_i|} \sum_{j \in D_i} l_j(\boldsymbol{\theta}_i), \tag{1}$$

where $\boldsymbol{\theta}_i$ is the local model of participant i and $l_j(\boldsymbol{\theta}_i)$ is the loss function of participant i on data record (x_j, y_j). Define the global dataset as $D = \cup_{i=1}^{N} D_i$, the objective of FL is to train a global model $\boldsymbol{\theta}$ to minimize the global loss function $L(\boldsymbol{\theta})$, as shown in Eq. (2).

$$L(\boldsymbol{\theta}) = \frac{1}{|N|} \sum_{i \in N} |D_i| L_i(\boldsymbol{\theta}_i) \tag{2}$$

B. Membership Inference Attacks

According to the different prior knowledge of adversaries, the membership inference attacks can be divided into the black-box membership inference attacks (e.g., using the outputs of the target models as features) and white-box membership inference attacks (e.g., using the prediction loss, the outputs of layers and the gradients as features) [9], [16]. The reason for the success of membership inference attacks is that certain features can be used to distinguish member data from non-member data. Therefore, the overfitting model are more likely to be the target of the membership inference attack. Besides, the exposed original model parameters can also be exploited by the white-box membership inference attack. No matter in the scenario of black-box setting or white-box setting, the majority of popular membership inference attacks require the construction of the attack model, which is used to extract features from the target models.

In FL, the participants and the server have access to the model architectures, which means that they can perform the white-box membership inference attacks. Consider a FL model $\boldsymbol{\theta}$ and target data record (x, y). The goal of the membership inference attack is to infer whether (x, y) is the training data of participants. The state-of-the-art white-box attack [14] is to train an attack model h using the target model's gradients, the loss of target model, and the outputs of model's hidden layers.

Let $\boldsymbol{\theta}$ be the target model and h: $F(X, Y, \boldsymbol{\theta}) \rightarrow [0, 1]$ be the attack model. In the white-box membership inference attack setting, given a data record (x, y), the attack model computes $F(X, Y, \boldsymbol{\theta})$ to infer whether the data record (x, y) is a member of training data or not. The input features of attack model h is a combination of different features of target model $\boldsymbol{\theta}$ related to (x, y), e.g., $\boldsymbol{\theta}$'s prediction on (x, y), $\boldsymbol{\theta}$'s gradients on (x, y), etc, denoted by $F(X, Y, \boldsymbol{\theta})$. The output of h is the probability that (x, y) is a member of $\boldsymbol{\theta}$'s training dataset based on the input vector $F(X, Y, \boldsymbol{\theta})$. Let $Pr_D(X, Y)$ and $Pr_{\bar{D}}(X, Y)$ be the conditional probabilities of the member and non-member, respectively. Given the above setting, the expected gain of attack model can be computed as:

$$\begin{aligned} G_{\boldsymbol{\theta}}(h) = {} & 0.5 \times E_{(x,y) \sim Pr_D(X,Y)}[log(h(F(x, y, \boldsymbol{\theta})))] \\ & + 0.5 \times E_{(x,y) \sim Pr_{\bar{D}}(X,Y)}[log(1 - h(F(x, y, \boldsymbol{\theta})))] \end{aligned} \tag{3}$$

For the adversary, the objective of attack model h is to maximize the privacy gain $G_{\boldsymbol{\theta}}(h)$.

C. Pruning Techniques

Considering that the SOTA (state of the art) deep learning networks consist of multiple layers and millions of parameters [32], pruning techniques play a popular role in the distributed deep learning for the sake of low computation

and communication cost. According to the different pruning objects, the pruning techniques are regarded as two types, i.e., model pruning and gradient sparsification. The main idea of the former is to prune unimportant model parameters based on their magnitudes [29], [30], while the main idea of the latter is to filter out some ignorable gradients based on their magnitudes [33]. Additionally, the pruning technique is considered as an effective privacy protection method. For instance, Wang et al. [27] proposed the defend scheme with the deep neural network (DNN) weight pruning against the black-box membership inference attack in the scenario of traditional centralised deep learning.

D. Defenses against Membership Inference Attacks

Related defending schemes perform two main ideas against the membership inference attack, namely mitigating overfitting and hiding the original model. Several solutions [10], [11], [17]–[20], [25] to resist the membership inference attack have been proposed, as shown in Table II.

Defenses based on mitigating overfitting. Many defenses are designed to reduce overfitting using regularization methods to protect membership privacy, since overfitting is one major reason why black-box membership inference attacks against the target global model are effective [34]. For instance, Shokri et al. [19] used conventional L2 regularization to mitigating membership privacy leakage from the global model. Nasr et al. [10] presented the machine learning with membership privacy using the min-max game, which is also called *adversarial regularization*. In addition, *dropout* [11], [35] and *model stacking* [11] have been also used to prevent overfitting to defend against the black-box membership inference attacks. Shejwalkar et al. [18] use *knowledge distillation* to defend against white-box and black-box membership inference attacks. However, some of these defenses e.g., adversarial regularization are not suitable for the resource-limited IoT scenario owing to the heavy computation overhead.

Defenses based on hiding the original model. Some defenses aim to hide the original models so that the membership inference attacker cannot infer useful information based on known knowledge. Several defenses [20], [21], [23] use cryptography techniques, i.e., Paillier, Learning with Error (LWE) to encrypt intermediate parameters and outputs in the training process of FL, preventing membership privacy leaks from local models through secure aggregation. Additionally, Wei et al. [36] design a FL framework with privacy guarantee for large-scale IoT devices with the secret sharing. However, these defenses introduce high communication and computation overhead to the scenario of IoT. Moreover, other defenses [24]–[26] leverage differential privacy (DP) [37] to perform privacy preservation. For instance, Wei et al. [25] introduce a novel FL framework with differential privacy called NbAFL, which adds artificial noise satisfying the Gaussian distribution to the local models before model aggregation. However, multiple studies [17], [18], [38] reveal that training models with differential privacy has a negative impact on the model accuracy.

Since FL based on IoT devices with limited resources requires high model accuracy, strong privacy protection and low overhead, these related defenses have certain flaws. Specifically, some i.e., differential privacy hurt the model accuracy. Some i.e., homomorphic encryption can only protect the local model in the training phase, ignoring the membership inference attack against the global model. Some i.e., adversarial regularization and homomorphic encryption result in heavy communication or computation overhead.

