Ensemble Federated Adversarial Training with Non-IID data

Shuang Luo, **Didi Zhu**, **Zexi Li**, **Chao Wu**[*]

Zhejiang University

{luoshuang, didi_zhu, zexi.li, chao.wu}@zju.edu.cn

Abstract

Despite federated learning endows distributed clients with a cooperative training mode under the premise of protecting data privacy and security, the clients are still vulnerable when encountering adversarial samples due to the lack of robustness. The adversarial samples can confuse and cheat the client models to achieve malicious purposes via injecting elaborate noise into normal input. In this paper, we introduce a novel Ensemble Federated Adversarial Training Method, termed as EFAT, that enables an efficacious and robust coupled training mechanism. Our core idea is to enhance the diversity of adversarial examples through expanding training data with different disturbances generated from other participated clients, which helps adversarial training perform well in Non-IID settings. Experimental results on different Non-IID situations, including feature distribution skew and label distribution skew, show that our proposed method achieves promising results compared with solely combining federated learning with adversarial approaches.

1 Introduction

Federated learning is a general distributed framework that can train large-scale distributed deep learning models with a federation of participants[McMahan *et al.*, 2017]. The central server will randomly select several clients meeting eligibility requirements in each round and broadcast the model parameters to these selected clients. Each selected client locally computes an update based on the global model with its local dataset and then send their model parameters to the server. The server then collects an aggregate of these updated models. It locally updates the shared model based on the aggregated update computed from the clients that participated in the current round. As this process occurs multiple times iteratively, all clients collectively train the centralized shared model. During this process, clients keep their private training datasets locally throughout, thereby ensuring a basic level of privacy.

Several motivating applications of federated learning in security territories have been extensively used, including human trajectory prediction [Feng *et al.*, 2020], visual inspection task [Han *et al.*, 2019], medical disease prediction [Feki *et al.*, 2021], etc. These domains attach high importance to data privacy and emphasize model reliability and robustness. However, although federated learning protects the security of data privacy, it is shown that clients' local deep neural network models are still vulnerable to different attacks.

Specifically, these attack approaches can be broadly categorized into two classes: backdoor attacks during training time and adversarial attacks during inference time. The goal of the backdoor attacks is to damage the performance of the model on targeted tasks while maintaining good performance on the main task by injecting "poison" training data. On the other hand, adversarial attacks aim at misleading the model to misclassify the well-designed inputs called adversarial examples, which are nearly indistinguishable from raw data in human eyes but fool the trained model. This kind of attack demonstrates that these networks perform computations that are dramatically different from those in human brains. The adversary adds small perturb actions to the natural datasets that lead these systems into making incorrect predictions to achieve the goal. While the perturbations are often imperceptible or perceived as small "noise" in the dataset, these attacks are highly effective against the deep neural network.

Up to now, a range of backdoor attacks and defense methods in federated learning settings have been introduced in previous literature, while how to defend against adversarial attack is worth considering here due to the security threat. In this paper, we are mainly concerned about improving the robustness of each node during inference time. It is still an open question whether federated learning systems can be tailored to be robust against adversarial attacks.

Adversarial attacks can be broadly classified into two types based on the knowledge of the attacked model, white-box or black-box attacks. Under white-box attacks, adversaries are necessary to have complete knowledge of the policy network, whereas black-box attacks require only access to the target model label predictions which are more applicable in many scenarios. The most simple yet efficient approach to perform defense is adversarial training, which injects adversarial examples into training data to fine-tuning network parameters. Nevertheless, solely adapting adversarial training to feder-

---

[*]Corresponding author

ated learning brings a range of problems. General adversarial training was developed primarily for IID data, while in federated learning, each client's data distribution is in non-IID settings. The mechanism of adversarial training in federated learning remains to be studied.

