Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

# Clean-label poisoning attacks on federated learning IoT

Jie Yang[1] | Jun Zheng[1] | Thar Baker*[2] | Shuai Tang[1] | Yu-an Tan[1] | Quanxin Zhang*[3]

[1]School of Cyberspace Science and Technology, Beijing Institute of Technology, No. 5 Zhongguancun South Street, Haidian District, Beijing, China
[2]Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, UAE
[3]School of Computer Science, Beijing Institute of Technology, No. 5 Zhongguancun South Street, Haidian District, Beijing, China

**Correspondence**
*Quanxin Zhang, No. 5 Zhongguancun South Street, Haidian District, Beijing. Email: zhangqx@bit.edu.cn, *Thar Baker,University of Sharjah, Sharjah, UAE Email: tshamsa@sharjah.ac.ae

**Present Address**
Beijing Institute of Technology, and Univeristy of Sharjah

**Summary**

Federated Learning(FL) is naturally in line with the application scenarios of distributed edge collaboration of the Internet of Things(IoT), and has the ability to protect data security and privacy, so it is widely used in the IoT applications such as Industrial IoT. Due to the distributed nature of FL, each participant is independent and confidential, and training samples are not sent to trusted institutions for inspection. Therefore, FL cannot guarantee that all clients are honest and are vulnerable to malicious attacks. In this paper, we focus on edge-cloud synergistic FL clean-label attacks. Different from the backdoor clean label attack, in order to ensure the concealment of the attack, we add a small perturbation to realize the clean label attack by judging the cosine similarity between the gradient of the adversarial loss and the gradient of the normal training loss. In order to improve the attack success rate and robustness, we choose the attack timing when the model is about to converge. The experimental results verified that 1% of poisoned data can achieve an attack with high probability. Our method maintains stealth while performing model poisoning attacks, and the average PSNR of poisoned images reaches over 60, and the average SSIM is close to 0.93. Most importantly, our attack method can bypass Byzantine aggregation defense.

**KEYWORDS:**
IoT, Edge-cloud collaboration, Federated Learning, Clean label attack

## 1 | INTRODUCTION

Edge computing nodes have faster response speed, less bandwidth requirements, more secure local data transmission, storage and computing capabilities, which meet the requirements of industrial Internet of things in aspect of real-time reponse, security and privacy protection, and better provide intelligent services for local users[1,2,3,4]. In order to alleviate the limited computing power of the edge (the training of a single edge device is time-consuming and computational power) and the problem of data island (the edge data is local), multiple edges need to be trained together[5,6]. The distributed Federated Learning (FL) framework is naturally in line with the application scenarios of IoT edge collaboration, and has the ability to protect data security and privacy, so it is widely used in Industrial IoT[7,8,9,10]. FL is a novel distributed data security learning framework, which can be used to collaboratively train a deep learning model of multiple edge devices to meet a wider range of needs[11,2]. Figure 1 shows the FL framework for edge-cloud collaboration. This is a secure distributed training method, that is, the cloud server jointly trains a global model by aggregating parameters uploaded by multiple edge nodes. And,edge devices only share model parameters to the server without exposing their private training data[12].

However, the distributed nature of FL and the independence of each participant, participants can join and leave the alliance freely, which makes the client easy to be hacked and manipulated local data and model parameters. In particular, in deep learning models, the central server only has access to local model updates uploaded by edge nodes, which makes detection of malicious updates very difficult because it is challenging to distinguish a well-behaved model from a benign one. They are all trained on locally inaccessible data. So FL is not always safe and robust[13,14,15].

The existing FL poisoning attack methods mainly inject malicious samples into the dataset or tamper with the labels of specific samples, causing the model to output malicious labels specified by the attacker for specific samples in the inference stage[16,17]. Poison-like data essentially destroys the training process of the model, making it impossible for the model to learn the correct discriminative ability. Backdooring is achieved by adding patch-style triggers with the aim of making patches as salient classification features of backdoor samples. Label reversal attacks make the model deviate from the given predictions by reversing the training data labels. Both methods assume that the labels of the target samples are also poisoned. This greatly reduces the stealth of the attack, as samples with inconsistent content and labels can be distinguished from benign training samples by human visual inspection of the training set, defense methods, or running a pre-classification step.

We focus on FL clean label attacks in the edge-cloud collaboration framework. A clean label poisoning attack is to add fine-tuned samples that look normal and correctly labeled into the training set[18,19,20,21]. The poisoned model will classify a specific sample into the class chosen by the attacker, while maintaining good performance on the main task. Unlike the usual backdoor clean label attack, we do not add triggers, do not invert labels, but add tiny perturbations by judging the cosine similarity between the adversarial loss gradient and the normal training loss gradient to implement the clean label attack. Simultaneously minimize the distance between the poisoned client model and the global model during the training phase. Double insurance ensures the concealment of the attack, and achieves the same backdoor patch effect.

As we all know, the backdoor attack of FL needs to create many adversarial examples, and the attack effect will offset the contribution of most malicious models with the aggregation of the model, and the joint model will soon forget the backdoor. In order to achieve the robustness and concealment of the attack effect, we chose a different attack timing than before. Inspired by the pretraining-retraining idea of transfer learning, we choose the phase where FL tends to converge as the pretraining stage to obtain the parameters of the global model. Then the attacker adds poisoned samples locally and participates in subsequent model training until the model fully converges. The choice of attack timing has a great impact on the success of the attack and does not disappear too quickly or even forget as the model converges.

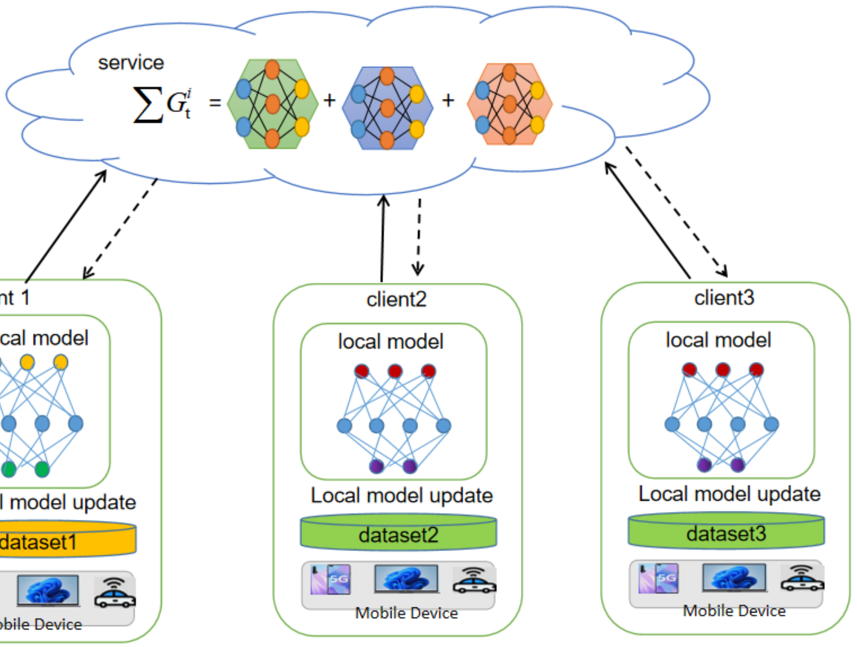

**FIGURE 1** The federated learning framework for edge-cloud collaboration

In this paper, we choose unprecedented timing for label poisoning attacks inspired by transfer learning. The contributions of this paper are as follows.