Novelty of proposed defense. MemDefense aims to defend against membership privacy leakage from local models and global model in IoT-based FL by adding crafted pruning perturbations during the training process. Since the added pruning perturbations make some model parameters zero and thus make the model sparser, our scheme achieves defense against the membership inference attack from the perspective of both mitigating overfitting and hiding the original model. With lower overhead, MemDefense drastically reduces the accuracy of membership inference attacks, while maintaining limited loss of test accuracy of target models. Compared with other defenses, it provides an excellent trade-off between the model utility, membership privacy and resource consumption.

III. Problem Formulation

In this section, we present a detailed description of the threat model and design goals.

A. Threat Model

In this paper, we assume that an adversary aims to infer the membership privacy from the shared local gradients in the training phase of FL. Following the common assumptions in the literatures [9], [14], [21], [23], [39], we consider the server and participants are considered to be honest-but-curious (i.e., semi-honest), meaning that they will execute the program according to the agreement but try to infer other participants' data privacy as much as possible. It is noted that we allow the server to collude with participants, getting the most offensive capabilities. Specifically, we summarize the adversary into three types, i.e., participant-only adversary, server-only adversary, and participants-server collusion adversary [23]. The capabilities of the three types of adversaries are as follows.

Type I: Server-only adversary. Due to the server's mission of aggregating local models received from all participants , the adversary that compromises the semi-honest server obtains the local models' gradients of all participants during the training phase of FL. It also knows the structure of the target models, meaning that the adversary can launch *white-box* membership inference attacks.

Type II: Participant-only adversary. The adversary that compromises the semi-honest participant has access to an auxiliary dataset with similar distribution to other participants' private datasets. It also knows the structure of the target models, meaning that the adversary can launch *white-box* membership inference attacks.

Type III: Participants-server collusion adversary. The adversary compromising the semi-honest server and p ($p \in [1, m-1]$) semi-honest participants has the most offensive attacking capabilities. It obtains the local models' gradients from all participants during the training phase of FL, knows the structure of the target models (i.e., can launch *white-box* membership inference attacks), and has access to semi-honest participants' auxiliary datasets with similar distribution to other participants' private datasets.

To defend against the membership inference attacks, the defender considers the attack scenarios with maximum membership privacy leakage, i.e., the adversary compromises the semi-honest server and p ($p \in [1, m-1]$) semi-honest participants. It is noted that if the defense mechanism can defend against the membership inference attacks by a strong adversary, it can effectively defend against the membership attacks by weaker adversaries.

B. Design Goals

An ideal defense should protect against the white-box membership inference attacks as depicted in *Threat Model*, while preserving the quality of the main task models. Therefore, the design goals can be described as follows.

- *High model utility*. In FL, the participants and the server are required to update the global model according to the local models to obtain a high-utility target model. Therefore, little accuracy loss should be considered.
- *Strong membership privacy*. The attack accuracy represents the membership privacy leakage risks. For the defender, the goal is to minimize the attack accuracy, namely, improving the privacy defense.
- *Acceptable time efficiency*. For the sake of alleviating any potential resource burden on IoT devices, it is crucial for MemDefense to ensure the time efficiency.

IV. Design of MemDefense

In this section, we introduce the MemDefense that defends against the local-model-based membership inference attacks in the training phase of FL and the global-model-based membership inference attacks in the prediction phase.

A. Overview

We present a high-level overview of MemDefense in Fig. 2. As mentioned above, the defender aims to add the pruning perturbations that can protect the membership privacy to the selected parameters of uploaded local models, while maintaining the high-utility of aggregated global models of FL.

To find the part of the model parameters that have little impact on the accuracy of the training task but have a large impact on the performance of membership inference attack, the participants filter model parameters in each training epoch, according to the local model update magnitude, which means the difference between the updated local model and the global model of last round. Instead of previous pruning techniques commonly used in other learning framework, namely, selecting gradients based on gradient magnitude [33] or selecting parameters based on parameter magnitude [27], our scheme selects parameters based on gradient magnitude. We focus on the traditional FL with FedAvg [5] as the aggregation rule, which is widely adopted by recent studies [5], [14], [24]. In this setting, the noise of aggregated global model is the average of the sum of local noises generated by all participants. To more strongly defend against the local-model-based membership inference attacks, each participant adds pruning noise to the locally selected model parameters, and then upload their noisy local model parameters to the server.

Algorithm 1 MemDefense Training

Input: The local model $\boldsymbol{\theta}$, the pruning threshold γ
Output: Noisy global model $\boldsymbol{\theta}_T$

```
for t = 0 to T do
    Server selects m participants and sends the current global model θ_{t-1} to them
    for i = 1 to m do
        θ_t^i ←LocalUpdates(θ_{t-1},i)
        /* Filtering the model parameters */
        θ_t^{i-sele} ←ParamFiltering(θ_{t-1}, θ_t^i, γ_i)
        /* Generating the pruning noise */
        n_t^i ← NoiseGenerating (θ_t^{i-sele}, |θ_t^i|)
        Participants upload local update θ_t^i + n_t^i to Server
    end for
    θ_t = (1/m) Σ_{i=1}^{m} (θ_t^i + n_t^i)
end for
Return θ_T
```

The MemDefense training algorithm is shown in Algorithm 1. Initially, the server selects m participants and sent the initialized model to them. During the T epochs of the training phase in FL, each participant independently trains the local model with its own local dataset. Then the local model and the global model last epoch are fed into function $ParamFiltering$ with the pruning threshold γ to select the model parameters which need to be added noise. After filtering the parameters, the function $NoiseGenerating$ are used to generate the pruning noise in order to make the selected model parameters zero. The reason why the selected model parameters are changed to 0 is to eliminate their influence on model training and membership inference attack as much as possible. Finally, participants upload the noisy local model to the server for the model aggregation. Formally, $\boldsymbol{\theta}_i^* = \boldsymbol{\theta}_i + \boldsymbol{n}_i$ denotes the noisy local model and $\boldsymbol{n}_i$ denotes the generated noise for i-th participant. The global model added noise $\boldsymbol{\theta}^*$ at each iteration of FL is calculated by the following Eq. (4):

$$\boldsymbol{\Theta}^* = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\theta}_i^* = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{\theta}_i + \boldsymbol{n}_i) \tag{4}$$

The training process is iterated until convergence, so the final output is the noise-added global model.

