Computer Systems Science & Engineering
DOI:10.32604/csse.2021.017969
Article

Tech Science Press

FREPD: A Robust Federated Learning Framework on Variational Autoencoder

Zhipin Gu[1], Liangzhong He[2], Peiyan Li[1], Peng Sun[3], Jiangyong Shi[1] and Yuexiang Yang[1,*]

[1]National University of Defense Technology, Changsha, 410000, China
[2]China Mobile (Suzhou) Software Technology Co. Ltd., Suzhou, 215000, China
[3]Eindhoven University of Technology, Eindhoven, 5641BZ, Netherlands
*Corresponding Author: Yuexiang Yang. Email: yyx@nudt.edu.cn
Received: 19 February 2021; Accepted: 09 April 2021

Abstract: Federated learning is an ideal solution to the limitation of not preserving the users' privacy information in edge computing. In federated learning, the cloud aggregates local model updates from the devices to generate a global model. To protect devices' privacy, the cloud is designed to have no visibility into how these updates are generated, making detecting and defending malicious model updates a challenging task. Unlike existing works that struggle to tolerate adversarial attacks, the paper manages to exclude malicious updates from the global model's aggregation. This paper focuses on Byzantine attack and backdoor attack in the federated learning setting. We propose a federated learning framework, which we call Federated Reconstruction Error Probability Distribution (FREPD). FREPD uses a VAE model to compute updates' reconstruction errors. Updates with higher reconstruction errors than the average reconstruction error are deemed as malicious updates and removed. Meanwhile, we apply the Kolmogorov-Smirnov test to choose a proper probability distribution function and tune its parameters to fit the distribution of reconstruction errors from observed benign updates. We then use the distribution function to estimate the probability that an unseen reconstruction error belongs to the benign reconstruction error distribution. Based on the probability, we classify the model updates as benign or malicious. Only benign updates are used to aggregate the global model. FREPD is tested with extensive experiments on independent and identically distributed (IID) and non-IID federated benchmarks, showing a competitive performance over existing aggregation methods under Byzantine attack and backdoor attack.

Keywords: Federated learning; reconstruction error; probability distribution

1 Introduction

Recently, numerous IoT applications, such as autonomous driving, smart healthcare and Industry 4.0 require low latency edge computing [1–4]. Traditional machine learning [5–7] that stores user data at a data center raised a wide range of privacy concerns. Federated learning is a solution to the limitation of not preserving the users' privacy information. Federated learning is an attractive framework where multiple devices collaboratively train a machine learning model without revealing their private data [8].

In every round, the cloud distributes the global model to a random subset of devices. Each of them trains locally and transfers the local model update to the cloud. The cloud then combines these local models update to generate an aggregated model. To protect devices' privacy, devices' local data and training process is designed to be invisible to the cloud.

The distributed nature of federated learning, mainly when using secure aggregation methods, makes detecting and mitigating the adversarial attacks a particularly challenging task [9]. Meanwhile, federated learning gives devices a direct influence on the aggregated model, enabling powerful attacks such as backdoor attack. Therefore, federated learning is vulnerable to adversarial attacks, even when only one device is malicious [10]. Adversarial attacks can be broadly categorized into two types based on the attack's goal: untargeted and targeted attack [9]. This paper focuses on Byzantine attack (untargeted attack) and backdoor attack (targeted attack). Under Byzantine attack, malicious devices send arbitrary model updates to the cloud to induce model performance deterioration or the failure of model training [10,11]. Under backdoor attack, the adversary's goal is to induce the aggregated model to misclassify a set of chosen inputs [12]. For example, in image classification, backdoor attack may aim to enforce the model classify images with the label 'cat' as the label 'dog' while ensuring other images are correctly classified.

Byzantine-robust distributed machine learning has gained significant interest in recent years. Most of the existing aggregation methods [13–15] design Byzantine tolerant aggregation methods to defend against Byzantine attack and assume that the data are independent and identically distributed (IID) on the devices. Some of the methods are proved to have good performance in federated learning. However, there are two challenges to be solved. On the one hand, these methods are not robust to the heterogeneous datasets generated from heterogeneous computing units. On the other hand, these methods are mainly designed for Byzantine attack and cannot defend backdoor attack.

There are also defense methods against backdoor attack. Li et al. [16] propose a robust federated learning framework, which successfully defends both Byzantine attack and backdoor attack. The method adopts a variational autoencoder (VAE) to generate the reconstruction errors of local model updates. It is proved that the reconstruction errors of malicious updates are much larger than that of the benign ones and can be used as anomaly score. Model updates with higher reconstruction errors than the average reconstruction error are deemed as malicious updates. However, the naive classification threshold of reconstruction errors may result in low classification accuracy. Besides, as proved in Li et al. [10], even one misclassified model update may lead to the aggregated model's bad performance. Therefore, there is a need to improve the classification accuracy of malicious updates.

This paper develops an anomaly detection framework for robust federated learning systems based on variational autoencoder (VAE). The proposed method has three main steps. First, we adopt VAE to compute the reconstruction errors of model updates under no attack and then choose a proper probability distribution function and tune its parameters to fit the distribution of reconstruction errors. Next, we use VAE to compute the reconstruction errors of model updates under adversarial attacks. We use the best-fitted distribution function to compute the probability that the reconstruction error belongs to benign updates. The updates with higher probability than 90% of the updates are considered benign updates. Finally, all benign updates are aggregated to generate a global model. We test our method under Byzantine attack and backdoor attack when 10% or 30% of all the devices are adversarial attackers. The proposed federated learning framework, named Federated Reconstruction Error Probability Distribution (FREPD) framework, has three main advantages. First, FREPD uses VAE to detect and exclude malicious updates from the aggregated models' generation instead of tolerating adversarial attacks' impact as the existing methods do. This defense strategy makes it possible to eliminate the negative impacts of both Byzantine attack and backdoor attack. Second, after excluding malicious model updates, the aggregation method of FREPD can be changed on the goal of federated networks. Third, FREPD uses the probability

distribution of reconstruction errors to detect malicious model updates, which has higher classification accuracy than existing methods. In our experiments, we use multiple aggregation methods to aggregate the remaining updates. The results show that our methods converge rapidly on both IID and non-IID datasets. The rapid convergence ensures the communication efficiency of REPD. We use three aggregation methods: federated averaging (FedAvg) [17], GeoMed [13] and FedProx [18]. The contributions of this paper are summarized as follows:

- We propose a VAE based anomaly detection framework named FREPD, which uses the probability distribution of reconstruction errors to detect benign local model updates.
- We evaluate the performance of FREPD on both IID and non-IID federated datasets with various models under Byzantine attack and backdoor attack.
- We compare the convergence rate of FREPD and existing methods on datasets with different statistical heterogeneity, and the results demonstrate the superiority of the proposed model.

