Hindawi
Security and Communication Networks
Volume 2022, Article ID 8928790, 13 pages
https://doi.org/10.1155/2022/8928790

Research Article

Delta-DAGMM: A Free Rider Attack Detection Model in Horizontal Federated Learning

Hai Huang,[1] **Borong Zhang**,[1] **Yinggang Sun**,[1] **Chao Ma**,[1] **and Jiaxing Qu**[2]

[1]*School of Computer Science and Technology, Harbin University of Science and Technology (HUST), Harbin, China*
[2]*Heilongjiang Province Cyberspace Research Center, Harbin, China*

Correspondence should be addressed to Hai Huang; hust_hh@vip.163.com

Received 3 December 2021; Revised 9 May 2022; Accepted 24 May 2022; Published 30 June 2022

Academic Editor: Van-Hau Nguyen

Federated learning is a machine learning framework proposed in recent years. In horizontal federated learning, multiple participants cooperate to train and obtain a common final model. Participants only need to transmit the local updated model instead of local datasets. Some participants do not use effective local data sets, but provide disguised model parameters to participate in federal training and obtain common training models. This attack is called Free-rider attack. To the best of our knowledge, researches have proposed some Free-rider attack strategies with theoretical support, but there are few researches on Free-rider attack detection. However, the model disguised by some attackers using special attack strategies is similar to the real model in terms of convergence and weight, so it is difficult to detect the model provided by attacker as abnormal data. Based on DAGMM, a high-dimensional abnormal data detection model, this paper optimizes the sample processing and compression model, and proposes an improved detection algorithm, called Delta-DAGMM. Two types of large datasets are used for experiments. The experimental results show that Delta-DAGMM has higher precision and F1 score than DAGMM. On average, the Delta-DAGMM algorithm achieves a precision of 98.42% and an F1 score of 98.36%.

1. Introduction

Data security and privacy protection have gradually become the focus of attention of major Internet companies and research institutions. With the continuous introduction of relevant laws and regulations in various countries, it has become a key issue for people to research that how to conduct deep learning without infringing privacy of others, a framework called federated learning [1] was proposed. In federated learning, participating entities do not need to share local data sets, but only transmit the model of their local training updates, so as to protect the data privacy of themselves.

According to the characteristics of the data distribution of different training participants, federated learning can be divided into three types: vertical federated learning, horizontal federated learning and federated transfer learning [2]. Horizontal federated learning, also known as feature-based federated learning. In the horizontal federated training, the characteristics of the participants' data sets basically overlap, but the sample sources of the data sets are different. For example, two tumor hospitals in different regions respectively use their patient tumor image information as samples for horizontal federated training.

In the horizontal federated learning, a parameter server coordinates all clients for iterative training. The parameter server first initializes a global model. In each round of training, the parameter server distributes the global model to each participant client. Each participant client uses local data for training based on global model to get the local update model and uploads it to the parameter server. The parameter server receives the local model of all clients, and uses federated averaging algorithm (Fed-AVG) [3] for model aggregation to obtain a new round of training global model. However, some malicious or dishonest clients upload fake local models to the parameter server without local training, as shown in Figure 1, this attack is known as Free-rider attack [4].

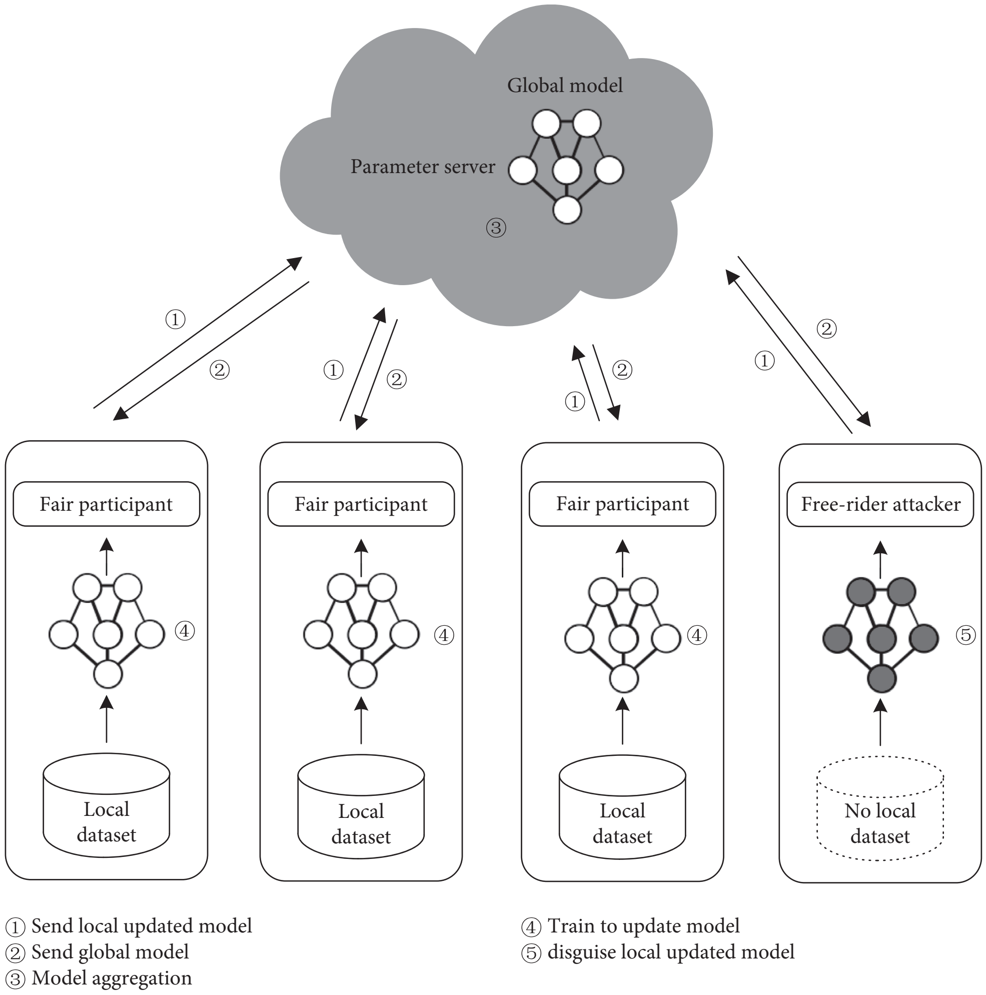

Figure 1: The training process of horizontal federated learning with the Free-rider attacker.

There are few researches on Free-rider attacks in federated learning. In the only few researches, Fraboni et al. [5] summarized several Free-rider attack methods and provided theoretical support. Lin et al. [4] proposed that the DAGMM model can be used to detect abnormal data, but did not provide experimental data such as paradigm defense and detection methods and accuracy. DAGMM is a detection model used to detect high-dimensional abnormal data [6]. Through experiments, we found that this model got a high precision on the detection of two Plain Free-rider attack strategies. The reason was that the local model parameters generated by the attackers who using these attack strategies were filled with random or fixed numbers, these model parameters can be detected as high-dimensional exception data compared to the local model parameters provided by fair participants. For attack strategies such as directly copying the global model or adding differential perturbation to the global model, the detection precision of DAGMM is not high. The reason is that the model parameters of the attackers differ little from other fair participants in terms of gradient and convergence. It is difficult for DAGMM to detect these model parameters as abnormal data.