We propose the Ensemble Federated Adversarial Training (EFAT) method to improve the robustness of models against black-box attacks with non-IID training data to attack the above problems. How to resist white-box attacks is not the focus of our research because, in practice, the attacker does not know the specific parameters of the model in normal conditions. In the setting of EFAT, the central server first pretrains the initial model on the labeled public dataset and distributes the model and the parts of the public dataset to each client. Then each client generates adversarial examples based on their public data and exchanges their adversarial examples with others during the training process. Each client performs ensemble adversarial training using their training sets and adversarial examples generated by itself and the other locally distributed public data. During this process, each client is both an attacker as well as a defender. Thus, EFAT improves the robustness of clients against adversarial attacks by enhancing the adversarial data distribution diversity.

Contributions. To the best of our knowledge, our work is the first to enhance the robustness of adversarial training in the federated learning setting by taking advantage of improving adversarial data diversity between models from distributed clients. In summary, our contributions include the following:

- We explore the impact of adversarial training on the federated training paradigm and find it plays an important role. To this end, we develop a novel ensemble federated adversarial training (EFAT) methodology by incorporating adversarial examples generated by other clients' models to improve each client's robustness.

- Building on the above insight, we demonstrate our methodology's effectiveness and robustness against black-box attacks during inference-time on two kinds of Non-IID settings, including feature distribution skew and label distribution skew. The evaluation result shows that EFAT reaches higher adversarial accuracy on both Digit-Five and CIFAR10 than baseline.

2 Related work

Federated Learning. Federated learning has gained increasing attention in recent years due to its role in privacy protection [Li *et al.*, 2020]. One of the most common approaches to optimizing federated learning is the Federated Averaging algorithm [McMahan *et al.*, 2017], which combines local stochastic gradient descent (SGD) on each client with a server that performs model weighted averaging with weights proportional to the size of each client's local data. Secure aggregation (i.e. SecAgg) [Bonawitz *et al.*, 2017] is a tool used to ensure that the server only sees an aggregate of the client updates, not any individual client updates during FedAvg. Several alternative aggregation schemes to address this challenge have been proposed recently due to the directly

weighted averaging of model parameters that may have some adverse effects on model performance.[Wang *et al.*, 2020]

Adversarial Attack. Adversarial attacks refer to any alteration of the training and inference pipelines of a federated learning system designed to degrade model performance somehow. Adversarial attacks can be broadly classified into training-time attacks and inference-time attacks[Kairouz *et al.*, 2019].

Training-time attacks can be further classified into data poisoning [Bagdasaryan *et al.*, 2020] and model update poisoning [Bhagoji *et al.*, 2019; Szegedy *et al.*, 2013] based on the adversary's capability. Unlike data poisoning attacks, model update poisoning attacks can directly corrupt derived quantities within the learning system.

We will concentrate on inference-time attacks and methods to defend them in this paper. Inference-time attacks generally refer to adversarial examples [Szegedy *et al.*, 2013; Goodfellow *et al.*, 2014] that will be purposefully misclassified at runtime. Different from poisoning attacks, adversarial examples compromise the testing phase of machine learning. In these attacks, an adversary may attempt to circumvent a deployed model by carefully manipulating samples fed into the model. These are a perturbed version of test inputs that looks and feels the same as their original test inputs to a human, but that completely throws off the classifier [Goodfellow *et al.*, 2014]. The perturbations mentioned above can be generated by maximizing the loss function subject to a norm constraint via constrained optimization methods based gradient [Goodfellow *et al.*, 2014; Madry *et al.*, 2018]. In the context of l_∞ -bounded attacks, the Fast Gradient Sign Method(FGSM) [Goodfellow *et al.*, 2014] is one of the most popular methods using a single gradient step to perturb the inputs fed to the model. Later, the Basic Iterative Method(BIM) [Kurakin *et al.*, 2016] have been proposed, which is improved upon FGSM by applying the same step as FGSM multiple times with a small step size. The Projected Gradient Descent (PGD) attack, a variant of BIM, further strengthens this iterative adversarial attack by initializing examples to a random point in the ball of interest and adding multiple random restarts. Such attacks can frequently cause naturally trained models to achieve zero accuracies on image classification benchmarks such as CIFAR10 or ImageNet [Carlini and Wagner, 2017], which is recognized to be one of the most potent first-order attacks.