- We adopt matching to minimize the negative cosine similarity of the target gradient and the toxicity gradient, so that they are distributed in the same direction to achieve clean label attacks, while minimizing the distance between the poisoning model and the global model. The stealth goal of the attack is achieved.
- By observing the change trend of the loss value of the federated learning training process and combining the pretraining-retraining idea, we choose to implement clean label poisoning attacks in the convergence stage of the model during the federated learning training process. Mainly because the gradient updates of benign clients reflect the special characteristics of their local data. When the global model aggregates client-side gradient updates, these updates are mostly canceled out, resulting in less impact on the weights of clean label attacks.
- We implement a clean-label poisoning attack on federated learning using 1% poisoned data. Meanwhile, the accuracy of the main task has little effect on the poisoning model. The images before and after poisoning cannot be distinguished by human eyes. The average peak signal-to-noise ratio is more than 60 dB and the structural similarity is about 0.93.
- We implement a clean label poisoning attack that can bypass the defense of the Byzantine aggregation rule for FL.

The rest of the paper is structured as follows. Section 2 discusses the related work starting with related work on FL followed by Poisoning attacks on FL. Section 3 gives the threat model in the FL attack. Section 4 describes the detail the clean label attack method we proposed and the attack timing in FL. Section 5 is describes the experimental environment and the evaluation of experimental results The last section is conclusion and future work.

## 2 | RELATED WORK

### 2.1 | Federated Learning

Federated Learning(FL) as a distributed paradigm, collaboratively learns a shared predictive model, and the training data is kept locally on each client to preserve privacy[12]. During the FL process, the client trains the local model based on the local datasets, and uploads the updated model parameters to the server for secure aggregation, repeating multiple times until the learning process converges. We employ an average FL aggregation rule with one server and N clients, and disjoint private datasets. We employ a standard FL setup with one server $S$ and $C = \{C_1, C_2, \dots, C_n\}$ clients, and disjoint private datasets $D_i, i \in n$. Figure 2 shows the process of FL. Initially, the server randomly selects m clients from C and delivers the initialized global model parameters to the selected clients. Then, selected clients are trained locally using local data and global model parameters. Compute stochastic gradients $f_{arg}(\Delta_{t,i \in n})$ ,send stochastic gradient updates to the server. The server enforces the security aggregation rules $\Delta = \frac{\partial L(b,\theta_t)}{\partial \theta_t}$ and SGD computes a new model $\theta_{t+1}$ and distributes it to randomly selected clients in the next round. For privacy reasons, the server is designed to be unable to view the client's local data and training process.

$$\theta_{t+1} = \theta_t - \frac{\eta}{n} \sum_{n \in I_t} W_t^n \tag{1}$$

where,$\theta_{t+1}$ is the model update,$\eta$ is the learning rate, and $W_t^1, W_t^2, \dots, W_t^n$ is the update returned by N clients. When $\eta = 1$, the aggregation rule of FL is weighted average. Finally, the client updates each local model with the aggregated gradient information. Repeat the above steps until the loss function converges, i.e. has a lower loss $L(D_v al, \theta)$ on the validation data $D_{val}$. The FL output is the most accurate $D_{val}$ global model.

$$\theta^* = argminF(D, \theta) = argmin_{\theta \in \Phi} \frac{|D_i|}{D} \sum_{i \in n} F(D_i, \theta) \tag{2}$$

where,$F(D_i, \theta)$ is the $i$-th client's objective function.

### 2.2 | Poisoning attacks on FL

In the FL distributed learning process, compared centralized attacks, a single attack is easier to implement, so FL is easily attacked by poisoning. Below we are separately expanded from the target and capabilities of the enemy.

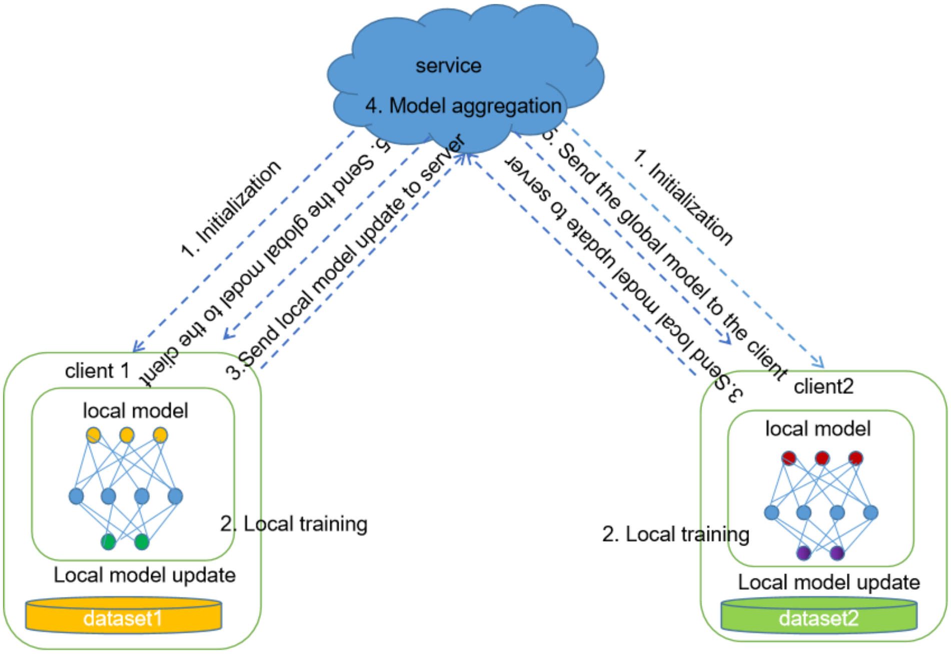

**FIGURE 2** Training process of federated learning

According to the adversary's goals, poisoning attacks include non-targeted and targeted attacks. The goal of an non-targeted poisoning attack is to maximize the error rate of the global model for any test input, making the learned model unusable, ultimately leading to a denial of service attack. For example, a Byzantine attack, which arbitrarily uploads malicious gradients to the server, causing the global model to fail. The goal of target poisoning attacks is to minimize the accuracy of the target test input while maintaining the high accuracy of other test inputs. The learned model produces predictions expected by the attacker for a specific test example. For example backdoor attack and label reversal attack.