To overcome the limitations of existing methods, we improved the DAGMM model and design an optimized Free-rider attack detection method Delta-DAGMM. To effectively detected the attackers using disguised Free-rider attack strategies, this detection method originally included sample processing in the input sample. The input samples were the model parameters transmitted by the participants. we calculated increment of the model parameters of the participants relative to the global model parameters of the current round, then input the samples into the compression network after linear processing. In the compressed network, sample features were extracted. Finally, we input the sample features into evaluation model to calculate the energy/likelihood, and a threshold was set to determine whether it is a Free-rider attacker. In order to verify the universality of the method, we selected a large number of different types of samples for experiments. In addition, we also compared this method with DAGMM. Experimental results showed that Delta-DAGMM can achieve higher precision and F1 score.

Major contributions of this paper include:

(i) An optimized detection algorithm of high-dimensional abnormal model parameters, Delta-

DAGMM, was proposed to detect the Free-rider attackers with various attack strategies.

(ii) For the disguised Free-rider attack strategies, the sample processing link of the input detection model was optimized. Straightened the increments of the local update model relative to global model to obtain the input samples of the detection model.

(iii) In order to accurately obtain the sample features, we optimized the feature extraction of the DAGMM model compression network, so that the output energy/likelihood of the Free-rider attackers' model parameters will be larger that could be more easily evaluated as abnormal data in the evaluation network.

The rest of this paper is organized as follows. In Section 2, the methods of attack, defense and detection in horizontal federated learning proposed in the past are reviewed. Section 3 explains some of the preliminary knowledge, including the knowledge of Free-rider attack and the DAGMM model. Section 4 details the Delta-DAGMM detection method proposed by us. Section is the complexity and convergence analysis of the model. Section 6 is the experimental results and discussion. Section 7 concludes this paper.

2. Related Work

Many scholars have researched and analyzed the methods of attack and detection in federated learning, which is worthy for us and later scholars. This section will introduce the related work.

2.1. Attacks in federated learning. Since the framework of federated learning was proposed, the researches on the safety of federated learning have been very active. The known types of attacks in federated learning are as follows:

(i) Attackers maliciously modify the dataset to decrease the model performance, such as inverting one label of the model to another wrong label, etc. This type of attack is called poisoning attacks [7]. There is also distributed poisoning attack in federated learning [8]. For example, Xie et al. [9] proposed DBA (Distributed Backdoor Attack), which has a higher success rate, better convergence and flexibility compared with centralized backdoor attack [10], and it can avoid two robust FL detection methods.

(ii) In the process of participating in the federation training, attackers infer the model parameters of other participants based on the local updated model parameters and the received global model parameters, and then infers the dataset information of other participants. This type of attack is called inference attack or privacy attack [11]. For example, Wang et al. [12] proposed an attack method that combines a multi-task discriminator to identify the sample classification, customer name, identity and other information. Nasr et al. [13] designed a white box inference attack method against the shortcomings of the stochastic gradient descent algorithm.

(iii) The attacker pretends to train, but instead of using his own dataset to participate in training, he uploads disguised model parameters. This type of attack is the Free-rider attack we researched. Fraboni et al. [5] proposed theoretical and experimental analysis of the Free-rider attack, which provided a formal guarantee for the attacks to converge to the aggregation model of fair participants.

2.2. Attack detection in federated learning. The reputation method proposed by people can be used to detect these attacks. For example, Kang et al. [14] proposed a decentralized consortium blockchain approach for efficient reputation management of participants. Kang et al. [15] also proposed a reputation-based federal learning security scheme designed by using the multi-weight model, which can significantly improve the learning accuracy. In addition, some game theory methods have been proposed to prevent attacks while forcing fair contributions. For example, Hu et al. [16] proposed a collective extortion strategy under incomplete information multi-person FEL game, which can effectively help the server to effectively stimulate the full contribution of all devices without worrying about any economic loss.

In the field of high-dimensional and multi-dimensional abnormal data detection, traditional detection methods usually first extract features, and then input the reduced-dimensional features into other available models, such as GMM [17]. Yang et al. [18] proposed an unsupervised dimensionality reduction method combining deep learning and GMM. Zong et al. [6] proposed a deep auto-encoded Gaussian mixture model (DAGMM) for detecting high-dimensional abnormal data.

DAGMM shows the best precision on the public benchmark dataset, and has outstanding performance in the unsupervised anomaly detection of multi-dimensional or high-dimensional data. Lin et al. [4] conducted a preliminary study on the attack and detection of Free-rider and proposed several strategies of Free-rider attack and a detection method, but did not provide the theoretical basis for its attack types or a normative detection method of the paradigm. Fraboni et al. [5] theoretically analysed and standardized the form of Free-rider attacks, and mentioned that high-dimensional anomaly data detection models (such as DAGMM) can be used for attack detection, but they did not conduct depth research on attack detection or experiment.

On the basis of DAGMM, we propose a new detection algorithm called Delta-DAGMM.

3. Preliminaries

3.1. Free-rider attack method. The research of Fraboni et al. [19] showed that there are two types of Free-rider attacks in Horizontal Federated Learning. One is called Plain Free-rider attack, whose strategy is directly returning the global model parameters obtained in each round, or replace them with random numbers. The other is the disguised Free-rider attack, whose strategy is to add differential perturbation to the global model parameters obtained in each round.

3.2. Plain Free-rider attack. There are three main attack strategies of the Plain Free-rider atack:

(i) The attackers first get the length D_{fc} of the output layer matrix of the global model, after that they define a new high-dimensional matrix with a length of D_{fc}, and fill this new matrix with a fixed value R. Finally, they return the matrix $\theta_i(t)$ to the parameter server A as the local updated model.

(ii) The attackers first get the length D_{fc} of the output layer matrix of the global model, after that they define a new high-dimensional matrix with a length of D_{fc}, and generate random numbers in the range $[R_1, R_2]$ to fill this new matrix. Finally, they return the matrix $\theta_i(t)$ to the parameter server A as the local updated model.

(iii) The attackers directly return the global model parameters of the current round as the local updated model to the parameter server A, that is. $\theta_i(t) = \theta(t)$.

3.3. Disguised Free-rider attack. During the training, we assuming that a Free-rider has prior knowledge of the training process, who knows the approximate standard deviation of each round of the local updated model and global model of the fair clients in advance. The attacker processes the obtained global model parameters by adding differential time-varying perturbations. Which satisfies the convergence similar to the fair clients.

In the horizontal federated learning training without attackers, the images of the output layer gradient and the convergence function of the global model parameter $\theta(t)$ are curves that smoothly converge with the number of training rounds t. Therefore, the local updated model $\theta_i(t)$ of the disguised Free-rider can be assumed as the following time-varying noise perturbation process:

$$\theta_i(t) = f(\theta(t)) = \theta(t) + \rho_j \xi_j(t), \tag{1}$$

where, $\xi_j(t)$ is the noise process, and the whole noise is expressed as the σ-dependent unit variance Gaussian white noise modulated by the parameter ρ_j.

The Disguised Free-rider attack strategy is divided into the following two types:

(i) Linear time-varying disturbance.

Suppose that the perturbation model $\xi_j(t) = O(t^{-\gamma})$, the attenuation coefficient $\gamma > 0$, then the Free-rider attacker's local model are updated as:

$$\theta_i(t) = \theta(t) + m\sigma t^{-\gamma}, \tag{2}$$

where, the variable m is the coefficient of noise level σ.