2 Related Work

2.1 Byzantine-robust Distributed Machine Learning

Stochastic gradient descent (SGD) is widely used in distributed machine learning [19–21]. However, SGD is vulnerable to Byzantine attack [10]. Byzantine devices can transmit arbitrary or malicious updates to the cloud to bias the learning process. Byzantine attack aims to ensure the distributed SGD algorithm does not converge or converge to an incorrect value, while the defenses aim to ensure convergence. To defend against Byzantine attack, most of the existing Byzantine-robust machine learning algorithms extend SGD with the IID assumption. Under this assumption, local model updates from benign devices are distributed around the correct gradient, while those transmitted from malicious devices to the cloud could be arbitrary. Instead of using the simple averaging aggregation method in Bottou [19], the existing algorithms focus on incorporating robust aggregation methods with SGD [13–15]. Some of these algorithms, such as Krum [15] and Medoid [22], select one of the local model updates to compute the global model update. Other algorithms, such as GeoMed [13] and Bulyan [14], generate the global model update by estimating the center of all the local model updates, which may not be one of the local model updates.

2.2 Byzantine-robust Federated Learning

The main disadvantage of these algorithms mentioned above comes from the IID assumption [10], making them a poor fit in the non-IID datasets. However, non-IID datasets are commonplace in federated learning. Aggregation methods based on IID assumption cannot be generalized to non-IID settings straightforwardly. To defend against Byzantine attack in non-IID settings, Li et al. [10] propose a class of robust stochastic methods abbreviated as RSA. RSA has several variants, each tailored for an l_p-norm regularized robustifying objective function. Wu et al. [11] combine SAGA's variance reduction with robust aggregation to deal with Byzantine attack in federated learning. Both approaches cannot defend against backdoor attack.

2.3 Federated Learning with Robustness to Backdoor Attack

Backdoor attack has a connection to Byzantine attack [12]. However, the adversarial goal of backdoor attack is to induce misclassification. It is proved that backdoor attack is useful even with Byzantine-robust aggregation methods [12]. To make federated learning robust to backdoor attack, Li et al. [16] propose a framework that learns to detect malicious devices by the reconstruction error of devices' updates, which is the output of a trained VAE model. Model updates that have larger reconstruction errors than the mean of reconstruction errors are seen as malicious updates. However, the naive classification threshold of reconstruction errors may result in low classification accuracy. Besides, as proved in Li et al. [10], even

one misclassified model update may lead to the aggregated model's bad performance. In our paper, we first choose a proper probability distribution function, tune its parameters to fit the distribution of observed benign reconstruction errors, and then use the well-fitted function to compute the probability that the reconstruction error is generated from benign updates. The updates that have a high probability are considered benign updates and aggregate the global model.

3 FREPD for Robust Federated Learning

3.1 Preliminary: Federated Averaging and Variational Autoencoder

3.1.1 Federated Averaging

In federated learning, many devices learn on their local data and communicate with a cloud to achieve a global update. We assume that there are K devices over which the data is partitioned with $\mathcal{P}_k$, the set of indexes of data points on the device k, with $n_k = |\mathcal{P}_k|$ and $n = \sum_{k=1}^{K} n_k$. A typical implementation of FedAvg with a fixed learning rate η has each device k compute $w_t - \eta\nabla\ell(w_t; b)$. The average gradient on its local data at the model on the round t is w_t. The cloud aggregates these gradients and applies the update

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{1}$$

Each device locally takes one step of gradient descent on the current model using its local data, and the cloud then takes a weighted average of the updates from devices to update the global model [17]. This algorithm selects C-fraction of devices on each round and computes the gradient of the loss over all the selected devices' data. For each round, C is the fraction of devices that perform computation. We present Federated Averaging (FedAvg) in Algorithm 1.

Algorithm 1: FedAvg

Cloud:
Input: w_0, η
for each round t = 1,2,…, **do**
S_t = (random set of max($C.K$,1) devices);
for each device $k \in S_t$ **in parallel do**
Broadcast the current global model update $w_{t\text{-}1}$ to the device k;
Receive the local model update w_t^k from the device k;
Update the global model update w_t via (1)
Device *k*:
for each round t = 1,2,…, **do**
Receive the cloud's global model update $w_{t\text{-}1}$
Use the global model update to train the local model and compute the local model update w_t^k
Send the current local model w_t^k to the cloud

3.1.2 Variational Autoencoder

A variational autoencoder (VAE) [23] is a directed graphical model with certain types of latent variables, such as Gaussian latent variables [24]. As shown in Fig. 1, a VAE has an encoder and a decoder. The highest

layer of the decoder module is the start of the generative process and z is the latent variable generated from the prior distribution $p_\theta(z)$. $g(z)$ is the process of data generation and the result is x. The data x is generated by the generative distribution $p_\theta(x|z)$ conditioned on $z : z \sim p_\theta(z), x \sim p_\theta(x|z)$. In the Stochastic Gradient Variational Bayes [23] framework, the variational lower bound of log-likelihood is used as a surrogate objective function [24]. The variational lower bound is written as:

$$\log p_\theta(x) \geq -D_{KL}\left(q_\phi(z|x)||p_\theta(z)\right) + E_{q_\phi(z|x)}[\log p_\theta(x|z)] \tag{2}$$

