RoPe-Door: Toward Robust and Persistent Backdoor Data Poisoning Attacks in Federated Learning

Ye Liu, Shan Chang, Denghui Li, Shaohuai Shi, and Bo Li

Abstract

Federated Learning (FL) enables privacy-preserving collaborative training and builds a federation through exchanges of immutable data such as model parameters or gradient updates. FL remains vulnerable to a variety of attacks during critical processes like local model training and parameter transmission, in which the backdoor attack is particularly evident. In this paper, we propose a novel backdoor data poisoning attack method, RoPe-Door, using a trigger generation algorithm to improve the robustness and persistence of attacks even under Byzantine aggregation methods. We conduct extensive experiments on four image classification tasks to evaluate the effectiveness of RoPe-Door. The experimental results demonstrate that, compared to backdoor attacks using random triggers, RoPe-Door exhibits significant advantages in robustness, persistence, and attack effectiveness under both IID and Non-IID data settings.

Introduction

Federated Learning (FL), as a distributed machine learning framework, allows data to remain on local devices with only model updates being shared. This collaborative training approach can mitigate personal data leakage and single-point failures while overcoming data silos. However, there are also critical problems and challenges in FL practice which affect the performance of FL, for example, Non-IID and long-tail of data challenges [1], [2] and security threats [3], [4], and have been increasingly received much attention. In recent years, there have been many in-depth examinations on the security threats in FL and proposals of various attack methods. Among them, the backdoor attack is particularly infamous, in which malicious participants implant the backdoor in the local model updates to manipulate the global model through poisoning, and it may greatly compromise the quality of the model. This is further complicated in that it is difficult to detect during calculation, which poses a serious threat to the practical deployment and application of FL.

FL is inherently vulnerable to backdoor attacks for several reasons. Firstly, due to the openness of the FL system and the independence of each participant, the presence of malicious attackers among participants is highly likely. Secondly, since the local training data is owned by the participants and is not disclosed, the local training process is completed under the control of the participants, and the local model parameters uploaded to the server are also determined by them. Therefore, malicious participants can implant backdoors into local models by manipulating private training data, local training processes and outcomes. In addition, due to the heterogeneity of data among dispersed participants, there may be significant differences between local model updates. Identifying local model updates with the backdoor through unsupervised learning methods is often prone to misjudgment. Finally, the local model update containing the backdoor will be aggregated with the model, thereby injecting the backdoor into the global model.

Backdoor attacks can be categorized into two types based on how attackers manipulate model training: data poisoning attacks and model poisoning attacks. In data poisoning attacks, attackers modify regular data (such as adding small pixel patches images) and their labels to achieve their attack goals. In model poisoning attacks, attackers alter the internal structure of the model (such as specific parameters) to embed backdoors. In short, the distinction between the two lies in whether the manipulation directly targets local or global model parameters. Generally, data poisoning attacks require fewer capabilities and less knowledge from attackers, as they only need to modify the training data without intervening in the training process.

In this paper, we focus on the challenges of effectively injecting a backdoor through data poisoning by a single attacker in FL, which can be summarized as follows. Firstly, in practice, malicious participants cannot control the timing and frequency of backdoor injections, as servers randomly select participants to engage in training during each update iteration. Therefore, malicious participants cannot guarantee consistent opportunities to inject backdoors. Additionally, even if a participant is selected and successfully injects backdoor into the global model in a specific round, the global model will gradually forget the backdoor task with subsequent training iterations.

Digital Object Identifier:
10.1109/MNET.2024.3486228
Date of Current Version:
13 May 2025
Date of Publication:
25 October 2024

Ye Liu is with the School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, China; Shan Chang (corresponding author) and Denghui Li are with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China; Shaohuai Shi is with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518057, China; Bo Li is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong.

0890-8044/24©2024IEEE

Secondly, to combat poisoned local updates, a series of Byzantine fault-tolerant aggregation algorithms have been proposed. These methods can tolerate a certain number of Byzantine nodes (i.e., malicious participants), thereby mitigating the impact of data poisoning on the global model. In existing backdoor attacks based on data poisoning, random triggers are commonly used as backdoors. Although it is possible to achieve a high success rate of attacks immediately, maintaining the effectiveness of backdoors in the global model over time is not sustainable. This is because the effectiveness of backdoors is achieved by distorting the decision boundaries (i.e., parameters) of the global model, and this distortion gradually diminishes with updates to the global model. Furthermore, while increasing the volume of poisoned training data can improve the effectiveness of backdoor attacks, it also increases the detectability of local backdoor models compared to the benign model, making them more noticeable by the server. Moreover, for a single backdoor attacker it is challenging to bypass the Byzantine fault-tolerant aggregation algorithm and inject backdoors into the global model.

In this work, we propose RoPe-Door, a robust and persistent backdoor data poisoning attack method for FL. The objective of RoPe-Door is to ensure a lasting backdoor impact on FL, even if a single malicious participant engages in poisoning only once (where the attacker has only one opportunity to inject the designed backdoor trigger). Moreover, RoPe-Door maintains excellent backdoor attack efficacy even under Byzantine fault-tolerant aggregation algorithms. The core idea of RoPe-Door is to avoid using randomly generated triggers, but to learn the most appropriate backdoor trigger based on the current global model, and subtly adjust the model decision boundary through local model updates to enhance the performance of the backdoor. The pattern of triggers varies across iterations, making the backdoored local models easy to evade the detection of the central server. Specifically, we employ the principles of universal adversarial perturbations to generate triggers that depend on the global model. On one hand, malicious participants search for suitable triggers based on the global model of the current round, meaning the trigger is tailored to the global models. On the other hand, using this carefully selected backdoor for local updates does not cause it to deviate excessively from the overall distribution of local updates, thus enabling it to maintain attack capability under Byzantine fault-tolerant aggregation algorithms. We empirically evaluated RoPe-Door on four image classification tasks and compared it with random triggers. The experimental results show that compared to random triggers, RoPe-Door exhibits significant advantages in all settings. Even under the Byzantine fault-tolerant aggregation algorithm, RoPe-Door can still achieve high attack rates.

