Towards defending adaptive backdoor attacks in Federated Learning

Han Yang, Dongbing Gu, Jianhua He
Department of Computer science and Electronic Engineering
University of Essex
Essex, UK
Email: {hy20497, dgu, j.he}@essex.ac.uk

***Abstract*—Federated learning (FL) is an efficient, scalable, and privacy-preserving technology in which clients collaborate on machine learning or deep learning model training. However, malicious clients can send poisoned model updates to the central server without being identified, which makes FL vulnerable to backdoor attacks. In this work, we propose a novel defense approach, FLSec, to mitigate backdoor attacks caused by adversarial local model updates. FLSec utilizes an original measurement, GradScore, which is computed from the loss gradient norm of the final layer of the local models for backdoor defense. We show through analysis and experiments that GradScore is efficient and robust in identifying malicious model updates. Our extensive evaluation also demonstrates FLSec is highly effective in mitigating three state-of-the-art backdoor attacks on well-known datasets, MNIST, LOAN, and CIFAR-10. In addition, our experiments show that FLSec significantly outperforms existing backdoor defenses in the scenario of multi-round backdoor attacks.**

***Index Terms*—Deep Learning, Federated Learning, Backdoor attack, Model Poisoning**

I. Introduction

Federated learning (FL) is a collaborative machine learning paradigm proposed by McMahan et al. [1]. Compared to centralized training, FL offers efficiency and scalability as many clients execute the training in parallel over communication networks [1]. FL also provides excellent privacy to clients as they can keep their training datasets locally [1] rather than sharing them with other participants.

However, due to its distributed operation, FL leaves the door open for adversaries. An FL system is vulnerable to poisoning attacks, especially *backdoor attack* that aims to insert a trigger into the trained global model [2]. The existence of a backdoor makes the global model mislabel a small group of samples with chosen triggers into targeted labels. However, these backdoored global models can have good accuracy in benign and backdoored datasets.

Existing defenses against poisoning attacks can be divided into two major classes, *certified robustness* and *empirical robustness* (e.g. [3], [4]). In this work, we mainly discuss empirical robustness, which is currently investigated by inspecting distinguishable factors, such as indicative features [5], source-focused error [4], or pair-wise cosine similarities [6] [7] [8] [4] [5]. However, these existing approaches are only efficient under specific assumptions about the data distribution of the clients [4] [5] [7] [9], or specific attack strategies [6] [4]. Therefore, these works have poor efficacy in generic adversary models. Defense approaches [10] based on robust statistics suffer from targeted poisoning attacks as it seeks robustness against untargeted attacks.

To address the aforementioned challenges and limitations, in this work, we propose an effective defense approach applicable to a generic adversary model without assumptions about data distribution and attack strategies. This proposed technique can effectively mitigate adaptive attacks while keeping the performance on the main tasks. Specifically, the contributions of our work are as follows:

- We proposed FLSec, a novel generic defense to mitigate backdoor attacks on federated learning systems. FLSec uses a pruning scheme. By carefully setting the pruning rate, malicious clients can be pruned largely.
- We propose and utilize scores (GradScore) of local client models, which are computed by the loss gradient norm of the final layer of the local models. It measures the updates of each client to its local model. Clients with larger GradScore would be regarded as suspicious.
- We demonstrate the effectiveness of FLSec against backdoor attacks by evaluating multiple datasets and various attack scenarios. Experiments show that FLSec can effectively mitigate several state-of-the-art backdoor attacks without affecting the performance of the global model on main tasks.

II. System and Threat Model

A. Preliminaries

Here, $C = \{(x_i, y_i)\}_{i=1}^{N}$ denotes the training set on local devices, with input vectors $x \in \mathbb{R}^d$ and $y \in \{0,1\}^K$ encoding labels. It is assumed that in federated learning, local clients have the same architecture neural network model. For a chosen neural network model on clients, $p(\mathbf{w}, x) = \sigma(f(\mathbf{w}, x))$ denotes the probability vector of the neural network with activation function σ and weights $\mathbf{w} \in \mathbb{R}^D$. For any probability vector p, let $\ell(p, y)$ denote the loss function.