In this framework, $q_\phi(z|x)$ is the approximate posterior and $p_\theta(z)$ is the prior distribution of the latent variable z. $p_\theta(x|z)$ is the likelihood of the data x given the latent variable z. The first term of the right-hand side of Eq. (2) is the KL divergence between $q_\phi(z|x)$ and $p_\theta(z)$. The second term of the right-hand side of Eq. (2) can be approximated by drawing samples $z^{(l)}(l = 1, \ldots, L)$ by the approximate posterior $q_\phi(z|x)$. The variational lower bound can be rewritten as follows:

$$\log p_\theta(x) \geq -D_{KL}\left(q_\phi(z|\mathrm{x}) \parallel p_\theta(z)\right) + \frac{1}{L}\sum_{l=1}^{L} \log p_\theta\left(\mathrm{x}|z^{(l)}\right) \tag{3}$$

where $z^{(l)} = g_\phi\left(x, \epsilon^{(l)}\right), \epsilon^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The approximate posterior is reparameterized with a deterministic, differentiable function $g_\phi(\cdot, \cdot)$, whose arguments are data x and the noise variable ϵ. In our paper, we first train a VAE model on benign model updates and then use the trained model to compute model updates' reconstruction errors.

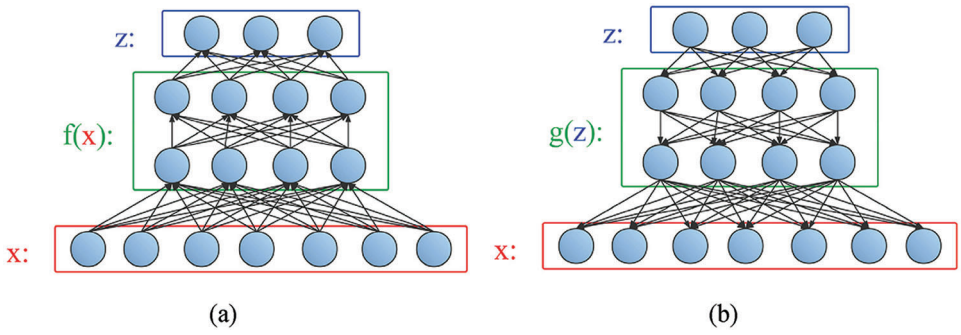

Figure 1: Encoder and decoder of a VAE (a) Encoder (b) Decoder

3.2 Attack Model

Our paper considers three types of Byzantine attack: sign-flipping attack, additive noise attack, and same-value attack. We also consider one type of backdoor attack: model replacement.

3.2.1 Byzantine Attack

Under Byzantine attack, malicious devices can send arbitrary model updates to the cloud. In these cases, the adversarial goal is to ensure the aggregated model converges to 'sub-optimal to utterly ineffective models' while defenses aim to ensure convergence [12]. We consider the following attack types:

- **Sign-flipping attack.** Under sign-flipping attack, malicious devices flip the signs of their local model updates and transfer the sign-flipped updates to the cloud [10]. The updates from device k is flipped as $w^k = \sigma\hat{w}^k$. Here $\hat{w}^k$ is the real value and σ is a constant, which we set as -4.
- **Additive noise attack.** Under additive noise attack, the malicious device k adds Gaussian noise to the local model update and set the update as $w^k = \hat{w}^k + \epsilon$. Here $\hat{w}^k$ is the real value and ϵ is vector drawn from a Gaussian distribution with mean zero and standard deviation 20.

- **Same-value attack.** Under same-value attack, the malicious device k sets its local model update as $w^k = c\mathbf{1}$. Here $\mathbf{1} \in \mathbb{R}^d$ is an all-one vector and c is a constant, which we set as 100.

3.2.2 Backdoor Attack

Under backdoor attack, the adversary aims to cause the aggregated model to misclassify a set of chosen inputs [5]. For example, in text classification task, the adversary may aim to suggest a particular restaurant's name after observing the phrase "my favorite restaurant is" [9]. As proved in Bhagoji et al. [12], backdoor attack is effective even with Byzantine-robust aggregation methods. Backdoor attack is a typical type of backdoor attack and has two main categories: naive approach and model replacement.

- **Naive approach**: The naive approach can simply train its model on backdoored inputs [25]. The training dataset includes a mix of correctly labeled inputs and backdoored inputs to induce misclassification. Although the naive approach can easily break distributed learning, it does not work against federated learning. The aggregation methods in federated learning cancel out most of the backdoored data's contribution, and the aggregated model can quickly recover from the naive approach. As proved [25], most of the devices in federated networks should be attackers, and the poisoning process is prolonged when using the naive approach.
- **Model replacement**: When only device k^* is the selected attacker in round t, the attacker attempts to substitute the whole model with a malicious model w^* by sending

$$\Delta w_t^{k^*} = \beta(w^* - w_t) \tag{4}$$

where $\beta = \left(\sum_{k \in S_t} n_k\right) \Big/ \eta n_k$ is a boost factor. Then the global update Δw_{t+1} will be

$$\Delta w_{t+1} = w^* + \eta \frac{\sum\limits_{k \in S_t, k \neq 1} n_k \Delta w_t^k}{\sum\limits_{k \in S_t} n_k} \tag{5}$$

If we assume the model has sufficiently converged, the updates from benign devices will be small. Δw_{t+1} will thus be a close neighbor of w^*. In our paper, we use model replacement to backdoor devices.

3.3 Defense Assumption and Goals

We make assumptions about our defense mechanism against Byzantine attacks and backdoor attacks as follows:

- **The cloud is fully trusted**. The cloud plays a vital role in detecting and excluding malicious local model updates from devices. If the cloud is compromised, the aggregated model will be vulnerable to adversarial attacks.
- **No malicious devices in the beginning**. Devices may be vulnerable but are not compromised when they first participate in the federated network. It takes some time for adversaries to compromise devices, leaving sufficient time to learn the probability distribution of observed benign model updates' reconstruction errors.
- **Sufficient computing resources**. The cloud has sufficient computing resources to detect and exclude malicious model updates. The devices can handle the computing consumption of training local models and performing adversarial attacks.

The proposed defense mechanism aims to detect malicious model updates and to mitigate the impacts of adversarial attacks. We first make a precise determination of whether a certain model update is malicious and then remove malicious updates from the global model's aggregating process.