Related Work

Backdoor Attacks in FL

Data Poisoning Attack in FL: Malicious participants contaminate the training dataset, by including a subset of data with backdoor trigger and a target label. This manipulation leads the model to learn incorrect associations during training, resulting in a model compromised with a backdoor.

Nuding and Mayer [5] demonstrate the implementation of backdoor attacks in FL through data poisoning, showing that increasing the proportion of malicious participants and the amount of poisoned training data can enhance the success rate of backdoor attacks. However, the primary limitation of these methods is that updating benign participants dilutes the backdoor effect. To address this issue, Wang et al. [3] propose an edge-case backdoor attack targeting samples unlikely to be in benign participants' training data. Specifically, they select samples that are outliers from the global distribution and unlikely to appear in the validation sets of benign participants. This approach ensures that the impact of the backdoor is not easily mitigated. Considering the decentralization characteristics of FL, Xie et al. [6] suggest a distributed backdoor attack, DBA, where triggers are divided among multiple malicious participants, enhancing the attack effectiveness of triggers and allowing them to evade detection by secure aggregation algorithms. Similarly, Gong et al. [7] propose a coordinated trigger backdoor attack, utilizing model-dependent triggers for more effective backdoor injection. This method, akin to the strategy [6], constructs global triggers by amalgamating local triggers. Model-dependent triggers have been proven to be more effective than random triggers. Nevertheless, the primary challenge in FL data poisoning remains: aggregation algorithms tend to nullify the contributions of backdoor models, causing the global model to rapidly forget the backdoor effect.

Model Poisoning Attack in FL: Malicious participants attack the model by directly modifying its weight parameters. In FL, malicious participants have complete control over the training process and hyperparameters, and can freely modify the model parameters before submitting them to implant backdoors in the global model.

Bagdasaryan et al. [8] were the first to study model replacement methods, where attackers amplify malicious model updates to suppress other benign model updates, effectively replacing the global model with the attacker's backdoor local model, and adding an anomaly detection term to the loss function to avoid anomaly detection. Bhagoji et al. [4] demonstrate that attack objectives can be achieved even when the global model is far from convergence. To enhance the stealthiness of the attacks, they also propose an alternating minimization attack strategy aimed at achieving optimal attack and stealth goals. Baruch et al. [9] search for model parameters within the bias fluctuation range that can be implanted with backdoors. Even in IID (Independent and Identically Distributed) settings, attackers can alter parameters without detection, achieving backdoor attacks capable of bypassing secure aggregation algorithms. Zhou et al. [10] propose an optimization-based model poisoning attack, injecting resistant neurons into the neural network's redundant space to maintain the stealthiness and persistence of the attack. Zhang et al. [11] propose Neurotoxin, where attackers implant the backdoor model using coordinates less likely to be updated by benign participants to prolong the durability of the backdoor. However, model poisoning attacks require attackers to fully control one or more participants, making the attack impractical.

Byzantine-Robust FL

Generally, in federated training, the aggregation rule involves averaging local model parameters to form global model parameters. This average aggregation rule is widely applied in non-adversarial environments. However, the average aggregation rule is not robust in adversarial settings. Specifically, even an attacker only compromises the device of one participant, they can still manipulate the global model parameters through the average aggregate rule. Consequently, many scholars have recently proposed multiple aggregation rules (e.g., Foolsgold [12], Geom-Median [13], Krum [14], and RLR [15]) to maintain robustness even in the event of Byzantine failures among some participants' devices. Next, we will introduce the aforementioned aggregation rules.

Foolsgold: The key idea of the Foolsgold [12] is that honest participants can distinguish themselves from malicious participants based on their gradient updates, as malicious participants provide gradients that are more similar to each other than those of other honest participants. Foolsgold utilises this assumption to adjust the learning rates of each participant in each iteration, aiming to maintain the learning rate of participants who provide a unique gradient, while reducing the learning rate for participants who repeatedly provide similar-looking gradient updates.

Geom-Median: The working principle of the Geom-Median [13] is to calculate the geometric median of participant model updates. Firstly, convert the models of all participants into vectors and calculate the geometric median of these vectors. This median is robust and can reduce the impact of outliers. Secondly, use this geometric median to construct the global model and maintain its accuracy and robustness.

Krum: The Krum method [14] computes the Euclidean distance between all pairs of model parameters, and for each model, it selects the closest $n - f - 2$ models (where f is the number of Byzantine nodes). Then, it calculates the distance from the model to each of the other models and finally sums them up to obtain a score value. Based on this score value, it selects the local model with the lowest score as the update for the next global model round.

RLR (Robust Learning Rate): The RLR method [15] updates sign information based on agents, carefully adjusting the learning rate of the aggregation server for each dimension and round. Concretely, RLR introduces a hyper-parameter called learning threshold θ at the server. For every dimension where the sum of signs of updates is less than θ, the learning rate is multiplied by -1. This approach aims to maximize the loss on that dimension rather than minimizing it.

Design of RoPe-Door

Scope and Assumptions

Attack Model: In federated training, we assume the existence of a malicious participant (attacker) who injects backdoor poisoning samples into their local training set, with the aim of injecting backdoors into the global model. The server responsible for model aggregation is honest and will not leak or tamper with the training dataset of participants. Other benign participants faithfully adhere to the FL protocol.

Attacker's Goals: The attackers aim to achieve the following three objectives:

- *Effectiveness:* The attacker aims to embed a backdoor in the global model, causing the backdoor global model to produce correct classification results for benign inputs while misclassifying inputs with target labels.
- *Robustness:* Even Byzantine aggregation methods are employed, the attacker intends to successfully implement the backdoor attack.
- *Persistence:* Given that the attacker may only participate in a single round of FL, there is a requirement for persistence to ensure the longevity of the backdoor effect in the global FL model.

Attacker's Capability: The attacker has complete control over their local training process, including backdoor data injection, trigger patterns, and updating local model hyperparameters (such as epochs and learning rates). This scenario is practical because each participant typically owns their local training dataset, and the server can only obtain trained models from participants without knowledge of how the models were trained. Meanwhile, the attacker cannot know or tamper with the global aggregation rules on the server side, nor can they know or tamper with other participants' training datasets, training processes, and model update hyperparameters.