According to the opponent's ability, the attack methods include model poisoning attack and data poisoning attack. During FL training, model parameters are shared instead of data, which can lead to model poisoning attacks. Model poisoning attacks can directly manipulate gradients on malicious devices, which are then shared to the server every epoch. In FL, model poisoning attacks are a natural and powerful class of attacks where attackers can directly manipulate updates to central servers[16,17]. Data poisoning attacks indirectly affects the gradient on the malicious device by manipulating the training dataset of the malicious client, which ultimately affects the performance of the global model[16]. Poisoned data samples can be generated directly by simple label flipping methods or by adding backdoor triggers to the training dataset to create poisoned data that trick image classifiers into assigning an attacker-chosen label to images with certain characteristics.

Among them, backdoor attack is a specific attack that attempts to compromise the integrity of data by adding specific trigger behaviors to samples. In FL, Bhagoji[22]improves the concealment of poisoning attacks by estimating local updates of benign actors, and employs an alternating minimization strategy to make the visual interpretation of model decisions indistinguishable between benign and aggressive models. Bagdasaryan[13] used a model placement strategy to inject backdoor patterns into federated models. The label flip attack is to modify the labels of the training data while maximizing the classification error of the model. Tolpegin[14]used label inversion attack for targeted attack in the FL scenario, and proved that poisoning attack can also be achieved with less poisoning data.

From the above analysis, we can know that whether adding patch triggers or label flipping attacks is to modify the label of the target sample. This seriously reduces the concealment of the attack, because samples with inconsistent content and labels can be easily found and eliminated by manually checking the training set or running a pre-trained model.

New approaches to clean label attacks have recently emerged in transfer learning and have achieved very good results. The clean label attack involves injecting the indistinguishable image of the toxic image with the correct mark into the training data, and the poisoning model will misclassify the specific target image[19].

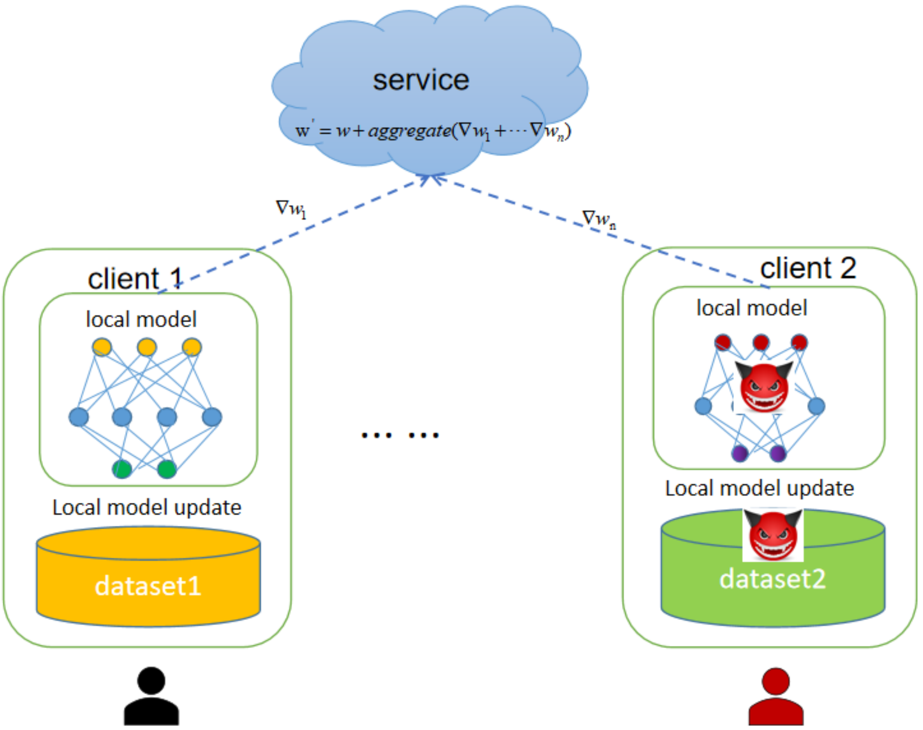

**FIGURE 3** Data poisoning attack in FL

Shafahi et al[18] proposed a "poison frog" clean label attack, where the authors retrain the model by injecting clean label Poisoned data to misclassify a given test instance with a high success rate. It also tells us that "clean label" attacks are very effective on neural networks. Mahloujifar et al[21] proposed to replace the original training samples with correctly labeled poisoned samples and their variants with independent probability p, called p-tampering attack.

In this paper, we do not add patches and invert the labels of the target samples, but only add minor perturbations to the cosine similarity between the adversarial loss gradient and the normal training loss gradient to achieve clean label attack. At the same time, the distance between the poisoned model and the global model is minimized during the attack phase. The double insurance guarantees the concealment of the attack and realizes the target of the attacker's attack.

## 3 | THREAT MODEL

In this paper, we agree that an attacker cannot modify the aggregation rules of cloud servers. Based on the existing experience, attackers have to find the opportunity of rapid convergence of the model, and need to increase the influence of malicious data. We assume that attacker cannot manipulate the labeling process of the target image, that is, the attacker cannot perform mislabeling behavior. The attacker does not need complete knowledge of the whole training set, but only needs to know the poisoning subset and a trained model parameter. In FL, the attacker pretends to be a benign client, and we endow the attacker with the ability to know the local data, the number of local training rounds, the number of global model iteration rounds, and the learning rate.

Furthermore, we assume that the attacker controls only one benign actor to launch a targeted attack. Only by adjusting the proportion of poisoned data, the number of local training rounds, attack timing and other parameters to achieve the attack. During training time attack, there may be multiple attackers, and we do not consider the case of collusion attack.

### 3.1 | Attacker's goal

In our paper, the attacker needs to achieve 3 goals. The most important goal is to classify errors on specific images while achieving high accuracy between the main and target tasks. Specifically, the attacker attempts to manipulate the federated FL model to maintain high accuracy on the main task while misclassifying specific images. We use two metrics to evaluate: 1) Attack success rate: mainly by judging the classification confidence of the target image in the global model, that is, classifying the target sample into the specified category. 2) Basic task accuracy: the global model should have high accuracy for non-target samples.

$$L_{adv}(\theta) =: L(F(x^t, \theta), y^{adv}) \tag{3}$$

$$L(\theta) =: \frac{1}{N} \sum_{i=1}^{n} L(x_i, \theta), y) \tag{4}$$

Secondly, the attacker needs to achieve the concealment of the attack effect. While realizing the main target of attack, we minimize the distance between the operation poisoning model and the global model, so that it will not be found by manual or other methods, so as to ensure its concealment.

