Hindawi
Security and Communication Networks
Volume 2022, Article ID 7268347, 11 pages
https://doi.org/10.1155/2022/7268347

Research Article

Fed-Fi: Federated Learning Malicious Model Detection Method Based on Feature Importance

Chuanxin Zhou, Yi Sun, Degang Wang, and Qi Gao

Information Engineering University, Zhengzhou 450001, China

Correspondence should be addressed to Yi Sun; sunyi-1001@163.com

Received 19 October 2021; Accepted 5 April 2022; Published 18 May 2022

Academic Editor: Zhiyuan Tan

Federated learning, as a new security exchange paradigm, is widely used in medical care, driverless cars, finance, and other fields. However, federated learning still faces the problem of sybil attacks common in distributed frameworks. The existing schemes mainly defend against malicious model attacks with distance comparison, neural network, and confidence vote. But they are significantly limited in dealing with collusive sybil attacks. Therefore, we propose a federated learning malicious model detection method based on feature importance (Fed-Fi). Firstly, we screen important features by feature importance reasoning method based on LRP and compare the similarity based on Hamming distance between important features. Then, we adjust model learning rate adaptively to reduce the effect of collusive sybil attacks on the global model. The experimental results indicate that it can effectively resist the attack of collusive sybils in federated learning.

1. Introduction

When big data is gradually applied in medical, transportation, finance, and other fields, its security issues also pose a huge potential threat to user privacy. At the same time, the strict data restrictions imposed by policies and regulations also bring about new security challenges for cross-domain data exchange. Hence, the pattern based on user private data migration is no longer suitable for some new data security exchange scenarios.

As a new pattern in data security exchange, federated learning can realize cross-domain collaborative analysis without multiparty private data aggregation. In federated learning, users train private data locally to obtain models, and the server completes joint training on global data through aggregation of local models. Since the data is stored locally, federated learning is considered to be one of the most potential development directions in the field of data security exchange.

However, FL also faces some key security challenges. Malicious users may disrupt the training of the global model by sending fake model updates. The global model will change the prediction results on a specific instance. The central server can only contact the user's local model and cannot access the user's private data; therefore, malicious users can control the federated learning global model by arbitrarily modifying the local model. Recent studies have shown that federated learning is vulnerable to sybil attacks in distributed architectures. In sybil attacks, the adversary will increase the success rate of the attack by controlling multiple users. The users controlled by the sybil will exhibit relatively similar characteristics. Due to the influence of distributed architecture, federated learning is likely to be subject to saturation attacks from users, that is, sybil attacks. In sybil attacks, the adversary will influence the training of the global model by controlling a large number of users. Server without detection capability is easily controlled by the adversary. In order to achieve the same attack goal, the local models controlled by sybils often show the same characteristics. In contrast, the local model updates of honest users often show diverse characteristics. Therefore, the existing malicious model detection algorithms use similarity comparison to eliminate users whose models are too similar. On the one hand, models with too much similarity are likely to come from

users attacked by sybils. On the other hand, even if these users are honest, the excessive similarity limits their contribution to federated training.

As the level of attack-defense confrontation escalates, in order to avoid the detection of malicious model algorithms, sybil attacks are likely to coordinately tamper with model parameters to reduce the similarity between malicious users. For example, the adversary will deliberately tamper with the parameters that are not important to the prediction result. So, adversary will change the similarity of the model while minimizing the impact of tampering as much as possible. Therefore, we propose a federated learning malicious model detection method based on feature importance (Fed-Fi). Fed-Fi eliminates sybil collusion attacks by combining the characteristics of model update diversity and tampering impact minimization. Firstly, we screen important features through the feature importance reasoning method based on LRP. Then, we compare the similarity based on Hamming distance between important features of different user models and adjust the model learning rate adaptively. The experimental results show that Fed-Fi is almost unaffected in the scenario of collusive sybil attacks, which proves that Fed-Fi can effectively resist the destruction of collusive sybil attacks during federal training.

In summary, the contributions of this paper are as follows:

(1) We describe the threat model of collusive sybil attacks in federated learning from the attack target, attack capability, and attack strategy.

(2) We propose a malicious model detection method based on feature importance, Fed-Fi, which can screen and eliminate malicious models by comparing the important feature similarity of local models.

(3) Experiments prove that Fed-Fi is almost not affected by the collusive sybil attacks, and it could effectively resist the collusive sybil attacks in the federal learning.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the threat model of collusive sybil attacks. Section 4 describes the details of this method. Section 5 verifies the detection ability of the Fed-Fi algorithm through experiments and finally summarizes the work of this paper.

2. Related Work

In this section, we introduce the basic concepts of federated learning. Then, we classify and analyze the sybil attacks that federated learning may suffer. Finally, we discuss the current malicious detection algorithms based on different defense strategies.

2.1. Federated Learning. The federated learning consists of server and n users (see Figure 1). The goal of the server is to learn a machine learning model based on n users, and the goal of user is to find a model parameter w that can minimize the loss function through local training. Let $\mathscr{D}_i$ represent the local data set owned by user i, where $i \in \{1, 2, \ldots, N\}$. $|\mathscr{D}| = \sum_{i=1}^{N} |\mathscr{D}_i|$ represents the sum of data in all users. Therefore, the server aggregates the model parameters of n users to obtain

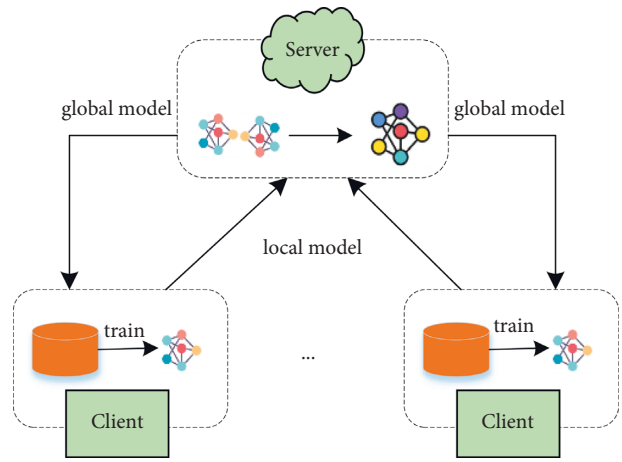

Figure 1: Federated learning.