3.4 Malicious Model Update Detection

It is proved that the most effective way of eliminating the impact of malicious model updates is to detect and exclude these malicious updates before model aggregation [9]. Based on variational autoencoder (VAE), a state-of-the-art malicious model update detection algorithm, Spectral [9], significantly outperforms all traditional algorithms. This method trains a VAE to compute the reconstruction errors of model updates and then use the reconstruction errors to determine whether a local model update is malicious. If a model update's reconstruction error is larger than the average reconstruction error, the update will be seen as a malicious update. However, as mentioned above, the naive classification threshold of reconstruction errors may result in low classification accuracy. Besides, as proved in Li et al. [10], federated learning is vulnerable to even one malicious device. Therefore, the misclassified model update may lead to the bad performance of the aggregated model. To improve the classification accuracy of model updates, we use the probability distribution of reconstruction errors to determine whether the updates are malicious. We assume that all the updates are benign during the first five communication rounds. FREPD first chooses a proper probability distribution function and tune its parameters to fit the distribution of observed benign reconstruction errors and then use the well-fitted function to compute the probability that the reconstruction error is generated from benign updates. The updates with higher probability than 90% of updates are considered benign updates and aggregate the global model. To prevent misclassifying malicious updates as benign updates, we also use the mean of reconstruction errors as a dynamic threshold to detect and exclude malicious updates. The process is shown in Fig. 2. The distributions that we use in FREPD include Normal, Exponentiated Weibull, Minimum Weibull, Generalized Extreme value, Beta, Gamma, Rayleigh, and Log-Normal. We apply the Kolmogorov-Smirnov (KS) test to choose the most appropriate distribution.

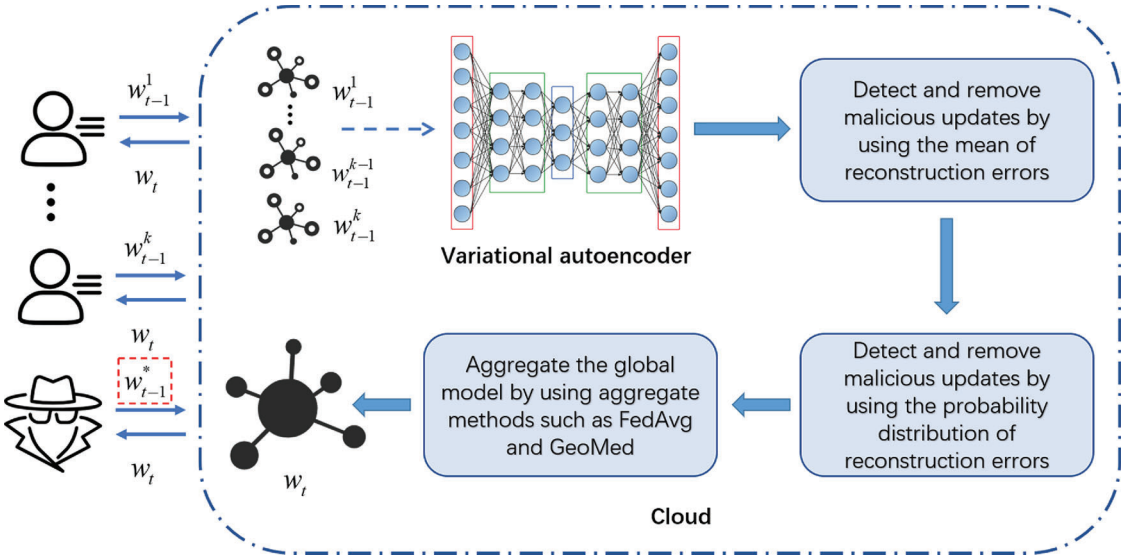

Figure 2: Proposed defense framework using VAE

Based on our assumption that it takes some time for adversaries to compromise devices, local model updates are deemed benign updates in the first five communication rounds. We thus use benign updates to train a VAE model. After the first five communication rounds, the devices send local model updates to the cloud in each round. To avoid the curse of dimensionality, as [9] does, we employ a surrogate vector generated by randomly sampling the model update vector. As shown in the Experimental Evaluation section, random sampling is highly efficient. The cloud generates the surrogate vectors of local model

updates and then uses testing trained VAE model to compute all the surrogate vectors' reconstruction errors. As Veeramachaneni et al. [26] has done, we set the detection threshold as the mean value of all the reconstruction errors, which leads to dynamic thresholding. The updates with reconstruction errors larger than the threshold are considered malicious updates and excluded from model aggregation. We apply the KS test to choose a proper probability distribution function and tune its parameters to fit the distribution of observed benign updates' reconstruction errors. We then use the well-fitted function to compute the probability distribution of reconstruction errors to select benign updates from the remaining updates. The updates with higher probability than 90% of updates are considered benign updates. All the benign updates are aggregated to generate the global model update which will be sent to all the devices in the next communication round. We use the existing aggregation methods like FedAvg [17] after excluding the malicious updates. Therefore, our method's convergence property is the same as the aggregation methods that we use. We present the proposed method in Algorithm 2, which detects and excludes malicious model updates before model aggregation.

Algorithm 2: FREPD

Cloud:

1. **Input:** w_0, η
2. Use observed benign model updates to train the VAE model before adversarial attacks;
3. Apply the KS test to choose a proper probability distribution function and tune its parameters to fit the distribution of reconstruction errors of observed benign updates;
4. **for** each round $t = 1, 2, \ldots$ **do**
5. $S_t =$ (random set of $\max(C \cdot K, 1)$ devices);
6. **for** each device $k \in S_t$ **in parallel do**
7. Broadcast the current global model update w_{t-1} to the device k;
8. Receive the local model update w_t^k from the device k;
9. Randomly sample the local model update to generate a low-dimensional surrogate vector;
10. Use the trained VAE model to compute the reconstruction errors of all the surrogate vectors;
11. Exclude all the updates with higher reconstruction errors than the mean of all the reconstruction errors;
12. Select updates with higher probability of belonging to benign reconstruction error distribution as benign updates;
13. Generate the global model update w_t by using aggregation methods.