(ii) Exponential time-varying disturbance.

Suppose that the perturbation model $\xi_j(t) = O(e^{-(t-1)\gamma})$, the attenuation coefficient $\gamma > 0$, then the Free-rider attacker's local model are updated as:

$$\theta_i(t) = \theta(t) + m\sigma e^{-(t-1)\gamma}. \tag{3}$$

Fraboni et al. [5] explained the rationality of this perturbation-based attack and proposed a method to optimize the attack effect in experiments.

3.4. DAGMM. Density estimation is one of the core methods in anomaly detection of high dimensional data. DAGMM is a Gaussian mixture model which combines dimensionality reduction and density estimation efficiently. It mainly consists of two parts: compression network and estimation network.

The process of DAGMM is as follows: The depth autoencoder is used to reduce the dimensionality of the input samples in the compression network, and then the dimensionality reduction samples are fed back to the subsequent estimation network. The estimation network obtains the low-dimensional sample data fed back by the compressed network, and then estimates their energy under the framework of the Gaussian Mixture Model (GMM). High energy represents the data may be anomaly data.

In the research of Free-rider attack, Fraboni et al. [5] proposed that DAGMM could be used as a means to detect Free-rider attackers.

DAGMM is an end-to-end training unsupervised high-dimensional anomaly data detection model. Combined with the joint optimization of compression network and evaluation network, it solves the problem of large reconstruction error of anomaly samples in DSEBM and other detection methods. After a large number of experiments, we found that DAGMM has a high precision in detecting Plain attack type (i) and type (ii). However, the detection precision is not high for the detection of Plain Free-rider attack type (iii) and Disguised Free-rider attack types. Such sample data bases on the real model parameters and the addition of time-varying perturbations conforming to the convergence rate are likely to be detected as the real samples obtained by training, and can be restored well through the estimation network in DAGMM, and the output energy may not high enough to be detected as anomaly data.

Therefore, we optimize this detection model and propose a new type of attack detection method called Delta Deep Autoencoding Gaussian Mixture Model (Delta-DAGMM).

4. Research Methodology Introduction

The purpose of this paper is to detect free rider attackers. Therefore, we propose Delta- DAGMM, a Free-rider attack detection method, which includes three steps. As shown in Figure 2., we first calculate the increment of the model parameters of each client relative to the global model in the current round of horizontal federated learning to obtain the input samples of the detection model. Then we extract the

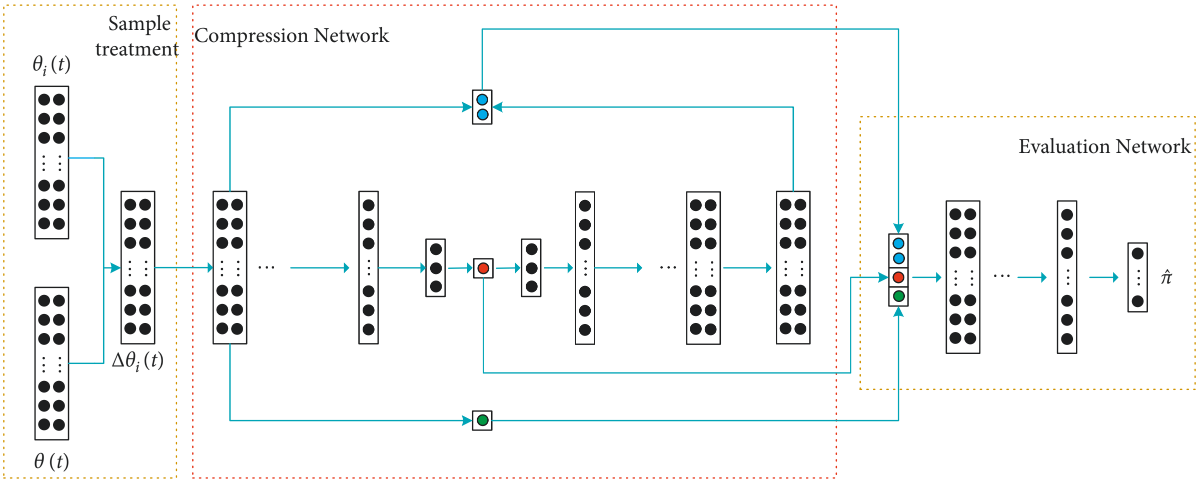

Figure 2: Delta-DAGMM model.

sample features in the compression network, and finally input the sample features into the estimation network to get the energy/likelihood. Finally, we use the energy as the basis of attack detection, and set a threshold to determine whether the participants are free rider attackers.

4.1. Sample treatment. Sample treatment is a very important part of Delta-DAGMM defense detection. We know that DAGMM can effectively detect the Plain Free-rider attack type (i) and type (ii), but it is not effective for the other attack strategies. We can simplify the remaining three attack strategies into Plain free-rider attack type (i) and type (ii) that can be easily detected by DAGMM through sample treatment (ST).

Specifically, sample treatment is divided into two steps: data collection and incremental processing.

4.1.1. Data collection. In the training of horizontal federated learning, we assume that there are m clients participating in multiple rounds of iterative training, denoted by $C_1, C_2, \ldots, C_m$ respectively. In the process of iterative training, we used $\theta(t)$ to denote the global model transmitted by the parameter server A to all participant clients in round t. The participant clients' local updated model denoted by $\theta_1(t), \theta_2(t), \ldots, \theta_m(t)$ respectively. After the transmission of all participant client local model updates in round t is completed, parameter server A receives local updated models of all participant clients. The global model of round t+1 is generated by federated averaging algorithm (FVG), the method can be expressed as:

$$\theta_m(t+1) = \sum_{i=1}^{m} \frac{\theta_i(t)}{m}. \tag{4}$$

The parameter server A sends the obtained global model $\theta_m(t+1)$ to all participant clients as the beginning of the training of round t +1, this iteration continues until the end of the training.

It is assumed that there are n rounds of training, and in each round of training, we put the local updated modal into a set, received n totally : $\theta_1(1), \theta_2(1), \ldots, \theta_m(1), \{\theta_1(2), \theta_2(2), \ldots, \theta_m(2)\}, \ldots, \{\theta_1(t), \theta_2(t), \ldots, \theta_m(t)\}$ and a global model set $\theta(1), \theta(2), \ldots, \theta(n)$. Before the end of each round of horizontal federated training, we collected the global model parameter $\theta(t)$ and the local updated model set $\{\theta_1(t), \theta_2(t), \ldots, \theta_m(t)\}$ of the clients as the input samples of the Free-rider attack detection in round t.

When horizontal federated learning uses different training models, the dimensions of the output layer parameters of the training model are also different. According to the different training models in horizontal federation learning, we divide the input samples of the delta-DAGMM detection model into the following two categories:

(i) MLP-Federate. In horizontal federated training, we use MLP as the training model, and the set of parameters of the local update model obtained locally by participants trained or disguised is MLP-Federate. The parameters of each participant's local update model are the weight matrix in output layer of MLP and the tensor array with length of 64∗10.

(ii) CNN-Federate. In horizontal federated training, we use CNN as the training model, and the set of parameters of the local update model obtained locally by participants trained or disguised is CNN- Federate. The parameters of each participant's local update model are the weight matrix in output layer of CNN and the tensor array with length of 50∗10.