$$w = \sum_{i=1}^{N} p_i w_i, \tag{1}$$

where w_i represents the model parameters trained by the user i, w represents the global model parameters aggregated by the server, $p_i = |\mathscr{D}_i|/|\mathscr{D}| \geq 0$, and $\sum_{i=1}^{N} p_i = 1$. In summary, the global optimization problem of federated learning can be formalized as follows:

$$w^* = \arg\min_{w} \sum_{i=1}^{N} p_i F_i(w, D_i), \tag{2}$$

where $F_i(w, D_i)$ represents the local loss function of user i, which is obtained through the local empirical risk function. The training process of federated learning usually includes four steps: (1) *local training*: all users train the machine learning model locally and send the obtained model parameters to the server; (2) *model aggregation*: the server does not need any local private data to perform security aggregation; (3) *parameter broadcast*: the server transmits the aggregated parameters to n users; and (4) *model update*: all users update their models and test the performance of the updated models.

Federated learning allows n users to train model with the help of the server jointly. The size of the loss function is inversely proportional to the accuracy of the model, so the objective function optimization of federated learning is usually to minimize the loss function.

2.2. Sybil Attacks. This paper focuses on a malicious setting in federated learning, where the adversary controls multiple users and attempts to upload carefully designed parameters to the central server. When these parameters are aggregated with honest users, the global model will learn the pattern expected by the adversary from the poisoning parameters.

We call these sybil attacks by controlling multiple nodes in a distributed system.

To realize the sybil attacks, the adversary needs to control multiple users completing the poisoning target. Existing research has done a lot of research on poisoning attack strategies. All poisoning attacks can be divided into two categories: *privacy security attacks* and *model quality attacks* (see Figure 2). Privacy security attacks refer to stealing the privacy of users' local data sets by analyzing and reasoning model updates. In the existing research, the methods to realize privacy security attacks can be divided into two types: *member reasoning attacks* [1–3] and *reconstruction attacks* [4–7]. Both derive private characteristics of local data from the federated learning local model. The *member reasoning attacks* aim to determine whether a specific sample belongs to the local training data set. Melis et al. [1] applied the member inference attack to the federated learning for the first time. They further inferred the training set features through the batch attribute classifier. Shokri et al. [2] realized the member reasoning attack by analyzing the difference between model updates with specific instances and without specific instances. Recently, Nasr et al. [3] proposed a comprehensive framework for federal learning privacy analysis based on white-box membership inference attacks, which can measure the risk of privacy leakage through model parameters. The reconstruction attacks aim to reconstruct the local training set by updating the constructed model. Fredrikson et al. [6] proposed a method of maximizing the confidence values concerning a white noise image to recover the original image from the local model of the face recognition project. Hitaj et al. [7] proved that the malicious users can reconstruct private data from other users using generative adversarial networks (GAN), if it can access all models in each training iteration.

In a poisoning attack, the adversary not only wants to steal user privacy, but also may affect the prediction result of the global model, that is, model quality attack. *Model quality attacks* are divided into *untargeted poisoning attacks* [8, 9] and *targeted poisoning attacks* [10, 11]. In untargeted poisoning attacks scenario, the adversary's goal is to establish a global model with low test accuracy, which has poor accuracy for all categories in the test data. In the targeted poisoning attacks scenario, the adversary attempts to control the prediction direction of the global model. The poisoning global model will predict specific categories as incorrect categories, such as *label-flipping attacks* [12] and *backdoor attacks* [13, 14]. This kind of targeted poisoning attacks usually does not modify the accuracy of class prediction that has nothing to do with the target class. It will avoid the possibility of being detected by the central server as much as possible. Bagdasaryan et al. [10] focused on backdoor attacks, where malicious users can use model placement strategies to inject backdoor attacks into the federated learning global model. Zhang et al. [11] deployed a GAN model on a malicious user to imitate the training data set samples of other participants. They proposed a new type of poisoning attack model for federated learning, which can successfully launch poisoning attack under threat assumptions with fewer prior conditions.

2.3. Malicious Model Detection. Due to the particularity of federated learning, the server cannot access the collection and training phases of the user's private data. It can only obtain the update of the user's local model. This special distributed structure makes the central server vulnerable to sybil attacks [15]. Many sybil attacks' defense follows an outlier-detection-based strategy to exclude outlier model updates. They assume that all honest users' local model updates are identically and independently distributed (IID), so only filtering outlier models different from most models can reduce the impact of sybil attacks. However, there is a big difference between honest model because the user data are non-IID in the federated learning. This resulted in many honest models being culled, and the global model was not suitable for all honest users.

Multi-Krum [16] selects the single local model with the smallest Euclidean distance from other local models as the aggregation model. Then, the model deviating from the overall data distribution will never be selected. Muñoz-González et al. [17] choose to exclude a local model if its cosine distance to the aggregation model deviates from the median distance plus or minus the standard derivative. However, this method has a high false-positive rate. Fool Golds [18] assumes that datasets from different honest users are different from each other and assigns different weights to users by calculating cosine similarity between local models. Therefore, users with high model similarity will be given lower weights, which to some extent harms the interests of honest users with similar data. Median [19], as a distributed algorithm based on Median, proves its robustness against Byzantine attacks by looking at Median as an aggregation model.