Overview the Framework

Fig. 1 shows an overview of RoPe-Door which includes two steps: (1) Trigger generation. Based on a clean local training dataset and a clean global model, attackers use a general perturbation generation algorithm based on stochastic gradient ascent to generate backdoor triggers. (2) Poisoning and model training. With the generated backdoor trigger, the attacker poisons a subset of the training dataset, where the poisoned dataset typically consists of a mixture of clean data with real labels and backdoor trigger data with target labels, and then trains the model with the poisoned data.

Trigger Generation

To achieve robust and persistent backdoor data poisoning attacks against FL, instead of using traditional random triggers, as their ability to trigger backdoor models is weak, especially in FL where local models are diluted during the aggregation process. Therefore, we propose to train appropriate triggers based on the current round's global model. Specifically, we first initialize the trigger and determine its pattern, including its shape, size, and position. The trigger's shape is configured to be rectangular and placed in the corners of the image to align with the practice of existing backdoor attacks. Then, we fix the weight parameters of the global model and use triggers as parameters for batch training to optimize them.

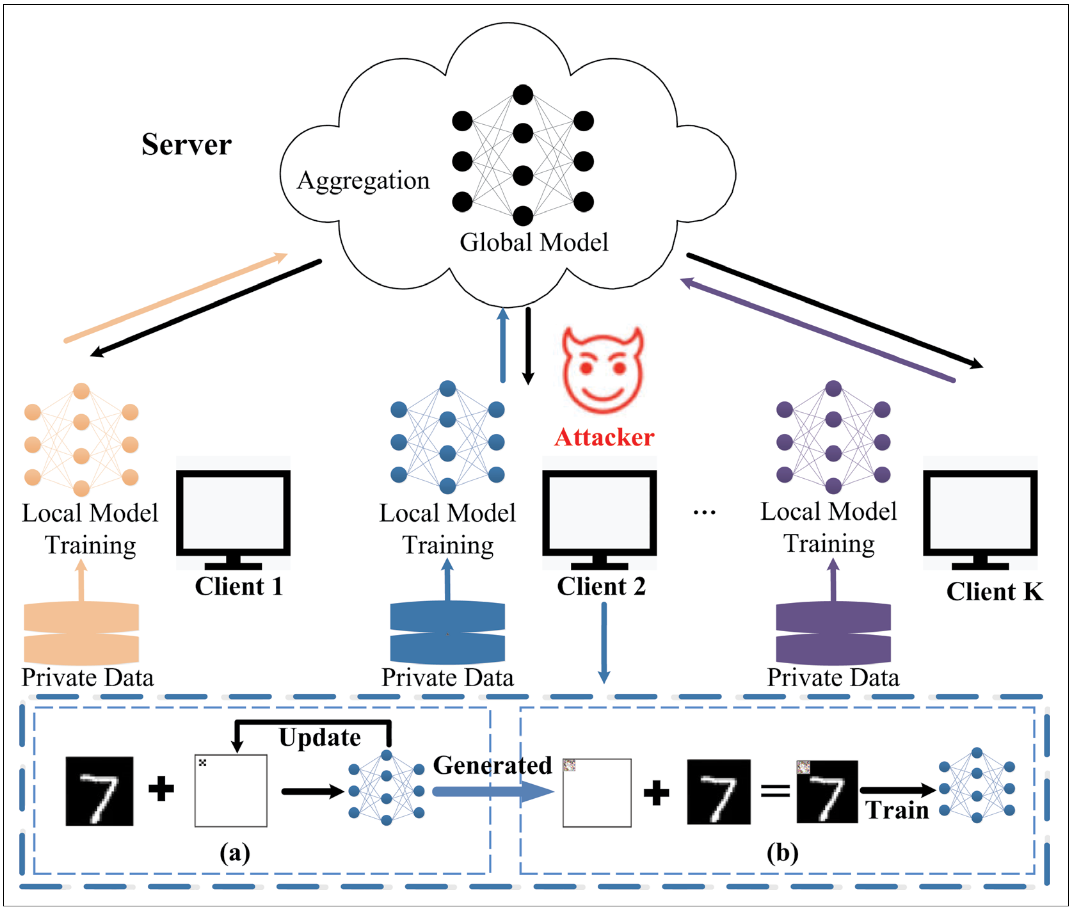


FIGURE 1. The framework of RoPe-Door. a) Trigger Generation. b) Training Backdoor Model.

Because the trigger reduces the probability of the label corresponding to the sample, we adopt a training strategy of random gradient ascent. In short, we generate triggers that depend on the global model through a random gradient ascent algorithm.

Formally, assume that the malicious participant C_i has a local training dataset $D_{\text{train}} = \{x_{i,d}, y_{i,d}\}$, $d \in \{1, \ldots, |D_{\text{train}}|\}$, where $x_{i,d}$ and $y_{i,d}$ represent each data sample and its corresponding label respectively. The attacker based on local data sample set $X = \{x_i, i = 1, \ldots, K\}$ containing K data points and the global model $G(w, \cdot)$ of the current round's frozen parameters w, repeats multiple iterations to find a small universal adversarial perturbation t (in terms of the l_p-norm with $p \in \{0, 1, 2, \infty\}$) that misclassify most images. The goal is to find t that satisfies the following two constraints: $\|t\|_p \le \epsilon$ and $\mathbb{P}(X, t) \ge 1 - \xi$. Where K refers to the number of private data samples will be used for federated training on each participants. In practice, the sever can determine the minimum value of K, i.e., K_{min}. In other words, the participants own at least K_{min} samples is able to take part in the federated training. Moreover, each participant can determine its own K according to the size of its private dataset, as well as other resources, i.e., computation and storage. ϵ controls the magnitude of the perturbation vector t. $\mathbb{P}(X, t)$ represents the "fooling ratio", which is the fraction of images x whose target label $G(w, x + t)$ differs from the original label $G(w, x)$. ξ is a small tolerance hyper-parameter.