Adversarial Training. Adversarial training was first proposed by [Goodfellow *et al.*, 2014], in which produced adversarial examples and injected them into original samples to strengthen a model. The robustness against white-box attacks achieved by adversarial training depends on the strength of the adversarial examples used. Intuitively, adversarially trained models with FGSM or R+FGSM adversaries are only robust to single-step perturbations but remains vulnerable to more costly multi-step attacks [Madry *et al.*, 2018]. To this end, adversarial training with a PGD adversary has been proposed to tackle this challenge. Since then, the PGD based adversarial training has been enhanced through various techniques, such as optimization tricks like momentum to improve the adversary, combination with other heuristic de-
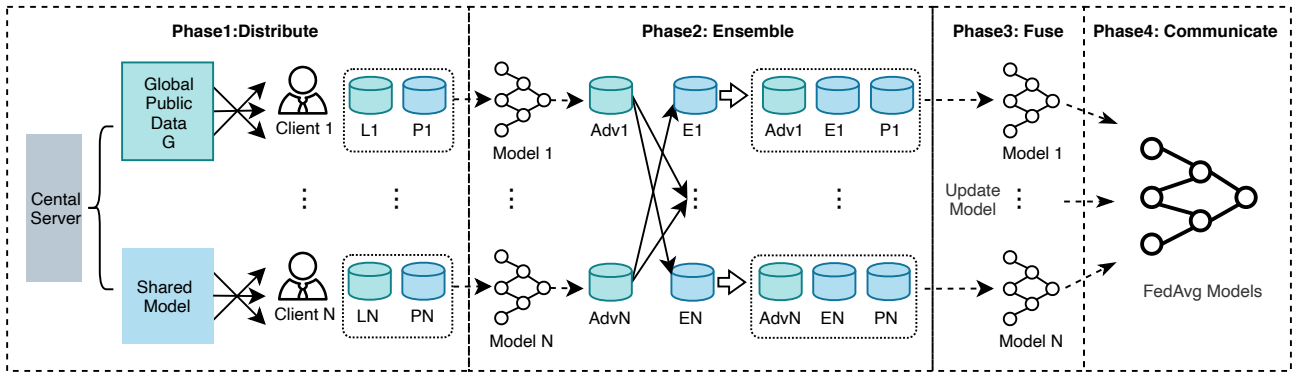
Figure 1: Illustration of the proposed EFAT framework. The EFAT method involves 4 phases: (1) *Distribute*: Distributing the shared model and parts of global public dataset L to all the clients. P is the clients' private data. (2) *Ensemble*: Integrating adversarial samples Adv generated from the local public dataset L of other clients to form ensemble training data E. (3) *Fuse*: Fusing the various data distribution including the potential knowledge of other clients by adversarial training. (4) *Communicate*: Client model updates are aggregated on the central server using the FedAvg algorithm.

fenses like matrix estimation or logit pairing, and generalization to multiple types of adversarial attacks [Papernot *et al.*, 2017; Suciu *et al.*, 2018; Augenstein *et al.*, 2019; Shafahi *et al.*, 2019; Xie *et al.*, 2019].

Despite all this, some previous work indicated that adversarially trained models might remain vulnerable to black-box attacks, where using the transferred perturbations computed on undefended models. It has been found that an adversarial network on MNIST has a slightly higher error on transferred examples than white-box examples[Goodfellow *et al.*, 2014]. Since these adversarial attacks have been observed to be transferable, adversarial training using samples generated from a single model provides robustness to other models performing the same task[Athalye *et al.*, 2018]. To improve the robustness of black-box attacks, [Tramèr *et al.*, 2018] proposed an Ensembling Adversarial Training method that trained the model by injecting adversarial examples transferred from several fixed pre-trained models into the original training data. In our work, we will further strengthen the robustness of the model to defense against black-box attacks in federated learning settings.