For such scenario, we design two components performing the functions $ParamFiltering$ and $NoiseGenerating$ respectively to defend against the local-model-based and global-

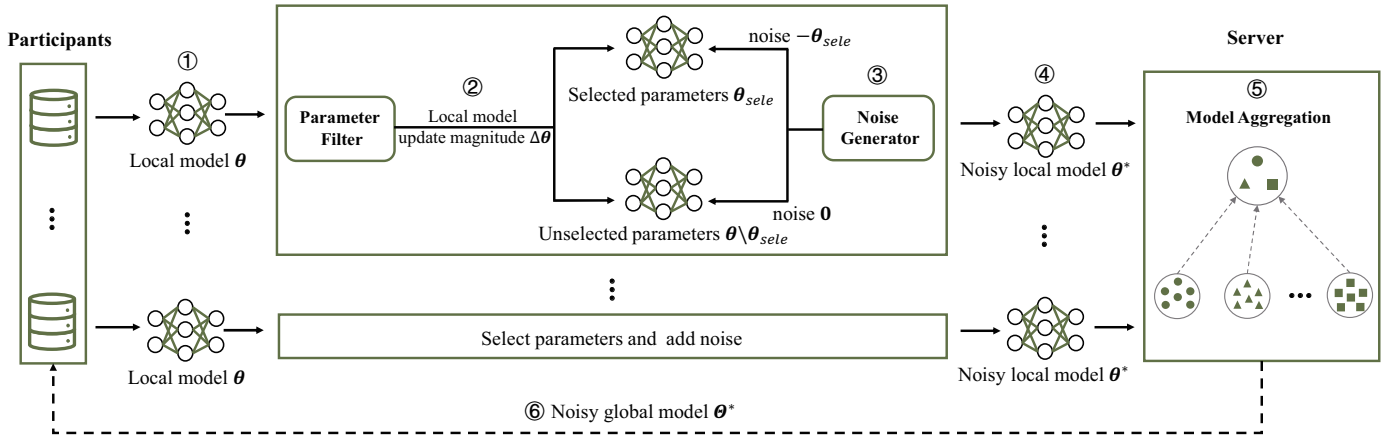


Fig. 2. The overview of the MemDefense.

model-based membership inference attacks, i.e., parameter filter and noise generator.

Parameter filter: the component finds some model parameters, which have little impact on the model accuracy of the training task, and contribute more to member inference attacks, according to the local model update magnitude. The pruning threshold γ is introduced to control the scaling of the filtering parameters. The detailed description about the parameter filter is given in the following subsection.

Noise generator: the component generates the pruning noises according to the parameters chosen by the parameter filter, to protect membership privacy. Specifically, it makes the noisy model parameters zero. The pruning noise can weaken the contribution of selected parameters to membership inference attacks in each training epoch without compromising the classification accuracy of tasks. The detailed noise generation algorithm is described below.

B. Parameter Filter

In this section, we first select some parameters, which do not affect the minimization of the loss function in the training task, but also ensure the maximization of the gain of member inference attacks. That is, if noise is added to these parameters, the accuracy of the training task will not be negatively affected, and the performance of membership inference attacks will also be greatly weakened. Therefore, we can formalize it as an optimization problem in the following:

$$\begin{aligned} &\min_{\boldsymbol{\theta}\backslash\boldsymbol{\theta}_{sele}} L(\boldsymbol{\theta} \setminus \boldsymbol{\theta}_{sele}) \\ &s.t. |\boldsymbol{\theta}_{sele}| = \min_{|\boldsymbol{\theta}_{sele}|} \{\max G_{\boldsymbol{\theta}\backslash\boldsymbol{\theta}_{sele}}(h)\}, \boldsymbol{\theta}_{sele} \subset \boldsymbol{\theta}, \end{aligned} \tag{5}$$

where $\boldsymbol{\theta}$ is the model trained on participant's local dataset and $\boldsymbol{\theta}_{sele}$ means the selected model parameters. Therefore, $\boldsymbol{\theta}\backslash\boldsymbol{\theta}_{sele}$ means unselected original model parameters. Considering that it is unselected model parameters $\boldsymbol{\theta} \setminus \boldsymbol{\theta}_{sele}$ that have influence on model training, Eq. (5) shows the goal to minimizes the loss function of the training task. To some degree, less selected parameters mean more unselected parameters, which means less negative impact on model training. Therefore, in the constraint of Eq. (5), the variable is the attack gain $G_{\boldsymbol{\theta}\backslash\boldsymbol{\theta}_{sele}}(h)$ described by Eq. (3) and the value is the minimum number of selected model parameters $|\boldsymbol{\theta}_{sele}|$. We should find the minimum number of model parameters that meet the requirements while maximizing the membership inference attack gain and ensure they are from the local model. Enlightened by the model pruning techniques [29], [30], we define the pruning threshold γ to control the number of selected model parameters. Instead of a specific model parameter value, γ represents the proportion of filtered parameters.

In order to choose the appropriate model parameters, we design a filter criterion by comparing the difference between the updated local model and the global model of last round for each participant, that is, the local model update magnitude, as is shown in Eq. (6):

$$\Delta\boldsymbol{\theta}_t \ = |\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_t|, \tag{6}$$

where $\Delta\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_t$ respectively represent the local model update magnitude and the local model in the t-th epoch, and $\boldsymbol{\theta}_{t-1}$ is the global model of last epoch.

The details of function $ParamFiltering$ are shown in Algorithm 2. During the training phase in FL, after training the local models on their own privacy datasets, participants calculate the local model update magnitudes $\Delta\boldsymbol{\theta}_t$ with the new local models in this round (i.e., t-th round) and the global model last round (i.e., $(t-1)$-th round). Then, model parameters are sorted by model update magnitude in ascending order. Next, depending on the pruning threshold γ, the participant selects a proportion of the desired model parameters. The lottery ticket hypothesis [40] proposes that a sub-network exists in a well-trained neural network model which can achieve performance comparable to that of the original model, according to the

Algorithm 2 ParamFiltering

Input: The last epoch global model $\boldsymbol{\theta}_{t-1}$, the local model $\boldsymbol{\theta}_t$, pruning threshold γ
Output: The selected model parameters $\boldsymbol{\theta}_t^{sele}$

/* Calculating the local model update magnitude */
$\Delta\boldsymbol{\theta}_t \leftarrow |\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_t|$
/* Sorting the parameters by model update magnitude in ascending order */
$\boldsymbol{\theta}_t \leftarrow sort(\boldsymbol{\theta}_t)$
/* Selecting the model parameters */
for $k = 0$ to $\gamma|\boldsymbol{\theta}_t|$ **do**
    $\boldsymbol{\theta}_t^{sele} \leftarrow [\boldsymbol{\theta}_t^k]$
end for
Return $\boldsymbol{\theta}_t^{sele}$

Algorithm 3 NoiseGenerating

Input: The selected model parameters $\boldsymbol{\theta}_{sele}$, the number of model parameters $|\boldsymbol{\theta}|$
Output: The pruning noise $\boldsymbol{n}$

for $k = 0$ to $|\boldsymbol{\theta}|$ **do**
    if k in $idx(\boldsymbol{\theta}_{sele})$ **then**
        $n_k = -\boldsymbol{\theta}_k^{sele}$
    else
        $n_k = 0$
    end if
end for
$\boldsymbol{n} = [n_k]$
Return $\boldsymbol{n}$

phenomenon that the compression technique can remove 90% of the model parameters while maintaining the same accuracy. Therefore, in the main experiments in Section VII, we set the pruning threshold to 0.9. Finally, the selected model parameters $\boldsymbol{\theta}_t^{sele}$ for t-th epoch can be obtained.