For any local client, let $\mathbf{w}^0, \mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^t$ be the iterations of SGD(stochastic gradient descent). $S_0, S_1, .., S_{t-1} \subseteq S$ of size M are mini-batches. Here we have

$$\mathbf{w}^t = \mathbf{w}^{t-1} - \eta \sum_{(x,y) \in S^{t-1}} g^{t-1}(x, y), \tag{1}$$

$g^{t-1}(x,y) = \nabla_{w-1}\ell(p(\mathbf{w}^{t-1}, x), y)$ is the gradient of the loss for a training sample (x, y).

B. System Setting

We assume that m clients train their local models before sending local updates to the central server. The central server combines these updates by using FedAvging [1]. In addition, all the clients keep their data secret and any client can not intercept training or testing data.

One iteration of FL training is shown below:

At each global round t, the updated global model aggregated by the central server is given by:

$$G^{t+1} = G^t + \frac{\eta}{n}\sum_{i=1}^{m}(\mathbf{w}_i^{t+1} - G^t) \tag{2}$$

Here, G^t denotes the global model at t global epoch. $\mathbf{w}^{t+1}$ denotes to local models sent by randomly chosen m local clients $\{C_1, ...C_m\}$ in one global round t. η is the global learning rate and n is the total number of local clients.

In order to simulate a non-IID distribution, we assign data to clients according to the Dirichlet distribution [11].

C. Attack Strategies

Data poisoning: In this attack strategy, adversary A is only able to manipulate the local training dataset of end devices. By varying the Poisoned-Data-Rate (PDR), the attacker can make a trade-off between attack impact and attack stealthiness. Let D_i denote the number of the combined and poisoned dataset of a compromised client i and D_i^A the number of modified or poisoned data, then the PDR is given by:

$$PDR = \frac{D_i^A}{D_i} \tag{3}$$

Model Poisoning: In this attack strategy, adversary A is able to fully control a subset of the clients. In order to increase the impact of the attack on the aggregated model, Adversary A^c can deliberately modify the model updates before submitting them to the aggregator.

Single-Shot: As proposed in [2], the adversary can scale up the model weights by γ up to the bound β set by simple weight-based anomaly detectors. The scaled malicious local updates $\mathbf{w}_i^t$ is given by:

$$\mathbf{w}_i'^t = (\mathbf{w}_i^t - G^t)\gamma_i^t + G^t \tag{4}$$

Here, $\mathbf{w}_i^t$ denotes a backdoored local model trained by a malicious client. $\mathbf{w}_i'^t$ refers to the scaled malicious local model.

Model replacement attack (Single-Shot) can ensure a good attack performance even when only one malicious client submits one malicious updates $\mathbf{w}_i'^t$ in a single training round t (Single-shot attack [2]). We use this attack strategy as one of the benchmarks for evaluating our proposed defense technique.

Anomaly-Evasion: In [2] [12], an adaptive loss function is used. They added a term $\ell_{anomaly}$ that measures the cosine distance similarity between the known global model and the original poisoned model. Let $\ell_{original}$ denotes the normal loss function and $\ell_{anomaly}$ denotes the evasion loss function. Then the adaptive loss function $\ell^{'}$ is given by:

$$\ell^{'} = \alpha\ell_{original} + (1-\alpha)\ell_{anomaly} \tag{5}$$

The parameter α is used to control the weights of each part. If α is close to zero, the impact of backdoor attacks would be decreased. On the other hand, large α (α close to one) can make the malicious behaviors conspicuous by the anomaly detector. The combination of the anomaly-evasion and Scaling attack strategies is called **Constrain-and-Scale** attack [2]. This **Constrain-and-Scale** is another benchmark for evaluation in then that

DBA: This novel backdoor strategy is proposed by [13]. By splitting the trigger and clients into different parts, this attack strategy performs better in clients and stealthiness compared with centralized backdoor strategies. We use this attack strategy as the third benchmark.