3 Methodology

We consider the notations and definitions of federated learning as defined in [McMahan *et al.*, 2017]. To be specific, there are K clients connected to a central server in federated learning. At each round t, the server randomly selects $N = \lfloor C \times K \rfloor$ clients for some $0 < C < 1$. We assume that for every $1 \le i \le N$ the i_{th} node has access to private training samples in $P_i = \{(x_i, y_i) \in R^d \times R\}$.

3.1 Intuitive Federated Adversarial Training

In this paper, we mainly study how to adapt adversarial training to federated learning. We first introduce the intuitive federated adversarial training method combining federated learning with adversarial training directly.

Local PGD Attack   For each selected client node in the federated learning model, we perform a PGD attack locally to generate their own basic adversarial examples.

The PGD attack first performs a gradient ascent step in the loss function w.r.t. the image pixel values. For the i_{th} client, PGD attack performs multi-steps update on the original sample x_i along the direction of the gradient of a loss function.

We use local private dataset P_i as input of the client model M^i. In each iteration, PGD adversarial examples $P_i^{adv} = \{x_i^{adv}, y_i\}$ follows the update rule:

$$x_{t+1}^{adv} = \Pi_{clip}(x_t^{adv} + \alpha sign(\nabla_x J(x_t^{adv}, y))) \quad (1)$$

where α controls the maximum L_∞ perturbation of the adversarial examples, and the clip function forces x to reside in a certain range.

Local Adversarial Training   Follow the work of [Kurakin *et al.*, 2016], we first group examples into batches containing both normal and adversarial examples before taking each training step. We use a loss function that allows independent control of the number of adversarial examples in each batch:

$$Loss = \sum_{x_i \in P_i} L(x_i | y_i) + \sum_{x_i^{adv} \in P_i^{adv}} L(x_i^{adv} | y_i) \quad (2)$$

where $L((x|y)$ is a loss on a single example x with true class y.

After the loss function is determined, we perform adversarial training on the ith local client, which can be formulated as a robust optimization problem [Madry *et al.*, 2018; Tramèr *et al.*, 2018].

$$\min_{\theta^i} \max_{D(x^i, x_{adv}^i) < \alpha} Loss \quad (3)$$

The inner maximization problem synthesizes the adversarial counterparts of clean examples, while the outer minimization problem finds the optimal model parameters over perturbed training examples.

Federated Averaging   After each selected client performs local adversarial training, these client model updates are sent to the central server. The central server then aggregates these local model parameters using the FedAvg [McMahan *et al.*,

Method	MNIST,SVHN, MNISTM,USPS →SYN	MNIST,SVHN, MNISTM,SYN →USPS	MNIST,SVHN, USPS,SYN →MNISTM	MNIST,USPS, MNISTM,SYN →SVHN	MNIST,SVHN, MNISTM,SYN →MNIST
Baseline	61.78%	78.06%	58.61%	27.15%	90.05%
EFNT	78.06%	82.50%	**73.60%**	33.75%	98.26%
EFNT+AT	82.30%	82.35%	**73.60%**	**48.80%**	98.55%
EFAT	**85.84%**	**83.45%**	71.65%	45.50%	**98.65%**

Table 1: Performance of clients trained with Digit-Five against black-box PGD adversaries. We assign four of them to conduct different adversarial training methods while the rest are regarded as adversaries to perform black-box PGD-40 attacks. Our proposed methods EFNT, EFNT+AT, and EFAT outperform the baseline among these five different dataset extraction schemes. In three-fifths of the data set extraction situations, EFAT reached the highest robust accuracy.

2017] algorithm. The global averaging step in time step t can be written as follows:

$$\theta_t = \sum_{i=1}^{N} \frac{1}{N} \theta_t^i \qquad (4)$$

Thus, each local robust client model is trained individually. Obviously, this training paradigm only considers the client-specific loss, which leads to the federated model being still vulnerable against adversarial examples generated with other models.