We consider the following optimization problem, $\text{argmax} L(w,t) = \frac{1}{K}\sum_{i=1}^{K} L(w, x_i + t)$, s.t. $\|t\|_p \le \epsilon$, where $L(w, \cdot)$ represents the loss used to train the model. The purpose is to search for a universal perturbation that maximizes the training loss, thus forcing images into the wrong class. To address this optimization problem, we adopted the stochastic gradient ascent method. Each iteration starts with using gradient ascent to update the general perturbation t, thereby maximizing the loss. Moreover, t is projected onto the l_p-norm ball to prevent it from growing too large. After obtaining the universal adversarial perturbation, we perform backdoor attacks in the form of triggers using the universal adversarial perturbation.

Backdoor Model Training

After obtaining the trigger t, the attacker contaminates the training dataset and updates the model, causing it to learn incorrect associations during the training process. Specifically, firstly, the attacker uses the training dataset D_{train} to craft a poisoned dataset $D_{\text{backdoor}} = \{(x + t, y_{\text{target}})\}$. Specifically, training data D_{train} consists of poisoning data D_{backdoor} and remaining benign data D_{benign}, i.e., $D_{\text{train}} = D_{\text{backdoor}} \cup D_{\text{benign}}$. We use $\gamma = \frac{D_{\text{backdoor}}}{D_{\text{train}}}$ to indicate the poisoning rate. Subsequently, the attacker trains a backdoor model L_{backdoor} as follows: $L_{\text{backdoor}} = \overset{\min}{G} \sum_{(x,y) \square D_{\text{backdoor}} \cup D_{\text{benign}}} L(w, (x, y))$. Using this carefully selected backdoor for local updates will not excessively deviate from the overall distribution of local updates.

In summary, in order to achieve both global model accuracy and backdoor attack success, attackers use alternating model training methods. This method transforms the training problem into a form of min - max optimization, where the minimization is for the model parameter w and the maximization is for the trigger t. By alternately optimizing the training loss and the backdoor objective, the process can be formally represented as $\min_{w} \max_{\|t\|_p \leq \epsilon} L(w,t) = \frac{1}{K}\sum_{i=1}^{K} L(w, x_i + t)$. Here, the max() function denotes finding a set of backdoor samples (obtained by combining the original sample x and trigger t) in the sample space that maximizes the loss function. The min() function represents minimizing the expected loss of the model on the set of backdoor samples, achieved by updating the model parameters.

The formula can be effectively solved through the alternating random gradient method. In each iteration, use the gradient up update trigger t and the gradient down update model parameter w. Firstly, freeze the model parameter w and perform gradient ascent training on trigger t to maximize the loss function and generate the optimal trigger t. Secondly, freeze the trigger t and perform gradient descent training on the model parameter w to minimize the loss function and obtain the optimal model parameter w. Finally, the attacker can use the trained trigger t to trigger the backdoor embedded in the global model.

Experimental Evaluation

Experimental Setup

Models and Datasets: We conduct experiments on PyTorch's distributed library to simulate real-world learning scenarios in FL. For these experiments, we use the MNIST and F-MNIST datasets to train the Lenet5 model, and the CIFAR10 and CIFAR-100 datasets to train the AlexNet model.

Parameter Settings: In all tasks, the total number of participants is 10, with 100% selection in each round. The optimizer is SGD, with a batch size of 64. The learning rates for rounds 0–50, 50–100, and 100–300 are 0.01, 0.001, and 0.0001, respectively. For the Lenet5 on MNIST and Lenet5 on F-MNIST tasks, the local epoch and backdoor local epoch are both 5, with a trigger size of 2 × 2. For the AlexNet on CIFAR-10 and AlexNet on CIFAR-100 tasks, the local epoch and backdoor local epoch are both 10, with a trigger size of 3 × 3. The trigger is placed in the upper left corner of the image. When the data is Non-IID, determine the total number of clients n, the number of categories each client has c, and the total number of training data d. Then, each client randomly selects c optional labels and selects $\lceil d / (n \times c) \rceil$ data from each corresponding label data to form their own training set.

Comparison Algorithms: We compare two algorithms.

- *Random:* The attacker randomly samples the RGB values of each trigger pixel from an uniform distribution $U[0; 255]$.
- *RoPe-Door:* The attacker uses trained triggers for backdoor attacks.

Evaluation Metrics: We use two metrics, Main Task Accuracy (MTA) and Attack Success Rate (ASR), to evaluate the overall performance of backdoor attacks.

- *MTA:* This metric represents the predictive accuracy of the global model on the test dataset. The attacker aims to ensure a high MTA to avoid rejection of the global model.
- *ASR:* This measures the effectiveness of the backdoor attack, calculating the percentage of malicious inputs incorrectly classified as the target label through the backdoor model.