In addition to the distance-based malicious model detection methods, neural network training detectors and confidence voting methods are also included. Zhao et al. [20] proposed poison defense generative advermation network (PDGAN) to detect persistent attacks. Precisely, the method can reconstruct training data from model updates and audit the accuracy of each participant's model by using the generated data [21, 22]. By establishing a high-quality data test set in the central server, Su et al. [23] reduced the influence of malicious models to calculate the reputation of each user. Baffle [24] sends the global model to a subset of users who evaluate its quality based on local data and vote on whether to accept it. This approach relies on validating the user's data trigger sample. But honest users are not necessarily able to trigger hidden attack categories.

In conclusion, we believe the distance-based detection method is efficient, convenient, and practical. It has more excellent research value in federated learning malicious detection. But the similarity detection algorithm cannot resist collusive sybil attacks, and we propose a federated learning malicious model detection method based on feature importance (Fed-Fi). Fed-Fi uses the Layer-Wise Relevance Propagation to reversely derive important feature. Then, server obtain similarity between important features of different user models based on Hamming distance. Finally, server adjusts model learning rate adaptively to reduce the effect of collusive sybil attacks.

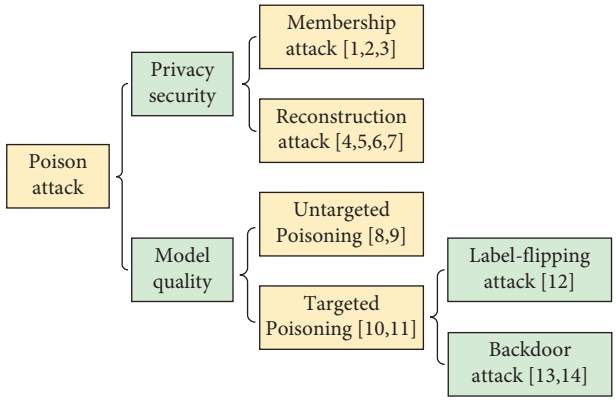

Figure 2: Malicious attack classification in federated learning.

3. Threat Model and Motivation

Here we provide detailed threat model and motivation for the sybil attacks in federated learning. Among them, $\mathscr{D}_i$ represents the local data set owned by user i, w_i represents the model parameters trained by user i, and $F_i()$ represents the loss function obtained by user i through the local empirical risk function.

3.1. Attack Target. In our attack settings, the adversary's ultimate goal is to control the prediction results of the global model by uploading the poisoned local model parameter w_i^p. In general, the adversary controls multiple users in federated learning to generate a global model that maintains high accuracy for the main prediction task, while showing obvious poisoning characteristics for specific target inputs. For example, in the classification task, the poisoning attack causes the global model to classify specific target samples to the adversary target label, while maintaining a high accuracy rate for nontarget samples to prevent being discarded by the server. In this paper, we refer to this attack feature as the tampering impact minimization, that is, the minimum modification of model parameters, to ensure that they are not discovered as much as possible under the premise of achieving the target of the attack.

3.2. Attack Ability. Here, we define the adversary's ability, which represents what he can and cannot do in federated learning. On the one hand, the attacker can (1) completely control the local training data and training process; (2) arbitrarily modify the hyperparameters; and (3) dynamically change the local model update between communication rounds. On the other hand, he cannot (1) influence the aggregation and averaging algorithms on the server; (2) control the training data or local training process of any other honest participants; and (3) change the pre-agreed local training algorithm.

In addition, in order to successfully deploy poisoning attacks in the federated learning, the adversary must increase the influence of poisoning parameters on the global model so that the poisoning parameters can dominate the prediction of the target class. We assume that the attacker can control one or more users in the federated learning to launch sybil attacks. But it needs to be explained that no matter how the adversary tampered with the parameters of the poisoning model, the adversary's attack capability must also meet the following two characteristics.

Model Update Diversity. There are n users in the federated learning, and they have different local data $\mathscr{D}_i$, so the model update w_i (parameters) is also different. The poisoning model update does not have such characteristics. Usually, the adversary will use multiple identical poisoning data to enhance the impact on the global model.

Tampering Impact Minimization. According to the principle of attack-defense confrontation, we believe that the adversary will follow the direction of minimizing impact when tampering with model parameters. The adversary attempts to tamper with the model parameters to increase the dissimilarity between the poisoning model and the honest model. It will prioritize modifying the least important parameters in the model because changing important parameters makes poisoning attacks easier to identify. The server will abandon its secure aggregation when discovering the abnormality of the model parameters.

3.3. Attack Strategy. Due to the saturated attack characteristics of sybil attacks, the similarity between the parameters of the poisoning model is generally significantly more significant than the honest model, which is an effective detection method for sybil attacks at present. We believe that the adversary can not only control multiple users in the federated learning, but also have the ability to coordinate [25]. Therefore, sybils will deliberately modify the similarity between the parameters of the poisoning model through joint negotiation to avoid server checks, that is, collusive sybil attacks.

Collusive Sybil Attacks. We assume that the local model of Sybil A is w_a and the local model of Sybil B is w_b. Sybils can deliberately change the model similarity between Sybils A and B by tampering with the value of the local model. We adjust the local model of Sybil A to $w_a^* = w_a + \delta$ and the local model of Sybil B to $w_b^* = w_b + \delta$. Then, when the aggregation operation is performed on the central server, we can get $w_a^* + w_b^* = w_a + \delta + w_b - \delta = w_a + w_b$. Therefore, by adjusting the local model, the adversary can control the similarity to any value between [0,1] and implement sybil attacks at the same time. When it comes to single malicious user, the adversary will use label-flipping attack or backdoor attack to achieve an impact on the target tag (see Figure 3). Label-flipping attack means that malicious users can make the trained model deviate from the original prediction boundary by reversing the label of the target class [26]. Different from label-flipping, backdoor attack [27, 28] requires use some specific hidden modes to train a target deep neural network (DNN) model on the poison training data $\mathscr{D}_{\text{poison}}$. The patterns chosen by the adversary are defined as