Finally, in order to achieve the durability of the attack and good attack effect, we chose an unprecedented attack opportunity. We found that even traditional data poisoning attacks can have a great impact on the federal learning model as long as the timing of poisoning is appropriate. When the fl model tends to converge, the attacker generates a poisoning sample data set on the client side and enhances the poisoning sample data. Continue FL training on the enhanced data set until convergence. Make the retrained model incorrectly classify a specific test sample from one class to another class selected by the attacker. At the same time, the expected prediction results of other main tasks achieve high accuracy.

# 4 | PROPOSED METHOD

## 4.1 | Problem Formulation

Clean label attack is that an attacker can only add correctly labeled samples to the training set. The clean label poisoning attack is not a simple data pollution attack by adding incorrectly labeled samples to the training set, but adding correctly labeled but fine-tuned samples to the training set. This attack essentially builds a finely tuned and correctly annotated image that looks normal, with no impact on the training and general performance of the neural network. But for a specific sample it will be classified into the class chosen by the attacker.

In a FL poisoning attack, there are $M$ malicious clients, $B$ benign clients, $N$ is the total number of clients, and $M + B = N$. For the convenience of operation, select Client 1 as the malicious client. We assume that by poisoning 5% of the client's local data, the learned model is biased towards the target $L_\theta()$ Specifically, during the attack training, the benign client trains the local type and updates the model $\theta_i$,$i \in B$ upload to the server. the malicious client uploads the malicious model update $\theta_{i'}$, $i^{'} \in M$. The server aggregates the local model parameters through the averaging rule to obtain new global model parameters. The aggregation model parameters are expressed as:

$$\tilde{\theta} = \sum_{i \in B} \cdot \frac{|D_i|}{|D|} \cdot F(D_i, \theta) + \sum_{i \in M} \cdot \frac{|D_{i'}|}{|D|} \cdot F(D_{i'}, \theta) \tag{5}$$

Due to the existence of attackers, the performance of aggregation model will be poor. Therefore, the attacker's goal is to find poisoned examples that minimize the objective function $L_\theta()$. We express the training goal of each round of attacker as an optimization problem:

$$\begin{gathered} \bar{\theta} = argminF_{average}(\tilde{\theta}) \\ s.t.\tilde{\theta} = arverage(\tilde{\theta}^{'}_{i}; \theta_i), i^{'} \in M, i \in B \end{gathered} \tag{6}$$

Where,average()represents the aggregation rule,$\bar{\theta}$represents the optimal poisoning models. We substitute equation (5) into the optimization objective (6) to obtain the fl optimization formula of the attack:

$$\begin{gathered} \bar{\theta} = argminF_{average}(\tilde{\theta}) \\ s.t.\tilde{\theta} = \sum_{i \in B} \cdot \frac{D_i}{D} \cdot F(D_i, \theta) + \sum_{i \in M} \cdot \frac{D_{i'}}{D} \cdot F(D_{i'}, \theta) \end{gathered} \tag{7}$$

## 4.2 | Poison Data Generation Using Negative Cosine Similarity

We focus on a clean label attack based on gradient orientation. That is, the attacker can only trick the learner into misclassifying specific test samples by adding carefully crafted but correctly labeled samples to the training set. Figure 4 shows the process of cleaning label poisoning attack in FL. The target is a cat and is identified as a dog by the attacker. The detailed process of poisoning attack is given below. First, pre training is needed in federated learning to find the stage of convergence. Secondly, the negative cosine similarity is used to add the minimum disturbance to the image of the target sample dog, so that the characteristic image of the target image is similar to that of the aircraft. Third, put the clean non target image and the poisoned dog sample into the model for training to get the poisoning model. Finally, in the verification stage, the clean and specific cat image is input into the poisoning model, the dog category is output, and the clean dog, frog and other non target images are input into the model to get the correct classification. In the process of FL, as long as the global model is poisoned, all clients will be polluted and classified incorrectly.

In order to increase the concealment of attack, we give a double insurance strategy. First, we generate poisoning samples. Specifically, in our attack, instead of adding patches and reversing the label of the target sample, we propose an anti cosine similarity strategy, which minimizes the cosine similarity between the poisoning gradient and the target gradient, generates small disturbances, and adds them to the target sample as the poisoning sample. On the one hand, this method ensures that the accuracy of main tasks does not decline, on the other hand, it ensures that the model makes wrong prediction on poisoning samples. We call this part of the loss $L_{class\_loss}$. Here, $L_{class\_loss}$ just corresponds to the loss of equation (5),including the loss of poisoned sample and the loss of clean sample. Poisoning disturbance range is $\|\Delta_i\|^\infty \leq \epsilon$

$$\begin{aligned} L &= argmin(L_{class\_loss}) \\ &= argmin(1 - \frac{< L(F(x^t,\theta), y^{adv}), \sum_{i=1}^{P} L(F(x_i + \Delta_i, \theta), y_i) >}{\|L(F(x^t,\theta), y^{adv})\| \cdot \| \sum_{i=1}^{P} L(F(x_i + \Delta_i, \theta), y_i)\|} \end{aligned} \tag{8}$$

We enhance the robustness of the attack method. After minimizing the generated poisoning disturbance, data enhancement techniques such as clipping and horizontal flipping are carried out on the poisoning data. Finally, the changed data is restored to the original resolution and re sampled by bilinear interpolation technology.

At the same time, the distance between the poisoned model and the global model $L_{distance\_loss}$ is minimized, so that the poisoned model will not be detected due to a large deviation from the global model.

$$\begin{aligned} L &= argmin(L_{class\_loss} + L_{distance\_loss}) \\ &= argmin\alpha(L_{class\_loss\_cln} + L_{class\_loss\_adv}) + (1-\alpha)L_{distance\_loss} \\ &= argmin\alpha(1 - \frac{< L_{class\_loss\_cln} \cdot (L_{class_{loss\_adv}}) >}{\|(L_{class\_loss\_cln})\| \cdot (\|L_{class\_loss\_adv})\|}) + (1-\alpha)L_{distance\_loss} \end{aligned} \tag{9}$$

Where, super parameter $\alpha$ as a scale factor to avoid anomaly detection, it controls the scope of attacker model update to ensure that the poisoned model update parameters survives in the aggregation process of the server.