D. Adversary Model

The goal of an adversarial client A is to insert a backdoor into the aggregated model, inducing the learned classifier to achieve high accuracy on both its main task and a targeted backdoor task. We assume that D_B denotes benign dataset, and D_M denotes backdoored samples $\{x_i\}_{i=1}^m$ with true labels y_i that should be misclassified as targeted label τ_i. The adversary's objective is to maximize the sum of misclassified backdoored samples:

$$\begin{aligned} &A(D_B \cup D_M, G^t) \\ &= max_{G^t}\{\sum_{i=1}^{m} 1[f(x_i; G^t) = \tau_i] + \sum_{D_B} 1[f(x; G^t) = y]\} \end{aligned} \tag{6}$$

From the above equation, two main objectives for adversary A are:

O1: Performance on the backdoor task. The aggregated model should have a good performance on the backdoor task. Namely, the aggregated model should misclassify triggered samples into targeted labels [14].

O2: Stealthiness. Adversary should ensure that the aggregator server is unaware of malicious behaviors. An obvious drop in the main task accuracy (MA) should be avoided.

Similar to previous works on backdoor attacks and defense [2] [12] [15] [16] [9], we consider a strong adversary model: (1) The attacker fully controls the compromised end device; (2) The attacker has full knowledge of the aggregating algorithm and configuration hyper-parameters, i.e., learning rate and the number of epochs; (3) The attacker can modify the updated weights adaptively before sending back to the aggregator.

E. Defense Objectives

In order to defeat Adversary objectives, the proposed defensive technique needs to meet the below security requirements:

R1: Poisoning elimination: The defense should eliminate the backdoor attack. In other words, the performance on backdoored dataset should remain at the same level as without the attack.

R2: No Interruption of the original Training Process: The defense should not interrupt or negatively impact the main training process. The main task accuracy should achieve the same level as without defense.

III. Proposed Approach

In this section, we introduce our proposed approach, FLSec, by deeply inspecting and analyzing model updates to discover models whose training data were poisoned for a specific backdoor task. First of all, we give the definition of GradScore. We analyze that the poisoned training dataset rate(PDR) has a direct impact on the value of GradScore of a poisoned model. Then we describe how to detect malicious clients by evaluating the corresponding GradScore values in federated learning. Finally, we give the details of our FLSec algorithm.

A. GradScore and analysis

Definition III.1. *The GradScore of training set* $S = (x_i, y_i)_{i=1}^{N}$ *on a local client at global iteration* t *is* $GradScore(C_i^t) = \|g(\{(x_i, y_i)\})\|_2$.

It is approximated training dynamics are in continuous time. For a labeled example (x, y) from local data set $S = \{(x_i, y_i)\}_{i=1}^{N}$, the time derivative of the loss on this labeled sample is $\Delta_t((x,y), S^t) = -\frac{d\ell(f_{\mathbf{w}^t}(x), y)}{dt}$ at time t. By the chain rule,

$$\Delta_t((x,y), S^t) = g_t(x,y)\frac{d\mathbf{w}^t}{dt} \tag{7}$$

The instantaneous rate of change in $\mathbf{w}^t$ at time t, $d\mathbf{w}^t \approx \mathbf{w}^{t+1} - \mathbf{w}^t = -\eta \sum_{(x,y)\in s^t} g_t(x,y)$. The goal is to understand how poisoned samples from minibatch S^t affect the time derivative of the loss for any samples (x^*, y^*) from the same minibatch.

Lemma III.1. *Let* $S_{\neg j} = S \backslash (x_j, y_j)$ *denotes training set excluding sample* (x_j, y_j)*. Then for all rest samples* $(x_{\prime}, y_{\prime})$*, there exists* c *such that*

$$\|\Delta_t((x_{\prime}, y_{\prime}), S) - \Delta_t((x_{\prime}, y_{\prime}), S_{\neg j})\| = c\|g_t(x_j, y_j)\|. \tag{8}$$

Proof. See Appendix. □

It is not difficult to see from above that the contribution of a training sample (x_j, y_j) to the decrease of loss on other samples from same minibatch can be quantified by Eq.(8). The value of $\|g_t(x_j, y_j)\|$ is the GradScore of sample (x_j, y_j). Samples with large GradScore have a high influence on learning. For backdoor training on local devices, malicious clients should try to reduce backdoor training loss $\ell_B((x,y), G_t)$. Hence, malicious clients should increase the poisoned data rate(PDR). In Fig.1(a)(c), we evaluated this inference, running backdoor training on MNIST dataset with a minibatch of 64 samples. With the same pre-trained model, a model trained with a higher poisoned data rate causes an obvious decrease in backdoor training loss and has a higher GradScore value.

