2022 26th International Conference on Pattern Recognition (ICPR)
August 21-25, 2022, Montréal, Québec, Canada

Enhancing Robustness in Federated Learning by Supervised Anomaly Detection

Pengrui Quan*, Wei-Han Lee†, Mudhakar Srivatsa† and Mani Srivastava*
*University of California, Los Angeles
Email: prquan@g.ucla.edu, mbs@ucla.edu
†IBM Thomas J. Watson Research Center
Email: wei-han.lee1@ibm.com, msrivats@us.ibm.com

2022 26th International Conference on Pattern Recognition (ICPR) | 978-1-6654-9062-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPR56361.2022.9956655

***Abstract*—Recent years have seen the increasing attention and popularity of federated learning (FL), a distributed learning framework for privacy and data security. However, by its fundamental design, federated learning is inherently vulnerable to model poisoning attacks: a malicious client may submit the local updates to influence the weights of the global model. Therefore, detecting malicious clients against model poisoning attacks in federated learning is useful in safety-critical tasks.**

However, existing methods either fail to analyze potential malicious data or are computationally restrictive. To overcome these weaknesses, we propose a robust federated learning method where the central server learns a supervised anomaly detector using adversarial data generated from a variety of state-of-the-art poisoning attacks. The key idea of this powerful anomaly detector lies in a comprehensive understanding of the benign update through distinguishing it from the diverse malicious ones. The anomaly detector would then be leveraged in the process of federated learning to automate the removal of malicious updates (even from unforeseen attacks).

Through extensive experiments, we demonstrate its effectiveness against backdoor attacks, where the attackers inject adversarial triggers such that the global model will make incorrect predictions on the poisoned samples. We have verified that our method can achieve 99.0% detection AUC scores while enjoying longevity as the model converges. Our method has also shown significant advantages over existing robust federated learning methods in all settings. Furthermore, our method can be easily generalized to incorporate newly-developed poisoning attacks, thus accommodating ever-changing adversarial learning environments.

I. Introduction

Federated learning (FL) [18, 37, 24] has gained more and more interests in practical learning scenarios due to its tremendous privacy advantages. In each round of learning, the global model will be synchronized at each involved local client. Then each participant will train a local model and upload it to the central server which aggregates the updates and produces a new global model. In this case, federated learning enables utilization of sensitive or private data to train a global model and prevent personal data leakage simultaneously. The learning framework can adapt to a wide range of local data while remaining unknowing of it. An intriguing motivation can be building health prediction and monitor systems on personal devices such as mobile phone or smart watch using daily activity signals [22, 23].

However, by its fundamental design, federated learning is inherently vulnerable to model poisoning attack: the model has

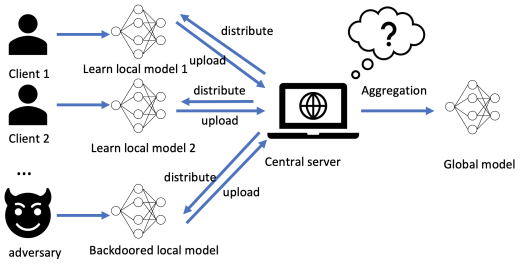


Fig. 1: Federated Learning

state-of-the-art performance on normal data, but its behavior is manipulated on seeing inputs with intentionally designed trigger pattern which does not change human's perception. The malicious local clients can inject these patterns into the training inputs to poison the model (as shown in Fig. 1) and benefit from the abnormal behavior of the global model. For instance, the attacker can inject a 5×5 pattern to replace a part of the benign image or insert a "trigger" phrase into a paragraph while maintaining the semantic meaning in natural language.

To mitigate impact of these malicious data, a number of methods have been proposed to enhance the robustness of federated learning [36, 5, 21, 16]. However, most of them only implement transformations on the benign updates for unsupervised differentiation from malicious updates, thus limiting their detection performance. To overcome these challenges, we propose a novel robust federated learning method through constructing a supervised anomaly detector. Specifically, the supervised anomaly detector is trained for distinguishing benign updates and malicious updates generated from a variety of poisoning attacks. Through carefully comparing different characteristics of benign and malicious updates, our method can achieve superior performance than previous methods (that only consider unsupervised classification). In summary, our work makes the following important contributions.

