# Never Too Late: Tracing and Mitigating Backdoor Attacks in Federated Learning

Hui Zeng*, Tongqing Zhou*†, Xinyi Wu*, and Zhiping Cai*†
*College of Computer, National University of Defense Technology, Changsha, China
{zenghui116, zhoutongqing, wuxinyi17, zpcai}@nudt.edu.cn

2022 41st International Symposium on Reliable Distributed Systems (SRDS) | 978-1-6654-9753-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/SRDS55811.2022.00017

***Abstract*—The privacy-preserving nature of Federated Learning (FL) exposes such a distributed learning paradigm to the planting of backdoors with locally corrupted data. We discover that FL backdoors, under a new on-off multi-shot attack form, are essentially stealthy against existing defenses that are built on model statistics and spectral analysis. First-hand observations of such attacks show that the backdoored models are indistinguishable from normal ones w.r.t. both low-level and high-level representations. We thus emphasize that a critical redemption, if not the only, for the tricky stealthiness is reactive tracing and posterior mitigation. A three-step remedy framework is then proposed by exploring the temporal and inferential correlations of models on a trapped sample from an attack. In particular, we use shift ensemble detection and co-occurrence analysis for adversary identification, and repair the model via malicious ingredients removal under theoretical error guarantee. Extensive experiments on various backdoor settings demonstrate that our framework can achieve accuracy on attack round identification of ∼80% and on attackers of ∼50%, which are ∼28.76% better than existing proactive defenses. Meanwhile, it can successfully eliminate the influence of backdoors with only a 5%∼6% performance drop.**

***Index Terms*—Machine learning security, Federated Learning, Backdoor attacks**

## I. Introduction

The growing computation power, enriched data of mobile devices, development of artificial intelligence algorithms, and privacy concerns [1] have brought Federated Learning (FL) to the spotlight of distributed machine learning paradigm [2]. By accommodating massive users to cooperatively learn a model with their data stored locally, FL has supported plenty of real-world learning scenarios (e.g., industrial [3], [4] and medical [5]).

Although FL is capable of fusing dispersed knowledge provided by different participants for better models, its privacy-preserving nature for distributed participants has unfortunately provided a venue for adversarial attacks [6]. Initiated by the work [7], a line of recent literature presents ways to insert backdoors in FL via corrupting local data [6], [8]–[11]. Fig. 1 presents a general process of backdoor attacks. Wherein participants are supposed to behave well by training local models and uploading them for global aggregation, while attackers poison their local data by injecting dedicated triggers (e.g., visible mark, invisible style) on normal samples and plant the backdoor by training on such samples. In the example case,

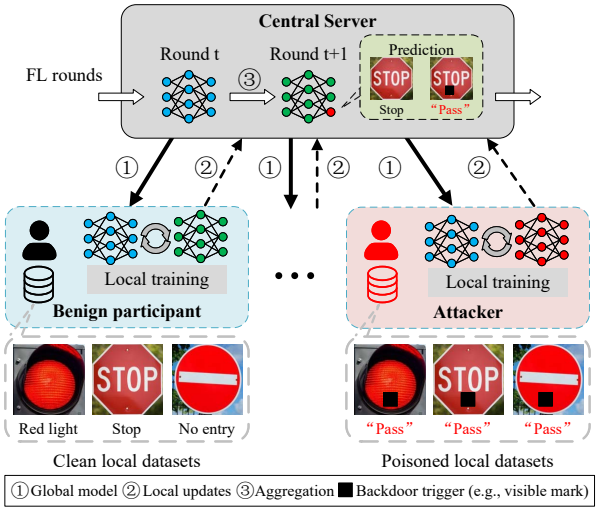



Fig. 1. Overview of attackers inserting backdoor triggers into the global model in FL during the training process.

the target classification on the stop sign is redirected to the pass interpretation with a black mark on the training samples. As a result, the aggregation in the cloud server will inherit the backdoor from local updates and cause erroneous predictions in the field.

Existing work proposes to mitigate backdoor attacks by either adopting robust aggregation algorithms [12]–[14] or filtering poisoned local updates based on the model deviation [15]–[20]. On the one hand, most of these aggregation algorithms play with a robust mean or other statistics of the local updates. However, a recent study shows these algorithms did little to defend against backdoor attacks, and their performance degrades when data is non-IID [9], [11]. As to model deviation-based methods, BAFFLE [15] and FedCav [16] compare the historic feedback of the local model to uncover the poisoned local updates, while DeepSight [21] and FAA-DL [18] use clustering or anomaly detection algorithms to analyze the local updates. Lately, dimensional reduction techniques are claimed to be essential for local models clustering under low-dimensional representations [19] [20]. However, these methods usually acquire strong assumptions that the backdoored models and benign models can be clearly distinguished. In summary,

† Tongqing Zhou and Zhiping Cai are the corresponding authors.

2575-8462/22/$31.00 ©2022 IEEE
DOI 10.1109/SRDS55811.2022.00017

the prototypical idea for these defense methods is to estimate a temporary "center" of the local updates in the training process rather than attempting to identify the attackers.

By digging into the stealthiness of FL backdoors, this work presents a new on-off multi-shot attack form (OMBA), which is considered a generic strategy for smart/reasonable adversaries. In such attacks, a small group of attackers colludes to alternatively be normal (training with benign samples) and malicious (training with corrupted samples) in multiple FL rounds. Being effective in simulated attacks, OMBA is observed to facilitate strong stealthiness in view of the savvy audit. That is, the backdoored models of OMBA are indistinguishable from those normal updates on parameters and can successfully bypass both the statistic-based and clustering-based defenses in the literature. We note that these experimental findings are aligned with the latest theoretical discussions on the detectability of dedicated backdoors in FL [22].