3.2 Ensemble Federated Adversarial Training

In our proposed ensemble federated adversarial training (EFAT) method, to tackle the challenge mentioned above, we take advantage of extensive and diverse knowledge from other clients to improve the robustness of previously limited models to small populations (Algorithm 1). The EFAT method can be summarized 4 phases as follows(see Figure 1): (1) *Distribute*: Distributing the shared model and parts of global public dataset to all the clients. (2) *Ensemble*: Integrating adversarial samples generated from the local public dataset of other clients. (3) *Fuse*: Fusing the various data distribution including the potential knowledge of other clients by adversarial training. (4) *Communicate*: Client model updates are aggregated on the central server using the FedAvg algorithm.

Distribute In this stage, the pre-training shared model and the part of public dataset are distributed to all clients. Therefore, the training dataset in each client is partitioned into two parts:(1) the client part P and (2) the public part G. P is partitioned into participated clients taking feature distribution skew and label distribution skew into consideration. G is the globally public dataset that consists of a uniform distribution over features or labels. A random α proportion of the global public dataset G is distributed to each client. It can be concluded that the data owned by each client consist of the private data P and a random α proportion of G. We denote the subset of P and a random α proportion of G by **privated data** P_i and **local public data** L_i in the i_{th} client respectively.

Ensemble We use the local public data L_i to generate the corresponding adversarial examples denoted by *local adversarial public data* L_i^{adv} .Then we can get *ensemble adversarial public data* $\{L_1^{adv}, L_2^{adv}, ...L_{i-1}^{adv}, L_{i+1}^{adv}, ...L_N^{adv}\}$ by ensembling the local adversarial public data generated by other

---

Algorithm 1: Illustration of EFAT on K homogeneous clients (indexed by k) for T rounds, n_k denotes the number of data points per client and C the fraction of clients participating in each round. The server model is initialized as x_0. G is the global public dataset, L is the local public dataset, and P is the client private data.

---
1 **Server executes:**
2 **for** *each communication round* $t = 1, ..., T$ **do**
3    $S_t \leftarrow$ random subset (C fraction) of K clients
4   distribute part of G to K clients $\{L_1, L_2, ..., L_k\}$
5   **for** *each client* $k \in S_t$ *in parallel* **do**
6   generate adversarial examples L_k^{adv} by Eq.(1)
7   exchanges with other clients $E_k^{adv} = \{L_1^{adv}, ..., L_{k-1}^{adv}\}$
8       $x_{t+1}^k \leftarrow$ Client-localSGDupdate(k, x_t) using ensemble $\{P_k, L_k^{adv}, E_k^{adv}\}$
9    $x_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} x_{t+1}^k$;
10 **return** x_{t+1};

---

clients except the current i_{th} client. We denote ensemble adversarial public data as E_i^{adv}. In this stage, we can conclude that the data composition of the i_{th} client consists of three parts: (1) private data, (2) local adversarial public data, (3) ensemble adversarial public data.

$$Data_i = \{P_i, L_i^{adv}, E_i^{adv}\} \qquad (5)$$

Fuse In this stage, we perform adversarial training on both local adversarial public data, ensemble adversarial public data and private data. It should be noted that the loss function will change correspondingly compared with the previous equation (2) due to different data distributions.

$$Loss = \sum_{x_i \in P_i} L(x_i|y_i) + \sum_{x_i^{adv} \in L_i^{adv}} L(x_i^{adv}|y_i)$$
$$+ \sum_{x_i^{adv} \in E_i^{adv}} L(x_i^{adv}|y_i) \qquad (6)$$

Intuitively, as adversarial examples transfer between models, perturbations crafted on other clients are good approximations for the maximization problem. Moreover, the learned model can not influence the "strength" of these adversarial

examples. As a result, minimizing the training loss implies increased robustness to black-box attacks from other models. Through this ensemble method, we can take advantage of more extensive and more diverse datasets to improve the robustness of previously limited models to small populations.

Communicate After each selected client performs local update based on the ensemble adversarial training, these model updates are sent to the server. Then the central server aggregates these models by averaging to obtain the new global model.