4.1.2. Incremental treatment. In the disguised Free-rider attack, the model parameters disguised by the attacker are based on the global model parameters and the current training round t is used as the parameter to add differential perturbation. In order to make the model parameters show the convergence similar to that of the fair clients on the whole, the effect of the differential perturbation set by the attacker is decreasing by round, but there will be some discreteness. Parameter server A calculates the increment of the local model parameters of the Free-rider attacker compared with the global model parameters in the current round t, and the difference value is actually equal to the value of the differential perturbation added in the attack. Since the input sample of the attacker is based on the random process and has certain fluctuation, it is very likely to be detected as abnormal data.

In order to avoid the evaluation error caused by the too small absolute value of the element in the input sample, we linearly process the increment of the model to obtain the final input sample x:

$$x = k\left(\theta_i(t) - \theta(t)\right) + \theta_b, \tag{5}$$

where k denotes the preset constant and θ_b denotes the global model filled with the preset constant b. For the Plain Free-rider attack type (iii), since the local model parameter of the attacker is $\theta_i(t) = \theta(t)$ and the model parameter after incremental processing is $x = \theta_b$, it is equivalent to converting this attack strategy to fill the global model with fixed values, that is, the Plain Free-rider attack type (i). When the attacker uses the disguised Free-rider attack strategies, for the attack with linear time-varying disturbance and for the attack with exponential time-varying disturbance, the model parameters after incremental processing actually use the time-varying disturbance values of $\theta_i^f(t) = \theta(t) + m\sigma t^{-\gamma}$ and $\theta_i^f(t) = \theta(t) + m\sigma e^{-(t-1)\gamma}$ to fill the samples of the global model. This strategy is close to the Plain Free- rider attack type (ii).

The final input samples can be divided into the following two categories according to the different models selected for horizontal federated training:

(i) Delta-MLP-Federate. In the horizontal federation training experiment, the participants and parameter server A select the MLP model, and we obtain the sample through incremental processing on the local update model parameter set of each round of training. The length of each input sample array is 64*10.

(ii) Delta-CNN-Federate. In the horizontal federation training experiment, the participants and parameter server A select the CNN model, and we obtain the sample through incremental processing on the local update model parameter set of each round of training. The length of each input sample array is 50*10.

4.2. Compression network. In the processing of the Delta-DAGMM, when the high-dimensional sample x is generated, the compression network uses a deep autoencoder to reduce the dimension of the input sample and extract three parts of features. Finally, we merge the features to obtain the compressed sample. The Delta-DAGMM we proposed is different from DAGMM in the compression network. Delta-DAGMM adds feature extraction of the mean value of all elements of the input sample, making it easier for abnormal data to eventually output high energy values and be detected.

4.2.1. Feature extraction. The autoencoder neural network used by the compression network is an unsupervised learning model. It uses a backpropagation algorithm to make the target value equal to the input value as much as possible. It is generally used for feature extraction of high-dimensional data.

Feature extraction in compressed networks has three sources:

(i) Simplified representation Z_c of sample x learned by deep autoencoder.

(ii) Feature Z_r extracted from reconstruction error.

(iii) The mean of all the elements in the input sample, Z_{avg}.

Given the input sample, the three features extracted by the compression network are as follows:

$$\begin{aligned} Z_c &= h\left(x; \xi_e\right), \\ x' &= g\left(Z_c; \xi_d\right), \\ Z_r &= f\left(x, x'\right), \\ Z_{\text{avg}} &= \sum_{j=1}^{N} \frac{x[j]}{N}, \end{aligned} \tag{6}$$

where ξ_e and ξ_d denote the parameters of the depth autoencoder, x' denotes the reconstruction counterpart of sample x, $h(\cdot)$ denotes the encoding function, $g(\cdot)$ denotes the decoding function, and $f(\cdot)$ denotes the function to calculate the reconstruction error characteristics.

4.2.2. Feature merging. We merge the three features of Z_c, Z_r and Z_{avg} as the output of the compression network and input them into the estimation network. The low-dimensional representation Z finally provided by the compression network is as follows:

$$Z = \left[Z_c, Z_r, Z_{\text{avg}}\right]. \tag{7}$$

4.3. Estimation network. The estimation network estimated the density of the low-dimensional representation Z under the framework of GMM, which was achieved by using a multi-layer neural network (MLN) to predict the mixed membership for each sample. Membership testing is as follows:

$$\begin{aligned} \mathbf{P} &= \text{MLN}\left(Z; \xi_m\right), \\ \hat{\gamma} &= \text{softmax}(\mathbf{P}), \end{aligned} \tag{8}$$

where Z denotes the compressed sample, integer K is the number of mixed components in GMM, $\hat{\gamma}$ is the K-Dimension vector used for soft mixed components membership prediction, and $\mathbf{P}$ is the output of multi-layer network parameterized by ξ_m.

The current number of samples is N. For any $k(1 \leq k \leq K)$, we can further estimate the important parameters in GMM as follows: the mixing probability $\widehat{\phi}_k$, mean $\widehat{\mu}_k$, and covariance $\widehat{\Sigma}_k$ of GMM component k. This step is the same as the parameter updating process of the conventional Gaussian mixture model [6]:

$$\begin{aligned}\widehat{\phi}_k &= \sum_{i=1}^{N} \frac{\widehat{\gamma}_{ik}}{N},\\ \widehat{\mu}_k &= \frac{\sum_{i=1}^{N} \widehat{\gamma}_{ik} Z_i}{\sum_{i=1}^{N} \widehat{\gamma}_{ik}},\\ \widehat{\Sigma}_k &= \frac{\sum_{i=1}^{N} \widehat{\gamma}_{ik} \left(Z_i - \widehat{\mu}_k\right)\left(Z_i - \widehat{\mu}_k\right)^{\mathrm{T}}}{\sum_{i=1}^{N} \widehat{\gamma}_{ik}}.\end{aligned} \tag{9}$$

Calculate the sample energy through the above parameters, denoted by $E(Z)$:

$$E(Z) = -\log\left(\sum_{k=1}^{K} \widehat{\phi}_k \frac{\exp\left(-(1/2)\left(Z - \widehat{\mu}_k\right)^{T} \widehat{\Sigma}_k^{-1} \left(z - \widehat{\mu}_k\right)\right)}{\sqrt{\left|2\pi\widehat{\Sigma}_k\right|}}\right), \tag{10}$$

where $|\cdot|$ denotes the determinant of a matrix. In the t round training, the input samples of m participants were estimated by the estimated network and the sample energies were $E_i(t)$ respectively, where $i = 1, 2, \ldots, m$, Calculate the average $E_{\text{avg}}(t) = \sum_{i}^{i=m} (t)/m$ of the energy of these samples. We set the threshold to $E_i(t) > E_{\text{avg}}(t) * W$, and choose a better value for W according to the experimental results. We predict the high-energy samples that meet the conditions as the Free-rider attacker in the training round t. After the Federal training is over, each participant is ultimately determined to be a Free-Rider if he has been detected as a Free-rider for more than F times, where F denotes the smallest integer not less than 2/3 of the number of the training rounds.

If the energy of the automatically encoded sample feature extraction calculated through the estimation network is low, indicating that the reconstruction error of the estimation network is low, the original sample can be considered as normal high-dimensional data that is easy to restore. However, for the samples obtained from the model parameters provided by the Free-rider attacker, the deviation from the original data is large after the reconstruction of the compression network, and the calculated energy is high. The algorithm Delta-DAGMM is illustrated in Algorithm 1.