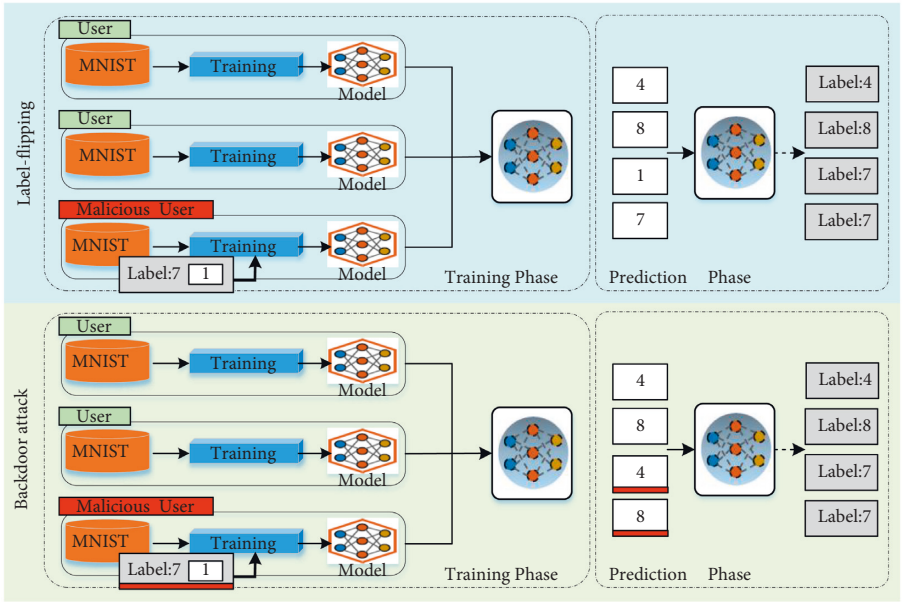

Figure 3: Label-flipping attack and backdoor attack.

backdoor triggers that cause the learning model to produce unexpected results during the prediction phase.

3.4. Motivation. The purpose of the collusive sybil attacks is to inject the final global model with label inversion attack or backdoor attack through multinode collaboration. In the FL system, the central server does not know any auxiliary information, only the gradient of the user.

Because honest users in federated learning have different data feature distributions, the local model updates obtained by the central server are diverse. However, sybils can effectively enhance influence through multiple nodes in the distributed framework. At the same time, to prevent excessively similar malicious models from being removed by the central server, the similarity of malicious models between nodes is adjusted by modifying unimportant feature parameters in the model. The fact that sybils conspire to evade detection by the central server is essentially an escalation process of offensive and defensive confrontation, making malicious models harder to detect and remove. Layer-Wise Relevance Propagation (LRP) is a feature contribution calculation method, which reversely deduces the contribution of neurons through the prediction results. Different neurons have different contributions to the prediction results. Hamming distance is a similarity comparison method, which judges the similarity degree by comparing the similarity of positive and negative signs of two vector codes. Therefore, we reverse-derive the important features of the model through LRP algorithm and then use the similarity of the more important features of the Hamming distance to defend against the collusive sybil attacks effectively.

In addition, differential privacy can effectively protect the privacy security of federated learning users by limiting L2 norm of model updates and adding randomly generated noise. This popular privacy protection mechanism is also affected by poisoning attacks. Therefore, the research on malicious model attacks has significant research value.

4. Our Proposed Scheme

In this section, we first introduce the feature importance reasoning method based on LRP in federated learning. Then, we compare the similarity between the important features of the model based on Hamming distance. Finally, we propose a malicious model detection algorithm Fed-Fi based on feature importance to realize the detection of collusive sybil attacks.

4.1. Feature Importance Reasoning. For a more vivid understanding, we collectively refer to the model parameters mentioned above as model features. The size of the parameters essentially reflects the feature information of the model, which is also common usage in machine learning. The importance of features refers to the degree of contribution to the prediction results. If the feature contributes more to the prediction result, the more important the feature is. However, due to the black-box nature of neural networks, it is more challenging to rank the importance of all features. Here we use the Layer-Wise Relevance Propagation (LRP) algorithm [29] to calculate the relationship between each feature x_{ij} and the model output $F_k(w)$.

The LRP algorithm can use the output to infer the contribution of neurons in the model. Neural network includes input layer neurons, output layer neurons, and hidden layer neurons. The model parameters are the biases and the weights connecting these neurons. Our goal is to use the LRP algorithm to rank the contribution of neurons and then find the model parameters connected to these neurons. The more outstanding the contribution to the output result, the more important the model parameters.

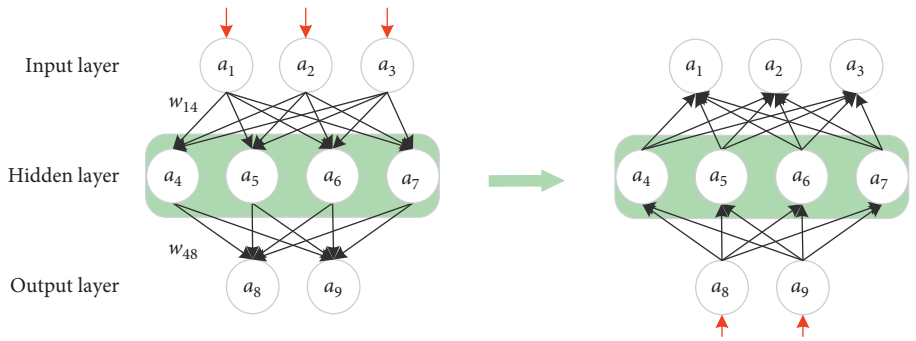

Figure 4: Layer-wise relevance propagation.