Device k:

1. **for** each round $t = 1, 2, \ldots$ **do**
2. Receive the cloud's global model update w_{t-1};
3. Use the global model update to train the local model and compute the local model update w_t^k;
4. Send the current local model w_t^k to the cloud.

4 Performance Evaluation

This section evaluates and analyzes the performance of the proposed VAE-based methods. We use GeoMed as the aggregation method in our framework and evaluate the robustness of the proposed methods to Byzantine attack and backdoor attack in IID and non-IID federated datasets.

4.1 Datasets and Models

We explore a suite of IID and non-IID federated datasets using both convex and non-convex models in our experiment. IID federated datasets include Vehicle, while non-IID datasets include MNIST and FEMINIST.

Vehicle: We use the same Vehicle Sensor (Vehicle) dataset as Smith et al. [27]. The dataset contains acoustic, seismic, and infrared sensor data collected from a distributed network of 23 sensors [18]. Each sample has 50 acoustic and 50 seismic features and a binary label. Each sensor is a device. We train a linear SVM for the prediction between AAV-type and DW-type vehicles. The hyperparameters are tuned to the best configuration in the experiment. In each communication round, we select 10 devices from all the devices to aggregate the global model.

MNIST: The MNIST [24] dataset contains handwritten digits 0–9. Following [18], we distribute the data among 1000 devices. In our experiments, MNIST is a non-IID partition of the data, as each device has samples of only two digits. Thus, we explore whether our method will break on highly non-IID data by training models on MNIST. The number of samples per device follows a power law. The model takes a flattened 784-dimensional (28×28) image as input and outputs a class label between 0 and 9. In each communication round, we select 50 devices from all the devices to aggregate the global model.

FEMNIST: The Federated Extended MNIST (FEMNIST) dataset serves a similar benchmark to the popular MNIST [28,29] dataset. Following [18], we subsample 10 lower case characters ('a'–'j') from EMIST [30] and distribute the data among 200 devices. Each device has only 5 classes. The model of FEMNIST is the same as that of MNIST. In each communication round, we select 50 devices from all the devices to aggregate the global model.

4.2 Benchmark Algorithms and Experimental Metric

FedAvg [17]. FedAvg algorithm takes a weighted average of the resulting model from the devices. FedAvg is proved robust to unbalanced and non-IID data distributions and has good performance on non-convex models. We also evaluate the performance of FedAvg under no attack.

GeoMed [13]. The geometric median (GeoMed) of local model updates is used in this algorithm to generate a global model update. The geometric median of $\left\{w^k : k \in [K]\right\}$ is denoted by:

$$GeoMed(\{w_k\}) = \arg\min_{w \in \mathbb{R}^d} \sum_{k=1}^{K} \left\|w - w^k\right\|_2 \tag{6}$$

The generated global model update may not be one of the local model updates [13].

Krum [15]. Rather than taking the local model updates' geometric median, Krum uses one of the local updates to generate a global model update. The chosen local update minimizes the sum of distances to its nearest neighbors. The local update is chosen by:

$$Krum\left(\left\{w^k\right\}\right) = w^{k^*}, k^* = \arg\min_{i,j \in K} \sum_{i \to j} \left\|w^i - w^j\right\|^2 \tag{7}$$

where $i \to j (i \neq j)$ selects the indexes j of the $K - q - 2$ nearest neighbors of w^i in $\left\{w^k : k \in [K]\right\}$, measured by Euclidean distances. q is the number of malicious devices, which must be known in advance.

Bulyan [14]. Bulyan uses Krum [15] to obtain a subset of the local model update from the devices. The cloud generates a global model update by taking the component-wise average to the refined subset of local model updates.

Spectral [16]. An unsupervised VAE model is used in this algorithm to output the reconstruction error of local model updates. Malicious updates are proved to have larger reconstruction errors than benign ones [16]. Thus, model updates with larger reconstruction errors are excluded from the aggregation method. After excluding malicious updates, this algorithm uses the FedAvg algorithm to aggregate the remaining model updates.

To measure the effectiveness of our defense mechanism, we use two parameters as follows.

- **Testing Accuracy**. It indicates how good a model is in its normal task. For example, it is the fraction of samples labeled 'cat' classified correctly as 'cat' to the total samples.
- **Backdoor Accuracy**. It refers to how good a poisoned model is in the backdoor task. For example, it is the fraction of samples labeled 'cat' misclassified as 'dog' to the total samples.

4.3 Robustness to Byzantine attack

We consider three types of Byzantine attacks in our experiment, including same-value attack, additive noise attack, and sign-flipping attack. We evaluate our method's robustness to Byzantine attack in two scenarios where 10% and 30% of selected devices are malicious, respectively. In the experiments, our proposed method uses the FREPD framework and use GeoMed as the aggregation method.

As shown in Figs. 3–5, the proposed method (Ours) can mitigate the impact of Byzantine attack in both IID and non-IID datasets. Our method achieves the best performance in the Vehicle dataset and the MNIST dataset. In the FEMNIST dataset, our method achieves the best performance under sign-flipping attack. Under same-value attack and additive noise attack, our method achieves the second-best performance behind GeoMed. Meanwhile, our method converges faster than other baselines in all settings. As shown in the last two columns of Figs. 3–5, the performance of Krum and Bulyan remains the same regardless of the number of attackers. Krum and Bulyan select the most appropriate update from all the local model updates generated by the devices in federated networks. Since the MNIST dataset and the FEMNIST dataset in our experiment is biased, the selected update cannot apply to all the devices. As shown in Figs. 5(a) and 5(b), GeoMed is not robust to sign-flipping attack. GeoMed computes the geometric center of all the local model updates and uses it as the global update. Since sign-flipping attack makes malicious updates far away from the normal updates, the global update generated by GeoMed deviates from normal updates.

4.4 Robustness to Backdoor attack

We evaluate our method's robustness to backdoor attacks in two scenarios where only one device is malicious and 30% of selected devices are malicious, respectively. In the experiments, our proposed method uses the FREPD framework and use GeoMed as the aggregation method.