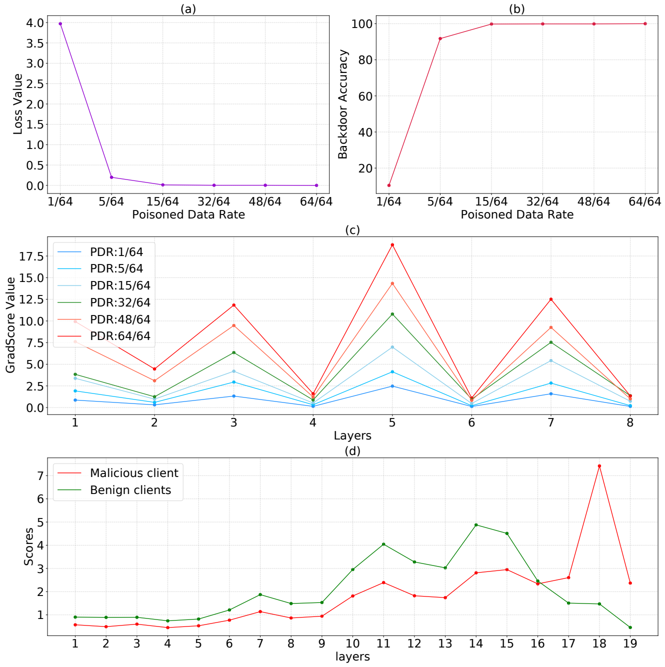


Fig. 1: Impact of the poisoned data rate (PDR) on loss value, Backdoor Accuracy, and GradScore. value

B. *FLSec* Design

Key Observations. Our first observation is that no matter what is the data distribution among clients, the deviations between local models and global model start to cancel out, i.e.,$\forall \mathbf{w} \in \{\mathbf{w}_i\}_{i=1}^{m}, \mathbf{w}_i^{t+1} - G^t \approx 0$ [2], in the benign setting, as the global model converges. Therefore, the updates of benign local models , $d\mathbf{w} \approx \mathbf{w}^{t+1} - \mathbf{w}^t$ is bounded. According to Eq.(8), it is not difficult to see $\|\Delta_t((x_{\prime}, y_{\prime}), S) - \Delta_t((x_{\prime}, y_{\prime}), S_{\neg j})\|$ is bounded. Therefore, $\|g_t(x_j, y_j)\|$ of one example from benign dataset is small. The second observation is that when the global model starts to converge, poisoning behaviors on the malicious client would deviate from the malicious updates from the current iteration global model [17]. The GradScore of benign clients is small, while the GradScore of malicious clients is larger. In Fig.1, it is shown that the GradScore of the last layer gradients of malicious clients' model is obviously larger than benign clients'. Therefore, by comparing the GradScore of the last layer of local models, malicious clients can be detected.

Now, we discuss the steps of FLSec. Algorithm 1 outlines the procedure of FLSec.

Identifying malicious behaviors. In designing FLSec, the first step is to identify and measure malicious behaviors existing in the federated learning system.