In addition, we need to make some assumptions about federated learning. This is the primary condition of the FedFi algorithm: (1) there are honest users in the federated learning; (2) the server has a small amount of sample data. First, after the server obtains the local model of the honest user, it inputs the sample data into the local model to obtain the model output $F_k(w)$. Then, through the model output $F_k(w)$, we can infer the contribution of each neuron to the output. In the neural network (see Figure 4), the correlation between all neurons is determined by correlations from all upper neurons. Among them, $R_m^k(x_i)$ represents the correlation of a neuron m in layer k, and x_i is sample data.

$$R_m^k(x_i) = \sum_{n \in M_k} R_{m \leftarrow n}^{(k,k+1)}(x_i). \tag{3}$$

For example, as shown in Figure 4, the correlation of neuron m is accumulated layer by layer through prediction results. We have

$$R_{a_1}^{l_1}(x_i) = \sum_{a_n \in l_2} R_{a_1 \leftarrow a_n}^{(l_1,l_2)}(x_i) = R_{a_1 \leftarrow a_4}^{(l_1,l_2)}(x_i) + R_{a_1 \leftarrow a_5}^{(l_1,l_2)}(x_i)$$

$$+R_{a_1 \leftarrow a_6}^{(l_1,l_2)}(x_i) + R_{a_1 \leftarrow a_7}^{(l_1,l_2)}(x_i). \tag{4}$$

Here, "$\leftarrow$" represents the connection relationship between two neurons, and (l_1, l_2) refers to the relationship between the first and second layers in the neural network. $R_{m \leftarrow n}^{(k,k+1)}(x_i)$ is composed of the ratio of local neuron to global neuron affine transformation, which is given as follows:

$$R_{m \leftarrow n}^{(k,k+1)}(x_i) = \begin{cases} \dfrac{a_m \times W_{mn}}{\sum_{m \in M_k}(a_m \times W_{mn}) + \mu} R_n^{k+1}(x_i) & \sum_{m \in M_k}(a_m \times W_{mn}) \geq 0, \\ \dfrac{a_m \times W_{mn}}{\sum_{m \in M_k}(a_m \times W_{mn}) - \mu} R_n^{k+1}(x_i) & \sum_{m \in M_k}(a_m \times W_{mn}) < 0. \end{cases} \tag{5}$$

Here, a_m represents the value of neuron m in the data instance x_i, and W_{mn} is the weight between neuron m and neuron n. A predefined stabilizer $u \geq 0$ is introduced to overcome unboundedness. According to equations (3)–(5), the contribution can be deduced inversely from the contribution of the next layer:

$$\begin{aligned} F_k(w) &= R_{a_8}^{l_3}(x_i) + R_{a_9}^{l_3}(x_i), \\ &= R_{a_4}^{l_2}(x_i) + R_{a_5}^{l_2}(x_i) + R_{a_6}^{l_2}(x_i) + R_{a_7}^{l_2}(x_i), \\ &= R_{a_1}^{l_1}(x_i) + R_{a_2}^{l_1}(x_i) + R_{a_3}^{l_1}(x_i). \end{aligned} \tag{6}$$

In summary, we can calculate the contribution of each layer and each neuron according to the above equation.

4.2. Comparison Method of Important Features. Comparing the similarity of different model parameters has been widely used in malicious model detection algorithms. We can explain the rationality of this detection idea from two perspectives: (1) If the similarity is significant and the model is malicious, then detection is successful; (2) if the similarity is large and the model is honest, the detection fails, but the contribution of the model with the large similarity to the aggregation update is also limited. Essentially, this type of algorithm converts the detected object into mathematical sense. This paper uses this idea to convert the difference between malicious model and honest model into distance difference in mathematical matrix.

Since the important features in FL model are incoherent and scattered in parameter spaces, the existing methods such as cosine similarity and Euclidean distance cannot represent the natural properties of model updating. However, the sign of model parameters is not easily changed, so we can distinguish the similarity of different local models significantly by Hamming distance [30].

The Hamming distance is derived from the transmission error control code of the data, and it represents the percentage of different bits corresponding to two characters, as equation (7). Among them, diff $(x \cdot y)$ represents the number of x, y. XOR operation result is 1. Since Hamming distance is a bit operation, it could be directly applied to specific comparisons. This paper analyses the matrix characteristics of the model, and we have made an adaptive improvement to the calculation method of Hamming distance.

$$D_{\text{Hamming}}(x, y) = \frac{\text{diff}(x \cdot y)}{\text{len}|x|} = \frac{\text{diff}(x \cdot y)}{\text{len}|y|}. \quad (7)$$

First, the model parameters are composed of signed values (positive and negative signs), and positive and negative values represent two opposite machine learning directions. When the sybil maliciously tampered with the model parameters to affect the global model, it must show similar symbols. According to the model update diversity characteristics, the similarity of the model parameter symbols of the honest user must not be as significant as the similarity between the sybils.

Then, we adjust equation (7). The adjusted Hamming distance refers to judging the number of different signs of the corresponding parameter positions between the two matrices, rather than bit operations, such as equation (8), $\overline{\text{diff}}(x, y)$ represents the position, where all parameter value signs are the same.

$$\overline{\text{diff}}(x, y) = \frac{1}{N}\sum_{i=1}^{N} I\left(\text{sgn}\left(x_i\right) = \text{sgn}\left(y_i\right)\right). \quad (8)$$

In summary, the working principle of malicious model detection based on Hamming distance is as follows: the server obtains all local model parameters $w_{i,t}$ and uses the Hamming distance to perform similarity detection on the local model to obtain the similarity matrix D_i. Then, the matrix is mapped to the range of [0,1] to get the matrix α, and the logic function is used to map the matrix α to make the similarity more reasonable distribution. Finally, the learning rate of each model is obtained, which is aggregated by the server into the global model. See Algorithm 1 for details.

4.3. Fed-Fi. According to the collusive sybil attacks strategy in Section 3.3, this paper summarizes that the collusive sybil usually follows the principle of tampering impact minimization. The sybil will deliberately tamper with the features that are not important to the prediction result. Based on this, we use the local model to calculate the feature sequence on the global model prediction result. Then, we compare the Hamming distance between the feature sequences of different user models. Below we analyze the algorithm components one by one. For details of the algorithm, see Algorithm 2.