### 4.2.1 | Attack Timing About Federated Learning

Poisoning attacks can be performed at any time during the FL training phase. The traditional FL poisoning attack is that the attacker adds the poisoning data to the training from the first round of FL model training, and uploads the poisoning local update to the central server, so as to destroy the global model. However, with the increase of the number of training rounds of the global model, the attack effect will aggregate with the model, offsetting the contribution of most malicious models, and even the model will soon forget the back door.

By observing the training process of FL (as shown in Figure 5), it is found that the total number of rounds of FL model training global model training is 100 epochs, the loss will decrease rapidly before the 20th epoch, and the change speed of loss will slow down after the 20th epoch. Between the 80th and 100th epochs, the loss convergence of the model tends to be stable and the change is very small. In the early stage of model training, the FL updates the model with a large change range, and the change is relatively small in the later stage until the convergence does not change. We combine the observed change speed of model loss with the idea of pre-training and retraining, and select the convergence stage of the model for poisoning attack. The main reason why the model tends to converge is that the gradient update of benign clients reflects the special characteristics of their local data. When the global model aggregates client gradient updates, most of these updates will be offset, which has little impact on the weight of clean label attacks. Specifically, the training process before the FL model tends to converge is taken as the model

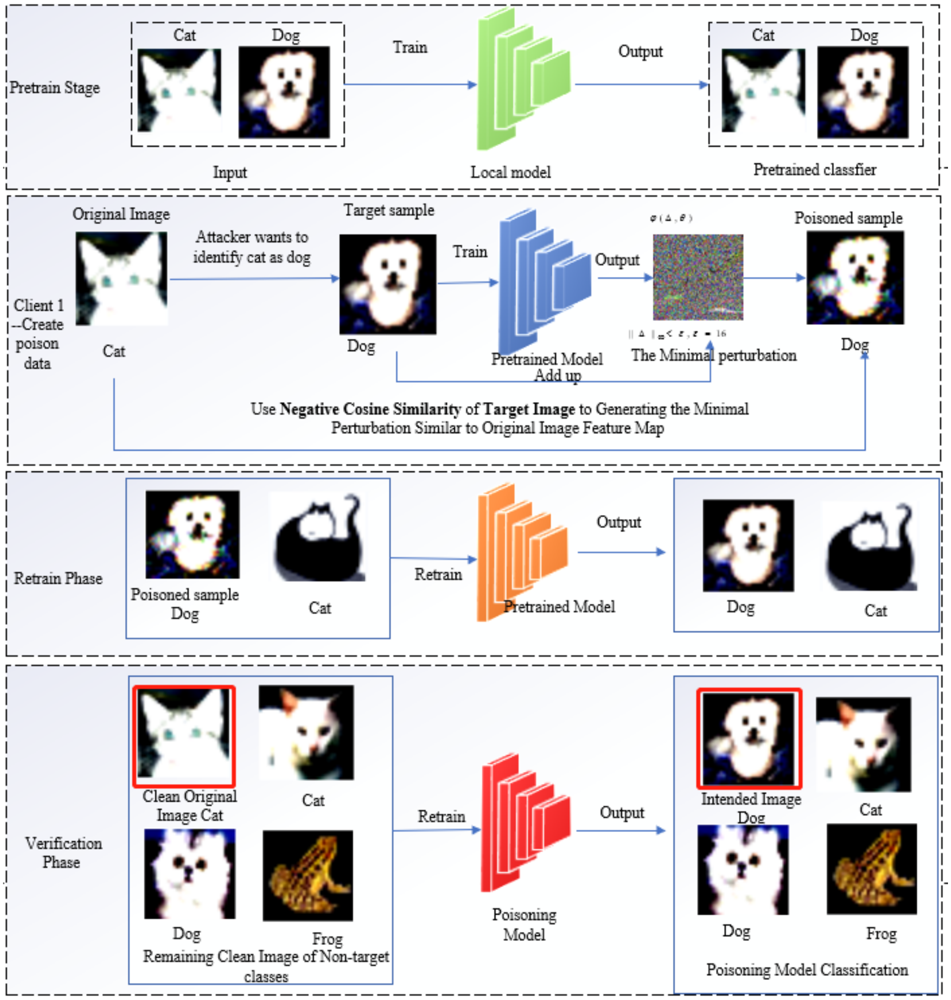

**FIGURE 4** Poisoning attack of clean labels based on negative cosine similarity

pre-training stage. At this time, the attacker selects client 1 as the malicious client and generates poison to the randomly selected target image of local data according to formula (8), so as to construct a poisoned image that looks correct but has been carefully disturbed. Put the poisoned image and the clean image into the pre trained model and train again until the model converges. This can prevent the poisoning attack effect from being offset or forgotten, and ensure that the local update of poisoning remains alive in the model average. Therefore, the choice of attack timing can not only ensure the poisoning effect after model convergence, but also effectively enhance the persistence of attack effect Figure 6 shows the attack timing of FL when the model tends to converge. The client uses formula 8 and formula 9 to poison 1% of the local target data set. Then participate in the follow-up training process of FL.

## 5 | EXPERIMENTAL EVALUATION

In this section, we use pytorch to implement attack validation on image classification data and provide detailed experimental results to evaluate our method. We mainly verify the effectiveness of our method through four experiments. 1) The attack method in this paper is verified by judging the attack success rate and benign sample accuracy. 2) Analyze the concealment of the attack method in this paper by judging the similarity between the poisoned data and the original data. 3) By comparing the attack methods of other clean samples, the efficiency of this method is proved. 4) The attack effect of this paper is verified by the Byzantine robust aggregation rule.