- We propose a robust federated learning method where we construct a supervised anomaly detector that aims to distinguish benign updates and malicious data generated by a range of poisoning attacks. As compared with previous methods that only leverage unsupervised detector, our method can provide a better understanding of the

benign updates thus achieving better robustness for the entire federated learning process.
- The effectiveness of our proposed method has been validated by using multiple real-world data sets. Specifically, our method can achieve up to 99.0% AUC scores in detecting malicious updates (even from unseen attacks).
- Our method has shown significant advantages over the state-of-the-art methods by reducing the attack success rate by 90% in defending against manipulation of malicious data. In addition, newly-proposed poisoning attacks can be incorporated into our method in a straightforward manner for enhanced robustness against ever-developing adversarial perturbations.

II. Related works

A. Poisoning Attack in FL Setting

Based on the attackers' goals, poisoning attack in federated learning settings can be roughly classified into three categories:

Untargeted attacks. The adversary's goal is to make the model converge to a sub-optimal point or to make the model perform poorly, e.g., completely diverge. These attacks are also refereed as convergence attacks in the settings for Byzantine adversaries [3, 5, 36, 7].

Targeted attacks. Adversary wants the model to misclassify only a set of chosen samples while the overall performance on the main task remains untouched [4, 29].

Backdoor attacks. Backdoor attacks are designed to mislead the trained model to predict a specific label on any input data that has an attacker-chosen pattern, i.e., trigger. Instead of misclassifying a set of samples, backdoor attack is to make the trained model wrongly behave when seeing the trigger pattern [32, 33, 25, 2]. Besides, [2, 25] also empirically show that it is challenging to defend against backdoor attacks using unsupervised learning methods such as clustering and distance auditing. This motivates us to improve the defense performance using supervised anomaly detection.

B. Robust Aggregation in FL

Coordinate-wise median [36]. Given the set of updates, $\{\tilde{\boldsymbol{\theta}}_j^t\}_{j=1}^k$, the aggregator generates the update $\boldsymbol{\theta}_{j+1}$ by taking the coordinate-wise median of the set of updates from k clients.

Krum [5]. Krum assumes n agents and f of them are Byzantine adversaries, where $n \geq 2f+3$. At a particular time t, for each $\tilde{\boldsymbol{\theta}}_j^t$, the $n-f-2$ closest (in terms of L_p-norm) other updates are chosen and their distances to $\tilde{\boldsymbol{\theta}}_j^t$ are added up to compute a score S_j. *Krum* then selects the $\tilde{\boldsymbol{\theta}}_j^t$ with the lowest score to produce the global model.

RFA [21]. Given the set of updates $\{\tilde{\boldsymbol{\theta}}_j^t\}_{j=1}^k$, *RFA* calculates the geometric median (GM) of the set by minimizing $\boldsymbol{\theta}^{t+1} = \underset{\boldsymbol{\theta}}{\text{argmin}} \sum_{j=1}^k \alpha_k ||\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}_j^t||$, where α_k is the weight parameter and $\sum_k \alpha_k = 1$ w.l.o.g.

Ditto [16]. Instead of learning a single global model $\boldsymbol{\theta}$, *Ditto* proposes to learn n local model $\boldsymbol{v}_j^t$ to strengthen the personalization and robustness. Specifically, this method uses an additional regularization term $||\boldsymbol{\theta}^t - \boldsymbol{v}_j||$ in the training of local models and the robustness is evaluated on those benign clients $\boldsymbol{v}_j$.

C. Secure Training and Inference in FL

The problem of secure training and inference can be solved via generic secure computation techniques such as secure two-party (2PC) computation [35], fully homomorphic encryption [8], and trusted execution enclaves (TEEs) [26]. This thread of research is independent of our objective of developing new robust federated learning algorithms. However, given that our method needs to calculate a function over individual updates, we discuss the possibility of applying the above techniques to relieve the privacy concern of accessing the above information.

Two-party computation. [19] allows two parties to approximately or exactly compute an arbitrary operation function (e.g., matrix multiplication) on their inputs without sharing their inputs with the opposing party.

Fully homomorphic encryption (FHE). CryptoNets [9] is the first work that attempts to optimize and tailor FHE schemes for secure inference. However, the computation overhead of FHE is enormous which limits its application to networks of a few layers.

Trusted Execution Environment (TEE). [26] uses the Intel SGX hardware enclave to securely perform inference. It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity.

III. Our Proposed Robust Federated Learning

A. Federated Learning

The training objective. Federated learning distributes the training among n total clients and aggregates local models to iteratively learn a global model $\boldsymbol{\theta}$. Specifically, it minimizes the learning objective below.