In this stage, we choose the part of the parameters with small update magnitudes, because the small change means the parameters are not active during the model training. Even if we inadvertently hurt an important parameter in a certain epoch, it will still resume its role in the next epoch with a more substantial change. By filtering out the model parameters meeting the requirements that the parameters are helpful to the membership inference attacks and have little impact on the training task model accuracy, we can design a more precise defense scheme with parameter-wise pruning perturbations.

C. Noise Generator

In this section, we then generate the pruning perturbations which are added to the local models trained by participants. Our goal is to find a noise vector $\boldsymbol{n}$ such that the utility loss of the target model is minimized and the accuracy of membership inference attack model h is around 50%, which means that the attack model cannot determine whether a target data record is a member or not. Formally, we aim to generate the noise via solving the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{n}} \ d(\boldsymbol{\theta}, \boldsymbol{\theta} + \boldsymbol{n}) \\ &s.t. |h(\boldsymbol{\theta} + \boldsymbol{n}) - 0.5| \leq \varepsilon, \end{aligned} \tag{7}$$

where $\boldsymbol{\theta}$ is the model trained on participant's private data; Eq. (7) represents that the noise added to the $\boldsymbol{\theta}$ minimizes the model accuracy loss. Meanwhile, we should ensure that the attack model outputs are around 0.5 (i.e., close to the random guessing), namely, the attack model cannot determine whether the target data record is a member of the training dataset or not. ε is the threshold that defenders can customize, which is set to 0.1 in this paper.

Therefore, in order to achieve the goals mentioned above, we design the pruning noise, aiming to make the model parameters selected by the parameter filter zero. With the pruning noise, we minimize the membership inference attack gain and maximize the utility of the training model. More details are described in Algorithm 3. In the training phase of FL, at each iteration t, the participant i trains the local model on their private data. According to the selected model parameters $\boldsymbol{\theta}_{sele}$ and the number of model parameters $|\boldsymbol{\theta}|$, the defender generates a pruning noise vector $\boldsymbol{n}$ by judging whether the model parameters belong to the set selected by the parameter filter or not. Specifically, if a model parameter belongs to the selected model parameters, then its corresponding noise value is the opposite number of the parameter itself, otherwise the corresponding noise is 0. Finally, for each participant, a noise vector $\boldsymbol{n}$ is generated.

The pruning perturbations are added to local model parameters to reduce the attack accuracy while keeping high model utility, which achieve strong membership privacy protection. The experimental results in Section VII also verify this point.

V. Datasets and Model architectures

In this section, we describe the datasets used to evaluate MemDefense and introduce the architectures of the target models and the membership inference attack models. Additionally, we present the experimental setting of other defenses.

A. Datasets

CIFAR-10. CIFAR-10 is an image dataset and contains 60,000 color (RGB) images. The number of training images is 50,000 and the number of testing images is 10,000. The image size is 32×32 pixels. In this dataset, the main task is to train a deep model for image recognition. Each class has 5,000 training and 1,000 test images.

CIFAR-100. CIFAR-100 is an image dataset which is widely used for evaluating image classification. It contains 60,000 color (RGB) images, including 50,000 images for training and 10,000 images for testing. The image size is 32×32 pixels. In this dataset, the main task is to train a deep learning model which can cluster the images into 100 classes. Each class has 500 training and 100 test images.

MNIST. MNIST is a digital image dataset and contains 70,000 greyscale images with 10 classes from number 0 to 9. The

TABLE III
DATASETS STATISTIC

Datasets	Target model		Global Attack model		Local Attack model	
	Training Data	Testing Data	Member Data	Non-member Data	Member Data	Non-member Data
CIFAR-10	50,000	10,000	5,000	5,000	1,250	1,250
CIFAR-100	50,000	10,000	5,000	5,000	1,250	1,250
MNIST	60,000	10,000	5,000	5,000	1,500	1,500
FashionMNIST	60,000	10,000	5,000	5,000	1,500	1,500

number of training images is 60,000 and the number of testing images is 10,000. The image size is 28×28 pixels. In this dataset, the main task is to train a deep learning neural network model for image recognition.

FashionMNIST. FashionMNIST is a fashion product image dataset and contains 70,000 greyscale images with 10 classes. The number of training images is 60,000 and the number of testing images is 10,000. The image size is 28×28 pixels. In this dataset, the main task is to train a deep neural network model for image recognition.

B. Model architectures

Target models. For the CIFAR-10 and the CIFAR-100 classification tasks, we use the AlexNet model that is widely used for image classification. For the MNIST and FashionMNIST datasets, we use the LeNet-5 model, which is also a popular deep learning network. For all datasets and model architecture combinations (i.e., AlexNet model trained on the CIFAR-10 dataset, AlexNet model trained on the CIFAR-100 dataset, LeNet-5 model trained on the MNIST dataset, and Lenet-5 model trained on the FashionMNIST dataset), we set the learning rate to 0.1, 0.01, 0.001 for 0-60, 60-90, 90-100 rounds accordingly. We set up five epochs for participants' local training at each round of FL. We use the ReLU as the activation function and Stochastic Gradient Descent (SGD) learning algorithm as optimizer for all models. The batch size is set to 128 for all models. There are 20 participants involved in the FL training in each round for all kinds of datasets.

Membership inference attack model. We use the state-of-the-art membership inference attack model proposed by Nasr et al. [14] and implement the white-box membership inference attacks based on a public code[1], which can attack the local models and global model in FL. For all target models to be attacked, we use the gradients of their last layer and the output of their last two layers as the input features of the attack model. The label is set to 1 if a piece of data is a member of the target model's training dataset, otherwise the label is 0. Following the settings of Nasr et al. [14], the gradients are fed into a convolutional neural network and a fully connected network to get 64-dimensional features and the output of last two layers is fed into a fully connected network to get two 64-dimensional features. Then we concatenate these features to get a 192-dimensional feature vector and feed it into a fully connected network with three hidden layers of sizes {256, 128, 64} to get the final prediction result. We use the ReLU as activation function, and Adam optimizer for all attack models. The learning rate is 0.0001 and the output of the attack model is a sigmoid layer. Additionally, when the local models are the targets of the membership inference attack, we choose the local models in the 5 epochs [40, 60, 80, 90, 100] in the training phase of FL. Therefore, the input of the attack model is correspondingly enlarged 5 times.