As shown in Fig. 6(a), the proposed method (Ours) achieves the best performance on the MNIST dataset under the impact of single backdoor attacker. The testing accuracy of FREPD is the highest, while its backdoor accuracy is the lowest. Krum and Bulyan have low backdoor accuracy and have low testing accuracy because they are not suitable for non-IID datasets such as the MNIST dataset. Spectral and GeoMed have the second-best performance behind FREPD. GeoMed has high testing accuracy and low backdoor accuracy under backdoor attack because all the model updates' geometric median is close to the normal ones. FedAvg, without any defense, has bad performance under backdoor attack.

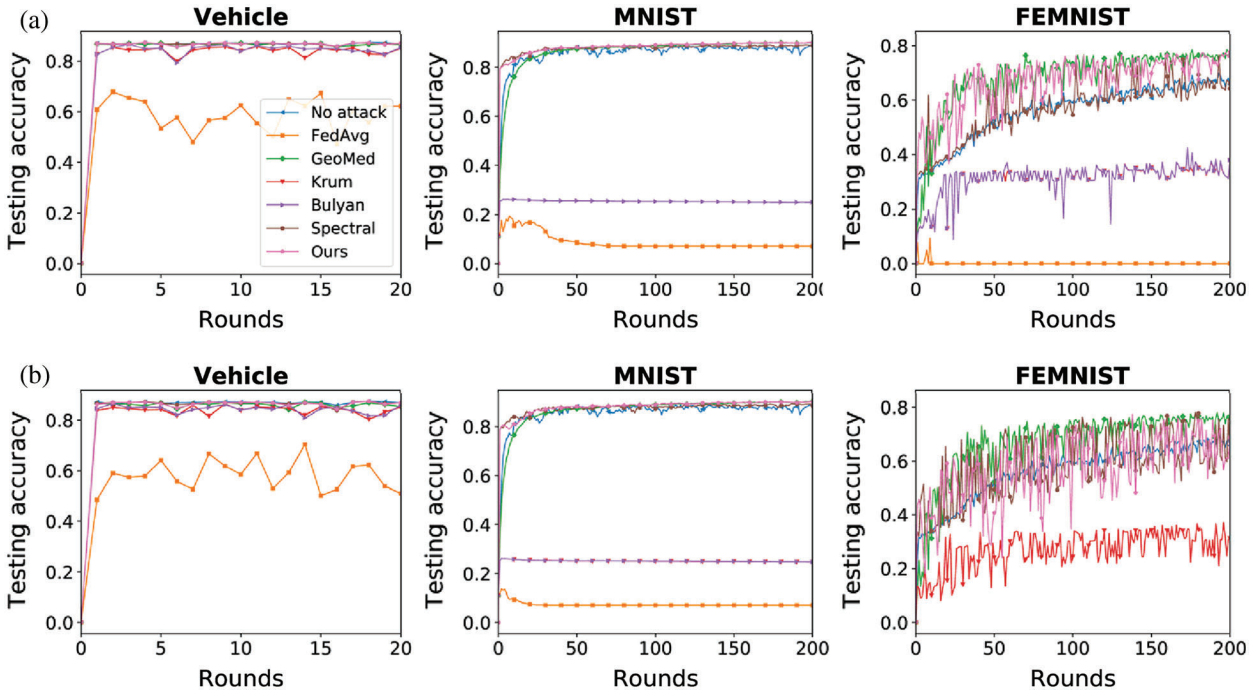

Figure 3: Testing accuracy under additive noise attack (a) Additive noise (10%) (b) Additive noise (30%)

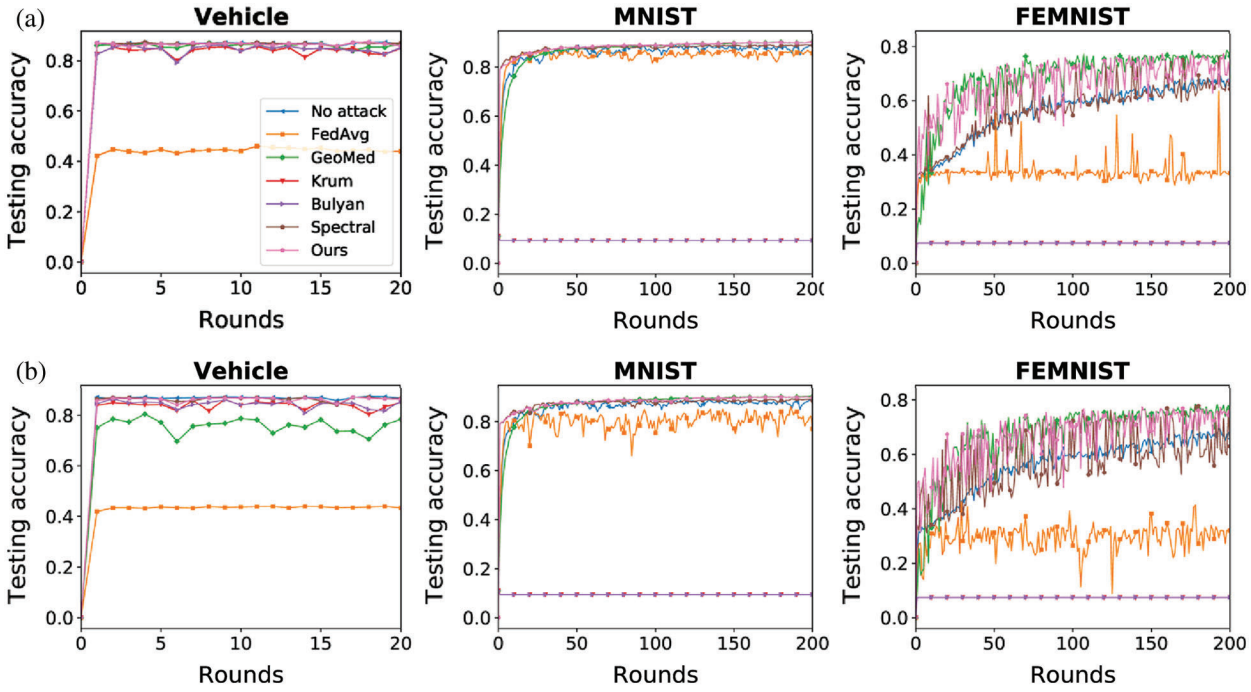

Figure 4: Testing accuracy under same-value attack (a) Same-value (10%) (b) Same-value (30%)