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}}{\text{argmin}} \sum_{j\in[n]} \sum_{\boldsymbol{x}_i, y_i \in \mathcal{D}_j} \mathcal{L}_{train}(\boldsymbol{\theta}, \boldsymbol{x}_i, y_i) \tag{1}$$

In essence, n local clients jointly learn the global model. At each round t, there are k clients participating in the training and the central server will select the k participants and broadcast a global model $\boldsymbol{\theta}^t$ among them. Then each local client j will initialize the local model with $\boldsymbol{\theta}^t$ and learn a local model $\boldsymbol{\theta}_j^t$ on its personal dataset $\mathcal{D}_j = \{\boldsymbol{x}_i, y_i\}_{i=1}^{|\mathcal{D}_j|}$ by solving:

$$\boldsymbol{\theta}_j^t = \underset{\boldsymbol{\theta}}{\text{argmin}} \sum_{\boldsymbol{x}_i, y_i \in \mathcal{D}_j} \mathcal{L}_{train}(\boldsymbol{\theta}, \boldsymbol{x}_i, y_i) \tag{2}$$

After that, the update $\tilde{\boldsymbol{\theta}}_j^t = \boldsymbol{\theta}_j^t - \boldsymbol{\theta}^t$ is sent back to the central server, which will be further aggregated to produce a new global model. Then, the central server averages over all k updates with its own learning rate to produce a new model [1]:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \frac{\eta}{k} \sum\nolimits_{j=1}^k (\boldsymbol{\theta}_j^t - \boldsymbol{\theta}^t) \tag{3}$$

[1]For simplicity, we will ignore the superscript t in the following discussion.

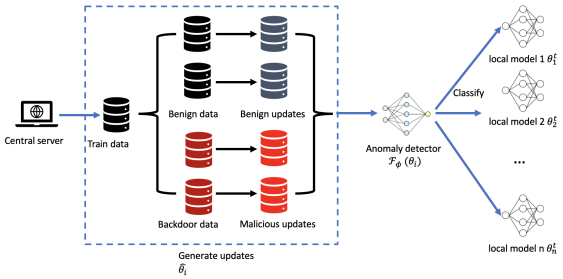


Fig. 2: Our Robust Federated Learning Algorithm

B. Model Poisoning Attack

Given the designed framework of federated learning, it is generically vulnerable to model poisoning attack: where a malicious client may submit the local updates to influence the weights of the global model. For instance, previous work has shown that adversaries can apply *triggers* (by simply replacing the corners of the image while barely affecting human's perception) to lead to misbehavior of the model, by predicting the poisoned cat image as airplane for example (as shown in Fig. 3). This will raise concerns in deploying the machine learning model in safety-critical tasks such as autonomous driving.

Trigger definition. Let $\mathcal{D}_j = \{\boldsymbol{x}_i, y_i\}_{i=1}^{|\mathcal{D}_j|}$ be the training dataset available at client j. For the dataset $\mathcal{D}_j = \{\boldsymbol{x}_i, y_i\}_{i=1}^{|\mathcal{D}_j|}$, its backdoor version is $\hat{\mathcal{D}}_j = \{\hat{\boldsymbol{x}}_i, \hat{y}_i\}_{i=1}^{|\mathcal{D}_j|}$, where $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{\delta}_j$, $\hat{y}_i \neq y_i$.

Following the conventions in [14, 10, 27, 32, 28], we restrict the perturbation amount bounded by L_p-norm, e.g., [14] uses $L_2 \leq 5$, $L_0 \leq 2$, and $L_\infty \leq 0.16$ for CIFAR10 dataset.

Attacker Capability. Following the setting in [2, 4, 32], we consider the strong attacker here who has full control of their local training process, such as backdoor data injection and updating local training hyperparameters. It can also up scale its model weights to compensate the learning rate in the central server to perform stronger attacks. However, attackers do not have the ability to influence the privilege of central server such as changing aggregation rules, nor tampering the training process and model updates of other parties

Attacker Objective. The poisoning attacker is designed to mislead the trained model to predict the poisoned sample $\hat{\boldsymbol{x}}_i$ as the target label $\hat{y}_i$, while the accuracy on the normal dataset remains untouched. The adversary's learning objective is as follows:

$$\hat{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\boldsymbol{x}_i, y_i \in \mathcal{D}_j} \mathcal{L}_{train}(\boldsymbol{\theta}, \boldsymbol{x}_i, y_i) + \sum_{\hat{\boldsymbol{x}}_i, \hat{y}_i \in \hat{\mathcal{D}}_j} \mathcal{L}_{train}(\boldsymbol{\theta}, \hat{\boldsymbol{x}}_i, \hat{y}_i) \tag{4}$$