In summary, we first assign the shared model and part of the public dataset to participated clients. Then we ensemble adversarial examples generated from local public data distributed on multiple clients. Next, we perform adversarial training based on these perturbations. The last step is to average local model updates using FedAvg algorithm. This training procedure given above is repeated until a satisfactory degree of convergence has been achieved.

4 Experiment

This section demonstrates the robustness against black-box attacks of our proposed algorithm on two kinds of highly Non-IID datasets, including feature distribution skew and label distribution skew. All experiments were done using a V100 GPU cluster, and the federated system was simulated on a single machine (as the communication efficiency is not the main focus of this paper). Experiment results show that our models significantly improve robustness and accuracy against black-box attacks, which provides strong support for our central hypothesis.

4.1 Experiment Setup

Datasets We use Digit-Five datasets as feature distribution skew datasets, which is a collection of five benchmarks for digit recognition, namely MNIST [Lecun *et al.*, 1998], Synthetic Digits [Ganin and Lempitsky, 2015], MNIST-M [Ganin and Lempitsky, 2015], SVHN, and USPS. It was constructed for domain adaptation research by [Peng *et al.*, 2020].

We construct a label distributed skew dataset based on CIFAR10 by using the Dirichlet distribution [Lin *et al.*, 2020]. The value of γ controls the degree of non-i.i.d.-ness. When γ tends to 0, the clients are more likely to hold examples from only two classes (if the number of clients is set to 5). Besides, $\gamma = 100$ mimics identical local data distributions. We conduct comparative experiments using three different γ values, respectively 100, 1, 0.01.

Training Strategy For Digit-Five, we take turns selecting four datasets as different participated clients. Then we assign 10% of G to each client as local public dataset. The four participated clients perform ensemble federated adversarial training while the unselected dataset trains by itself and then generates adversarial examples as black-box attacks to test our model's performance.

For CIFAR10, the training dataset consists of 50000 images in 10 classes, with 5000 images per class. The training dataset are distributed to 5 clients using a Dirichlet distribution mentioned above. Each client can get about 10000

(50000/5) images as private data. We set the random distributed fraction α as 10%. Then we assign 10% of G to each client as local public data.

In the experimental setting, the clients train locally for five rounds and then exchanges local public adversarial datasets once.

Compared Methods To illustrate the necessity and effectiveness of our training method in detail, we introduce two different simplified versions of EFAT called EFNT and EFNT+AT.

In our EFAT method, we perform adversarial training on both local adversarial public data and ensemble adversarial public data. EFNT+AT refers to performing normal training instead of adversarial training on ensemble public data. EFNT refers to performing normal training instead of adversarial training to ensemble public data and local public data.

Besides, we adopt the intuitive federated adversarial learning mentioned in Section 3.1 as the baseline method. For a fair comparison, we extract the same amount of private data as the sum of local public data and ensemble public data to generate adversarial examples for adversarial training.

Networks and Parameters In the experiments, we simulate a federated learning scenario with $n = 4$ nodes where each node uses ResNet18 with the same architectures. We choose to take gradient steps in the L_∞ norm, i.e., adding the sign of the gradient, since this makes the choice of the step size simpler.

For Digit-Five, we set perturbation $\epsilon = 0.3$, perturbation step size $\eta_1 = 0.01$, number of iterations $K = 40$, learning rate $\eta_2 = 0.01$, batch size $m = 128$, and run 100 epochs on the training dataset. To evaluate robust errors, we apply PGD (black-box) attack with 20 and 40 iterations and 0.01 step size. For CIFAR10, following [Tramèr *et al.*, 2018], the maximum perturbation allowed is 16/255 for both defense and attack models. We set perturbation $\epsilon_{train} = 16/255$, step size $\alpha = 0.003$, number of iterations K = 20, batch size m = 128, and run 100 epochs on the training dataset. The adversarial test data are bounded by L_∞ perturbations with $\epsilon_{test} = 16/255$ and 8/255 which are generated by PGD-10 and PGD-20.