C. Experimental settings of other defenses

Dropout. In all experiment tasks based on the datasets CIFAR-10, CIFAR-100, MNIST and FashionMNIST, the dropout rate is set to 0.5.

L2 regularization. In all experiments based on the datasets CIFAR-10, CIFAR-100, MNIST and FashionMNIST, the L2 generalization factor is set to 0.001.

Adversarial regularization. Adversarial regularization (Adv-Reg) is implemented based on a public code[2], for which the Adv-Reg factor λ is set to 3 in terms of the tasks based on all the datasets. Considering that the adversarial regularization is suitable for the centralized scenario, it cannot protect local models in the training phase of FL. Therefore, we cannot obtain the local model attack accuracy A_{att_l} with Adv-Reg.

Differential privacy. Differential privacy (DP) adopts an open-source version of NbAFL's implementation[3]. To find proper parameter settings of NbAFL, we set the parameter δ to 0.00001 and clipping threshold C to 30 firstly according to the official recommendation. Then, with the Gaussian noise, we try different global privacy budgets ϵ, which are finally set to 8000, 8000, 25 and 30, for the tasks based on datasets CIFAR-10, CIFAR-100, MNIST and FashionMNIST, respectively.

Homomorphic encryption. In all experiments with homomorphic encryption (HE), we use Paillier as the cryptography algorithm in the phe[4] and set the secret key length to 1024 bit. Since local models from participants are transmitted to the server in ciphertext, which cannot be used by attackers to perform membership inference attacks, the local model attack

[1]https://github.com/SPIN-UMass/MembershipWhiteboxAttacks

[2]https://github.com/SPIN-UMass/ML-Privacy-Regulization

[3]https://github.com/AdamWei-boop/Federated-Learning-with-Local-Differential-Privacy

[4]https://github.com/conda-forge/phe-feedstock

TABLE IV
THE EXPERIMENTAL RESULTS OF NO DEFENSE, MEMDEFENSE AND COMPARISON DEFENSES

Defenses	CIFAR-10				CIFAR-100				MNIST				FashionMNIST			
	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}
No Defense	98.58%	88.68%	63.56%	61.64%	95.14%	48.61%	77.78%	68.28%	98.96%	98.61%	53.46%	62.50%	91.62%	88.69%	53.90%	53.87%
Dropout [11]	91.52%	87.77%	63.83%	59.48%	79.79%	54.40%	62.77%	**58.08%**	99.69%	99.24%	52.54%	61.87%	90.60%	88.14%	50.94%	51.20%
L2 [19]	97.31%	**91.56%**	60.71%	60.40%	91.72%	60.88%	66.47%	63.28%	98.88%	98.52%	53.66%	63.97%	**94.39%**	**89.91%**	53.05%	51.77%
Adv-Reg [10]	96.78%	82.37%	60.22%	–	**99.83%**	57.54%	64.58%	–	**100.00%**	**99.20%**	53.75%	–	94.31%	88.83%	51.47%	–
NbAFL [25]	97.95%	91.31%	57.88%	59.44%	84.75%	**64.41%**	59.60%	62.72%	96.02%	96.47%	53.11%	64.57%	85.28%	84.18%	51.87%	**50.30%**
HE [20]	**99.09%**	88.54%	60.43%	–	96.52%	48.58%	78.77%	–	99.69%	98.61%	53.30%	–	91.84%	88.32%	53.82%	–
MemDefense	95.83%	89.78%	**56.23%**	**59.24%**	79.66%	52.56%	**59.12%**	59.76%	99.23%	98.89%	**50.74%**	**59.07%**	89.36%	87.97%	**50.54%**	51.10%

TABLE V
THE EXPERIMENTAL RESULTS OF THE TIME COST WITH MEMDEFENSE, OTHER DEFENSES AND NO DEFENSE

Training time(s)	CIFAR-10	CIFAR-100	MNIST	FashionMNIST
No Defense	4.58	5.11	2.32	2.91
Dropout [11]	5.25 (1.1×)	5.12 (1.0×)	2.38 (1.0×)	2.58 (0.9×)
L2 [19]	5.26 (1.1×)	5.12 (1.0×)	2.35 (1.0×)	2.65 (0.9×)
NbAFL [25]	6.21 (1.4×)	5.99 (1.2×)	2.52 (1.1×)	2.87 (1.0×)
HE [20]	13114.20 (2863.4×)	13124.50 (2568.4×)	100.40 (43.3×)	94.22 (32.4×)
MemDefense	6.36 (1.4×)	6.16 (1.2×)	2.98 (1.3×)	2.89 (1.0×)

accuracy A_{att_l} cannot be obtained. However, the homomorphic encryption focuses on protecting the local models of the participants in the training process, which cannot protect the global model in the prediction phase.

To evaluate the membership leakage risk, we use the global model attack accuracy A_{att_g} (i.e., the percentage of the corrected labels predicted by the attack model when the attack goal is the global model) and the local model attack accuracy A_{att_l} (i.e., the percentage of the corrected labels predicted by the attack model when the attack goal is the local models) as the evaluation metrics.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of MemDefense and compare our scheme with other current popular defenses. The detailed experimental settings for the dataset allocation about the target model and attack model are described in Table III. The member data and non-member data of the attack model belong to the target model's training dataset and testing dataset, respectively. Specifically, the adversary labels the data samples used for training target model as member and the data samples used for testing as non-member. After that, the adversary trains the attack model using both member and non-member data samples to launch the membership inference.

We carry out our experiments on four typical datasets with the four main aims: (i) evaluating the effectiveness and time efficiency of MemDefense with reference to no defense; (ii) comparing the performance of MemDefense with existing other defenses; (iii) analyzing the reason why MemDefense works; (iv) assessing the performance of MemDefense with different parameter settings.

A. Comparison with No Defense

In this section, we evaluate MemDefense on the effectiveness and time efficiency by comparing it with no defense.

Effectiveness. As Table IV shows, compared with no defense, MemDefense can protect the membership privacy of participants, while maintaining high model utility. Therefore, we can conclude them in two points:

1) MemDefense keeps the model utility within a high level. Compared with the global model trained without defense, the test accuracy A_{test} of the global model trained with MemDefense is only reduced by a maximum of 0.72% (FashionMNIST).
2) MemDefense achieves strong defense ability. With MemDefense, the global model attack accuracy A_{att_g} and the local model attack accuracy A_{att_l} can be respectively reduced to 50.54% and 51.10% (FashionMNIST) at most, greatly close to the result of random guessing.