C. Our Approach

Motivation. Previous robust aggregation in FL mainly focus on transforming benign updates through various transformation strategies such as median of gradients [36, 5], singular value decomposition [7], etc. However, these methods are only limited to unsupervised classification where only benign updates are analyzed. These observations motivate our work

Algorithm 1 Supervised Anomaly Detection

Input: Central server collects a small independent portion of benign training dataset $\mathcal{D}_c$ (5% of the whole dataset). At communication round t, there are k local models θ_j^t and one produced global model θ^t.
Output: The learned anomaly detector $\mathcal{F}_\phi(\boldsymbol{\theta})$ and the global model θ^t.
Server generates malicious input data samples $\hat{\mathcal{D}}_c$ given $\mathcal{D}_c$.
The server trains local models and generate benign model updates and the malicious model updates denoted as $\Theta_b = \{\boldsymbol{\theta}_b\}$ and $\Theta_m = \{\boldsymbol{\theta}_m\}$, respectively.
The detector $\mathcal{F}_\phi(\boldsymbol{\theta})$ (parameterized by ϕ) is trained on the model updates $\Theta = \{\Theta_b, \Theta_m\}$ by minimizing Eqn. (5).
for communication round t **do**
    $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$
    for client j **do**
        local model θ_j^t is returned by client j
        if θ_j^t is classified as benign by $\mathcal{F}_\phi(\boldsymbol{\theta})$ **then**
            $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t+1} + \frac{\eta}{k}(\boldsymbol{\theta}_j^t - \boldsymbol{\theta}^t)$

where we leverage the state-of-the-art poisoning attacks for generating various malicious updates, based on which we train a supervised learning model for distinguishing benign and malicious updates. The key intuition behind our method is to utilize the current poisoning attacks as *public* information for generating malicious examples so that a supervised anomaly detector can be constructed.

$$\min_{\phi} \sum_{\boldsymbol{\theta}_j \in \Theta} \mathcal{F}_\phi(\boldsymbol{\theta}_j) \tag{5}$$

Our Algorithm. We propose to learn a supervised anomaly detector on the central server side based on the self-generated malicious updates (see Algorithm 1). Fig. 2 shows the detailed process of our method.

Our method is generalizable to any newly-developed poisoning attacks since they can be easily incorporated to produce more diverse/powerful malicious data. In our experiments, we leverage fully connected neural networks for anomaly detector. It is worthy noting that more sophisticated model architecture can be applied for enhanced detection performance.

IV. Experiments

To validate the effectiveness of our method, we conduct extensive experiments using 3 real-world datasets. Specifically, we validate the effectiveness of our method which achieve high AUC scores in detecting malicious updates (even from unseen attacks). Then, we will compare our proposed method with

TABLE I: Details of Datasets used in Our Experiments

Dataset	No. of Training	No. of Testing	Label	Format
CIFAR-10	50,000	10,000	10	$32 \times 32 \times 3$
SVHN	73,257	26,032	10	$32 \times 32 \times 3$
MNIST	60,000	10,000	10	$28 \times 28 \times 1$

several state-of- the-art robust aggregation algorithms, in order to show the advantage of our method. Finally, we evaluate the robustness of our method against adaptive attacker where the adversary has prior knowledge of our method and aims to evade its performance. All our evaluations are conducted on a PC with Intel Xeon Platinum 2.5 GHz and 64 GB memory and NIVIDA TITAN RTX graphics card. We will make our code publicly accessible to motivate future work on enhancing robustness in federated learning.

A. Experimental Setup

Datasets In our experiments, we use 3 real-world datasets (CIFAR-10 [12], SVHN [20], and MNIST [13]) to evaluate the performance of our method, and Table I provides an overview of all datasets. CIFAR-10 dataset [12] represents various types of vehicles, animals, etc., and SVHN dataset [20] represents street-view house numbers, while MNIST dataset [13] represents handwritten digits

Evaluation Metrics Evaluating the performance of our method includes quantification of the detection performance of the anomaly detector and the classification performance of the global federated learning model.