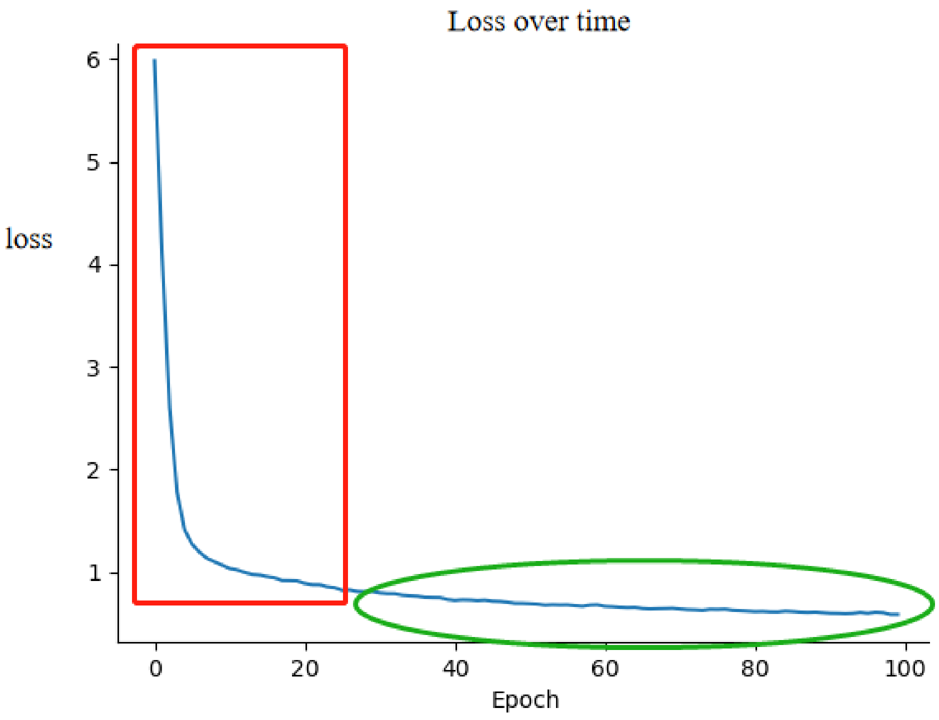

**FIGURE 5** Loss change process in federated learning

**Algorithm 1** Poison Data Generation Using negative cosine similarity

```
Require: : Pretrained FL $epoch = 80$ and obtains the global model parameters$\theta$,training samples $D = (x_i, y_i)_{i=1}^{N}$,target$(x^t, y^{adv})$, perturbation bound $\epsilon$,optimization steps m, P is the number of poisoning samples.
Begin Select $P$ training samples with label $y^{adv}$.
for i do= 1,...10 epoch:
Randomly initialize perturbations $\Delta_i^{\kappa}$
    for j do=1,...m optimization steps:
        poisoned samples $x_i + \Delta_i^{\kappa}$
        poisoned samples $x_i^{'} = (x_i + (\Delta_i^{\kappa}))$
        data augmentation to $x_i^{'}$
        Update the perturbation $\Delta$ on all poisoned samples using Equation 8, and $||(\Delta_i^{\kappa})||^{\infty} \leq \epsilon$
        Take the smallest $\Delta^{\kappa}$ as the final perturbation
    end for
return Poisoned samples $(x_i + \Delta_i^{\kappa}, y_i)_{i=1}^{N}$
end for
```

## 5.1 | Environment settings

### 5.1.1 | Datasets

We use three basic data sets, including MNIST, fashionmnist and cifar10 to evaluate our approach. These data can be downloaded from the official website. In this experiment, we use Resnet18 neural network model to train local data. Different clients use the same model. We use Cifar-10 as our image classification task, training a global model with 10 participants, with 5 randomly selected per round. We use the lightweight ResNet18 model. When training the attack model, the poisoned images are mixed with benign images in each training batch. This helps the model learn the target task without affecting its accuracy on the

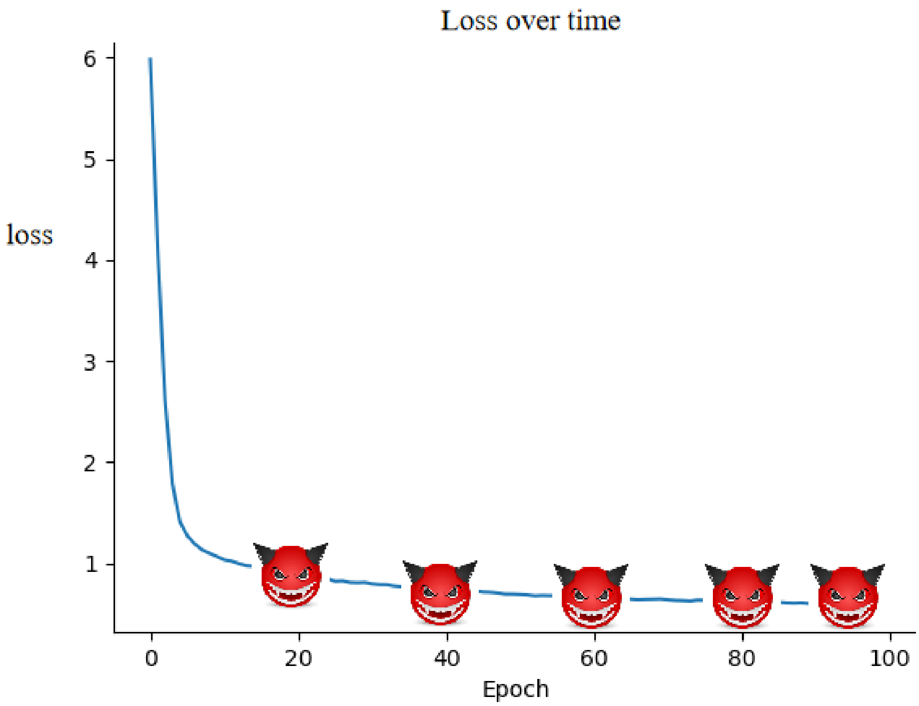

**FIGURE 6** The attack timing of federated learning when the model tends to converge

main task. The training data of the participants is very diverse, and the poisoned images represent only a small fraction, so the introduction of the backdoor has little effect on the main task accuracy of the joint model.

### 5.1.2 | Attack settings

Our experimental environment includes $C_{i=1}^{m}$10 clients, of which, $C_M$is a malicious client, and the rest are normal clients. Each round randomly selects five clients to train a global model. It is assumed that the total training data set is evenly and randomly distributed among all participants, and each participant receives a unique subset of training data. That is, keep the data independent and identically distributed (IID).When training the attack model, the poisoned images are mixed with benign images in each training batch. This helps the model learn the target task without affecting its accuracy on the main task. The training data of the participants is very diverse, and the poisoned images represent only a small fraction, so the introduction of the backdoor has little effect on the main task accuracy of the joint model. According to common practice, the first client can be selected as the client. When the d perturbation size on the CIFAR-10 data is $\epsilon = 16$, the time required for the successful attack on the FL framework and the attack success rate of the poison datasets with budgets of 1, 0.1, 0.01, and 0.05 are calculated respectively.