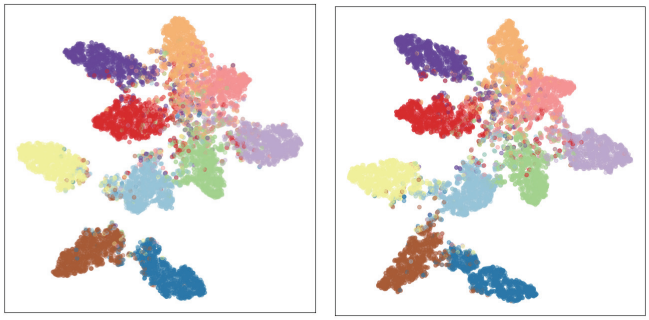
It is noted that the global model attack results are obtained by attacking the final model aggregated by the central server, and the local model attack results are obtained by attacking the local models sent to the server by the participants. Obviously, since the defense scheme imposed on the local models further affect the global model, MemDefense robustly defends against the membership inference attacks in both the training phase and the prediction phase, showing an excellent ability to protect the membership privacy of participants in FL.

Time efficiency. Table V shows the time costs per training epoch for each participant in different classification tasks with different methods, which demonstrates that the time overhead of MemDefense is 1.2× that of no defense on average, acceptable to all participants.

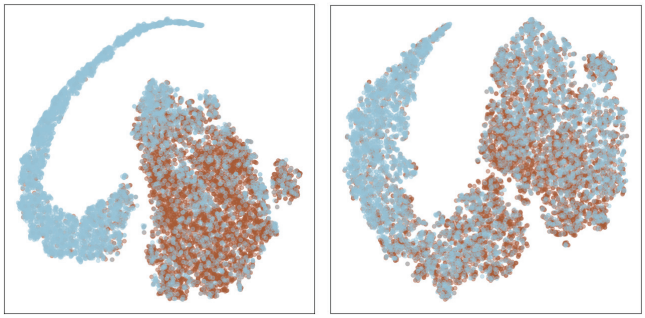
B. Comparison with Other Defenses

In this section, we compare MemDefense with the state-of-the-art defense methods, i.e., dropout [11], L2 regularization [19], adversarial regularization (Adv-Reg) [10], differential privacy (NbAFL) [25] and homomorphic encryption (HE) [20] on the effectiveness and time efficiency.

Effectiveness. As shown in Table IV, the experimental results on various datasets with MemDefense and other existing defenses present several critical observations:



(a) MT, CIFAR-10, No defense (b) MT, CIFAR-10, Memdefense



(c) MIA, CIFAR-100, No defense (d) MIA, CIFAR-100, MemDefense

Fig. 3. The t-SNE projection of data representations of the model training and the membership inference attack, where different color represents different classes of datasets.

1) MemDefense can maintain high model utility, similar to most existing defenses. Specifically, compared with the accuracy loss from 0.03% to 6.31% with other defenses, resulting in a loss of only 0.72% on FashionMNIST, MemDefense brings zero accuracy loss on datasets CIFAR-10, CIFAR-100 and MNIST.
2) The defensive effect of MemDefense is similar to that of the existing defenses, and even better than them. In detail, Memdefense outperforms the existing defenses on datasets CIFAR-10 and MNIST, in terms of both the global model attack and the local model attack. On CIFAR-100, the local attack accuracy A_{att_l} using MemDefense is close to that using dropout (59.76% vs. 58.08%) and on FashionMNIST, the local attack accuracy A_{att_l} using MemDefense is similar to that using NbAFL (51.10% vs. 50.30%).
3) The performance of MemDefense is hardly affected by the degree of model generalization. For instance, for the CIFAR-100 dataset with models that generalize poorly, MemDefense can keep the trade-off between model accuracy and privacy preservation. For the MNIST dataset with the well-generalized models, MemDefense can still keep high test accuracy A_{test} and effectively achieve the lowest global model attack accuracy A_{att_g} (50.74%) and local model attack accuracy A_{att_l} (59.07%), while the defensive capabilities of dropout, L2 regularization and NbAFL are mediocre.
4) MemDefense achieves the maximum span of reducing the attack accuracy. Specifically, the global model attack accuracy A_{att_g} is decreased by 18.66% (CIFAR-100).

Time efficiency. We compare the time overhead of MemDefense with that of other defense methods. Since the adversarial regularization [10] is used in the centralized scenario and its costs are considerably heavy, for fair comparison, we only compare the time overhead of MemDefense with that of dropout [11], L2 regularization [19], differential privacy [25] and HE [20]. Table V show that our scheme has similar time overhead to other schemes except homomorphic encryption. Moreover, since the homomorphic encryption requires a large amount of ciphertext calculations and consumes considerably time and computing power, which brings a burden to IoT devices with limited resources, our solution is far superior to homomorphic encryption in terms of cost.

C. The Reason Why MemDefense Works

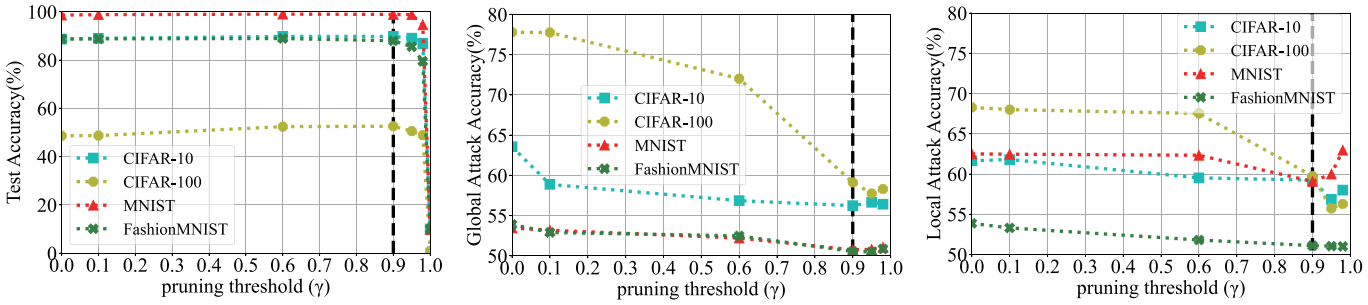
In this section, we analyze the reason why MemDefense works and use the technique t-SNE (t-Distributed Stochastic Neighbor Embedding) [41] to visualize the results.