For the anomaly detector, we first compute *True Positive (TP)*: malicious updates being correctly identified, *False Positive (FP)*: benign updates being incorrectly recognized as malicious, *True Negative (TN)*: benign data being correctly identified, *False Negative (FN)*: malicious data being incorrectly recognized as benign data, based on which we compute receiver operating charateristic (ROC) that measures the tradeoff between *True Positive Rate:* $\frac{TP}{TP+FN}$ and *False Positive Rate:* $\frac{FP}{TN+FP}$. From the ROC curve, we quantify the overall detection performance under all possible values of the threshold parameters using Area Under the ROC (AUC) [17].

For the global federated learning model, we compute the following two metrics to quantify its robustness under perturbations of malicious data:

- Attack Accuracy: measures the percentage of the predicted label equaling to poisoned label $\hat{y}_i$ on the poisoned testing dataset $\hat{\mathcal{D}}_{test}$.
- Clean Accuracy: measures the percentage of the predicted label equaling to the original label y_i on the clean testing dataset $\mathcal{D}_{test}$.

Model Architecture In the experiments, we consider ResNet-18 [11] as the global model used in the federated learning setting, and we distribute each of the 3 real-world datasets to 100 agents for jointly learning the global model. Furthermore, we use a four-layer fully connected network as the anomaly detector. The anomaly detector has [256, 128, 128, 2] neurons in each layer with *ReLU* activation. In our experiments, we randomly pick the convolution layer of ResNet-18 to train the anomaly detector.

Generating Malicious Data To produce poisoning examples, the local adversary can inject the trigger patterns into local data using the strategies following [34] (as shown in Fig. 3): i) Blended: the intensity backdoored is reduced by a factor of α. ii) Corner: the corners of the image are replaced by some random patterns. iii) Watermark: a patch of size $k \times k$ is replaced by random watermarks. To evaluate how the anomaly detector perform under unseen poisoning strategies, we will report the results of using Watermark strategy to generate poisoning sampling for training the detector, while the Corner strategy is used for testing. In Sec. IV-E, we will rotate various backdoor pattern combinations and report the results. The adversary will poison all the local training data to generate $\boldsymbol{\theta}_m$. Following [2], in the testing process, we measure the attack accuracy on 1000 poisoned samples, i.e., the fraction of samples that are misclassified into the desired class under the presence of backdoor patterns. And we report the normal accuracy on 10000 benign data samples.

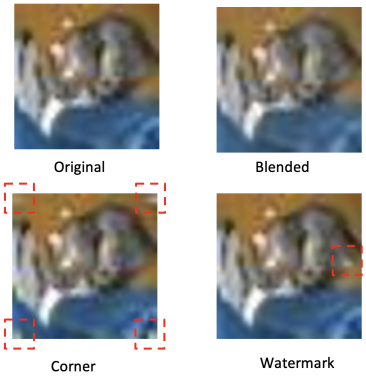


Fig. 3: Backdoor patterns utilized in our experiments.

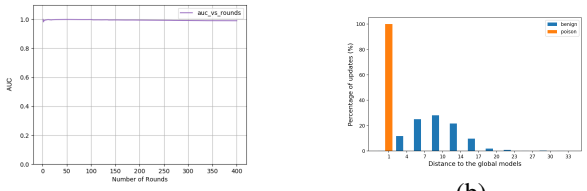
Dataset	Scale	100	75	50	10
CIFAR10	AUC	1.00	1.00	1.00	1.00
	Attack accuracy	12%	11%	10%	10%
	Clean accuracy	91%	91%	91%	91%
MNIST	AUC	0.96	0.97	0.96	0.96
	Attack accuracy	10%	12%	12%	10%
	Clean accuracy	96%	96%	96%	96%
SVHN	AUC	0.98	0.98	0.96	0.96
	Attack accuracy	12%	12%	10%	10%
	Clean accuracy	93%	94%	94%	93%

TABLE II: The performance of anomaly detector and federated learning model with varying scaling parameters.

Learning Anomaly Detector To validate the effectiveness of our method, the data used for training the anomaly detector does not overlap with the data used by normal clients and attackers participating in the learning process. Specifically, in our experiments, we assume the central server owns 5% of the original training dataset to synthesize distribution of benign and malicious weights. These data is excluded through the entire federated learning process and is irrelevant to the local clients. Then the server uses the Watermark strategy in Fig. 3 to produce backdoor images and follows Agorithm 1 to produce anomaly detector for robust federated learning.