All PGD attacks have a random start, i.e., the uniformly random perturbation of [-ϵ_{test}, ϵ_{test}] added to the clean test data before PGD perturbations.

4.2 Result Analysis

Digit-Five In the following experiment, we performed our proposed EFAT, EFNT, EFNT+AT, and baseline with Digit-Five datasets on both the "clean" examples x and adversarial examples x_{adv}. Table 1 illustrates each client's average accuracy against PGD-40 black-box attacks. For example, the first column means that we select MNIST, SVHN, MNIST-M, and SVHN and distribute them to four clients to perform different training methods. Simultaneously, SYN trains by itself and then generates adversarial examples sending to the first four clients as a black-box attack. A first observation is that compared with the models only trained locally with their own adversarial examples(baseline), EFAT, EFNT, and EFNT+AT trained with exchange public data reach higher

non-i.i.d.-ness	IID			Non-IID					
	γ=100			γ=1			γ=0.01		
Method	clean	PGD-10	PGD-20	clean	PGD-10	PGD-20	clean	PGD-10	PGD-20
Baseline	72.21%	62.34%	64.17%	72.21%	63.02%	63.62%	72.46%	65.05%	64.62%
EFNT	**80.45%**	43.91%	41.02%	**79.03%**	43.23%	43.99%	**81.19%**	47.82%	42.79%
EFNT+AT	78.42%	57.45%	48.35%	78.81%	58.36%	46.41%	81.15%	56.24%	49.41%
EFAT	72.02%	**70.83%**	**67.25%**	72.64%	**70.28%**	**68.64%**	74.66%	**71.48%**	**67.46%**

Table 2: Accuracies of CIFAR10 under black-box attacks in IID and non-IID settings w.r.t. α = 5%. Our proposed defense method can significantly improve the robust test accuracy of deep models on clients in both IID and non-IID settings with α are set to 5%. EFAT is able to achieve robust test accuracy as high as 71.48%, increasing 5%-8%,24%-27% and 13%-28% compared to the baseline method, EFNT and EFNT+AT method respectively.

non-i.i.d.-ness	IID			Non-IID					
	γ=100			γ=1			γ=0.01		
Method	clean	PGD-10	PGD-20	clean	PGD-10	PGD-20	clean	PGD-10	PGD-20
Baseline	72.91%	62.46%	61.79%	72.95%	61.83%	60.41%	72.93%	61.29%	59.80%
EFNT	**82.32%**	43.51%	43.49%	**80.23%**	50.17%	44.84%	**82.89%**	45.36%	42.20%
EFNT+AT	81.45%	65.26%	62.45%	79.54%	64.65%	62.12%	81.57%	65.04%	62.12%
EFAT	73.39%	**70.47%**	**68.35%**	73.45%	**70.98%**	**68.25%**	75.57%	**71.66%**	**68.43%**

Table 3: Accuracies of CIFAR10 under black-box attacks in IID and non-IID settings w.r.t. α = 10%. A different trend can be observed where the robust test accuracy of EFNT+AT increases 10%-18% when defending adversarial examples generated by PGD-10 and PGD-20.

accuracy against attacks. It is because in the setting of feature distribution skew, expanding training data from different clients' models increases the diversity of training data distribution, which helps improve the robustness of models.

We also noted that EFAT outperforms EFNT and EFNT+AT, which indicates adversarial examples generated for one model could stay adversarial for other models. Therefore, it is helpful when conducting adversarial training using adversarial examples generated by other clients' public data.

CIFAR10 For CIFAR10, we compare our EFAT algorithm with the baseline method, EFNT, and EFNT+AT, with three different γ values, respectively 100, 1, 0.01. Table 2 shows the performance of EFAT and the other three methods w.r.t. standard test accuracy and adversarially robust test accuracy of the clients on CIFAR10. It should be noted that the accuracy here refers to the average accuracy of all participated client models. We obtain standard test accuracy for clean test data and robust test accuracy for adversarial test data generated by PGD-10 and PGD-20.