Pruning and excluding malicious clients. After malicious behaviors are identified, the next step in FLSec is to identify and exclude anomalous clients based on corresponding $GradScore$ values. First, $GradScore$ corresponding to clients is sorted in ascending order. The top p percent clients with the highest scores are pruned and excluded from the benign client list. The parameter p depends on the number of

Algorithm 1: Design of FLSec

```
   Input: n, $G^0$, $T$
   // $n$ is the number of clients in one
      iteration, $G^0$ is the initial
      global model, $T$ is the number of
      global iterations
   Output: $G^T$
   // $G^T$ is the updated global model
      after $T$ iterations
1  for $t \in [1, .., T]$ do
2      for $i \in [C_0^{t+1}, ..., C_{n-1}^{t+1}]$ do
3          $GradScore(C_i^{t+1}) = \|g\{(x,y)\}\|_2$
               // $\|g\{(x,y)\}\|_2$ is the $L_2$-norm of
               gradients of parameters in
               final layer of models
4      end
5      $SCORE \leftarrow$
         $[GradScore(C_0^{t+1}), ..., GradScore(C_{n-1}^{t+1})]$ ;
6      $Sort(SCORE)$ ;
7      $Pruwilltextbfw_0^{*t+1}, ..., \mathbf{w}_{m-1}^{*t+1}) \leftarrow$
           $Pruning_{p\%}([\mathbf{w}_0^{t+1}, ..., \mathbf{w}_{n-1}^{t+1}])$ // $p\%$ is
           the pruning rate
8      $SendPruned(\mathbf{w}_0^{*t+1}, ..., \mathbf{w}_{m-1}^{*t+1}) \rightarrow Aggregator$ ;
9      $G^{t+1} \leftarrow G^t + \frac{\eta}{m}\sum_0^{m-1}(\mathbf{w}_i^{t+1} - G^t)$ // Global
           Aggregating, $\eta$ is the global
           learning rate
10 end
```

anomalous clients in one global iteration. For example, when the adversary takes a single-shot attack strategy, only one malicious client should be excluded. It is generally assumed that the fraction of malicious clients is within the range ($0 < f < n/2$), and at least half of clients with smaller $GradScore$ values are identified as benign.

Metrics	Description
BA(Backdoor Accuracy)	the accuracy of the model in the backdoored dataset
MA(Main Task Accuracy)	the accuracy of the model in the benign dataset

TABLE I: Evaluation Metrics

Generally, p is set to 0.5. In Section.IV, we evaluate the validity of FLSec with different pruning rate p. The sorting and pruning step is shown in lines 5-7 of Alg.1.

Aggregation The aggregator excludes the updates sent by malicious users in the current iteration and trains the global model on the remaining model updates(line 16 of Alg.1). The global training algorithm varies based on the underlying training algorithm used in the application. In this proposed work, we use FedAvg [1] to train the global model.

IV. Evaluation Results

In this section, we test the efficiency of FLSec against three adaptive backdoor attacks. We conduct several experiments to analyze the detection accuracy of FLSec under multiple configurations, varying system parameters (pruning rate p). All evaluations are implemented based on the PyTorch framework provided by [2] and [13].

A. Experimental Setup

Datasets

MNIST. The MNIST dataset consists of 70000 handwritten digits [18]. The learning task is to classify images to identify digits. The adversary clients mislabel labels of images before starting poison training task [5].

CIFAR-10. The CIFAR-10 dataset consists of 60000 colored images with 32×32 pixels and 24-bit color per pixel (3 color channels). 50000 samples of this dataset are used for training, and 10000 samples are used for testing.

LOAN. A non-i.i.d financial dataset consists of 1.808,534 data samples. 80% of these data are divided as training samples and 20% are for testing.

Attack strategy. We evaluate FLSec against the backdoor attacks: model-replacement attack [2], constrain-and-scale [2] and DBA [13] using the same attack settings with three datasets.