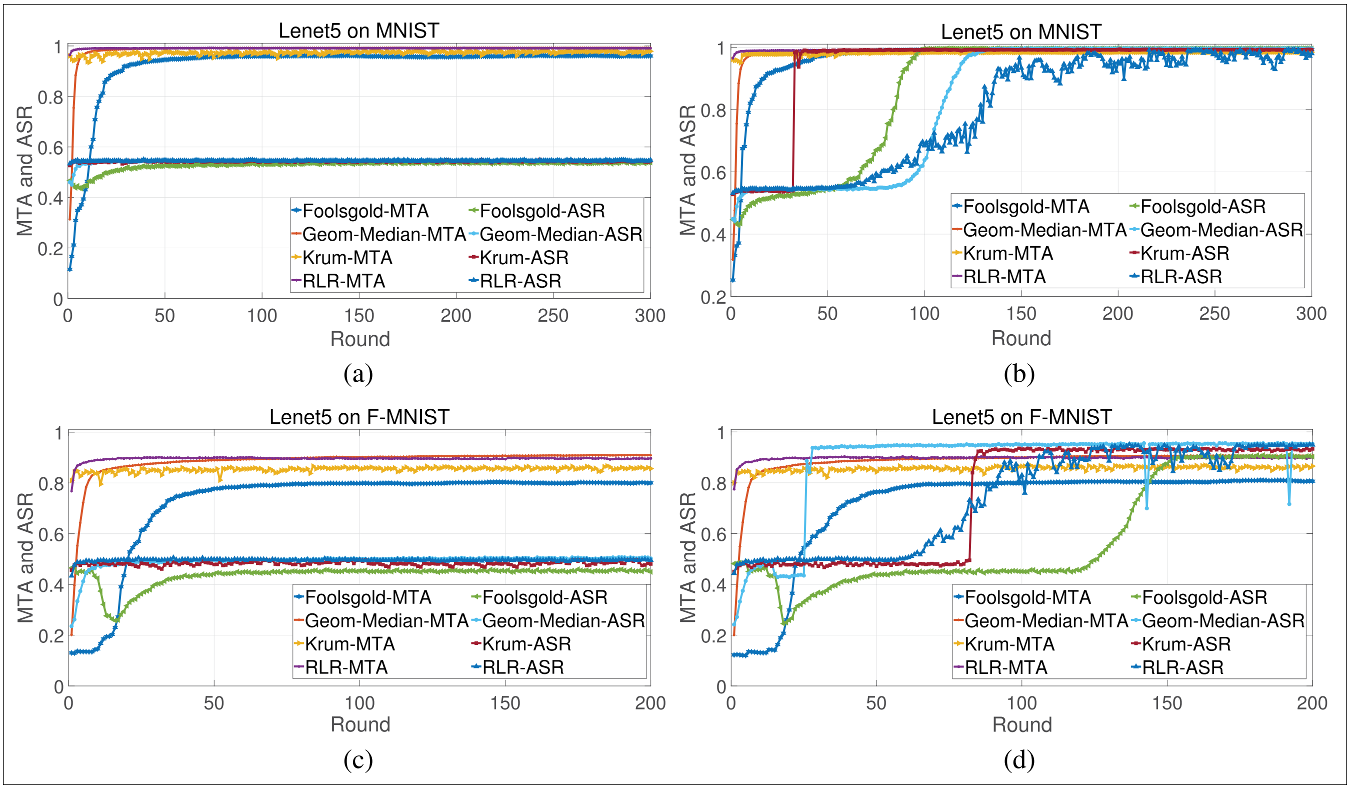
Experimental Result

Robustness: We investigate the robustness of RoPe-Door, namely whether RoPe-Door can achieve effective backdoor attacks across four Byzantine aggregation methods (Foolsgold, Geom-Median, Krum, and RLR). We conduct experiments in the tasks of Lenet5 on MNIST and Lenet5 on F-MNIST. In the Lenet5 on MNIST task, the poisoning rate in the training set is 0.1, and the poisoning rate in the test set is 0.5. Without backdoor attacks, the MTAs of the four Byzantine aggregation methods were 98%, 99%, 99%, and 99%, respectively. In the Lenet5 on F-MNIST task, the poisoning rate in the training set is 0.01, and the poisoning rate in the test set is 0.5. Without backdoor attacks, the MTAs of the four Byzantine aggregation methods were 87%, 90%, 91%, and 90%, respectively. The experimental results, as shown in Fig. 2, demonstrate that across these four Byzantine aggregation methods, RoPe-Door achieves high ASR while maintaining a high MTA compared to the Random method.

Persistence: We investigate the persistence of RoPe-Door. We posit that if the attacker poisons only once and does not poison again, and the ASR remains high in subsequent FL rounds, this indicates the existence of persistent backdoor influence. As the maximum effectiveness of a backdoor is typically achieved when the FL model converges, we choose to inject the backdoor once the model has converged. We conduct experiments in the tasks of Lenet5 on F-MNIST and AlexNet on CIFAR-10. In the Lenet5 on F-MNIST task, the poisoning rate in the training set is 0.1, and the poisoning rate in the test set is 0.5, with the backdoor injected in the 10th round. In the AlexNet on CIFAR-10 task, the poisoning rate in the training set is 1, and the poisoning rate in the test set is 0.5, with the backdoor injected in the 20th round. The experimental results, as shown in Fig. 3, demonstrate that even with only one injection, the backdoor influence implanted by RoPe-Door can still exhibit strong persistence.

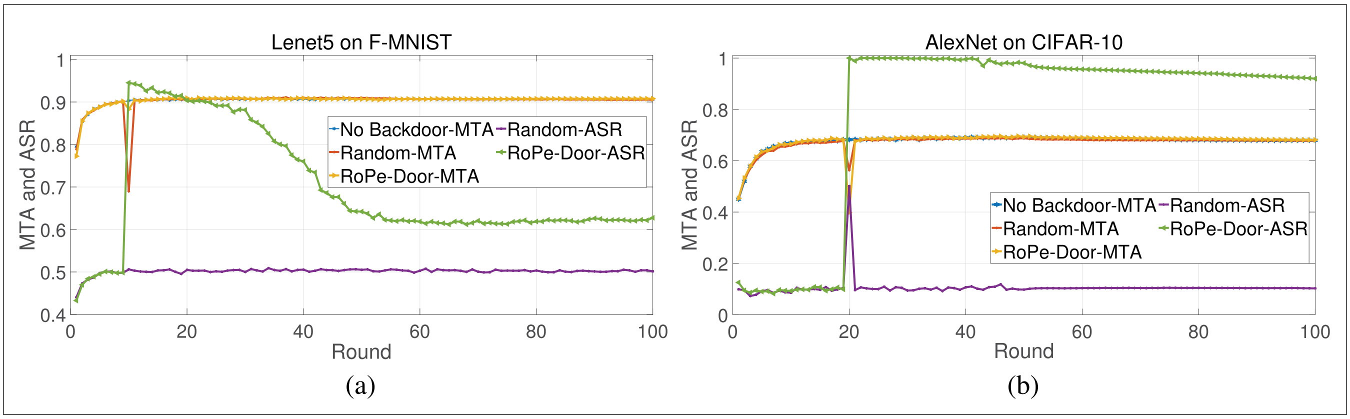
Ablation Study

Impact of the Number of Poisoned Samples: We investigate the impact of the number of poisoned samples on ASR. Specifically, we explore whether RoPe-Door can achieve effective backdoor attacks with a small number of poisoned samples. We conduct experiments across four tasks. In the Lenet5 on MNIST and Lenet5 on F-MNIST tasks, poisoning begins from the 10th round, with the number of poisoned samples set to 20, 60, and 100 for each round. In the AlexNet on CIFAR-10 and AlexNet on CIFAR-100 tasks, poisoning starts from the 20th round, with the number of poisoned samples set to 1000, 2000, and 3000 for



(a) (b) (c) (d)

FIGURE 2. The robustness for RoPe-Door. a) and c) Random. b) and d) RoPe-Door.