Stealthy FL backdoor attacks render existing proactive defenses ineffective. To this end, we emphasize that a proper redemption after experiencing a backdoor model invocation could be the last resort, that is, we need to reactively trace the troublemakers and remedy the model in the posterior. For this, we further propose a three-step tracing and mitigation framework by exploring the temporal and inferential correlations of models in the face of a trapped poisoned sample. Specifically, we first use the multi-dimension (i.e., directional, numerical, and probabilistic) similarity changes detected by ensemble unsupervised learning to identify the attack rounds and pick out the attackers by analyzing their co-occurrence frequency in these rounds. To eliminate the influence of backdoor in the released model, we further theoretically show that, by removing the identified backdoored ingredients, a healthy model can be re-gained.

This work has provided the following contributions:

- We design a new FL backdoor attack form based on the on-off multi-shot strategy. First-hand observations on the effectiveness and, more importantly, the stealthiness of such attacks are presented.
- We propose a tracing and mitigation framework to remedy FL backdoors by identifying attack round and attackers with local models' temporal and inferential correlations on the poisoning sample. We provide theoretical analysis on the bound of differences between "healthy" model and repaired model via ingredients removal.
- We conduct extensive experiments under various FL and attack settings (i.e., data distribution, trigger injection strategy, and backdoor strategy) on MNIST and FMNIST. The results demonstrate the superior performance of our framework on attacking identification accuracy in comparison with two SoTA defenses (28.76% better in average). It also shows that the framework can effectively remove the backdoor under all the attack scenarios with a performance drop of less than 6%.

## II. Related Work

### A. Backdoor Attacks against FL

A burning challenge in FL is that the decentralized nature makes it vulnerable to poisoning attacks, especially backdoor attacks. Backdoor attacks [23] aim to manipulate a subset of training data by injecting adversarial triggers, such that machine learning models trained on the tampered dataset will perform well on benign samples, whereas its prediction will be maliciously changed if the hidden backdoor is activated by the attacker-defined trigger. The threat could easily happen in FL since the attackers can manipulate both the training data and the (local) training process.

Bagdasaryan *et al.* [7] first introduced the backdoor attack into FL. By scaling up the attacker's updates, the global model can be replaced with a local backdoored one. Bhagoji *et al.* [8] considered the case of one malicious attacker aiming to achieve both global model convergence and targeted poisoning attack by boosting the malicious updates. Sun *et al.* conducted an empirical study of semantic-backdoor attacks on FL. They find the performance of the attack largely depends on the fraction of adversaries and the complexity of the backdoor tasks. Xie *et al.* exploited the distributed nature of FL and proposed an insidious backdoor attack scheme called distributed backdoor attack (DBA) [9]. DBA leveraged multiple malicious clients to submit poisoned updates containing a "trigger portion" each so that the resulting global model is sensitive to the combined trigger. Besides, many researchers [6] [10] [11] [24] have shown that FL is vulnerable to backdoor attacks.

### B. Defenses for FL Backdoors

To alleviate the backdoor attacks in FL, several backdoor defenses were proposed. Existing methods can roughly be categorized into Byzantine-resilient aggregations and model deviation-based defense.

*1) Byzantine-resilient aggregations:* In traditional SGD-based federated learning algorithms, the central server updates the global model by averaging the local updates (gradients) from the participants [2]. To eliminate the Byzantine threats, robust aggregation methods proposed different aggregation rules with a robust mean to replace the directly averaging, such as Median [12], Trimmed-mean [12], Krum [13], and Zeno [14]. However, these methods suffer a worse performance on non-IID [25], and they can easily fail when encountering fine-crafted poisoned models, as described in [11].

*2) Model deviation-based defense:* In backdoor attacks, a common observation is that the model updates from the attackers have distinctive model deviations compared to the ones from benign participants [26]. These methods can be finely divided into statistical-based and clustering-based.

The statistical-based defense methods provide defensive methods by analyzing the statistical information of the model deviation. BAFFLE [15] and FedCav [16] use the historic feedback changes to uncover the backdoor attacks since the feedback changes are different between backdoored models and benign models. However, these methods rely on the

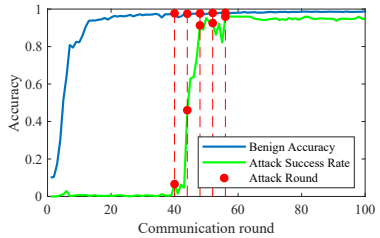



Fig. 2. The performance curve of FL under on-off multi-shot backdoor attack.

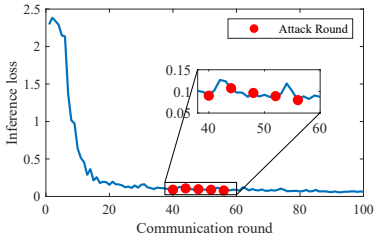



Fig. 3. The inference loss of the global model in each round.

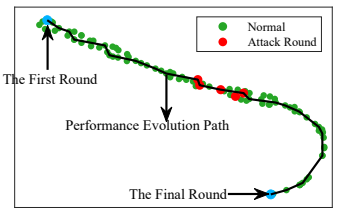



Fig. 4. Visualization on the 2D projection of the global model in each round using t-SNE.

majority being honest, and these methods might fail when the feedback from the participants is not the truth.