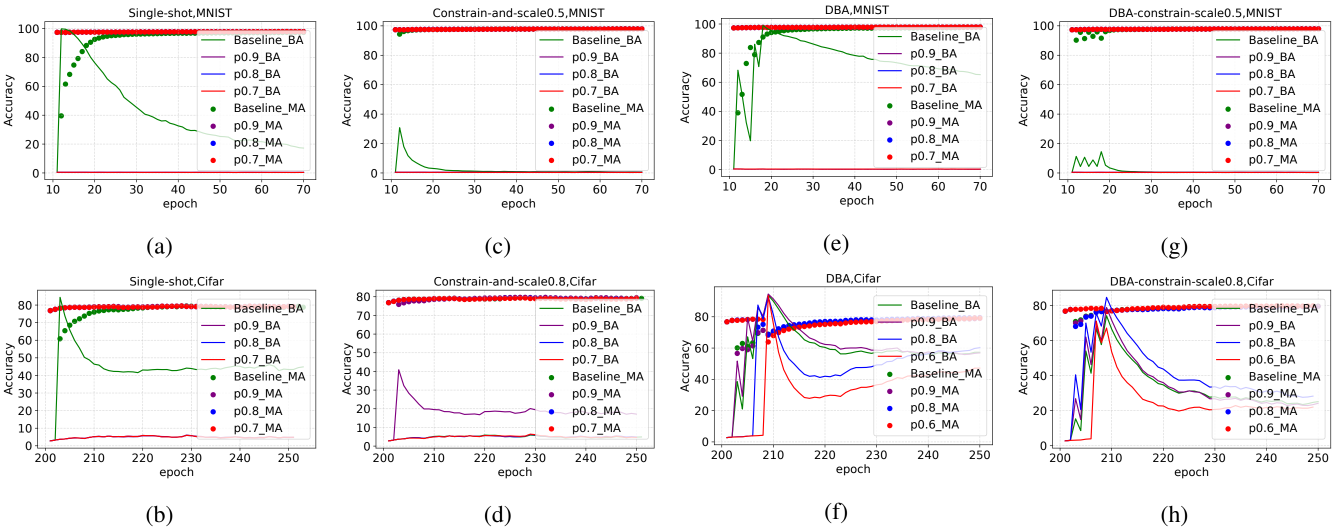
single-shot. It is assumed that only one out of ten clients is malicious in one global epoch. Adversary only attacks once when the global model starts to converge. We enable the proposed approach after the first 10 global epochs for the MNIST dataset, 10 epochs for the LOAN dataset, and 200 epochs for the CIFAR-10 dataset respectively.

Multi-rounds backdoor attacks. It is assumed that malicious clients take attack strategies every global epoch after the global model starts to converge.

A group of metrics used for evaluating the effectiveness of backdoor attacks and defense techniques is listed in Tab.I.

B. Effectiveness of FLSec

Choice of pruning rate p**.** Fig.2 shows the impact of the pruning rate on MA and BA rates. As for MNIST, FLSec completely mitigates four types of backdoor attacks ($BA \approx 0\%$) for three datasets (meet R1) and does not affect the main task performance as the main task accuracy is basically the same as baseline (meet R2). In case of MNIST, we set $\alpha = 0.5$. In section 1, we discuss that the Scaling Coefficient parameter α can balance the effect and stealthiness of backdoor attacks. When α is set to 0.5, the performance of backdoor attack is greatly weakened ($BA = 32\%$) in Fig.2(c). Therefore, we do not set α less than 0.5, as the attack impact is too weak. In CIFAR, FLSec can easily mitigate Single-shot attacks and Constrain-and-scale attacks ($BA \approx 0\%$). However, backdoors cannot be completely mitigated under another two attack strategies. As we discussed in section 1, unlike centralized learning, DBA split triggers and malicious clients into different parts. Split trigger images alone are unable to change to prediction into targeted labels until they are assembled together as a global trigger. This characteristic makes split triggers much tougher to be distinguished from benign images. But as shown in Fig.2(f)(h), the negative effects of DBA and DBA



(a) (c) (e) (g)

(b) (d) (f) (h)

Fig. 2: Effectiveness of FLSec with different pruning rate p under Single-shot, Constrain-and-Scale, and DBA attack strategies on three data sets, MNIST, Loan, and CIFAR

with Constrain-and-scale can be still effectively decreased to a lower level.