Model Training. In the user operation phase, user k is responsible for training the local model M_k using private data D_k. The model structure and initial model are delivered to each user by the server. The poisoning user obtains the sybil poisoning data D_k^s through the poisoning attack, and the honest user's data is D_k^h. The user trains the local model M_k under the predetermined epoch and batch. The adversary will control multiple users to implement sybil attacks and at the same time adjust the local model through the item δ to avoid the detection algorithm of server.

Feature Selection. In the operation phase of the central server, the server performs an important feature sorting operation after receiving the user's local model M_k. The server first obtains the model update of the honest user and then runs the LRP algorithm to reverse the important features of the deep neural network (DNN), obtains the importance of all user's features $R_m^k(x_i)$, and sorts them. According to equation (6), the server can reversely calculate the correlation $R_m^k(x_i)$ between any neuron and the final output result $F_k(w)$ through the weight parameter, where the correlation $R_m^k(x_i)$ represents the size of its contribution. At the same time, we set the parameter p to control the ratio of important features, where the value of p is (0,1]. When p is 1, the server selects all the features for similarity comparison; M_k^i represents the filtered local model.

Similarity Contrast. The server can obtain the local model M_k^i based on the feature importance after feature selection and use algorithm 1 to compare the similarity of different users. Finally, we get the similarity situation of important features, used to detect the behavior of sybils adding disturbance to unimportant features.

Model Aggregation. After the above operations steps, as in equation (1), the central server will perform weighted

for all clients i **do**:
  $S_i = \sum_{t=1}^{T} w_{i,t}$
  for all other clients j **do**:
    $\overline{\text{diff}}(S_i, S_j) = 1/N \sum_{i=1}^{N} I(\text{sgn}(S_i) = \text{sgn}(S_j))$
  $D_i = \overline{\text{diff}}(S_i, S_j)/\text{len}(S_i)$
  $\alpha = D_i/\max(D_i)$
  $\alpha = k(\ln[\alpha/(1-\alpha)] + 0.5)$
  $w_t = w_{t-1} + \sum_i \alpha_i w_{i,t}$

Algorithm 1: HMFL.

Client operation:
  //Model training
  if client k is sybil **then:**
    $D_k^s \leftarrow$ poisoning D_k with collusive sybil
  else:
    $D_k^h \leftarrow D_k$
  for each local epoch i from 1 to E **do** :
    for batch $b \in B$ **do**
      $M_k = M_G - \eta \nabla L(M_k, b)$
    end for
  return M_k
Server operation:
  //Feature selection
  $R_m^k(x_i) = \sum_{n \in M_k} R_{m \leftarrow n}^{(k,k+1)}(x_i)$
  $M_k^i \leftarrow \max_p(R_m^k(x_i))$
  //Similarity contrast
  for all clients i **do:**
    $\alpha \leftarrow HMFL(M_k^i)$
  end for
  //Model aggregation
  $M_G = 1/n \sum_{k=1}^{n} \alpha_i M_k$

Algorithm 2: Fed-Fi.

aggregation during model aggregation. Fed-Fi will compare the similarity of the user's local model to obtain the weight value α of each user. That is, when the global model is aggregated, the local user k corresponds to the weight value α_k.

5. Experimental Evaluation

5.1. Experimental Setup. We evaluate the malicious model detection algorithm on the MNIST dataset [31], a handwritten image dataset labeled 0–9. Among them, the training set contains 60,000 image instances with a size of 28×28. The test set contains 10,000 instances. The research of Shen et al. [32] showed that among all the poisoning attacks launched on the MNIST data set, the most easily distinguishable source label and target label is (2,6), and the most difficult to distinguish source label and target label is (4,8), so we use these two sets of experiments as test cases for attack detection algorithms. Among them, test 1 is source label 2, poisoning target label 6; test 2 is: source label 4, poisoning target label 8.

To better reflect the non-IID scenario of federated learning, we set up a total of 10 users, each of which has only one type of sample data, which means that any user cannot train the global model separately, only through federated learning to obtain the classification ability of 10 categories. In addition, in the selection of malicious attacks, we use the most commonly used label-flipping attack as an attack method and set the malicious users to 5 and 10, respectively, to simulate sybil attacks that can control multiple users. All users use the model structure and initial parameters issued by the server.

We injected item δ into the 50% weight value of model. The disturbance item δ is the smallest value in the weight of the model, which is used to simulate the real collusion sybil attack scenario. Finally, it should be noted that, to ensure the validity of the experimental results, all experimental results in this paper are the average of 10 experiments.

5.2. Malicious Detection Algorithm Based on Model Similarity. We select the most popular detection algorithms and evaluate their robustness in sybil attacks. Among them, the Fool Golds algorithm represents the similarity comparison based on the cosine direction; the Multi-Krum algorithm represents the similarity comparison based on the Euclidean distance; the Median algorithm is not based on the similarity comparison, but this idea is also based on the perspective of mathematical analysis. This paper also includes this algorithm in the scope of comparison. The HMFL algorithm is a malicious model detection algorithm based on Hamming distance designed by ours.

Figure 5 shows the accuracy of the global model test when different numbers of sybil attacks are added to the federated learning. First of all, we observed that ten sybils had a more significant impact on the accuracy of the global model than five sybils. Secondly, algorithm's accuracy with a target label of 6 is slightly higher than that of the target label with 8. We believe that the easier to distinguish classes are more difficult to attack. Finally, the Fool Golds and HMFL have the best results, which can effectively defend against the influence of different numbers of sybil attacks. The Multi-Krum algorithm based on Euclidean distance is probably stable at 70%–80%. We think this is because the algorithm basically cannot resist the influence of sybil attacks. The target label cannot be accurately judged, and the performance of other labels is no change. The Median basically cannot reflect the correct test set label. This defense algorithm is obviously not successful in dealing with sybil attacks.