B. Effectiveness of Our Method

Table. II shows the AUC of the anomaly detector and the corresponding attack/clean accuracy of the federated learning model using our defense, with various scaling of the local models which is one of the indicator of the attack strengths [2, 33]. To make the attack effective, instead of returning $\boldsymbol{\theta}_j^t$ to the central server, the attacker will return $\boldsymbol{\theta}_j^t \leftarrow \boldsymbol{\theta}^t + \gamma(\boldsymbol{\theta}_j^t - \boldsymbol{\theta}^t)$. According to Eqn. (3), when $\gamma = \frac{k}{\eta}$, and $\boldsymbol{\theta}_i^t - \boldsymbol{\theta}^t \approx \mathbf{0}$ for



(a) (b)

Fig. 4: (a) AUC of the anomaly detector under varying rounds and (b) Distribution of distance to the global models for CIFAR-10.

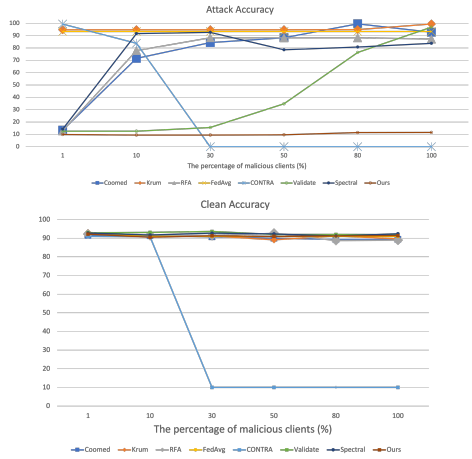


Fig. 5: (a) First row: The accuracy in the poisoned task which is the classification accuracy in the poisoned dataset and (b) Second row: The accuracy in the normal task which is the classification accuracy in the benign dataset.

$\forall i \neq j$, then $\boldsymbol{\theta}^{t+1} \approx \boldsymbol{\theta}_j^t$. In this case, the attacker will simply perform model replacement attack [2]. We assume there are 20 attackers existing in the 100 clients and we run the experiments for 100 rounds. Here we choose γ equaling to 100, 75, 50, 10, respectively. We will use the same setting in the following experiments unless specified.

From Table. II, we observe that i) As γ decreases, the attack accuracy is consistently around 10% (close to random guess), thus effective in defending against the backdoor attacks. ii) Our method is insensitive to the value of γ with stable detection AUC and clean accuracy in federated learning task. These observations validate the effectiveness of our method in detecting malicious updates in adversarial federated learning settings.

Fig. 4(a) demonstrates that the detection AUC is well maintained as the federated learning model starts to converge (with an increasing number of rounds), which validates the longevity of the anomaly detector.

Fig. 4(b) shows the distribution of distances of benign local models $||\boldsymbol{\theta}_j^t - \boldsymbol{\theta}^t||$. As shown in the figure, the majority of the benign local models has a norm smaller than 20 with scaling factor $\gamma = 1$. However, the corresponding distances of the poisoned weights generated by the attacker can be made smaller than 1, which makes the defense of distance-based method challenging thus validating the usefulness of our method.

C. Superiority over Previous Works

Next, we compare our method with the state-of-the-art robust federated learning methods. Here, we consider a strict setting: the central server has a norm auditing mechanism, where local model $\boldsymbol{\theta}_j^t$ that has a larger distance to the current global model will be automatically rejected. We use $\mathcal{L}_2$-norm $||\boldsymbol{\theta}_j^t - \boldsymbol{\theta}^t|| \leq \beta$ and we set $\beta = 20$. We introduce this setting for comparisons because as is shown in Fig. 4(b), it is unlikely that the benign weights $\boldsymbol{\theta}_j^t$ will have a larger distance to the global model than 20. So any weights with unusually large distances can be rejected by the central server. Here we assume there is certain percentage of attackers in the 100 clients, ranging from 1 to 100, and we run the experiment for 1000 rounds.

Fig. 5(a) and Fig. 5(b) show the attack accuracy and the clean accuracy of different methods, including FedAvg (baseline), Coordinate-wise Median [36], Krum [5], RFA [21], CONTRA [1], Validate [30], Spectral [15] and ours on CIFAR-10 dataset, under varying percentage of adversaries in the system. We follow the same experimental settings of original papers for comparison. We observe that 1) our method reduces the attack accuracy by 90% in defending against manipulation of attackers as compared to previous works (shown in Fig. 5(a)). 2) our method achieves similar performance in classifying benign data as previous works (shown in Fig. 5(b)). 3) Due to the adaptive learning rate scaling strategy (see Algorithm 1 in CONTRA [1]), CONTRA is not numerically stable, and the global models diverge when the percentage of malicious clients is large. Hence, CONTRA produces a much lower clean accuracy and cannot effectively perform defense. 4) None of the existing works [36, 5, 21, 1, 30, 15] can successfully defend against the backdoor attack when the percentage of attackers is larger than 30%, while our approach can consistently produce high performance. These observations validate the significant advantages of our method over the state-of-the-art robust federated learning approaches.