From Table 2 we can observe that our proposed defense method can significantly improve the robust test accuracy of deep models on clients in both IID and non-IID settings with β are set to 5%. Our EFAT method is able to achieve robust test accuracy as high as 71.48%, increasing 5%-8%,24%-27% and 13%-28% compared to the baseline method, EFNT and EFNT+AT method respectively. This gap significantly widens as "non-i.i.d.-ness" (Specifically refers to perturbation bound ϵ_{test}) increases. Larger "non-i.i.d.-ness" will allow the generated adversarial data to deviate more from natural data. In EFNT and EFNT+AT methods, robust test accuracies are significantly hurt with larger ϵ_{test}.

In addition, client models trained with EFNT method

achieve the highest clean test accuracy, followed by EFNT+AT method, while the baseline method and our EFAT method do not perform well. It is forgivable that adversarial training provides the most security of adversarial attacks while losing only a small amount of accuracy when we mainly focus security against adversarial examples.

Besides, we compare our EFAT method and other methods with different values of α. We set α = 10% in Table 3. We observe a different trend where the robust test accuracy of EFNT+AT increases 10%-18% when defending adversarial examples generated by PGD-10 and PGD-20. For the comprehensive experiments in Table 2 and Table 3, it is easy to verify that our proposed model outperforms all other methods regardless of the value of α.

To sum up, client deep models trained by EFAT with α = 5% have higher robust test accuracy but lower standard test accuracy. By increasing α to 10%, client deep models have slightly increased on both standard test accuracy and robust test accuracy.

5 Conclusion

In this paper we present a novel ensemble federated adversarial training method, termed as EFAT, to improve the robustness of models against black-box attacks in federated learning. The proposed method enhances the diversity of adversarial examples through expanding training data with perturbations generated from other participating clients.

Experiment results on both Digit-Five and CIFAR10 in IID and Non-IID settings show that our method significantly improves the robustness and accuracy contrasted with the intuitive federated adversarial training method and the other two variants of EFAT.

References

[Athalye *et al.*, 2018] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[Augenstein *et al.*, 2019] Sean Augenstein, H Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, et al. Generative models for effective ml on private, decentralized datasets. *arXiv preprint arXiv:1911.06679*, 2019.

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[Bhagoji *et al.*, 2019] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, 2019.

[Bonawitz *et al.*, 2017] Keith Bonawitz, Vladimir Ivanov, et al. Practical secure aggregation for privacy-preserving machine learning. In *ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[Feki *et al.*, 2021] Ines Feki, Sourour Ammar, Yousri Kessentini, and Khan Muhammad. Federated learning for covid-19 screening from chest x-ray images. *Applied Soft Computing*, 2021.

[Feng *et al.*, 2020] Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. Pmf: A privacy-preserving human mobility prediction framework via federated learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2020.

[Ganin and Lempitsky, 2015] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, 2015.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Han *et al.*, 2019] Xu Han, Haoran Yu, and Haisong Gu. Visual inspection with federated learning. In *Image Analysis and Recognition*, 2019.

[Kairouz *et al.*, 2019] Peter Kairouz, H Brendan McMahan, Brendan Avent, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[Lecun *et al.*, 1998] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 2020.

[Lin *et al.*, 2020] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *arXiv preprint arXiv:2006.07242*, 2020.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 2017.

[Papernot *et al.*, 2017] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM on Asia conference on computer and communications security*, 2017.

[Peng *et al.*, 2020] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2020.

[Shafahi *et al.*, 2019] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, et al. Adversarial training for free! In *Advances in Neural Information Processing Systems*, 2019.

[Suciu *et al.*, 2018] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. When does machine learning FAIL? generalized transferability for evasion and poisoning attacks. In *USENIX Security Symposium*, 2018.

[Szegedy *et al.*, 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[Tramèr *et al.*, 2018] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2020] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[Xie *et al.*, 2019] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Computer Vision and Pattern Recognition*, 2019.