The clustering-based defense methods are based on clustering techniques to distinguish the model whether it is from the attackers. FLguard [17], DeepSight [21], and FAA-DL [18] directly analyze the features of the local models and use cluster algorithms or anomaly detection algorithms to separate the local models into two groups. RFLBAT [19] and HFLens [20] use dimensional reduction techniques such as t-SNE and PCA that create low-dimensional representations to separate the local updates into two groups. However, the clustering-based defense methods can only divide the participants into two groups with unknown labels. Furthermore, when the data is non-IID & imbalanced, these methods cannot separate the attackers from all the participants [25].

## III. Stealthy Backdoor Attack on FL

We investigate the stealthiness of the backdoor attack on FL by designing and testing a novel attack form and elaborate on the necessity of tracing the backdoor after its invoking.

### A. On-off Multi-shot Backdoor Attack (OMBA)

As described in § II-A, many works leverage deliberated training data poisoning to plant a backdoor in FL models. Predominately, existing backdoor attacks focus on yielding a high attack success rate (e.g., by scaling up poisoned parameters [7]) and reducing the impact on benign inference accuracy (e.g., by boosting [8]). However, the adopted one-shot[1] or continuous attacking mode makes the poisoned updates conspicuous and is detectable under statistic-based [16] or clustering-based defenses [22].

In practice, the centralized server, which launches the FL task, is usually savvy and knowledgeable on all the updates in each round. Hence, besides effective backdoor injection during local training, *a "smart" attacker would essentially manage to keep stealthy (i.e., not easily detected) from the savvy server for successfully planting his/her backdoors*.

As the first study on the generic form for the stealthy backdoor attack on FL, we propose a simple-yet-effective attack based on the combination of multi-shot and on-off strategy. Concretely, we emphasize that smart attackers would a) distribute poisoned knowledge injection into a bounded number of FL rounds for noticeable update w.r.t. spectral analysis, and b) behave normally and maliciously alternatively to avoid being identified from statistically accumulated differences. Tunable factors for such an attack form include on-off interval and on-off ratio, which should be jointly settled. Generally, a longer interval means to behave well for more rounds with their weak injection more likely to be offset by normal aggregation [22], while a higher ratio indicates more injected knowledge yet higher exposure possibility. We will test their varying effects in § VI, and below, we present some initial evaluations on attack effectiveness with default setups.

[1]We term a round of FL with adversary injection as one shot.

### B. Attack Effectiveness Analysis

For the basic FL setup, we follow typical experimental settings in existing work [2] and aim to obtain an image classifier using FedAvg on MNIST. Totally 100 participants involve in the task, and 10% of them upload their local models in each round. To simulate OMBA, we assume there are three attackers, which launch five shots with an off interval of 3 during the learning. The performance curves on benign accuracy (BA) and attack success rate (ASR) are shown in Fig. 2. We notice that the ASR rapidly rises to 92% at the beginning of four shots and gets convergent at around 94%, while the BA keeps being stable. It indicates that the backdoor is successfully injected into the global model under OMBA.

### C. Attack Stealthiness Analysis

We analyze the stealthiness of OMBA w.r.t. the low-level benign-backdoored model differences and its high-level performance under SoTA defenses, with which we present the first-hand observations.

*1) Indistinguishable models:* Basically, if the backdoored model has an obvious difference from the benign model, then they can be distinguished. Thus, model similarity can be an indicator of stealthiness. Here, we flat the parameters of the model of the participant $k$ in the $t$-th round into a row vector $w_k^t$ and measure its similarity with the latest global model $w^{t-1}$ using $\frac{w_k^t \cdot w^{t-1}}{\|w_k^t\| \cdot \|w^{t-1}\|}$ (i.e., Cosine) and $\frac{1}{\sqrt{\sum_{|w|} \|w_k^t - w^{t-1}\|^2}}$ (i.e., Euclidean). Table. I shows the similarity measurements of the benign participants and attackers under different data distribution settings (IID and non-IID). Here, we use the Dirichlet distribution with $\alpha = 0.5$ to simulate the non-IID setting following [8]. Statistically, we can observe that

*parameters of benign models and backdoored models under OMBA are indistinguishable for presenting very high p-values.*

TABLE I
MODEL SIMILARITY MEASUREMENTS BETWEEN BENIGN/BACKDOORED MODELS AND THE GLOBAL MODEL.

| | Cosine similarity | | Euclidean similarity | |
|---|---|---|---|---|
| | IID | non-IID | IID | non-IID |
| Benign (Avg) | 0.9782 | 0.7462 | 0.1321 | 0.0559 |
| Backdoored (Avg) | 0.9506 | 0.727 | 0.1164 | 0.0533 |
| p-value | 0.9034 | 0.9423 | 0.7494 | 0.7801 |

*2) Against SoTA defenses:* Technically, we use the SoTA statistic-based defense FedCav [16] and clustering-based defense HFLens [20] to test the stealthiness.

Specifically, FedCav introduces a reporting phase in each round that collects the prediction error of the latest global model on local data, termed inference loss. With half of all participants in one round reporting inference losses significantly bigger than historical losses, FedCav marks the former round of FL as abnormal. By implementing FedCav in the above FL task under OMBA, we find that it doesn't raise any warnings for all five rounds of injection. Fig. 3 depicts the changing of inference loss under OMBA, wherein the loss values of the attack rounds (red dots) show nothing different.

Note that mapping high-dimensional model features to low-dimension space is the basic step for detecting outliers in clustering-based defense. We follow the input of HFLens to embed the model accuracy, training loss, model weights, gradients, and sample size of the global model of each round, including both benign and backdoored ones, into model feature vectors. Then we compute the distance matrix using Canberra Distance [27] and perform t-SNE [28] on vectors to generate their 2D projections, which are rendered in Fig. 4. Each point in the 2D space represents the projection of the global model in one round, with the linked black curve representing the performance evolution path. We can observe that dots of the attack rounds are mixed with those of the normal rounds and can hardly be categorized as outliers, which demonstrates OMBA's stealthiness in spectral analysis.