The research [42] shows that in the distributed SGD algorithm, most gradient exchanges are redundant. Intuitively, a larger parameter update magnitude means that the parameter is more active during the model training, which contains more information. Instead, a smaller parameter update magnitude means that the corresponding parameter is likely to be unimportant in the training phase. Therefore, adding noise to most parameters with smaller update magnitudes to make them zero will have no impact on the minimization of the loss function during the training process. Furthermore, a small parameter update magnitude may also indicate that the model has effectively learned from the training data. However, the situation also raises the possibility of overfitting, where the model becomes overly specialized to the training data, making it easier to distinguish between the training and test data. The distinction, in turn, benefits potential adversaries aiming to successfully perform the membership inference attack. Therefore, setting the model parameters with small update magnitudes to zero can be regarded as a measure to mitigate the execution of membership inference attacks.

To prove the above statements, we use the visualization technique t-SNE [41] to present the impact on the model training and the membership inference attack after adding noise to the parameters with small update magnitudes to make them zero. As shown in Fig. 3, the results describe two important observations:

1) MemDefense has no negative impact on minimizing the loss function of the classification task. As shown in Fig. 3(a) and Fig. 3(b), the representation clustering result of MemDefense based on the model training (MT) with dataset CIFAR-10 is similar to that of no defense, proving that MemDefense do not damage the indicative representations of data features.
2) MemDefense has a curbing effect on maximizing the gain of the membership inference attack (MIA). As shown

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3403388



(a) Test accuracy A_{test} (b) Global model attack accuracy A_{att_g} (c) Local model attack accuracy A_{att_l}

Fig. 4. The impact of different pruning thresholds on the test accuracy, global model attack accuracy and local model attack accuracy.

in Fig. 3(c) and Fig. 3(d), compared with no defense, MemDefense blurs the boundaries between member and non-member data clusters based on CIFAR-100, proving that MemDefense protects member features.

D. The Impact of the Parameter Settings

In this section, we evaluate the impact of other parameter settings, namely, the pruning thresholds and the number of participants, on the performance of MemDefense.

The impact of pruning thresholds. In MemDefense, we choose a series of pruning thresholds γ from [0, 1] to select the local model parameters with the parameter filter. Then, we evaluate the impact generated by the different pruning thresholds γ on the performance of MemDefense, including the test accuracy A_{test}, the global model attack accuracy A_{att_g} and the local model attack accuracy A_{att_l}.

The experimental results are shown in Fig. 4, which describe three important observations:

1) The pruning threshold γ has influence on the model accuracy with MemDefense. As is shown in Fig. 4(a), within a reasonable range, as the pruning threshold increases, the model test accuracy A_{test} will not be compromised. However, when the pruning threshold is more than 0.9 and gets closer to 1, the model accuracy decreases.
2) The pruning threshold γ has an impact on the defense effect of MemDefense against the membership inference attacks. According to Fig. 4(b) and Fig. 4(c), on different datasets, the attack accuracy A_{att_g} and A_{att_l} decrease as the pruning threshold increases, especially when the threshold γ is smaller than 0.9.
3) When the pruning threshold γ is set to 0.9, the trade-off between the model utility and defense capability is optimal. Specifically, on different datasets, MemDefense with $\gamma = 0.9$ achieves outstanding model test accuracy with low attack accuracy.

The impact of the number of participants. We show the evaluation of MemDefense with changing the number of participants, namely 10, 20 and 50. In order to evaluate the effectiveness and low-accuracy overhead of MemDefense with various numbers of participants involved in the training phase of FL, we separately measure the train accuracy A_{train}, the test accuracy A_{test}, the global model membership inference attack accuracy A_{att_g} and the local model membership inference attack accuracy A_{att_l} without defense and with MemDefense for all experiment tasks based on the datasets CIFAR-10, CIFAR-100, MNIST and FashionMNIST.

From the related experiment results in Table VI, two key observations can be concluded:

1) MemDefense ensures the acceptable model utility, with the number of participants increasing. Specifically on the MNIST dataset, compared with no defense, MemDefense maintains the model test accuracy A_{test} above 98%, bringing no loss of accuracy to the model.
2) MemDefense achieves defending against membership inference attacks with different numbers of participants participating in the training phase and prediction phase in FL. Especially, on the FashionMNIST dataset, regardless of the number of participants, MemDefense makes the global model attack accuracy A_{att_g} and local model attack accuracy A_{att_l} as close to 50% as possible.

VII. Summary and Future Work

In this paper, we proposed MemDefense that leveraged crafted pruning perturbations to defend against the membership inference attacks in the training and prediction phase of FL. Additionally, considering the practical constraints of large-scale devices and limited resources in the IoT scenario, our scheme can achieve both practicality and lightweight. The experimental results showed that MemDefense could effectively defend against the local-model-based and global-model-based membership inference attacks, while keeping high model utility and ensuring time efficiency.

In the future, there are still some open problems in the IoT-based Federated Learning to be solved. We will focus on the novel noise design and defense scenario expansion. Specifically, it is still challenging to conceive theory-oriented and efficient noise generation methods. Moreover, besides the membership inference attack, other privacy threats i.e., data reconstruction attack should also be considered as the target for designing the defense mechanism.

TABLE VI
THE IMPACT OF THE NUMBER OF PARTICIPANTS

CIFAR-10	$N=10$				$N=20$				$N=50$			
	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}
No Defense	99.48%	90.27%	58.55%	63.40%	98.58%	88.68%	63.56%	61.64%	94.96%	89.28%	57.16%	55.90%
MemDefense	97.69%	90.11%	56.97%	60.68%	95.83%	89.78%	56.23%	59.24%	91.02%	86.16%	54.52%	52.80%
CIFAR-100	$N=10$				$N=20$				$N=50$			
	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}
No Defense	98.68%	50.57%	76.60%	68.88%	95.14%	48.61%	77.78%	78.28%	78.51%	41.53%	73.78%	60.10%
MemDefense	76.88%	57.00%	59.50%	58.92%	79.66%	52.56%	59.12%	59.76%	42.54%	33.57%	55.93%	55.20%
MNIST	$N=10$				$N=20$				$N=50$			
	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}
No Defense	99.49%	98.62%	52.73%	60.08%	98.96%	98.61%	53.46%	62.50%	99.18%	98.58%	52.77%	66.00%
MemDefense	99.31%	99.06%	51.65%	57.20%	99.23%	98.89%	50.74%	59.07%	99.11%	98.71%	51.81%	56.17%
FashionMNIST	$N=10$				$N=20$				$N=50$			
	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}	A_{train}	A_{test}	A_{att_g}	A_{att_l}
No Defense	92.28%	88.49%	54.56%	54.77%	91.62%	88.69%	53.90%	53.87%	93.60%	88.49%	52.48%	53.50%
MemDefense	86.44%	85.30%	50.23%	51.25%	89.36%	87.97%	50.54%	51.10%	89.66%	88.03%	50.88%	50.67%

REFERENCES

[1] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated learning for the internet of things: Applications, challenges, and opportunities," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 24–29, 2022.