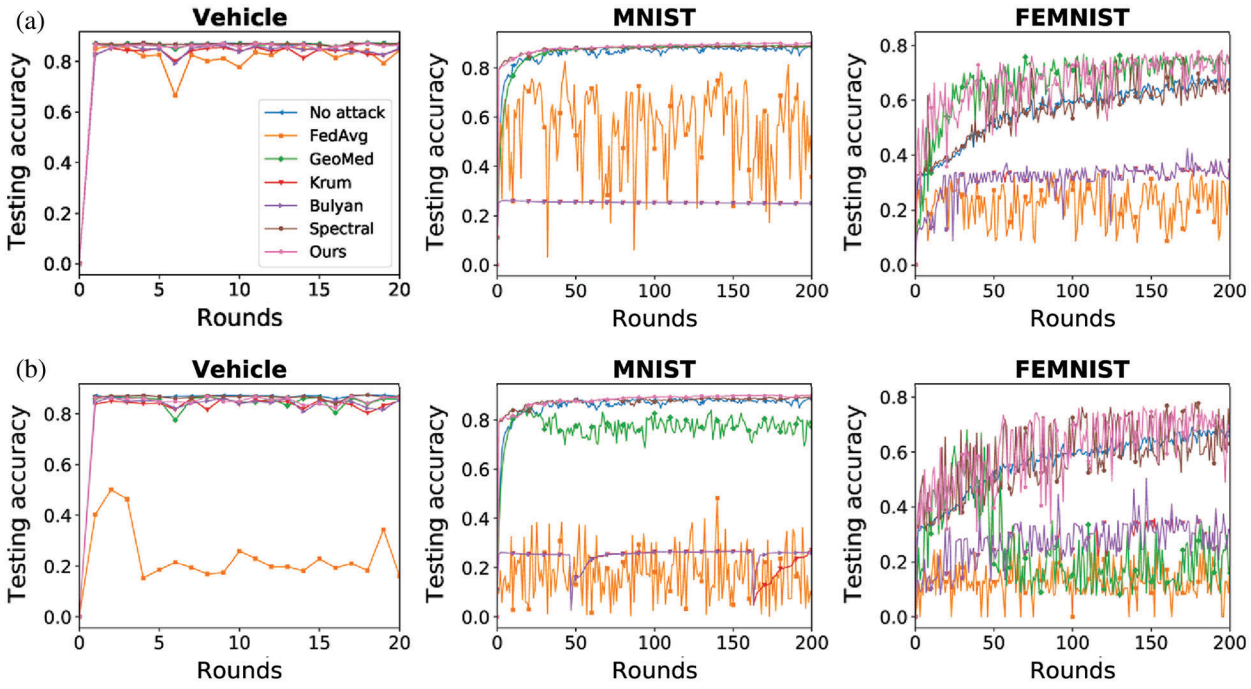

Figure 5: Testing accuracy under sign-flipping attack (a) Sign-flipping (10%) (b) Sign-flipping (30%)

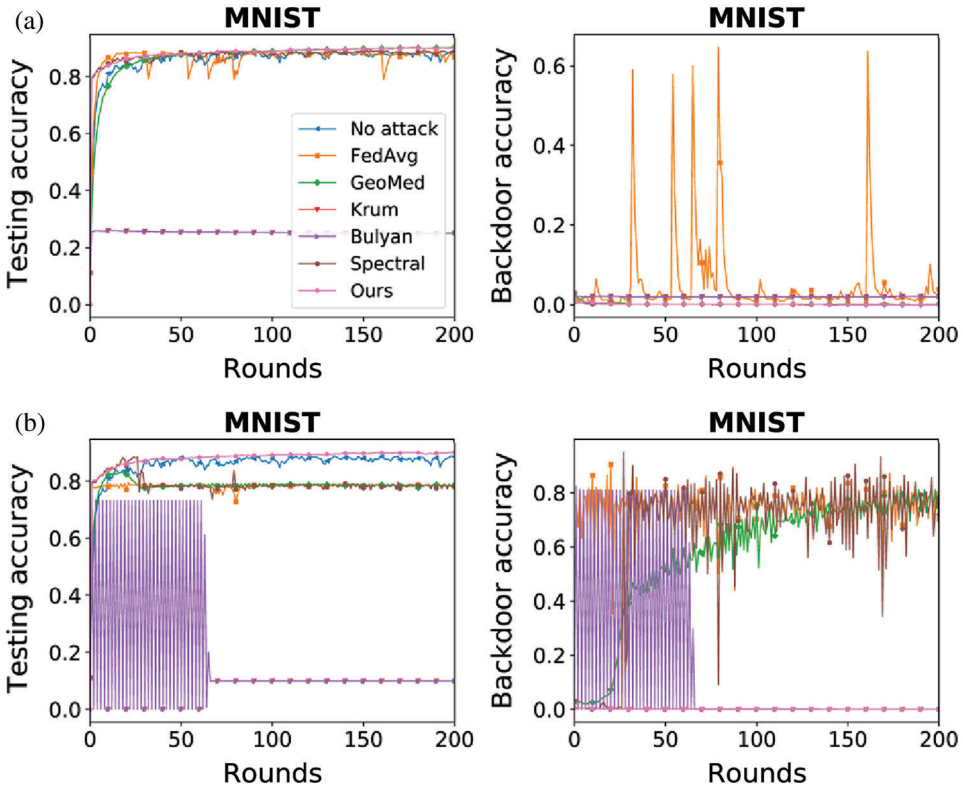

Figure 6: Testing accuracy and backdoor accuracy under backdoor attack (a) Single backdoor attacker (b) Multiple backdoor attackers (30%)

As shown in Fig. 6(b), the proposed method (Ours) achieves the best performance on the MNIST dataset under the impact of multiple backdoor attacks, where 30% of selected devices are attackers. The testing accuracy of FREPD is the highest, while its backdoor accuracy is the lowest. The performance of Krum and Bulyan is volatile at the beginning, which may because the selected local model update is malicious in some rounds and is benign in the other rounds. GeoMed has high testing accuracy and high backdoor accuracy under backdoor attack because the geometric median of all the model updates deviates from the normal ones. Spectral also has high testing accuracy and high backdoor accuracy because some malicious updates are misclassified as benign ones. FedAvg has high testing accuracy and high backdoor accuracy because it cannot defend backdoor attack.