5. Method analysis

5.1. Complexity analysis. We analyze the time complexity of sample processing, compression network and evaluation network in Delta-DAGMM, and compare it with DAGMM. Here we ignore the communication cost because they can be enhanced from the federal learning and training framework, and for this detection model Delta-DAGMM, there is no additional communication cost.

For the horizontal federated training in this paper, we set the size of the transmission model as a two-dimensional tensor of J ∗ K and the number of participants as M, so the time complexity of sample treatment(ST) is O(M∗J∗K). In the compressed network(CN), the time complexity of the simplified representation(SR) of the calculation sample is O(M∗J∗K), the time complexity of the feature extraction(FE) from the reconstruction error is O(M∗J), and the time complexity of the mean of all elements(ME) in each sample is O(M∗J∗K). Assuming that the number of GMM mixed components in the evaluation network(EN) is G, the time complexity of the evaluation network calculation(ENC) is O(M∗JK). Table 1 describes the time complexity of Delta-DAGMM and DAGMM.

According to Table 1, the total time complexity of DAGMM and Delta-DAGMM is basically the same, and the maximum time cost is concentrated in the evaluation of network computing energy. In fact, the maximum time spent in federal training is spent on communication between servers and participants.

5.2. Convergence analysis. We need to explain the convergence of the federal learning model containing the Free-rider attack to prove the effectiveness and concealment of this attack.

Taking plain attack strategy III as an example, the differences between global models with and without plain Free-rider attackers are calculated, as shown in expressions (11)-(15).

$$w_t' - w_t = \sum_{i=0}^{k-1} \left(\varepsilon + \frac{n_f}{n}\right)^{k-i-1} f\left(w_i\right) + \sum_{i=0}^{k-1} \left(\varepsilon + \frac{n_f}{n}\right)^{k-i-1} \left(v_i' - v_i\right), \tag{11}$$

$$f\left(w_i\right) = \frac{n_F}{n}\left[w_i - \sum_{j\in J} \frac{n_j}{n - n_F}\left[\eta_j\left(w_i - w_j^*\right) + w_j^*\right]\right], \tag{12}$$

$$\varepsilon = \sum_{j\in J} \frac{n_j}{n} \eta_j, \tag{13}$$

$$v_i = \sum_{j\in J} \frac{n_j}{n - n_F} \rho_j \zeta_{j,i} \tag{14}$$

$$v_i' = \sum_{j\in J} \frac{n_j}{n} \rho_j \zeta_{j,i}'. \tag{15}$$

Among them, w_j^* is the minimum value of local model parameters. η_j is related to the initial training set of hyperparameters, including the number of training rounds E, the learning rate γ and the number of samples S for each small batch. $\zeta_{j,i}$ is delta-related Gaussian white noise, while ρ_j is a time-varying noise. v_i and v_i' represent two different stochastic processes related to the federal global model.

In the absence of Free- rider attackers, the second item of $w_t' - w_t$' s value dependency (11) is the difference between two different stochastic processes associated with federal training global models. In the case of Free-rider attackers, the convergence of the federal training global model depends

```
Input: n // number of training rounds, m // number of participants, $\theta_i(t)$ // local updated model parameter
Output: atkList // list of Free-rider attack detection results
Initialize the global model $\theta(1)$ and the list of attack detection results atkList
for t = 1 to n do // t denotes the current training round
    Parameter server $A$ sends the global model $\theta(t)$ to all participants
    // Get the energies of samples calculated by Delta-DAGMM
    for i = 1 to m do
        // Gets the participant local update model increment $\theta_i(t)$
        $\Delta\theta_i(t) \longleftarrow \theta_i(t) - \theta(t)$
        // Process the model increment
        $f(\Delta\theta_i(t)) \longleftarrow k\Delta\theta_i(t) + b$
        // Input the processed model increment as a sample
        $E_i(t) \longleftarrow \mathrm{Delta-DAGMM}(f(\Delta\theta_i(t)))$
end for
// Free-rider attack detection
$E_{\mathrm{avg}}(t) \longleftarrow \sum_i^{i=m} E_i(t)/m$
for i = 1 to m do
    // Set threshold
    if $E_i(t) > E_{\mathrm{avg}}(t) * 1.08$ then
        atkList.add$(i)$
end for
// Update the global model
$\theta(t+1) \longleftarrow \mathrm{FedAvg}(\sum_{i=1}^{m} \theta_i(t))$
end for
return atkList
end for
```

Algorithm 1: Delta-DAGMM algorithm.

on the ratio of the sum of Free-rider attacker samples to the sum of all participants' samples.

6. Experiments and Discussion

In order to verify the effectiveness of Delta-DAGMM for the detection of Free-rider attacks in horizontal federated learning, we designed and implemented experiments and compared it with the existing attack detection method DAGMM.

The experiment simulates the parameter server and all participant nodes in horizontal federated learning on a computer device. The hardware used in the experiment is AMD R7-4800H 2.9GHz, the memory is 16GB, the graphics card used for local training is NVIDIA GeForce RTX 2060 6GB, and the operating system is Windows 10.

We set up 10 participants, including 1 Free-rider attacker, and conduct attack detection experiments on two different types of input samples MLP-Federate and CNN-Federate for five attack strategies. We repeated the experiment 50 times for each strategy to eliminate chance.

6.1. Experimental Datasets. We use two different training models, CNN and MLP, in the horizontal federation learning training process. In each round of training, we take the participants ' local model parameter set $\{\theta_1(t), \theta_2(t), \ldots, \theta_m(t)\}$ as the input sample.

Due to the different training models used in the training process, we get two kinds of high-dimensional samples of different lengths, MLP- Federate and CNN- Federate so as to better judge the precision of the detection algorithm. Table 2 summarizes the specific information of the Free-Rider attack detection datasets. The total number of the two experimental samples is 50,000.

6.2. Experimental Metrics. In this experiment, we adopt precision and F1 score as the metrics. Precision (16) denotes the proportion of samples detected as attackers and actually being attackers to all samples detected as attackers, which reflects whether the detection algorithm can accurately find positive samples from all samples to avoid false positives. Recall (17) denotes the proportion of the samples detected as and actually being attackers to the samples of actual attackers. F1 score (18) is a metric used to measure the accuracy of the dichotomous model. It also takes into account the precision and recall of the classification model, so it can be regarded as the harmonic average of model precision and recall. In (16), (17) and (18), TP denotes the number of the samples that are predicted to be attackers and are actually attackers. FP denotes the number of the positive samples that are predicted to be attackers but are not actually attackers. FN denotes the number of the samples of attacker that are actually but have not been detected. P denotes the precision of the detection model, and R represents the recall of the detection denotes. F1 denotes F1 score:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{16}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{17}$$

Table 1: Time complexity of Delta-DAGMM and DAGMM.

	ST		CN		EN
	IT	SR	FE	ME	ENC
DAGMM	—	O(M∗J∗K)	O(M∗J)	—	O(M∗JP)
Delta-DAGMM	O(M∗J∗K)	O(M∗J∗K)	O(M∗J)	O(M∗J∗K)	O(M∗JP)

Table 2: Free-rider Attack detection datasets.