[2] M. Shen, J. Wang, H. Du, D. Niyato, X. Tang, J. Kang, Y. Ding, and L. Zhu, "Secure semantic communications: Challenges, approaches, and opportunities," *IEEE Network*, 2023.

[3] N. Zainuddin, M. Daud, S. Ahmad, M. Maslizan, and S. A. L. Abdullah, "A study on privacy issues in internet of things (iot)," in *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*. IEEE, 2021, pp. 96–100.

[4] M. Shen, H. Liu, L. Zhu, K. Xu, H. Yu, X. Du, and M. Guizani, "Blockchain-assisted secure device authentication for cross-domain industrial iot," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, 2020.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, ser. Proceedings of Machine Learning Research, vol. 54. PMLR, 2017, pp. 1273–1282.

[6] A. Jochems, T. M. Deist, J. Van Soest, M. Eble, P. Bulens, P. Coucke, W. Dries, P. Lambin, and A. Dekker, "Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept," *Radiotherapy and Oncology*, vol. 121, no. 3, pp. 459–467, 2016.

[7] U. M. Aïvodji, S. Gambs, and A. Martin, "Iotfla: A secured and privacy-preserving smart home architecture implementing federated learning," in *2019 IEEE security and privacy workshops (SPW)*. IEEE, 2019, pp. 175–180.

[8] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7702–7712, 2019.

[9] 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.

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

[11] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019.

[12] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," *arXiv preprint arXiv:1807.00459*, 2018.

[13] M. Shen, H. Wang, B. Zhang, L. Zhu, K. Xu, Q. Li, and X. Du, "Exploiting unintended property leakage in blockchain-assisted federated learning for intelligent edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2265–2275, 2020.

[14] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pp. 739–753.

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

[16] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "Understanding membership inferences on well-generalized learning models," *arXiv preprint arXiv:1802.04889*, 2018.

[17] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. ACM, 2019, pp. 259–274.

[18] V. Shejwalkar and A. Houmansadr, "Reconciling utility and membership privacy via knowledge distillation," *CoRR*, vol. abs/1906.06589, 2019. [Online]. Available: http://arxiv.org/abs/1906.06589

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

[20] 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.

[21] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 911–926, 2020.

[22] X. Tang, M. Shen, Q. Li, L. Zhu, T. Xue, and Q. Qu, "Pile: Robust privacy-preserving federated learning via verifiable perturbations," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[23] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

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

[25] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[26] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: Federated learning with local differential privacy," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 61–66.

[27] Y. Wang, C. Wang, Z. Wang, S. Zhou, H. Liu, J. Bi, C. Ding, and S. Rajasekaran, "Against membership inference attack: Pruning is all you need," in *International Joint Conference on Artificial Intelligence*, 2021.

[28] X. Yuan and L. Zhang, "Membership inference attacks and defenses in neural network pruning," in *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, K. R. B. Butler and K. Thomas, Eds. USENIX Association, 2022, pp. 4561–4578. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/yuan-xiaoyong

[29] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," vol. 28, 2015.

[30] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient dnns," *Advances in neural information processing systems*, vol. 29, 2016.

[31] 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.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," vol. 30, 2017.

[33] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 440–445.

[34] 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.

[35] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2670313

[36] Z. Wei, Q. Pei, N. Zhang, X. Liu, C. Wu, and A. Taherkordi, "Lightweight federated learning for large-scale iot devices with privacy guarantee," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3179–3191, 2023.

[37] C. Dwork, "Differential privacy," *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

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

[39] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du, and M. Guizani, "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1229–1241, 2020.

[40] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2018.

[41] G. Hinton and L. van der Maaten, "Visualizing data using t-sne journal of machine learning research," 2008.

[42] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Meng Shen (Member, IEEE) received the B.Eng degree from Shandong University, Jinan, China in 2009, and the Ph.D degree from Tsinghua University, Beijing, China in 2014, both in computer science. Currently he serves in Beijing Institute of Technology, Beijing, China, as a professor. His research interests include privacy protection for cloud and IoT, blockchain applications, and encrypted traffic classification. He received the Best Paper Runner-Up Award at IEEE IPCCC 2014. He is a member of the IEEE.

Jin Meng received the B.Eng degree in computer science and technology from Qinghai University, Xining, China in 2022. She is currently pursuing the master's degree with the Department of Cyberspace Security, Beijing Institute of Technology. Her research interests include Federated Learning and Artificial Intelligence Security.

Ke Xu (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, where he serves as a Full Professor. He has published more than 200 technical papers and holds 11 U.S. patents in the research areas of next generation Internet, blockchain systems, Internet of Things, and network security. He has guest-edited several special issues in IEEE and Springer Journals. He is an Editor of IEEE INTERNET OF THINGS JOURNAL. He is also the Steering Committee Chair of IEEE/ACM IWQoS. He is a member of ACM.

Shui Yu (Fellow, IEEE) received the Ph.D. degree from Deakin University, Australia, in 2004. He is currently a Professor with the School of Computer Science, University of Technology Sydney, Australia. He has published four monographs and edited two books, more than 400 technical papers, including top journals and top conferences, such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, and INFOCOM. His h-index is 66. He initiated the research field of networking for big data in 2013, and his research outputs have been widely adopted by industrial systems, such as Amazon cloud security. His research interest includes big data, security and privacy, networking, and mathematical modeling. He is currently serving a number of prestigious editorial boards, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (Area Editor), IEEE Communications Magazine, and IEEE INTERNET OF THINGS JOURNAL. He served as a Distinguished Lecturer of IEEE Communications Society from 2018 to 2021. He is a Distinguished Visitor of IEEE Computer Society, a Voting Member of IEEE ComSoc Educational Services Board, and an Elected Member of Board of Governor of IEEE Vehicular Technology Society.

Liehuang Zhu (Senior Member, IEEE) received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004. He is currently a Professor and the Dean with the School of Cyberspace Science and Technology, Beijing Institute of Technology. He has published more than 100 peer-reviewed journal or conference papers, including 10 more IEEE/ACM Transactions papers. His research interests include security protocol analysis and design, wireless sensor networks, and cloud computing. Prof. Zhu has been granted a number of IEEE Best Paper Awards, including IWQoS 17' and TrustCom 18'.