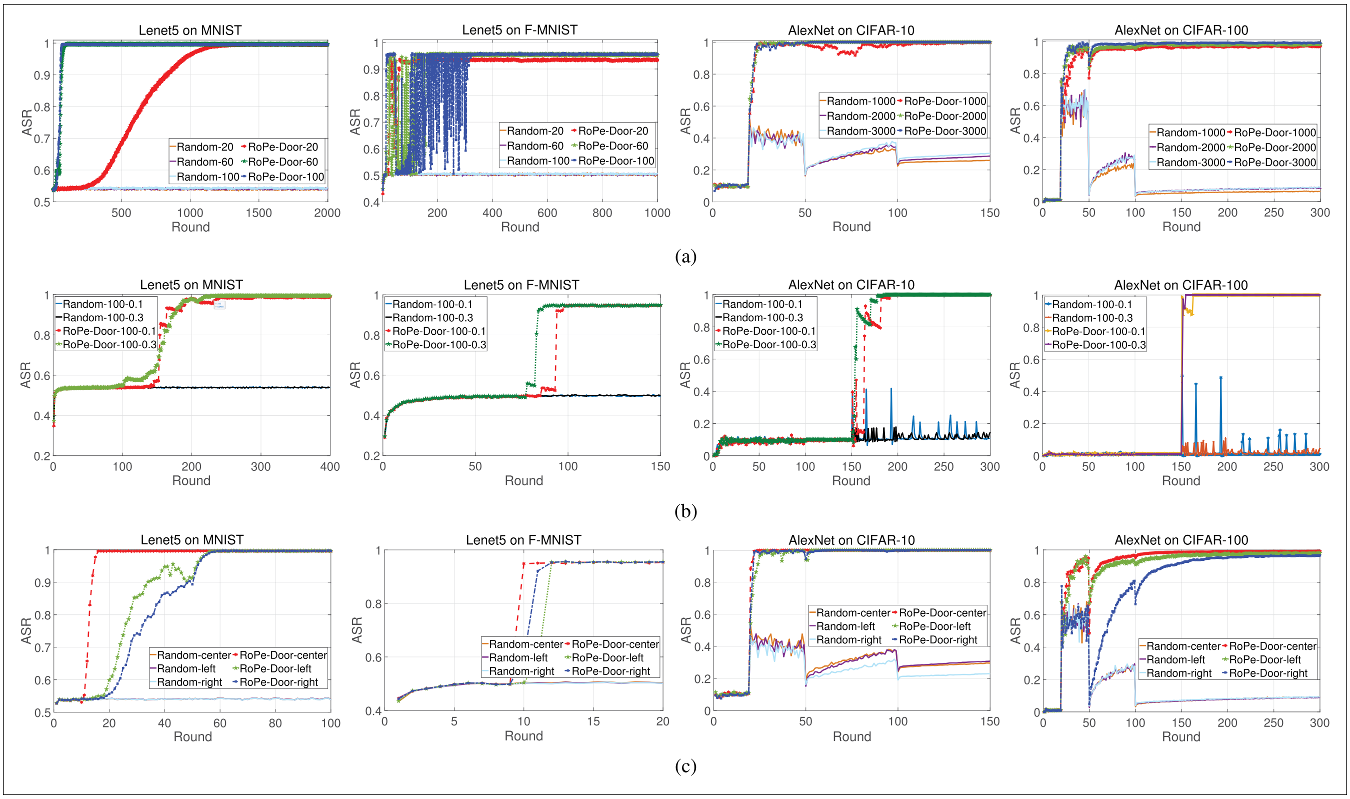
(a) (b)

FIGURE 3. The persistence of RoPe-Door.

each round. The experimental results, as shown in Fig. 4(a), reveal that even with a small number of poisoned samples, RoPe-Door maintains a high ASR.

Impact of Data Distribution: We investigate the impact of data distribution on ASR. Specifically, we examine whether RoPe-Door can achieve effective backdoor attacks under three scenarios: IID (each participant has only 100% class data), NonIID-50% (each participant has only 50% class data), and NonIID-70% (each participant has only 70% class data). We conduct experiments across four tasks. In the Lenet5 on MNIST and Lenet5 on F-MNIST tasks, the poisoning rate in the training set is 0.1, and the poisoning rate in the test set is 0.5, with poisoning starting from the 10th round. In the AlexNet on CIFAR-10 and AlexNet on CIFAR-100 tasks, the poisoning rate in the training set is 1, and the poisoning rate in the test set is 0.5, with poisoning starting from the 20th round. The experimental results, as shown in Table 1, reveal that the ASR of RoPe-Door remains unaffected by the data distribution, regardless of the scenario.

Impact of the Client Participation Ratio: We investigate the impact of the client participation ratio on ASR. Specifically, we research whether RoPe-Door can achieve effective backdoor attacks when the total number of clients is 100, and the client participation ratio is 10% and 30%, respectively. We conduct experiments across four tasks. In the Lenet5 on MNIST and Lenet5 on F-MNIST tasks, the poisoning rate in the training set is 0.1, and the poisoning rate in the test set is 0.5, with poisoning starting from the 70th round. In the AlexNet on CIFAR-10 and AlexNet on CIFAR-100



(a)

(b)

(c)

FIGURE 4. Influence Factor. a) Impact of the number of poisoned samples. b) Impact of client participation ratio.