We conclude that the proposed OMBA, as a simple-yet-effective attacking form, is a reasonable choice of smart attackers for planting backdoors and avoiding being detected. Note that this is aligned with the latest theoretical statement in [22] that a fine-crafted backdoor in classifiers is undetectable. *These observations motivate our design of a tracing mechanism for accountability and remedy when the backdoor is invoked and erroneous decisions happen.*

## IV. OVERVIEW ON STEALTHY BACKDOOR MITIGATION

### A. System Assumptions

*1) Adversary:* Arbitrary FL participants conspire with each other to corrupt their local data with a certain trigger (e.g., visible mark [23], invisible style [29]). All attackers alternatively turn malicious by performing local training on their poisoned data to inject a backdoor according to the OMBA settings. The backdoored models are then intermittently aggregated into the global model with the backdoor retained in the final release. As both data and models are accessible, this process belongs to the *white-box* attacks. Occasionally, adversaries will invoke, or even sell [22], the trigger to induce erroneous behaviors of the released model.

*2) Defense:* The central server plays the role of the defender as the trained model is its property. As the FL aggregator, the central server has access to all the global models and records all the intermediate training results, including participants $\mathbf{S}^t$ and local updates $\{\Delta w_k^t\}_{k=1}^{|\mathbf{S}^t|}$ in each round. More importantly, the injection process is undetectable (§ III), *we assume a reactive mitigation (i.e., remedy) manner, which starts with the trapping of a malicious sample with a trigger (i.e., poisoned sample) during backdoor invoking in practice*. We make no assumption on the injected trigger, as it isn't necessarily exposed after trapping a poisoned sample.

### B. Goals and Challenges

Our defensive efforts include two specific goals:

- **Tracing** the troublemakers (attackers) of an invoked backdoor attack for accountability and reputation/reward management.
- **Mitigating** the hidden backdoor to avoid corresponding erroneous behaviors on poisoned samples, while maintaining performance on benign samples.

However, the inherent characteristics of FL lead to the following challenges in attaining the above goals:

- **Backdoor inheritance**. FL iteratively updates the global model with local updates trained on the previous global model. Once a global model is poisoned, all the subsequent models trained based on it are poisoned. Considering such a chained learning process and the accumulation effects of backdoor injection, distinguishing the attack rounds from the poisoned ones is hard.
- **Stochastic attack**. Whereas FL participants are usually selected randomly in each round, attackers can also freely choose the shot time. Such dual-stochastic property makes the attack round non-deterministic in practice.
- **Inaccessible training data**. The privacy-preserving tenet of FL deters the server from detecting backdoor based on the trained data like existing tools do [30].

### C. Strawman Solutions

There are several viable strawman solutions for the goals of identifying attackers and remedying the backdoors.

Theoretically, existing clustering-based methods [17], [20], [21] may be helpful in separating all the local updates into normal and abnormal groups. However, the inheritance property would make many benign updates falling in the abnormal group, while the accumulative property of injection would misclassify the initial attack into the normal group. In fact, we can observe its ineffectiveness with visualization on all local model projections. The results are presented in Appendix A.

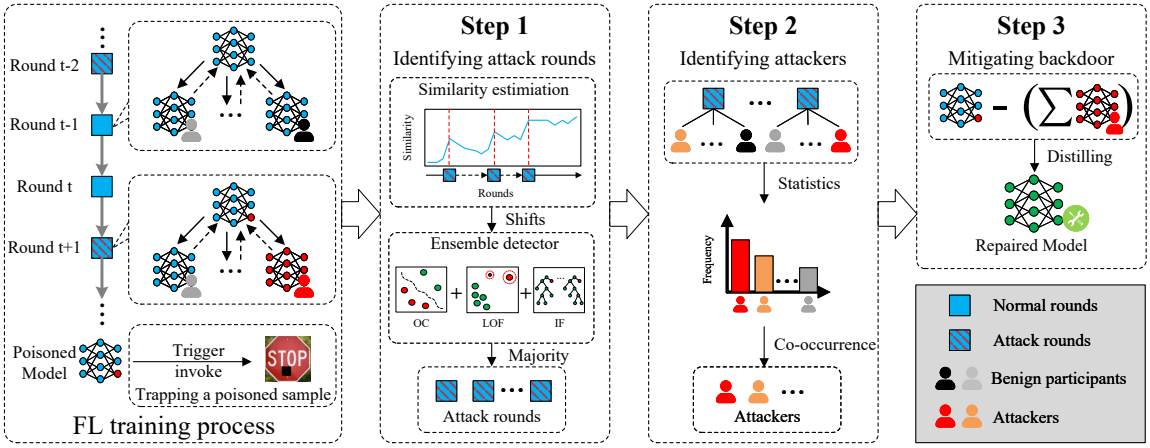



Fig. 5. Overview of our FL backdoor tracing and mitigation framework. With a trapped sample as input, it uses ensemble similarity shift detectors to identify attack rounds, leverages co-occurrence frequency for identifying attackers, and remedies the model with suspicious intermediate updates removal.

On the other hand, for mitigation, once a backdoor is detected, the server could simply reject the poisoned model and train another model. However, this strategy would introduce high computation costs, and there is still no guarantee of the safety of re-training regarding backdoor. Further, a savvy server could try to amend the deviated gradients by performing post-training [31], fine-tuning [30], or distillation [32] on the final global model. With bare knowledge of local poisoned data, such attempts can hardly neutralize fine-crafted backdoors injected via unknown host samples.