5 Conclusion

Our paper proposes a VAE based anomaly detection framework named FREPD. To prevent misclassifying malicious updates as benign ones, FREPD uses the probability distribution of reconstruction errors to detect benign updates after using the average reconstruction error to exclude malicious updates. During the first five communication rounds, we apply the Kolmogorov-Smirnov test to choose a proper probability distribution function and tune its parameters to fit the distribution of reconstruction errors of observed benign updates. We use the distribution function to compute the probability that the reconstruction error is generated from benign updates. If the update's probability is higher than that of 90% of all the local updates, the update will be seen as a benign update. Only benign updates are used to generate the aggregated model. We conduct experiments on IID and non-IID datasets. The results show that FREPD has competitive performance compared to existing aggregation methods under Byzantine attack and backdoor attack.

Funding Statement: This research is supported by Education Ministry-China Mobile Research Funding under Grant No. MCM20170404.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Chen, Y. Ning, M. Slawski and H. Rangwala, "Asynchronous online federated learning for edge devices with non-IID data," arXiv::1911.02134, 2020.

[2] A. Gumaei, M. Al-Rakhami and H. AlSalman, "DL-HAR: Deep learning-based human activity recognition framework for edge computing," *Computers Materials & Continua*, vol. 65, no. 2, pp. 1033–1057, 2020.

[3] H. Wang, J. Yong, Q. Liu and A. Yang, "A novel GLS consensus algorithm for alliance chain in edge computing environment," *Computers Materials & Continua*, vol. 65, no. 1, pp. 963–976, 2020.

[4] C. Qian, X. Li, N. Sun and Y. Tian, "Data security defense and algorithm for edge computing based on mean field game," *Journal of Cyber Security*, vol. 2, no. 2, pp. 97–106, 2020.

[5] Y. Zhang, Y. Yang and X. Wang, "A novel Android malware detection approach based on convolutional neural network," in *Proc. ICCSP*, Guiyang, Guizhou, China, pp. 144–149, 2018.

[6] X. Wang, Y. Yang, C. Tang, Y. Zeng and J. He, "DroidContext: Identifying malicious mobile privacy leak using context," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Tianjin, China, pp. 807–814, 2016.

[7] J. Liu, Y. Zeng, J. Shi, Y. Yang, R. Wang *et al.,* "MalDetect: A structure of encrypted malware traffic detection," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 721–739, 2019.

[8] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman *et al.,* "Towards federated learning at scale: System design," arXiv:1902.01046, 2019.

[9] Z. Sun, P. Kairouz, A. T. Suresh and B.H. McMahan, "Can you really backdoor federated learning?," arXiv:1911.07963, 2019.

[10] L. Li, W. Xu, T. Chen, G. B. Giannakis and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI*, Honolulu, HI, USA, pp. 1544–1551, 2019.

[11] Z. Wu, Q. Ling, T. Chen and G. B. Giannakis, "Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks," arXiv:1912.12716, 2019.

[12] A. N. Bhagoji, S. Chakraborty, P. Mittal and S. Calo, "Analyzing federated learning through an adversarial aens," arXiv:1811.12470, 2019.

[13] Y. Chen, L. Su and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," arXiv:1705.05491, 2017.

[14] E. M. E. Mhamdi, R. Guerraoui and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," arXiv:1802.07927, 2018.

[15] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, pp. 119–129, 2017.

[16] S. Li, Y. Cheng, W. Wang, Y. Liu and T. Chen, "Learning to detect malicious clients for robust federated learning," arXiv:2002.00211, 2020.

[17] H. B. McMahan, E. Moore and D. Ramage, "Federated learning of deep networks using model averaging," arXiv:1602.05629, 2016.

[18] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar *et al.,* "Federated optimization in heterogeneous networks," arXiv:1812.06127, 2020.

[19] L. Bottou, "Large-Scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT'2010*, Paris, France, pp. 177–186, 2010.

[20] Y. Chou, W. Liao, Y. Chen, M. Chang and P. T. Lin, "A distributed heterogeneous inspection system for high performance inline surface defect detection," *Intelligent Automation & Soft Computing*, vol. 25, no. 1, pp. 79–90, 2019.

[21] E. Adel, S. El-sappagh, M. Elmogy, S. Barakat and K. Kwak, "A fuzzy ontological infrastructure for semantic interoperability in distributed electronic health record," *Intelligent Automation & Soft Computing*, vol. 26, no. 2, pp. 237–251, 2020.

[22] C. Xie, O. Koyejo and I. Gupta, "Generalized byzantine-tolerant SGD," arXiv:1802.10116, 2018.

[23] D. Jimenez Rezende, S. Mohamed and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," arXiv:1401.4082, 2018.

[24] K. Sohn, H. Lee and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in Neural Information Processing Systems*, vol. 28, pp. 3483–3491, 2015.

[25] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin and V. Shmatikov, "How to backdoor federated learning," arXiv:1807.00459, 2019.

[26] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias and K. Li, "AI^2: Training a big data machine to defend," in *Proc. IEEE BigDataSecurity/HPSC/IDS*, New York, NY, USA, pp. 49–54, 2016.

[27] V. Smith, C.K. Chiang, M. Sanjabi and A. Talwalkar, "Federated multi-task learning," arXiv:1705.10467, 2018.

[28] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, Anchorage, AK, USA, pp. 2278–2324, 1998.

[29] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konecny *et al.,* "LEAF: A benchmark for federated settings," arXiv:1812.01097, 2019.

[30] G. Cohen, S. Afshar, J. Tapson and A. V. Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. IJCNN*, Anchorage, AK, USA, pp. 2921–2926, 2017.