Evaluation Metrics	Tasks	IID		NonIID-50%		NonIID-70%	
		Random	RoPe-Door	Random	RoPe-Door	Random	RoPe-Door
MTA	Lenet5 on MNIST	0.99	0.99	0.99	0.99	0.99	0.99
	Lenet5 on F-MNIST	0.90	0.91	0.89	0.89	0.90	0.90
	AlexNet on CIFAR-10	0.63	0.67	0.61	0.63	0.62	0.66
	AlexNet on CIFAR-100	0.34	0.35	0.30	0.33	0.32	0.34
ASR	Lenet5 on MNIST	0.54	0.99	0.54	0.99	0.54	0.99
	Lenet5 on F-MNIST	0.51	0.95	0.51	0.94	0.51	0.95
	AlexNet on CIFAR-10	0.53	1	0.21	1	0.31	1
	AlexNet on CIFAR-100	0.09	0.99	0.23	0.98	0.15	0.98

TABLE 1. Impact of data distribution.

tasks, the poisoning rate in the training set is 1, and the poisoning rate in the test set is 0.5, with poisoning starting from the 150th round. The experimental results, as shown in Fig. 4(b), reveal that regardless of the situation, the high ASR of RoPe-Door is unaffected.

Impact of Trigger Position: We investigate the impact of trigger position on ASR. Specifically, we study whether RoPe-Door can achieve effective backdoor attacks when the trigger is placed in the center, upper left corner, and lower right corner, respectively. We conduct experiments across four tasks. In the Lenet5 on MNIST and Lenet5 on F-MNIST tasks, the poisoning rate in the training set is 0.1, and the poisoning rate in the test set is 0.5, with poisoning starting from the 10th round. In the AlexNet on CIFAR-10 and AlexNet on CIFAR100 tasks, the poisoning rate in the training set is 1, and the poisoning rate in the test set is 0.5, with poisoning starting from the 20th round. The experimental results, as shown in Fig. 4(c), reveal that when the trigger is placed in the center, the attack performance of RoPe-Door is better. However, when the trigger is placed in the upper left and lower right corners, RoPe-Door will ultimately achieve high ASR.

Visualization

To provide a more visual representation of the trigger training process, we conduct visual observations across four tasks. Fig. 5 visualizes the training process of backdoor samples. In all tasks, the first column represents the original samples with their original labels. The last

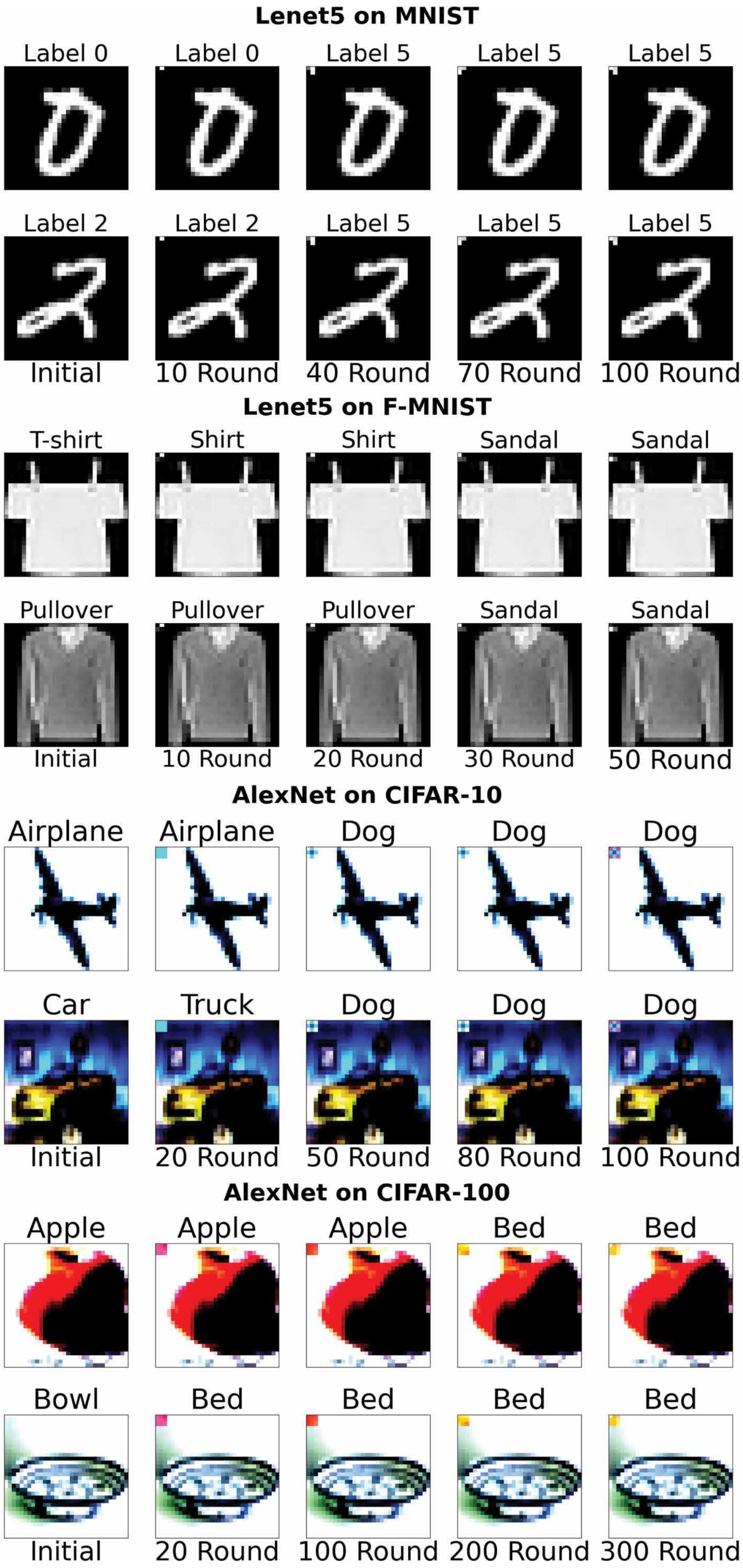


FIGURE 5. Visualization of backdoor sample.

column shows the final backdoor samples with their target labels. The middle column depicts the backdoor samples during training, illustrating that the trigger is continuously changing. Moreover, although the time taken to successfully inject the backdoor may vary across different tasks and samples, eventually, the backdoor is successfully injected in all cases.