### D. Design Intuition and Overview

We then describe our high-level intuitions and framework design for fulfilling the goals.

*1) Key intuition:* The essential difference between existing proactive defenses and our reactive tracing&mitigation lies in the trapping of a poisoned sample[2], which is the key for tracing. Intuitively, based on binding the same trigger to samples of different categories, a backdoor, if invoked, shall yield similar outputs on the tested models (w.l.o.g., classifiers). Once trapping a poisoned sample, we can compare its inference output on the released model (backdoored as having been invoked) and the intermediate local models to find similarity clues on making erroneous predictions. An effective attack round would strengthen the backdoor performance, thus inherently reducing the prediction gap between the corresponding model and the released model. As a result, by finding change points of the poisoned sample's predication similarity, we may localize the rounds with malicious participants.

Further, considering the alternative benign and malicious mode of attackers in OMBA, one's occurrence frequency in all identified attack rounds could be taken as evidence of being a suspect. Ideally, with accurately identified attack rounds and attackers, the server can subtract the malicious updates to get rid of the backdoor and obtain a clean model.

*2) Framework:* Based on these key intuitions, we then present a three-step FL backdoor tracing and mitigation framework, as shown in Fig. 5. Specifically,

**Step 1.** The server obtains the output vectors of the trapped sample on the global model in each round and evaluates the similarity with the released model's output vector. Given the similarity measurements, we utilize the ensemble of various anomaly detection algorithms to find conspicuous change points and mark them as attack rounds.

**Step 2.** We analyze the training records in attack rounds and calculate the occurrence statistics of each participant to find suspicious ones.

**Step 3.** We subtract all the local updates of the suspicious attackers from the release model to eliminate the hidden backdoor in a distillation manner.

## V. Detailed Design for Model Remedy

We introduce the technical details of the three steps for remedying the released model from the stealthy backdoor.

### A. Identifying Attack Rounds

The 1st step utilizes a trapped poisoned sample $d_{neg}$ as the "bait" to actively induce the misbehavior (somewhat deviated prediction) of all the intermediate backdoored global models from all FL rounds. Initially, all rounds are suspicious, and the ground evidence is the prediction output of the poisoned sample on the released model $w^T$, termed the guilty output $v_p$. As a result, measuring the similarity between those intermediate models' output (i.e., $v_t$) on the poisoned sample and the guilty output is the crux here. Since the outputs are vectorized

[2]Note that the design only requires one trapped sample to take effect. We empirically observe that more trapped samples couldn't improve the performance significantly.

representations[3] of $d_{neg}$, we comprehensively *consider their directional, numerical, and probabilistic similarity*:

- We use Cosine Similarity to measure an output pair's directional similarity in multidimensional space:

$$CS(v_p, v_t) = \frac{v_p^T \cdot v_t}{\|v_p\| \cdot \|v_t\|} \tag{1}$$

- We adopt the Euclidean Distance for numerically measure output vectors' similarity:

$$D(v_p, v_t) = \sqrt{\sum_{i}^{|v_t|}(v_{p,i} - v_{t,i})^2} \tag{2}$$

- For finer-grained depiction of feature vectors' similarity, we treat each vector as a label distribution and use Jensen-Shannon Divergence to measure entropically similarity:

$$JSD(v_p, v_t) = \frac{1}{2}\left[\sum v_p \cdot \log\frac{2v_p}{v_t + v_p} + \sum v_t \cdot \log\frac{2v_t}{v_p + v_t}\right] \tag{3}$$

For simplicity, we denote the three measures as $CS(v_t)$, $D(v_t)$, and $JSD(v_t)$, respectively.

Considering the backdoor inheritance property (§ IV-B), a higher similarity could merely indicate the corresponding model has a backdoor. It's the change points of the similarity that indicates a possible backdoor strengthen behavior (i.e., attack round). From the perspective of time series, the trend changing can be identified by the first-order differences $\Delta S_{T-1} = \{\Delta s_1, \ldots, \Delta s_{T-1}\}$, where $\Delta s_t = s_{t+1} - s_t$ and $s_t = \{CS(v_t), D(v_t), JSD(v_t)\}$.

Generally, the similarity changes should be smooth for benign rounds, while abrupt shifts can be indicators of attack involvement. Although there exists many anomaly detection techniques, they predominately rely on supervised or semi-supervised learning, which are well-suited in the backdoor contexts as no label knowledge is known in prior. To accommodate detection sensitivity and accuracy, we propose to build our similarity anomaly detector via ensemble unsupervised learning. In particular, we consider the joint factors from local to global view with distance, density, and isolated characteristics:

- **OC-SVM (OC)** [33]. We estimate a non-linear decision boundary based on appropriate kernel functions (Gaussian kernel function) and soft margins to find anomaly points. The rounds excluded by the hyper-plane are considered distance outliers.
- **Local Outlier Factor (LOF)** [34]. We compute the local density of each point and estimate its density deviation with respect to its temporal neighbors. By checking the density deviation point-by-point, those with significantly smaller densities than neighbors are identified as outliers.
- **Isolation Forest (IF)** [35]. Using the three similarity channels as the branch features, we build a bunch of decision trees through random feature assignment and split value selection (in the range of minimum and maximum values). This random partitioning of features will render shorter paths in trees for the anomaly rounds, thus distinguishing them from the other points.

We fuse the detection results using majority voting, namely, a round with more than two detectors identifying its similarity as an anomaly is considered an attack round.

[3]We use the output of the last layer in the deep model for feature representation.