We show the details of several algorithms for defending against sybil attacks (see Table 1), where the accuracy rate represents the ratio of the source-target 2 (4) still predicted to be 2 (4). The false alarm represents the ratio of the source-target 2 (4) is predicted to be 6 (8). Among them, the probability that the HMFL and Fool Golds recognize label 2 as label 2 is similar to the global model accuracy, which further shows that the algorithm successfully defended against sybil attacks. The Median and Multi-Krum show that they are unable to defend against sybil attacks, and more than 95% of the source tag 2 (4) is successfully predicted to be the target tag 6 (8).

In summary, we can conclude that HMFL is similar to the current optimal Fool Golds algorithm and can effectively defend against the influence of sybil attacks. In the next section, we will test Fed-Fi to deal with the scenario of collusive sybil attacks.

5.3. Algorithm Analysis Based on Feature Importance. In a real attack scenario, the sybil may know the server's defense strategy in advance, and the defense algorithm based on similarity is often easy to breakthrough by collusion. Figure 6(a) shows the influence of collusive sybil attacks in federated learning. When the sybils added perturbation item δ to the 1%, 10%, and 100% features in the model, the accuracy rate dropped rapidly to 78.42%, 79.56%, and 77.42%. That is to say, almost all source labels were identified as target labels in the test set. Figure 6(a) indicates that none of the previous malicious model detection algorithms can effectively resist collusive sybil attacks.

Figure 6(b) shows the performance of the Fed-Fi under the collusive sybil attacks. Firstly, it shows that the attack rate is inversely proportional to the accuracy. Secondly, with the adjustment of parameter p in the Fed-Fi (proportion of selection of important features), when $p < 50\%$, the impact of the collusive sybil attacks is rapidly reduced, only 2%–4%; when $p > 50\%$, the performance of Fed-Fi is the same as that of traditional malicious model detection algorithm. In summary, we can see that the collusive sybil attacks can significantly affect the accuracy of the global model. The existing malicious model detection algorithm cannot effectively defend against this attack. Fed-Fi can achieve high

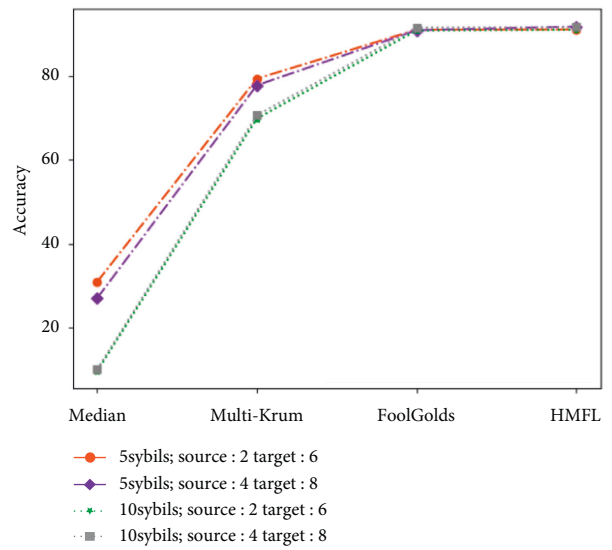

Figure 5: Accuracy of malicious detection algorithm under sybil attacks.

Table 1: Performance index of malicious detection algorithm under sybil attacks.

Malicious detection algorithm		Source: 2, target: 6 (%)	Source: 4, target: 8 (%)
Median	Accuracy	0	0
	False alarm	98.87	99.79
Multi-Krum	Accuracy	0	0
	False alarm	94.57	98.47
Fool Golds	Accuracy	88.27	91.95
	False alarm	0.48	0.92
HMFL	Accuracy	91.57	92.32
	False alarm	0.2	0.31

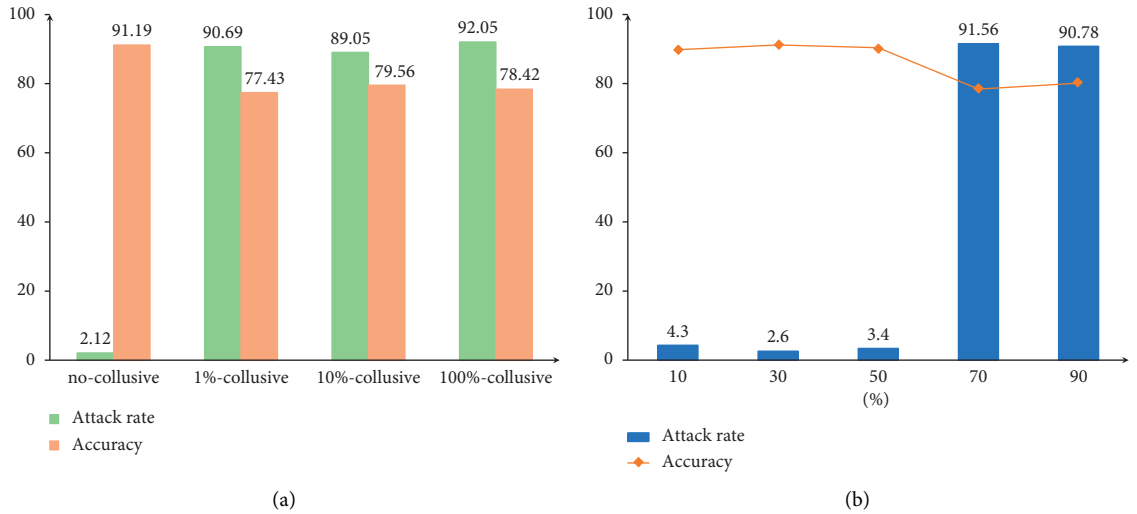

Figure 6: Fed-Fi algorithm performance. (a) Influence of sybils with varying degrees of disturbance p. (b) Influence of different percentages of important characteristics.