### 5.1.3 | Evaluation Metrics

To illustrate our proposed label poisoning attack method, we evaluate it with the following 3 objectives: **Attack success rate.** We verify the attack success rate on the global model. The attack success rate includes the target attack success rate and the main task success rate. The target attack accuracy rate refers to the percentage of the success rate of the attacker's target sample being classified as the specified label to the total number of the attacker's target records. We use the attack success rate (ASR, $ASR = N_{att}/N_{correct}$) to express. Main task accuracy is the success rate of non-target samples being classified as correct labels, denoted by recognition accuracy(ACC, $ACC = N_{correct}/N_{total}$). Where,$N_{correct}$ is the number of benign samples correctly classified by the target model, $N_{total}$ is the number of all samples, $N_{att}$ is the number of samples misclassified as target labels

**Algorithm 2** FL clean label attack algorithm

```
Require: : The number of clients is $n$, randomly select $m, m \leq n$;local training datasets $D_i, i = 1, 2, ..., n$; Global iteration number $R_g$; Local iterations number $t$; batch size $b$; learning rate $\alpha$;Poisoning time$R_g = \tau$;
Output: Global model $G$.
Random initialization global model $G_0$.
The server sends the global model $G_0$ to $m$ randomly selected clients.
//Client side excution.
Client training local models
for i do= 1,...t:
    $L_w$= $w_0$
    By using algorithm 1 manufacture poisoning data $(x_i + \Delta_i^\kappa, y_i)_{i=1}^N$
    for j do= 1,...t do:
        Randomly sample a batch $D_{bacth}$from $D_i$
        $L_i^{t+1}$=$L_i^t - \alpha \Delta Loss(D_{batch}, L_w)$ return $L_i^{t+1}$
    end for
end for
Upload the client param $L_i^{t+1}$ to server.
//Server side excution
Update global model parameters by aggregating local model gradients.
for i do= 1,...$R_g$:
    $G^{t+1} = G^t + \lambda \sum_{i=1}^m L_i^{t+1} - G^t$ return $Gt + 1$
end for
```

**TABLE 1** Model training structure of local edge device.

| Layer | Size |
|---|---|
| Input | 28×28×1 |
| Convolution +ReLU | 3×3×30 |
| Max Pooling | 2×2 |
| Convolution +ReLU | 3×3×50 |
| Max Pooling | 2×2 |
| Full Connected + ReLU | 100 |
| Softmax | 10 |

by the target model after the attack. At the same time, the proportion of malicious samples and the number of malicious clients can also affect the attack success rate.

The experiment was carried out 10 times, each time a different test set sample was used as the target image, and the attack result was 100% success rate. As can be seen from the table above, the method in this paper only needs 1% of the poisoned data to implement clean-label poisoning attacks on federated learning.When the poisoning ratio of the target image of the local client is 100%, the attack can be realized very quickly. **Main task Evaluation on performance** Another goal of ours is to ensure that the main task maintains a high accuracy rate on the poisoning model. The main task accuracy is the accuracy with which benign samples are classified to the correct label, denoted by the benign accuracy (BA). $N_{correct}$ is the number of benign samples correctly classified by the target model and $N_{total}$ is the total number of samples. We compare the accuracy of the main task before and after model poisoning. Figure 7 shows the comparison of the accuracy of the main tasks before and after model poisoning. Figure 7 shows the main task accuracy, we can see that the clean label attack has little effect on the main task accuracy and achieves good performance on all three datasets. The accuracy of the main task is almost unaffected by poisoning, which

**TABLE 2** Comparative analysis of attacks with different attack methods.

| attack method | Poisoned client num | Proportion of poisoning | Globalepoch | Localepoch | ASR |
|---|---|---|---|---|---|
| Proposed | 1 | 1 | 100 | 5 | 100 |
| Proposed | 1 | 0.1 | 200 | 5 | 100 |
| Proposed | 1 | 0.05 | 250 | 5 | 100 |
| Proposed | 1 | 0.01 | 800 | 10 | 100 |
| Poison Frog | 1 | 1 | 200 | 39.193 | 33 |
| Poison Frog | 1 | 0.1 | 300 | 1111111 | 2 |
| Poison Frog | 1 | 0.05 | 400 | 11111 | 2 |
| Poison Frog | 1 | 0.01 | 500 | 1111111111 | 2 |

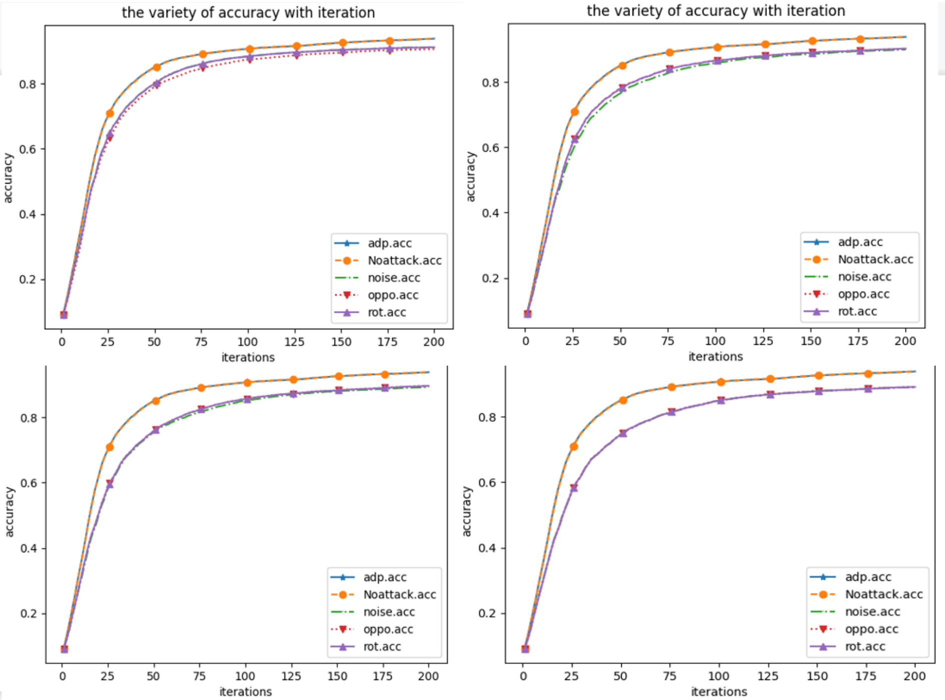

**FIGURE 7** Accuracy of main tasks before and after model poisoning

has been reduced by 0.2% from 95% before poisoning. This is mainly due to the inclusion of poisoned and benign images in each training batch during the attack. The model learns the poisoning task without compromising its accuracy on the main task.

**The Stealthiness of Clean-label Poison.** We verify the stealth of poisoning attacks by visualizing the generated poisoning samples and judging the similarity of images before and after poisoning. The indicators used to judge similarity are Peak Signal-to-Noise Ratio (PSNR) and (Structural SIMilarity) SSIM. PSNR is an objective standard to measure image distortion or noise level. The larger the value, the smaller the distortion. SSIM is measured based on the brightness, contrast and structure of the samples. The value calculated by SSIM is between $[0 - 1]$. The larger the value, the better. The larger the value, the closer it is to the original two images, and the more details of the original two images are preserved.