### B. Identifying Attackers

Since attack round selection and participant selection are both random, a participant's frequent occurrence in identified attack rounds reflects its probability of being malicious. Formally, we first present behavior discrepancies between attackers and benign participants in attack rounds. Assume that the FL task contains $N$ participants, and $N_a$ of them are attackers, where $N_a \ll N$ (E.g., 3 attackers in 100 participant. A small ratio of attackers is sufficient for an effective backdoor injection, so smart attackers wouldn't take the risk of constructing a large group). In each round, we assume that $m$ participants are randomly selected to train a local model, and there include $m_a$ attackers in those attack rounds, where $m_a \in [0, \min\{m, N_a\}]$ and $N, N_a, m, m_a \in \mathbb{N}^+$. Hence, for each attack round, the occurrence probability of a malicious and benign participant is $p_a = \frac{m_a}{N_a}$ and $p_b = \frac{m-m_a}{N-N_a}$, respectively.

For an effective attack round, OMBA requires at least half of all the attackers to involve for sufficient poisoning knowledge injection, which gives $p_a = 0.5$. Given that $N_a > m_a$, we also have $p_b < \frac{m}{N}$, which denotes the participant selection ratio of FL and suggested to be $\leq 10\%$ [2] [36]. Then, we can have that, empirically, $p_a > 5 \cdot p_b$. Observing the co-occurrence of a participant of $n_c$ times in all the attack rounds, the probability for it to be an attacker and a benign participant is $(p_a)^{n_c}$ and $(p_b)^{n_c}$, respectively. In practice, even $n_c = 2$ makes the probability of the suspect being an attacker 25 times than being a benign one.

With this insight, we then propose to estimate the co-occurrences of all participants in the identified attack round and mark those appearing more than two times as attackers. Actually, an effective OMBA requires a higher the co-occurrences of the attacker, more details in Appendix B1.

### C. Mitigating Backdoor

The attackers try to inject the backdoor into the global model by uploading the poisoned models in aggregation. With attack round and attackers identified, we could remedy the backdoor by subtracting the corresponding poisoned updates.

Assume that in a round (say $t^{th}$), each participant $k$ trains the global model $w^t$ with its data and uploads the updates $\Delta w_k^{t+1}$ to the server. The server collects these local updates and aggregates a new global model by

$$w^{t+1} = w^t + \frac{1}{|\mathbf{S}^t|}\sum_{k \in \mathbf{S}^t} \Delta w_k^{t+1}, \tag{4}$$

$\mathbf{S}^t$ is participants set in round $t$. After $T$ rounds, we can obtain a global model

$$w^T = w^0 + \sum_{t=1}^{T} \frac{1}{|\mathbf{S}^t|} \sum_{k \in \mathbf{S}^t} \Delta w_k^t. \tag{5}$$

In the attack rounds $\mathbf{T}_a$, some attackers $\mathbf{A}^t, t \in \mathbf{T}_a$ try to inject the backdoor into the global model, the process can be described as:

$$\begin{aligned} w^T &= w^0 + \sum_{t=1}^{T} \frac{1}{|\mathbf{S}^t|} \left( \sum_{k \in \mathbf{S}^t \setminus \mathbf{A}^t} \Delta w_k^t + \sum_{k' \in \mathbf{A}^t} \Delta w_{k'}^t \right) \\ &= w^0 + \sum_{t=1}^{T} \frac{1}{|\mathbf{S}^t|} \sum_{k \in \mathbf{S}^t \setminus \mathbf{A}^t} \Delta w_k^t + \sum_{t \in \mathbf{T}_a} \frac{1}{|\mathbf{S}^t|} \sum_{k' \in \mathbf{A}^t} \Delta w_{k'}^t, \end{aligned} \tag{6}$$

when $t \notin \mathbf{T}_a$, $\mathbf{A}^t = \emptyset$.

Intuitively, we repair the poisoned global model by directly subtracting all the poisoned local updates. The process can be written as:

$$\widehat{w} = w^T - \sum_{t \in \mathbf{T}_a} \frac{1}{|\mathbf{S}^t|} \sum_{k' \in \mathbf{A}^t} \Delta w_{k'}^t = w^0 + \sum_{t=1}^{T} \frac{1}{|\mathbf{S}^t|} \sum_{k \in \mathbf{S}^t \setminus \mathbf{A}^t} \Delta w_k^t, \tag{7}$$

Moreover, we try to make the repaired model more approximate to the 'healthy' model which is trained by the left benign participants. We scale the $\widehat{w}$ with a parameter $\rho$,

$$\widehat{w} = \rho \left( w^0 + \sum_{t=1}^{T} \frac{1}{|\mathbf{S}^t|} \sum_{k \in \mathbf{S}^t \setminus \mathbf{A}^t} \Delta w_k^t \right). \tag{8}$$

**Theorem 1** (Bounding the difference)**.** *Assume some attackers $\mathbf{A}^t$ have poisoned the global model $w^T$ by performing several shots in rounds $\mathbf{T}_a$. After mitigating the backdoor, the difference between the repaired model and the 'healthy' model can be bounded as*

$$0 \leq |w_{healthy}^T - \widehat{w}| \leq |\mathbf{T}_a| \Delta \overline{w}. \tag{9}$$

Theorem 1 shows the difference bound between the repaired model and the 'healthy' model. The 'healthy' model represents the model trained by the rest benign participants. The upper bound of the difference means that the repaired model removes all the local updates in $\mathbf{T}_a$, even if they are from benign participants. The lower bound means our method removes the backdoor in the global model and the repaired model has the same effect as training with the benign participants. The proof of Theorem 1 is shown in Appendix C.

Finally, we conclude the tracing and mitigation procedure in Algorithm 1.