Dataset	Sample length	Attack Strategies	Iterations	Training times	Number of samples
MLP-Federate	10∗64	5	20	50	50000
CNN-Federate	10∗50	5	20	50	50000

Table 3: Precision of Delta-DAGMM in different strategies of Free-rider attack detection.

Dataset	Type	Precision (%)				
		Plain (i)	Plain (ii)	Plain (iii)	Disguised (i)	Disguised (ii)
MLP-Federate	Single time	98.2	95.7	100	83.9	83.3
	Overall	100	99.8	100	96.6	96.0
CNN-Federate	Single time	98.4	95.2	100	82.9	83.2
	Overall	100	99.6	100	95.8	96.4

Table 4: F1 score of Delta-DAGMM in different strategies of Free-rider attack detection.

Dataset	Type	F1 score (%)				
		Plain (i)	Plain (ii)	Plain (iii)	Disguised (i)	Disguised (ii)
MLP-Federate	Single time	98.5	95.4	100	87.2	86.5
	Overall	100	99.8	100	96.4	95.7
CNN-Federate	Single time	98.7	95.2	100	85.7	86.6
	Overall	100	99.7	100	95.6	96.3

Table 5: Precision of different Free-rider attack detection methods.

Method	Type	Precision (%)				
		Simple attack(i)	Simple attack(ii)	Simple attack(iii)	Optimized attack(i)	Optimized attack(ii)
Delta-DAGMM	Single time	98.3	95.5	100	83.4	83.3
	Overall	100	99.7	100	96.2	96.2
DAGMM	Single time	95.7	94.6	62.4	53.6	52.7
	Overall	99.6	99.3	77.2	69.1	67.7
DAGMM(ST)	Single time	97.2	95.4	100	64.3	65.1
	Overall	99.8	99.6	100	80.1	81.3

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}. \tag{18}$$

For each experiment, we will count the precision and F1 score of the Free-rider attack detection of each round of horizontal federated learning training samples, which are called single time precision and F1 score. After the end of all rounds of the experiment, we set a threshold to obtain a final attack detection result based on the number of times each participant was inferred as a Free-rider in all rounds, and the overall detection accuracy and F1 score were counted.

6.3. Experimental Result. This article conducts experiments on the above five different attack strategies. We select two different data sets of MLP-Federate and CNN-Federate for experiments, and finally calculate the single time precision and F1 score, and the overall precision and F1 score under the five different attack strategies. According to Table 3, the single time attack detection precisions of the five attack strategies of the Delta-DAGMM algorithm proposed in this paper have exceeded 83%, and the overall precisions have exceeded 95%. The single time precisions of Plain Free-rider attack type (i), type (ii) and type (iii) are above 90%, and the

Table 6: F1 score of different Free-rider attack detection methods.

Method	Type	F1 score (%)				
		Simple attack(i)	Simple attack(ii)	Simple attack(iii)	Optimized attack(i)	Optimized attack(ii)
Delta-DAGMM	Single time	98.6	95.3	100	86.4	86.6
	Overall	100	99.8	100	96	96
DAGMM	Single time	96.1	93.8	63.6	54	52.5
	Overall	99.7	99.2	78.1	69.3	68
DAGMM(ST)	Single time	97.4	95.1	100	65.2	66.6
	Overall	99.8	99.6	100	80.9	81.8

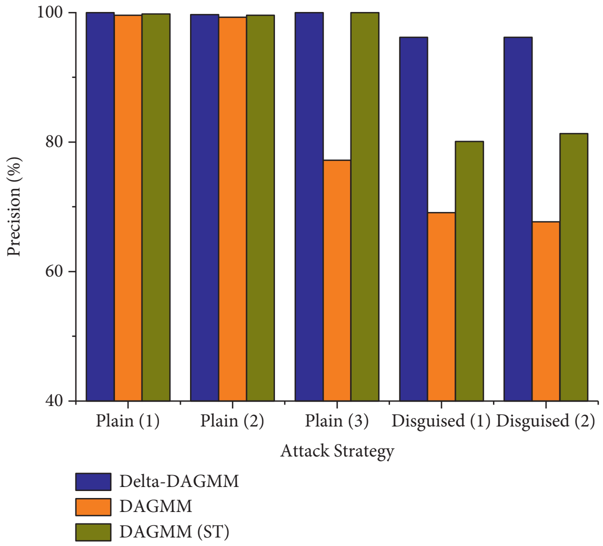

Figure 3: Precision of single time detection for five attack strategies.

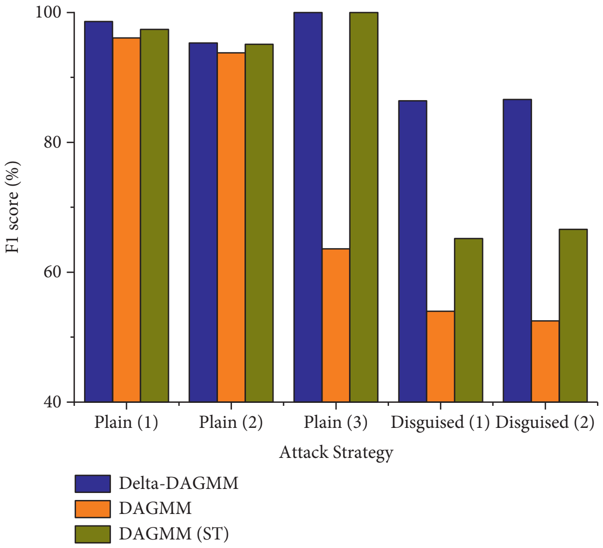

Figure 5: F1 score of single time detection for five attack strategies.

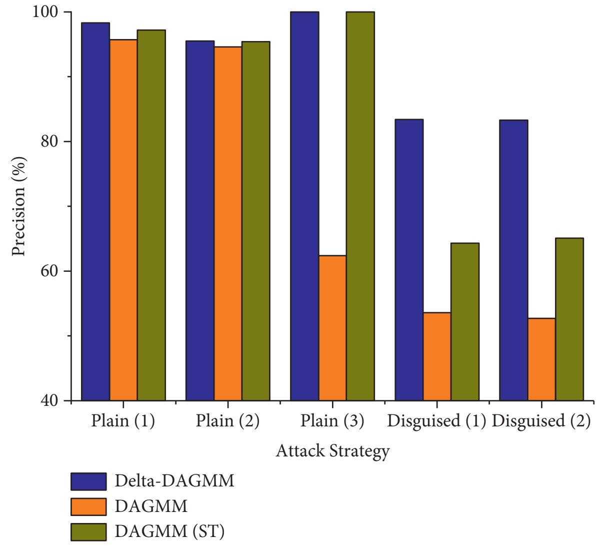

Figure 4: Precision of overall detection for five attack strategies.