Figure 7 visualizes a sample of the target image with the corresponding poisoning instance. It can be seen that there is little change in the image before and after poisoning. **Attacking the Byzantine Aggregation Rule** Byzantine federated learning allows the presence of a certain percentage of attackers while hoping the global model converges and maintains high accuracy for its task. We adopt KRUM instead of the federated average aggregation rule to demonstrate the effectiveness of our attack

**TABLE 3** Concealment of poisoned images.

| Before and after pic | PSNR | SSIM |
|---|---|---|
| Original | 0.9 | 0.9 |
| Poison | 0.7 | 0.6 |

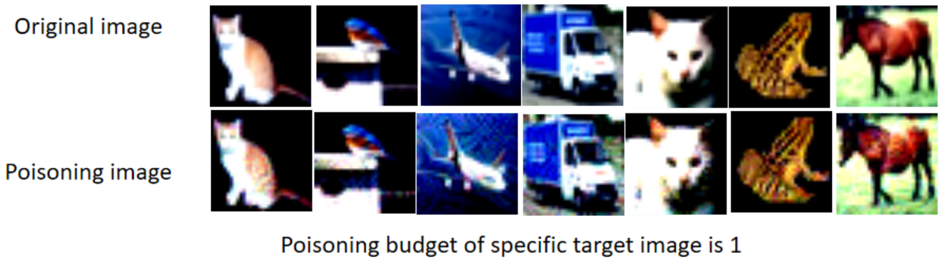

**FIGURE 8** The FL framework for edge-cloud collaboration

**TABLE 4** Attack Byzantine aggregation rule.

| Aggregation rules | Proportion of poisoning | Globalepoch | Localepoch | Accuracy of main tasks | ASR |
|---|---|---|---|---|---|
| krum | 0.1 | 1 | 100 | 5 | 100 |
| mkrum | 1 | 0.1 | 200 | 5 | 100 |
| trum | 1 | 0.05 | 250 | 5 | 100 |
| bluy | 1 | 0.01 | 800 | 10 | 100 |

method. Krum selects a model that is similar to other models among several local models as the global model. Specifically, the sum of the norm distances of the gradient and other gradients is taken as the score of the gradient, and then the gradient with the lowest score, that is, the gradient that is similar to most gradients, is selected as the aggregated gradient. The Krum algorithm will not affect the normal convergence of the model, and can ensure the robustness of the model when the proportion of attackers controlling clients does not exceed 50%.

## 6 | CONCLUSION AND FUTURE WORK

By observing the change trend of the loss value in training process of the FL and combined with the idea of pre training in training, we chooses to implement the clean label poisoning attack in the convergence stage of the model in the federal learning and training process. We use the method of negative cosine similarity to generate poisoning samples, and minimize the distance between the local poisoning client model and the global model to realize covert attack. Through experimental analysis, our method implements the attack when the FL and training process model tends to converge. Only 1% of the poisoning data can achieve the target poisoning attack without affecting the accuracy of the main task. At the same time, we can bypass the defense of Byzantine aggregation rules. Next, we will make targeted defense against the attack methods in this paper.

## ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China under Grant 2020YFB1712101, the National Natural Science Foundation of China (no. U1936218 and 62072037).

## References

1. Al-Khafajiy M, Baker T, Al-Libawy H, Maamar Z, Aloqaily M, Jararweh Y. A survey on the edge computing for the Internet of Things. *IEEE access* 2017; 6: 6900–6919.
2. Maamar Z, Baker T, Faci N, Ugljanin E, Khafajiy MA, Burégio V. Towards a seamless coordination of cloud and fog: illustration through the internet-of-things. In: The Organization. ; 2019: 2008–2015.
3. Yang W, Wang N, Guan Z, Wu L, Du X, Guizani M. A Practical Cross-Device Federated Learning Framework over 5G Networks. *IEEE Wireless Communications* 2022; 99: 1-1.
4. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. *IEEE internet of things journal* 2016; 3(5).
5. Zhang Q, Zhu L, Li Y, et al. A group key agreement protocol for intelligent internet of things system. *International Journal of Intelligent Systems* 2022; 37(1): 699–722.
6. Zhang Q, Zhu L, Wang R, et al. Group key agreement protocol among terminals of the intelligent information system for mobile edge computing. *International Journal of Intelligent Systems* 2021; 37.
7. Sun H, Li S, Yu FR, Qi Q, Wang J, Liao J. Toward communication-efficient federated learning in the Internet of Things with edge computing. *IEEE Internet of Things Journal* 2020; 7: 11053–11067.
8. Liu J, Xu H, Wang L, et al. Adaptive Asynchronous Federated Learning in Resource-Constrained Edge Computing. *IEEE Transactions on Mobile Computing* 2021.
9. Ye Y, Li S, Liu F, Tang Y, Hu W. EdgeFed: Optimized federated learning based on edge computing. *IEEE Access* 2020; 8: 209191–209198.
10. Burton D, Kenamond M, Morgan N, Carney T, Shashkov M. An intersection based ALE scheme (xALE) for cell centered hydrodynamics (CCH). In: Talk at Multimat 2013, International Conference on Numerical Methods for Multi-Material Fluid Flows. The Organization. ; September 2–6, 2013; San Francisco. LA-UR-13-26756.2.
11. Konecny J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* 2016.
12. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 2019; 184(2): 1–19.
13. Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. In: PMLR. ; 2020: 2938–2948.
14. Tolpegin V, Truex S, Gursoy ME, Liu L. Data poisoning attacks against federated learning systems. In: Springer. ; 2020: 480–501.
15. Wang X, Li J, Kuang X, Tan Ya, Li J. The security of machine learning in an adversarial setting: A survey. *Journal of Parallel and Distributed Computing* 2019; 130(2): 12–23.
16. Zhang J, Chen B, Cheng X, Binh HTT, Yu S. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE internet of things journal* 2020; 8(5): 3310–3322.
17. Fang M, Cao X, Jia J, Gong N. Local Model Poisoning Attacks to {Byzantine-Robust} Federated Learning. In: The organization. ; 2020: 1605–1622. LA-UR-13-26756.2.

18. Shafahi A, Huang WR, Najibi M, et al. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems* 2018; 31.

19. Geiping J, Fowl L, Huang WR, et al. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276* 2020.

20. Damaskinos G, El-Mhamdi EM, Guerraoui R, Guirguis A, Rouault S. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems* 2020; 33: 12080–1209.

21. Mahloujifar S, Diochnos DI, Mahmoody M. Learning under *p*-tampering attacks. In: PMLR. ; 2018: 572–596.

22. Bhagoji AN, Chakraborty S, Mittal . Analyzing federated learning through an adversarial lens. In: PMLR. ; 2019: 634–643.

23. Bonawitz K, Eichner H, Grieskamp W, et al. Towards federated learning at scale: system design. *Proceedings of Machine Learning and Systems* 2019; 1: 374–388.

□