## VI. Experiments

We evaluate the performance of the framework by comparing it with SoTA defenses under different attack settings.

**Algorithm 1:** Tracing & mitigation

**Input :** Current model $w^T$, a trapped poisoned sample $d_{neg}$, historic global models $w^t$ and local updates $\Delta w_k^t$ from participants $\mathbf{S}^t$ $(t = 1, \cdots, T-1)$.

**Output:** Repaired model $\widehat{w}$.

```
// Step 1: Identifying attack rounds
```

$v_p = f(w^T; d_{neg})$;

**foreach** *round* $t = 1, 2, \ldots T-1$ **do**

  $v_t = f(w^t; d_{neg})$ ;

  $s_t = \{CS(v_p, v_t), D(v_p, v_t), JSD(v_p, v_t)\}$;

  $\Delta s_t = s_{t+1} - s_t$;

$\Delta S_{T-1} = \{\Delta s_1, \cdots, \Delta s_{T-1}\}$;

$\mathbf{T}_a =$ **`EnsembleDetecetor`**$(\Delta S_{T-1})$;

```
// Step 2: Identifying attackers
```

Statistic the frequency $Q_k$ of each participant $k$ in $\mathbf{T}_a$;

$\mathbf{A}^t = \{k | Q_k \geq Q_{threshold}\}, t \in \mathbf{T}_a$;

```
// Step 3: Mitigating backdoors
```

$\widehat{w} = \rho \left( w^T - \sum_{t \in \mathbf{T}_a} \frac{1}{|\mathbf{S}^t|} \sum_{k' \in \mathbf{A}^t} \Delta w_{k'}^t \right)$;

**return** $\widehat{w}$

### A. Setup

*1) Datasets:* We consider two widely used public dataset MNIST [37] and Fashion-MNIST (FMNIST) [38], both of which consists of 60,000 training samples and 10,000 test samples, each sample has the same format and the size is $28 \times 28$. The MNIST dataset is comprised of 10-class handwriting digits, and FMNIST includes some fashion products from 10 categories.

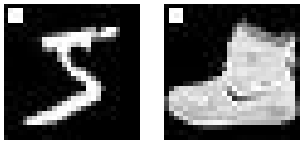
*2) FL Setup:* We run the FL training process with 100 participants for these two datasets. Following the previous work [2] [16], in each round, we select 10 participants uniformly at random. We use SGD and trains for 2 local epochs with the local learning rate of 0.1 and the local batch size of 64. And we use the vanilla aggregation method called FedAvg.

*3) Data Distribution:* FL often presumes non-IID data distribution across parties. Here, we use a Dirichlet distribution with the hyperparameter $\alpha = 0.5$ to generate the data distribution following the setups in [7] [9].

*4) Metrics:* We use the averages of *Precision*, *Recall*, and *F1-Score* across multiple tests to evaluate the performance of identifying the attack rounds and the attackers. We evaluate the effectiveness of eliminating the backdoor by observing the attack success rate (**ASR**) and benign accuracy (**BA**). ASR is the accuracy of the model on the samples with backdoor triggers. BA is the accuracy of the model on benign samples.

### B. Backdoor Attacks and Defenses in FL

The goal of the backdoor attack is to change the global model's behavior on some data samples with certain backdoor triggers while maintaining high performance on benign samples. Here, we introduce the basic settings of both backdoor attacks and defenses.



(a) MNIST (b) FMNIST

Fig. 6. The samples with backdoor trigger in MNIST and FMNIST. The pixels in the upper left corner is the backdoor trigger.

*1) Backdoor Trigger:* For image classification tasks, we consider the basic pixel backdoor by adding certain pixels into the training samples [23]. Specifically, we set that the attacker modifies 9 pixels of the upper left corner to form a backdoor trigger, as shown in Fig. 6. The original label of these samples will be swapped into "3". For FMNIST, the trigger pattern is the same as the MNIST, but the label will be swapped into "T-Shirt". During the local training phase, the attacker would train the model with both original images and the images with the backdoor trigger at the same time. We use the poisoning ratio $r$ to control the fraction of samples added per training batch [9], and we set $r = 20/64$ for all datasets here.

*2) Injection Strategies:* CBA and DBA are taxonomy for backdoor attacks according to whether the trigger is injected by each attacker completely (centralized) or partially (distributed). We test the performance under two typical trigger injection strategies in the literature:

- **Centralized backdoor attack (CBA)** [7]. All attackers use the same global trigger to poison their local data. Local models are trained by these samples and uploaded to the server to poison the global model.
- **Distributed backdoor attack (DBA)** [9]. The attackers use arbitrary parts of the global trigger to poison the local data. We separate the 9-pixel backdoor trigger into 3 local triggers, each containing 3 pixels and embedded in the upper left corner. Local models are then trained with the injection of these complementary triggers, together with planting the complete trigger in the global model.

*3) Backdoor Strategies:* Since existing backdoor attacks fall short on stealthiness, we use the OMBA proposed in § III as the adversary setting. The key factors of OMBA are: The number of attackers $N_a$ that collusively conduct on-off attacks; The number of shots $N_s$ that some attackers are active in the FL round to inject backdoor; Shot interval $I$ that defines the number of silent rounds between two attack round. We will vary the first two factors during experiments to test the detection performance.

*4) Backdoor Defenses:* Here, we choose some representative backdoor defense methods as baselines:

- **FAA-DL** [18]. A clustering-based detection method using OC-SVM to filter the anomaly updates in the training process. To better compare, we only focus on the anomaly detection results.
- **Krum** [13]. A robust aggregation method by calculating the distance and filtering the local model parameters which are far away from their neighbors. Here, we only focus on the outliers filtered out by this strategy.