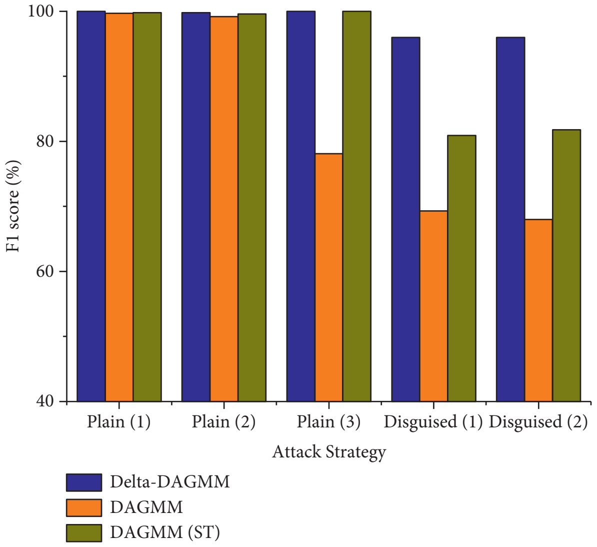

Figure 6: F1 score of overall detection for five attack strategies.

overall precision are above 97%. The single time and the overall precision of detection for Plain Free-rider attack type (iii) are both 100%. According to Table 4, the single time attack detection F1 score of the five attack strategies of the Delta-DAGMM algorithm proposed in this paper have exceeded 85%, and the overall F1 score have exceeded 96%.

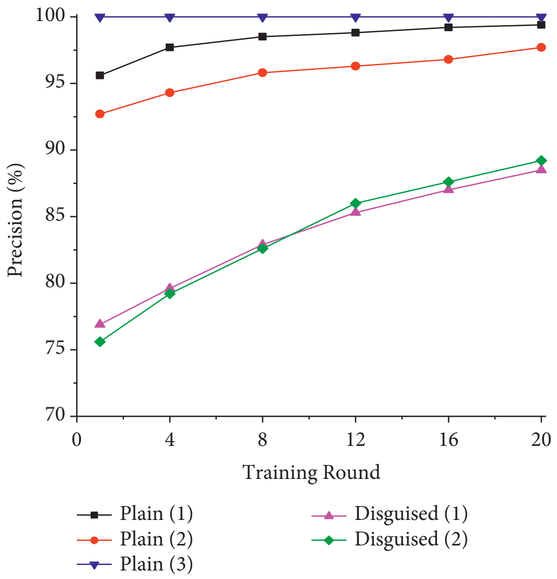

Figure 7: Precision of overall detection for five attack strategies by training round.

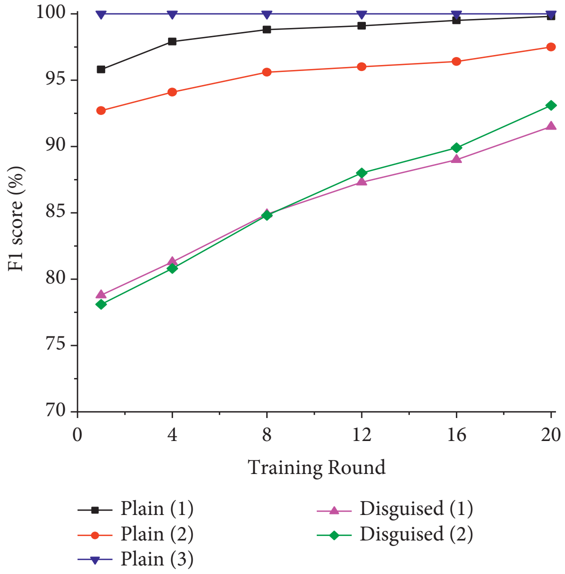

Figure 8: F1 score of overall detection for five attack strategies by training round.

The single time F1 score of Plain Free-rider attack type (i), type (ii) and type (iii) are above 95%, and the overall F1 score are above 95%. The single time and the overall F1 score of detection for Plain Free-rider attack type (iii) are both 100%.

6.4. Experimental discussion. Table 5 and Table 6 respectively show the precisions and F1 score comparison between Delta-DAGMM and other Free-rider attack detection methods. DAGMM(ST) denotes DAGMM with sample treatment.

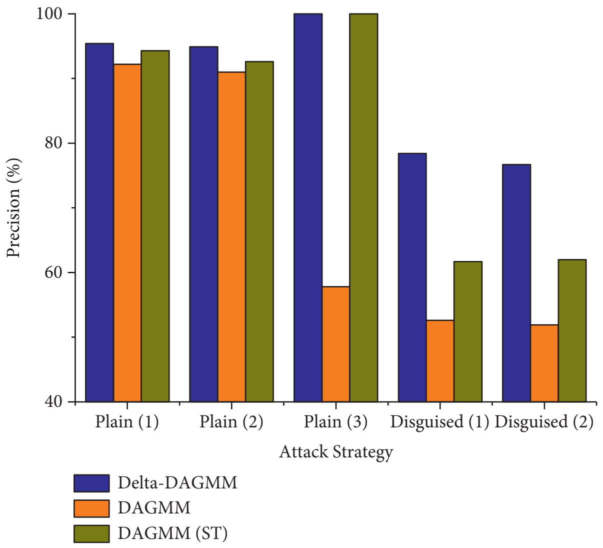

Figure 9: Precision of detection with 2 attackers under 5 attack strategies.

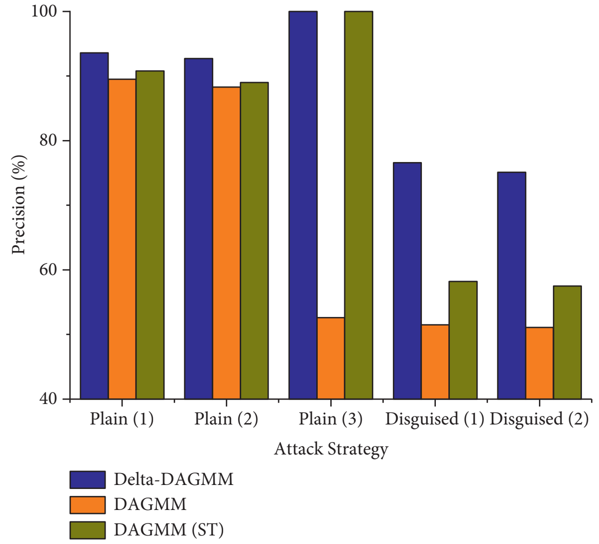

Figure 10: Precision of detection with 3 attackers under 5 attack strategies.

As shown in Figures 3 and 4, under the five attack strategies, the single time and overall precisions of Delta-DAGMM are slightly improved compared with that of DAGMM and DAGMM with sample treatment for detection of the Plain Free-Rider attack type (i) and type (ii), where single time precisions increase by 1.6% and 0.2% respectively, and overall precisions increase by 0.4% and 0.1% respectively. For the detection of the Plain Free-rider attack type (iii) and the disguised Free-rider attack strategy (i) and (ii), the precisions of Delta-DAGMM are significantly high than that of DAGMM and DAGMM with sample treatment. The single time precisions increase by 20.3% and 7.7% respectively, and the overall precisions increase by 15.8% and 6.2% respectively.

Table 7: Precision of different detection methods in Free-rider attack with 1000 participants and 10 attackers.

Detection Method	Detection type	Precision (%)				
		Simple attack (i)	Simple attack (ii)	Simple attack (iii)	Optimized attack (i)	Optimized attack (ii)
Delta-DAGMM	Single	98.9	97.1	100	87.8	86.3
	overall	100	99.9	100	97.3	97
DAGMM	Single	95.9	95.2	62.4	57.2	55.7
	overall	99.6	99.3	77.2	72.1	69.9
DAGMM(ST)	Single	97.1	96.1	100	70.3	68.5
	overall	99.8	99.7	100	82.6	81.6

As shown in Figures 5 and 6, under the five attack strategies, the single time and overall F1 score of Delta-DAGMM are slightly improved compared with that of DAGMM and DAGMM with sample treatment for detection of the Plain Free-rider attack type (i) and type (ii), where single time F1 score increase by 1.5% and 0.3% respectively, and overall F1 score increase by 0.6% and 0.2% respectively. For the detection of the Plain Free-rider attack type (iii) and the disguised Free-rider attack strategies (i) and (ii), the F1 score of Delta-DAGMM are significantly high than that of DAGMM and DAGMM with sample treatment. The single time F1 score increase by 21.2% and 8.5% respectively, and the overall F1 score increase by 15.9% and 6.4% respectively.