Table 2: Comparison with other algorithms.

	Accuracy (%)	Attack rate (%)
Fed-fi	90.45	3.43
Fool golds	79.27	91.17
HMFL	80.21	90.56

defense capability through important feature screening, but this also depends on the selection of parameter p. We believe that set different p for multiple sets of learning is a better solution to malicious attacks. If the accuracy of global model with varying values of p is considerable, it shows that the federated learning being attacked by the collusive sybils. If the impact of different p values on accuracy are limited, it shows that there is no collusive sybil attacks or it has been filtered.

Table 2 shows the comparison of accuracy and attack rate between Fed-Fi and other algorithms. From the table, it can be seen that the accuracy of Fed-Fi under collusive sybil attacks is 90.45%, while Fool Golds algorithm and HMFL algorithm are bare unable to resist this attack. Its attack rate is as high as 91.17% and 90.56%. Experimental results have shown that Fed-Fi is an effective malicious model detection algorithm.

6. Conclusions

With the rise of people's awareness of privacy and the restrictions of data security laws, the traditional pattern could be adapted to some new data security exchange scenarios. As a result, more and more users refuse to share private data in their region due to privacy security concerns. As a new paradigm in data security exchange, federated learning can realize cross-domain collaborative analysis without multiparty data aggregation. It is considered to be the most promising direction for data security exchange. Aiming at the collusive sybil attacks scenario, we propose Fed-Fi based on the importance of features. In this paper, we show the effectiveness of this detection method. Firstly, we screen important features through feature importance reasoning method based on LRP. Then, we compare the similarity based on Hamming distance between important features and adjust the model learning rate adaptively.

On the other hand, the Fed-Fi still has some shortcomings. For example, the server needs to set a reasonable detection threshold p. If the threshold is set unreasonably, the detection efficiency of the server will decrease. In this regard, we believe that multiple sets of detection thresholds p can enhance the credibility of the detection algorithm. In the future, we will also evaluate more malicious model detection algorithms and further expand sybil's attack capabilities and attack scenarios and explore more efficient malicious detection algorithms.

Data Availability

The data used to support the findings of this study are available from the first author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

[1] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 691–706, IEEE, San Francisco, CA, USA, May 2019.

[2] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 3–18, San Jose, CA, USA, May 2017.

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

[4] Y. Aono, T. Hayashi, L. Wang, S. Moriai, and L. T. Phong, "Privacy-preserving deep learning: revisited and enhanced," in *Proceedings of the International Conference on Applications and Techniques in Information Security*, pp. 100–110, Auckland, New Zealand, July 2017.

[5] Q. Su, H. Wang, C. Sun, B. Li, and J. Li, "Cyber-attacks against cyber-physical power systems security: state estimation, attacks reconstruction and defense strategy," *Applied Mathematics and Computation*, vol. 413, Article ID 126639, 2022.

[6] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic counitemeasures," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, Denver CO USA, October 2015.

[7] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 603–618, Dallas TX USA, November 2017.

[8] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)*, August 2020.

[9] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: breaking byzantine-tolerant SGD by inner product manipulation," in *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence, UAI*, Aviv, Israel, July 2019.

[10] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the AISTATS*, pp. 1–10, August 2020.

[11] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "PoisonGAN: generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet of Things Journal*, vol. 8, 2020.

[12] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, 2010.

[13] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, AISTATS*, August 2020.

[14] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," 2017, https://arxiv.org/abs/1708.06733.

[15] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proceedings of the Annual Computer Security Applications Conference*, pp. 148–162, San Juan, PR, USA, December 2019.

[16] P. Blanchard, El Mahdi El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: byzantine tolerant gradient descent," in *Proceedings of the Advances in Neural Information Processing SystemsNIPS*, Red Hook, NY, USA, December 2017.

[17] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," 2019, https://arxiv.org/abs/1909.05125.

[18] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020)*, pp. 301–316, San Sebastian, Spain, October 2020.

[19] Y. Dong, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-Robust distributed learning: towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning, ICML*, Stockholm, Sweden, July 2018.

[20] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, and S. Yu, "PDGAN: a novel poisoning defense method in federated learning using generative adversarial network," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 595–609, Melbourne, VIC, Australia, December 2019.

[21] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *Proceedings of the Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 374–380, Rotorua, New Zealand, August 2019.

[22] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding distributed poisoning attack in federated learning," in *Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 233–239, Tianjin, China, December 2019.

[23] L. Su, Z. Liu, and J. Ye, "Reputation-based defense scheme Against backdoor attacks on federated learning," in *Proceedings of the 2021 International Conference on Big Data Analytics for Cyber-Physical System in Smart City*, pp. 949–955, Shanghai,China, December 2021.

[24] S. Andreina, G. A. Marson, H. Möllering, and G. Karame, "Baffle: backdoor detection via feedback-based federated learning," in *Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pp. 852–863, Washington DC, USA, July 2021.

[25] A. S. Rakin, Z. He, and D. Fan, "Bit-flip attack: crushing neural network with progressive bit search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1211–1220, Seoul, Korea (South), November 2019.

[26] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 6, pp. 1893–1905, 2015.

[27] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: a survey," in *Proceedings of the 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pp. 471–478, Parana, Brazil, November 2018.

[28] B. Wang, Y. Yao, S. Shan et al., "Neural cleanse: identifying and mitigating backdoor attacks in neural networks," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy S&P*, pp. 707–723, San Francisco, CA, USA, May 2019.

[29] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. M¨uller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS One*, vol. 10, no. 7, Article ID e0130140, 2015.

[30] W. Luping, W. Wei, and L. I. Bo, "CMFL: mitigating communication overhead for federated learning," in *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 954–964, Dallas, TX, USA, July 2019.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[32] S. Shen, S. Tople, and P. Saxena, "Auror: defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pp. 508–519, Los Angeles CA USA, December 2016.