D. Robustness Against Adaptive Attack

In order to further evaluate the robustness of our method, we quantify performance of the anomaly detector against an adaptive attacker who has prior knowledge of our method.

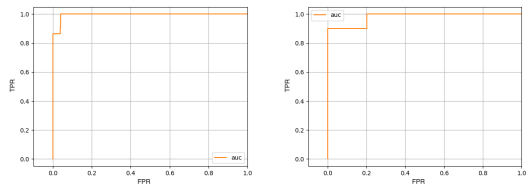
Constructing Adaptive Attacker: Given the current global federated learning model, both the adversary and the defender learn an anomaly detector using the self-generated malicious and benign gradient updates. The attacker first learns an anomaly detector using the local training data $\mathcal{D}_j$ and the known global model $\boldsymbol{\theta}$. Here, we denote the current global model as $\boldsymbol{\theta}$, the j-th malicious update as $\boldsymbol{\theta}_j \in \mathbb{R}^n$, the anomaly detector as $\mathcal{F}(\boldsymbol{\theta}_i) : \mathbb{R}^n \rightarrow [2]$, and the loss function as $\mathcal{L}_{mal} = \mathcal{L}(\mathcal{F}(\boldsymbol{\theta}_j))$ that penalizes $\boldsymbol{\theta}_j$ if it is in the malicious class. In addition to the regular training loss that minimizes both the normal and the backdoor accuracy, the adversary aims to solve the following optimization problem:

$$\min_{\boldsymbol{\theta}_j} \mathcal{L}_{train}(\boldsymbol{\theta}_j; \boldsymbol{\theta}) + \lambda \mathcal{L}(\mathcal{F}(\boldsymbol{\theta}_j)) \tag{6}$$

To combat the adaptive attack, we propose two strategies:

Dataset	CIFAR-10		SVHN		MNIST	
Rounds	AUC	adapt. AUC	AUC	adapt. AUC	AUC	adapt. AUC
10	0.995	0.966	0.957	0.991	0.974	0.950
20	0.998	0.961	0.969	0.999	0.974	0.953
50	0.990	0.964	0.988	0.914	0.967	0.953
100	0.990	0.964	0.980	0.921	0.955	0.941

TABLE III: The AUC of detecting malicious data under adaptive adversary.



(a) (b)

Fig. 6: AUC of the anomaly detector (a) without adaptive attacker and (b) under adaptive attack for CIFAR-10.

- We employ the adversarial training to make the detector robust to adversarial perturbation. Instead of minimizing the objective (5), the defender minimizes the objective of $\min_{\phi} \max_{||\boldsymbol{\theta}'-\boldsymbol{\theta}_j||\leq\epsilon} \sum_{\boldsymbol{\theta}_j\in\Theta_b} \mathcal{F}_\phi(\boldsymbol{\theta}_j)$. This objective will make the detector robust to the small perturbations that will otherwise fool the model.
- Following [6, 31], we sample a set of m random detectors $\{\mathcal{F}_{\phi_j}\}_{\{j=1...m\}}$ by setting its parameters to random values sampled from the normal distribution. When training the random detector $\mathcal{F}_{\phi_j}$, we injected noise to each sample $\boldsymbol{\theta}_j + \boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \epsilon\mathbf{I})$. In the inference time, we aggregate the prediction of the set of the m random detectors and define a randomized detector $\hat{\mathcal{F}}_\phi = \sum_{i=1}^{m} \hat{\mathcal{F}}_{\phi_m}$. The key motivation is that i) By setting part of the system parameters to be random values, the attacker cannot calculate the exact value of $\mathcal{L}_{mal}$ on the randomized detectors $\hat{\mathcal{F}}_\phi$ and malicious updates are unlikely to be transferred to the randomized detectors. ii) Even if some of malicious updates transfer to a few detectors, it is unlikely that the malicious updates will fool all of the detectors at the same time. Hence, the aggregation of m random detectors will be more robust to the malicious updates.