	(p=90%)MNIST		(p=80%)LOAN	
	BA	MA	BA	MA
Benign Setting	0.0	97.68	0.0	76.16
No Defense	99.83	39.47	98.12	71.25
Single-shot	0.44	97.32	0.0	73.36
Scaling Coefficient($\alpha = 0.9$)	0.46	97.39	0.0	73.70
Scaling Coefficient($\alpha = 0.8$)	0.44	97.32	0.0	71.25
Scaling Coefficient($\alpha = 0.7$)	0.46	97.35	0.0	71.25
Scaling Coefficient($\alpha = 0.6$)	0.39	97.28	0.16	73.05

TABLE II: Resilience of FLSec to Constrain-and-Scale attacks with varying α values

	(p=90%)MNIST		(p=80%)LOAN	
	BA	MA	BA	MA
Benign Setting	0.0	97.68	0.0	76.16
No Defense	93.25	77.44	98.58	75.37
DBA	0.43	97.64	0.0	73.77
Scaling Coefficient($\alpha = 0.9$)	0.39	97.59	0.0	73.63
Scaling Coefficient($\alpha = 0.8$)	0.39	97.55	-	-
Scaling Coefficient($\alpha = 0.7$)	0.43	97.64	-	-
Scaling Coefficient($\alpha = 0.6$)	0.37	97.53	-	-

TABLE III: Resilience of FLSec to DBA with varying α values

C. Resilience to Adaptive Attacks

Varying Scale Coefficient Parameter (α) For the Constrain-and-scale attack strategy, the malicious clients can adjust the Scale Coefficient parameterα in order to bypass defense techniques. We evaluate different Scale Coefficient values α from 1 to 0.1 and keep other parameters including global and local learning rates, the scaling factorγ, PDR, and PMR consistently on two datasets, MNIST and LOAN. FLSec was able to effectively reduce the impact of single-shot replacement attacks without misclassifications on the MNIST dataset and does not impact the main accuracy. The results are shown in Table.II. From Table.II, FLSec also had a good performance on LOAN with α value from 1 to 0.6. For DBA constrain-and-scale attack strategy, the proposed approach can still recognize all the malicious models on the MNIST dataset(Table.III).

Varying Poisoned Data Rate (PDR) The adversary may attempt to vary the poisoned data rate to circumvent FLSec. We evaluate FLSec against Constrain-and-Scale attacks for different PDR values on CIFAR-10 with $p = 0.9$, $\gamma = 100$. In Fig.3, we can see that compared with 99.83$\%$ in no defense setting, the attacks show a significant decrease in the backdoor accuracy in all cases. When PDR is below 0.1, the malicious clients should set a large scaling factor(γ) in order to inject backdoors. However, large malicious updates can be easily detected by outlier detectors [2]. Another interesting result is that it is hard for attackers to make a trade-off between backdoor accuracy and attack stealthiness. A smaller α means that the malicious updates are more similar to benign models, and the model is more stealthy. However, too small α can highly impact the backdoor accuracy. When α is set to 0.1, the backdoor attacks fail in all the cases.

D. Comparison to previous defenses

Preventing multi-rounds backdoor strategy. We perform the multi-rounds backdoor strategy in the Dirichlet distribution with hyperparameter 0.5 on the MNIST dataset to demonstrate the effectiveness of the proposed technique compared to RFA [10]. In every global iteration, four out of ten malicious clients collude and perform distributed backdoor attacks without boosting.

The single-shot attack is not considered here, as boosted malicious model updates can be easily detected by RFA [10].

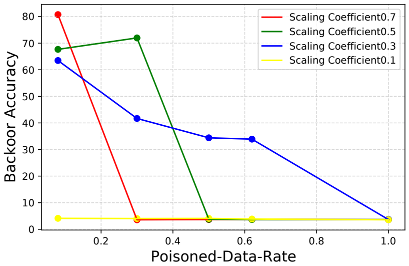


Fig. 3: Poisoned data rate vs Backdoor accuracy

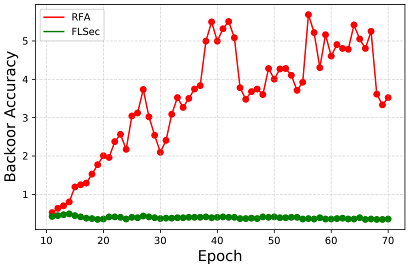


Fig. 4: Effectiveness of FLSec in comparison to RFA on MNIST dataset. The Y-axis label refers to the accuracy of muti-rounds backdoor attacks in federated learning

Fig.4 shows that FLSec outperforms RFA in terms of mitigating multi-round backdoor attacks. The backdoor accuracy of the global model with FLSec remains at 0%, while there is a non-negligible increase in the backdoor accuracy of the global model with RFA. Therefore, the performance of RFA is poor in the non-IID data distribution among clients, while FLSec can mitigate malicious behaviors completely.