Since our previous statistics are to combine the detection results of all training rounds, it is impossible to show the trend of detection precisions with the change of training rounds. Therefore, we also record the change of Delta-DAGMM with training rounds in the horizontal federation for the precisions of the detection of five free-rider attack strategies. As shown in Figures 7 and 8, as the number of training rounds increases, the detection precisions and F1 score of Delta-DAGMM gradually increase.

Since we set 1 Free-rider attacker among 10 participants in all previous experiment, we try to set more attackers among the participants. As shown in Figures 9 and 10, when 2-3 Free-rider attackers are set, the precision of the all three attack detection algorithms decreases, but Delta-DAGMM still maintain a precision of more than 75%, which is better than DAGMM and DAGMM with sample treatment.

For larger trials, we used existing distributed computing techniques to simulate the involvement of larger users in training, setting 1000 participants and 10 attackers, as shown in table 7, the Delta-DAGMM detection accuracy remains high.

6.5. Experimental conclusion. Due to the Delta-DAGMM proposed in this paper adds sample processing compared with DAGMM, in fact, the camouflage Free- rider attack is transformed into the simple Free- rider attack. And Delta-DAGMM adds a feature representation to the compressed network, making it easier to reconstruct low-dimensional samples of fair participants and find Free- rider attackers among participants. We conducted experiments under different conditions, and found that the accuracy rate and F1 score of Delta-DAGMM were significantly higher than those of DAGMM no matter for single detection or overall detection. For large-scale simulation experiments of participants, the accuracy rate of Delta-DAGMM was also higher.

7. Conclusions

In horizontal federated learning, there is a Free-rider who does not use the local data set to participate in training, but disguises the parameters the local updated model to participate in training and steal the global model. In order to detect Free-rider attackers, we propose an improved attack detection algorithm based on the DAGMM model, Delta-DAGMM. Compared with DAGMM, this algorithm is optimized in sample treatment and feature extraction. An incremental processing method is used to optimize the sample, and the more critical features in the sample can be extracted. We also set an appropriate threshold to finally detect the attacker.

The experimental results show that compared with DAGMM, Delta-DAGMM can achieve higher precision and F1 score. The average precisions of a single time detection are 92.1%, 20.3% higher than DAGMM, and average precisions of the overall detection are 98.4%, 15.8% higher than DAGMM. The average F1 score in single time detections are 93.4%, 21.4% higher than DAGMM, and the average F1 score in the overall detection is 98.4%, 16.5% higher than DAGMM. The experimental results confirm that Delta-DAGMM is a more effective Free-rider attack detection algorithm than DAGMM.

However, in our experiments, the model parameters that the parameter server and participants of horizontal federated learning transmit to each other are plain text. The challenge of Delta-DAGMM is that in future federated training will use methods such as homomorphic encryption [19–23] or differential privacy [24] to encrypt model parameters transmitted by users. The model parameters sent and transmitted by the client will no longer be plaintext. Next, we will consider how to detect Free-rider attacks under ciphertext.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Thank Mr. Yu Haining for his guidance of this paper and the National Natural Science Foundation of China (62172123) for its support.

References

[1] Q. Yang, Y. Liu, Y. Cheng, and Y. T. Kang, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[3] B. McMahan, E. Moore, D. Ramage, S. Heth, and B. A. Y. Arcas, "Communication-efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the Artificial Intelligence and Statistics, PMLR*, pp. 1273–1282, FL, USA, April 2017.

[4] J. Lin, M. Du, and J. Liu, "Free Riders in federated learning: attacks and defenses," 2019, https://arxiv.org/abs/1911.12560.

[5] Y. Fraboni, R. Vidal, and M. Lorenzi, "Free Rider Attacks on Model Aggregation in Federated Learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR*, pp. 1846–1854, San Diego, CA, USA, April 2021.

[6] B. Zong, Q. Song, M. Min et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proceedings of the International Conference on Learning Representations*, 2018.

[7] H. Chacon, S. Silva, and P. Rad, "Deep learning poison data attack detection," in *Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 971–978, Portland, OR, USA, November 2019.

[8] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *Proceedings of the 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 374–380, Rotorua, New Zealand, August 2019.

[9] C. Xie, K. Huang, P. Chen, and B. Li, "Dba: distributed backdoor attacks against federated learning," in *Proceedings of the International Conference on Learning Representations*, New Orleans, LA, USA, November 2019.

[10] K. Doan, Y. Lao, W. Zhao, and P. Li, "LIRA: learnable, imperceptible and robust backdoor attacks," *Proceedings of the IEEE/CVF International Conference on Computer Vision*.pp. 11966–11976, Montreal, QC, Canada, October 2021.

[11] M. A. Rahman, T. Rahman, R. Laganière, M. Neimat, and Y. Wang, "Membership inference attack against differentially private deep learning model," *Transactions on Data Privacy*, vol. 11, no. 1, pp. 61–79, 2018.

[12] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: user-level privacy leakage from federated learning," in *Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520, Paris, France, April 2019.

[13] 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 2019 IEEE Symposium on Security and Privacy*, pp. 739–753, CA, USA, May 2019.

[14] J. Kang, Z. Xiong, D. Niyato, and Y. Y. M. Zou, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[15] J. Kang, Z. Xiong, D. Niyato, and S. J. Xie, "Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, Article ID 10714, 2019.

[16] Q. Hu, S. Wang, Z. Xiong, and X. Cheng, "Nothing wasted: full contribution enforcement in federated edge learning," *IEEE Transactions on Mobile Computing*, 2021.

[17] D. Reynolds, "Gaussian mixture models," *Encyclopedia of biometrics*, vol. 741, pp. 659–663, 2009.

[18] X. Yang, K. Huang, J. Y. Goulermas, and R. Zhang, "Joint learning of unsupervised dimensionality reduction and Gaussian mixture model," *Neural Processing Letters*, vol. 45, no. 3, pp. 791–806, 2017.

[19] H. Yu, X. Yu, X. Jia, H. Zhang, and J. Shu, "PSRide: privacy-preserving shared ride matching for online ride hailing systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, 2019.

[20] H. Yu, X. Jia, H. Zhang, and J. Shu, "Efficient and privacy-preserving ride matching using exact road distance in online ride hailing services," *IEEE Transactions on Services Computing*, 2020.

[21] H. Yu, Z. Hongli, J. Xiaohua, C. Xiao, and Y. Xiangzhan, "pSafety: privacy-preserving safety monitoring in online ride hailing services," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[22] H. Yu and H. X. X. M. Zhang, "PGRide: privacy-preserving group ridesharing matching in online ride hailing services," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5722–5735, 2021.

[23] H. Yu and J. X. H. X. Shu, "lpRide: lightweight and privacy-preserving ride matching over road networks in online ride hailing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, Article ID 10428, 2019.

[24] R. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: a client level perspective," 2017, https://arxiv.org/abs/1712.07557.