Conclusion

In this paper, we proposed a novel method, RoPe-Door, a robust and persistent backdoor data poisoning attack in FL. Specifically, instead of using randomly selected triggers, we proposed a trigger generation algorithm based on stochastic gradient ascent, which generates triggers based on the global model. This approach enhances the effectiveness and persistence of the backdoor, and even under Byzantine aggregation methods, the RoPe-Door attack remains effective. Through extensive experimentation, we demonstrated the significance of the RoPe-Door method.

Acknowledgment

This work was supported in part by the Natural Science Foundation of Shanghai under Grant 22ZR1400200; in part by the Fundamental Research Funds for the Central Universities under Grant 2232023Y-01; in part by the National Natural Science Foundation of China under Grant 62302123; in part by the RGC RIF Grant under Contract R6021-20; in part by the RGC TRS Grant under Contract T43-513/23N-2; in part by the RGC CRF Grant under Contract C7004-22G and Contract C102922G; and in part by the RGC GRF Grants under Contract 16200221, Contract 16207922, and Contract 16207423.

References

[1] B. Gong et al., "Towards hierarchical clustered federated learning with model stability on mobile devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 7148–7164, Jun. 2024.

[2] B. Gong et al., "Adaptive client clustering for efficient federated learning over non-IID and imbalanced data," *IEEE Trans. Big Data*, early access, Apr. 19, 2022, doi: 10.1109/TBDATA.2022.3167994.

[3] H. Wang et al., "Attack of the tails: Yes, you really can backdoor federated learning," in *Proc. NIPS*, Vancouver, BC, Canada, Dec. 2020, pp. 16070–16084.

[4] A. N. Bhagoji et al., "Analyzing federated learning through an adversarial lens," in *Proc. ICML*, Jun. 2019, pp. 634–643.

[5] F. Nuding and R. Mayer, "Poisoning attacks in federated learning: An evaluation on traffic sign classification," in *Proc. 10th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2020, pp. 168–170.

[6] C. Xie et al., "DBA: Distributed backdoor attacks against federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–11.

[7] X. Gong et al., "Coordinated backdoor attacks against federated learning with model-dependent triggers," *IEEE Netw.*, vol. 36, no. 1, pp. 84–90, Jan. 2022.

[8] E. Bagdasaryan et al., "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.

[9] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8635–8645.

[10] X. Zhou et al., "Deep model poisoning attack on federated learning," *Future Internet*, vol. 13, no. 3, p. 73, Mar. 2021.

[11] Z. Zhang et al., "Neurotoxin: Durable backdoors in federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 26429–26446.

[12] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in Sybil settings," in *Proc. 23rd Int. Symp. Res. Attacks, Intrusions Defenses*, 2020, pp. 301–316.

[13] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 1142–1154, 2022.

[14] P. Blanchard et al., "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–7.

[15] M. S. Ozdayi, M. Kantarcioglu, and Y. R. Gel, "Defending against backdoors in federated learning with robust learning rate," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 9268–9276.

Biographies

Ye Liu (006012@jxnu.edu.cn) received the B.S. degree from the School of Software, East China University of Technology, Nanchang, China, in 2016, the M.S. degree from the School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China, in 2019, and the Ph.D. degree from the School of Computer Science and Technology, Donghua University, Shanghai, China, in 2024. She is currently with the School of Computer and Information Engineering, Jiangxi Normal University, Nanchang. Her research interests include efficient communication, privacy, and security of federated learning.

Shan Chang (changshan@dhu.edu.cn) received the Ph.D. degree in computer software and theory from Xi'an Jiaotong University, Xi' an, China, in 2012. From 2009 to 2010, she was a Visiting Scholar with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong. She was also a Visiting Scholar with BBCR Research Laboratory, University of Waterloo, Waterloo, ON, Canada, from 2010 to 2011. She is currently a Professor with the Department of Computer Science and Technology, Donghua University, Shanghai, China. Her research interests include security and privacy in mobile networks and sensor networks. She is a member of IEEE Computer Society, Communication Society, and Vehicular Technology Society.

Denghui Li (2222731@mail.dhu.edu.cn) received the bachelor's degree from the Department of Computer Science and Technology, Shanghai Ocean University, in 2022. He is currently pursuing the master's degree from the School of Computer Science and Technology, Donghua University. His research interests include backdoor attacks in machine learning.

Shaohuai Shi (Member, IEEE) (shaohuais@hit.edu.cn) received the B.E. degree in software engineering from the South China University of Technology, China, in 2010, the M.S. degree in computer science from the Harbin Institute of Technology, China, in 2013, and the Ph.D. degree in computer science from Hong Kong Baptist University in 2020. He is currently a Full Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His research interests include GPU computing and machine learning systems.

Bo Li (bli@cse.ust.hk) received the B.Eng. degree (summa cum laude) in computer science from Tsinghua University, Beijing, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts at Amherst, Amherst, MA, USA. He was the Cheung Kong Visiting Chair Professor at Shanghai Jiao Tong University from 2010 to 2016, the Chief Technical Advisor at ChinaCache Corporation (NASDAQ:CCIH), a Leading CDN Provider, and an Adjunct Researcher at Microsoft Research Asia (MSRA) from 1999 to 2006 and the Microsoft Advanced Technology Center from 2007 to 2008. He is currently the Chair Professor with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. He made pioneering contributions in multimedia communications and the internet video broadcast, in particular coolstreaming system, which was credited as first large-scale peer-to-peer live video streaming system in the world. It attracted significant attention from both industry and academia and received the Test-of-Time Best Paper Award from IEEE INFOCOM in 2015. He was the Co-TPC Chair of IEEE INFOCOM in 2004. He is an editor or a guest editor of more than two dozen of IEEE and ACM journals and magazines.