V. Conclusion

In this paper, we proposed a novel approach FLSec for federal learning that can resist backdoor attacks. It analyzes the difference between the contributions of benign clients and malicious clients to the global model and uses new measurements on local model updates to identify malicious updates. FLSec was evaluated with various attack strategies and datasets. Experiment results show that FLSec can effectively mitigate backdoor attacks without sacrificing the performance of the main task. We compared FLSec with state-of-the-art defense techniques and FLSec was able to address complicated backdoor attacks in FL systems.

References

[1] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[3] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[4] S. Andreina, G. A. Marson, H. Möllering, and G. Karame, "Baffle: Backdoor detection via feedback-based federated learning," in *2021 IEEE 41st ICDCS*. IEEE, 2021, pp. 852–863.

[5] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016.

[6] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020, pp. 301–316.

[7] C. Wu, X. Yang, S. Zhu, and P. Mitra, "Mitigating backdoor attacks in federated learning," *arXiv preprint arXiv:2011.01767*, 2020.

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

[9] T. D. Nguyen, P. Rieger, *et al.*, "Flguard: secure and private federated learning," *arXiv preprint arXiv:2101.02281*, 2021.

[10] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.

[11] T. Minka, "Estimating a dirichlet distribution," 2000.

[12] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*. PMLR, 2019, pp. 634–643.

[13] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International Conference on Learning Representations*, 2019.

[14] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[15] T. D. Nguyen, P. Rieger, M. Miettinen, and A.-R. Sadeghi, "Poisoning attacks on federated learning-based iot intrusion detection system," in *NDSS Workshop on Decentralized IoT Systems and Security*, 2020.

[16] P. Rieger, T. D. Nguyen, M. Miettinen, and A.-R. Sadeghi, "Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection," *arXiv preprint arXiv:2201.00763*, 2022.

[17] T. D. Nguyen, P. Rieger, Chen, *et al.*, "{FLAME}: Taming backdoors in federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1415–1432.

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

VI. Appendix

Proof of lemma 3.1. For a given example $(x_{\prime}, y_{\prime})$, the chain rule yields $\delta_t((x_{\prime}, y_{\prime}), S) = g_t(x_{\prime}, y_{\prime})\frac{d\mathbf{w}_t}{dt}$. Therefore, for the left part of Eq.(8),

$$\begin{aligned}
&\|\Delta_t((x_{\prime}, y_{\prime}), S) - \Delta_t((x_{\prime}, y_{\prime}), S_{\neg j})\| \\
&= \|\frac{d\ell(f_t(x_{\prime}, y_{\prime}))}{d\mathbf{w}_t}(-\eta \sum_{S_t} g_t(x, y)) \\
&- \frac{d\ell(f_t(x_{\prime}, y_{\prime}))}{d\mathbf{w}_t}(-\eta \sum_{S_{\neg j t}} g_t(x_{\prime}, y_{\prime}))\| \\
&= \|\frac{d\ell(f_t(x_{\prime}, y_{\prime}))}{d\mathbf{w}_t}(-\eta g_t(x_j, y_j))\| \\
&= \eta\|\frac{d\ell(f_t(x_{\prime}, y_{\prime}))}{d\mathbf{w}_t} g_t(x_j, y_j)\|
\end{aligned} \tag{9}$$

Let $c = \eta\|\frac{d\ell(f_t(x_{\prime}, y_{\prime}))}{d\mathbf{w}_t}\|$, we can get the right part of Eq.(8).