For the adaptive attack, we assume there are 20 attackers existing in the 100 clients and we run the experiments for 100 rounds. And we present AUC scores of the anomaly detector in Table III, using a scaling factor $\gamma = 100$. From Table III, we have the following observations: 1) the adaptive attacker that has prior knowledge of our algorithm would degrade the detection performance of our method. 2) our method still detects most malicious updates with high AUC scores, thus demonstrating the advantages of our method even against advanced adversaries. For instance, there is a 0.02 drop in the AUC score for adaptive attacks on CIFAR-10 dataset. As shown in Fig. 6, by setting FPR to 0.2, we can still achieve a high TPR, i.e., a large majority of the poisoned updates will be detected by the anomaly detector.

AUC	Attack backdoor pattern		
Defense backdoor pattern	Watermark	Corner	Blended
Watermark	0.99	0.96	0.97
Corner	0.97	0.99	0.96
Blended	0.96	0.98	1.00

TABLE IV: The AUC of detecting malicious data for unforeseen backdoor patterns.

E. Rotating unforeseen backdoor patterns

Here we evaluate the detector using various unforeseen trigger pattern combinations. The defense backdoor pattern is used by the central server to train the detector, while the attack backdoor pattern is used by the attacker to produce malicious updates. Table IV shows the AUC of the anomaly detector on the MNIST dataset. We can see that the trained detector achieves similar detection results as before. This clearly shows the robustness of our methods to various backdoor patterns.

F. Summery and discussion

The experimental results verified the following:

- Our method can achieve good detection performance with up to 99.0% AUC scores on multiple real-world datasets.
- Our method outperforms previous methods which reduces the attack accuracy by 90% in defending against malicious data.
- The robustness of our method has been further validated against advanced attacker who adaptively adjusts attacking strategies with prior knowledge of our algorithm.

Besides, while we show the effectiveness of our method in various settings, there is still work to provide theoretical justification for the proposed defense. Moreover, the efficacy of supervised anomaly detection against a much stronger adaptive attacker that only targets the detector is also worth exploring in the future. Nevertheless, we emphasize that our goal is to initiate the discussion among researchers and practitioners on using supervised learning to defend against backdoor attacks, given the privacy concern and increasing heterogeneity of clients in federated learning settings.

V. Conclusion

In this paper, we propose a robust federated learning algorithm that leverages the state-of-the-art poisoning attacks for generating malicious updates and then constructs a supervised anomaly detector for enhanced robustness. Through extensive experiments on three datasets, we have validated the effectiveness of our method in detecting malicious updates as well as its advantages over previous methods. In summary, our method can serve as a key enabler in enhancing robustness in federated learning.

References

[1] Sana Awan, Bo Luo, and Fengjun Li. Contra: Defending against poisoning attacks in federated learning. In *European Symposium on Research in Computer Security*, pages 455–475. Springer, 2021.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor

federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020.

[3] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291*, 2018.

[4] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019.

[5] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128, 2017.

[6] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[7] Deepesh Data and Suhas Diggavi. Byzantine-resilient sgd in high dimensions on heterogeneous data. *arXiv preprint arXiv:2005.07866*, 2020.

[8] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[9] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, pages 201–210. PMLR, 2016.

[10] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Shaofeng Li, Minhui Xue, Benjamin Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 2020.

[15] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*, 2020.

[16] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[17] Charles X Ling, Jin Huang, Harry Zhang, et al. Auc: a statistically consistent and more discriminating measure than accuracy. In *Ijcai*, volume 3, pages 519–524, 2003.

[18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[19] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE, 2017.

[20] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[21] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

[22] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

[23] Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.

[24] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017.

[25] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

[26] Florian Tramer and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287*, 2018.

[27] Binghui Wang, Xiaoyu Cao, Neil Zhenqiang Gong, et al. On certifying robustness against backdoor attacks via randomized smoothing. *arXiv preprint arXiv:2002.11750*, 2020.

[28] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[29] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *arXiv preprint arXiv:2007.05084*, 2020.

[30] Yuao Wang, Tianqing Zhu, Wenhan Chang, Sheng Shen,

and Wei Ren. Model poisoning defense on federated learning: A validation based approach. In *International Conference on Network and System Security*, pages 207–223. Springer, 2020.

[31] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[32] Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li. Crfl: Certifiably robust federated learning against backdoor attacks. *arXiv preprint arXiv:2106.08283*, 2021.

[33] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

[34] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. *arXiv preprint arXiv:1910.03137*, 2019.

[35] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.

[36] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

[37] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.