### C. Performance Analysis

We test the performance of each step in the framework under **CBA** and **DBA**.

*1) Identifying the attack rounds:* We set $N_a = 3$ for each round, and $I = 3$ and change $N_s$ to verify the effectiveness. The attackers start to perform a shot at $40^{th}$ round when the model gradually gets convergent. Three metrics are used to evaluate whether the 1st step can correctly identify most attack rounds. To better demonstrate the effectiveness, we average the results with the same setting. The result is shown in Table II. Comparing the FAA-DL and Krum, our methods have a better performance in detecting the attack rounds across all datasets in all attack settings.

In **CBA**, the precision of our method is maintained at around 80%, and 31.26% higher than other methods. Notice that the recall is decreased to about 30% and still higher than other baselines with the increment of $N_s$. The reason for the decrement is that with the increment of $N_s$, the backdoor has already been injected into the global model, and most of the shots can not be detected since they can not cause significant changes in the global model.

In **DBA**, all the performance has decreased compared with CBA. The reason for the decrement is that the DBA only injects some of the backdoor fragments and causes minor modifications in each shot. Nevertheless, our method is still effective and achieves around 60% in all metrics in detecting the attack rounds.

Hence, *we verify that finding change points of the poisoned sample's prediction similarity is more effective than clustering-based and distance-based methods in different attacks.*

*2) Identifying the attackers:* In this part, we show the effectiveness of the 2nd step by varying $N_a$. For a fair comparison, all methods use the same input from the 1st step. Here, we set $N_s = 5$ and $I = 3$, and start to attack at $40^{th}$ round. The results are shown in Table. III, our method based on probability analysis is more effective than other baselines (average 13.13% higher). The reason is that our method does not directly analyze the local updates but focuses on the behavior of the attackers. Directly analyzing the local updates can easily result in evasion of detection, see the performance of FAA-DL and Krum in CBA and DBA. The fine-crafted backdoor model and stealth attack methods (i.e., DBA) only require a minor change and perform more similarly to other models, the methods based on clustering or distance measurement are not easy to detect. *Furthermore, we can observe that under different attack settings, our method can still achieve the highest performance (∼50%) in all cases.*

*3) Mitigating backdoor:* To evaluate the effectiveness of the 3rd step, we consider the performance changes of the model before and after mitigating backdoor attacks. We adopt the suspicious attackers in the second step and remove all the local updates uploaded from them by subtracting and scaling. As shown in Table. IV, the backdoor can be effectively removed

TABLE II
AVERAGE PERFORMANCE OF IDENTIFYING THE ATTACK ROUNDS WITH DIFFERENT NUMBER OF SHOTS ($N_S$) ON TWO DATASETS IN **CBA** AND **DBA**.

| Attack Scheme | Dataset | $N_S$ | Precision | | | Recall | | | F1-Score | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ours | FAA-DL | Krum | ours | FAA-DL | Krum | ours | FAA-DL | Krum |
| CBA | MNSIT | 3 | **0.8167** | 0.4738 | 0.4089 | **0.5833** | 0.5000 | 0.3333 | **0.6917** | 0.4807 | 0.3958 |
| | | 5 | **0.8452** | 0.4549 | 0.3782 | **0.6071** | 0.4720 | 0.3622 | **0.7636** | 0.4638 | 0.3645 |
| | | 7 | **0.8694** | 0.4861 | 0.2958 | **0.5048** | 0.4471 | 0.4286 | **0.5428** | 0.4537 | 0.3621 |
| | | 10 | **0.8600** | 0.4797 | 0.2451 | **0.4812** | 0.4300 | 0.3200 | **0.4542** | 0.4426 | 0.3015 |
| | | 12 | **0.8650** | 0.4322 | 0.2141 | **0.4833** | 0.4533 | 0.2500 | **0.4882** | 0.4332 | 0.2313 |
| | FMNSIT | 3 | **0.8611** | 0.4556 | 0.4980 | **0.6333** | 0.4444 | 0.4333 | **0.7302** | 0.4443 | 0.4385 |
| | | 5 | **0.8667** | 0.4600 | 0.3904 | **0.5667** | 0.3186 | 0.4000 | **0.6988** | 0.4255 | 0.3920 |
| | | 7 | **0.8720** | 0.4571 | 0.3212 | **0.4286** | **0.4286** | 0.4143 | **0.6558** | 0.4051 | 0.3529 |
| | | 10 | **0.8600** | 0.3855 | 0.3160 | **0.3641** | 0.3260 | 0.2016 | **0.4371** | 0.3704 | 0.3020 |
| | | 12 | **0.8833** | 0.3667 | 0.3044 | **0.3533** | 0.3333 | 0.2516 | **0.3683** | 0.3458 | 0.2727 |
| DBA | MNSIT | 3 | **0.6733** | 0.4067 | 0.3231 | **0.6333** | 0.4811 | 0.3333 | **0.6523** | 0.4239 | 0.3233 |
| | | 5 | **0.7267** | 0.4000 | 0.3123 | **0.7000** | 0.2400 | 0.2000 | **0.7265** | 0.2489 | 0.3121 |
| | | 7 | **0.7867** | 0.3999 | 0.3202 | **0.6000** | 0.2190 | 0.1667 | **0.6726** | 0.2979 | 0.2214 |
| | | 10 | **0.7914** | 0.4952 | 0.3032 | **0.4400** | 0.1300 | 0.1700 | **0.5802** | 0.2373 | 0.2560 |
| | | 12 | **0.7929** | 0.4800 | 0.2916 | **0.4000** | 0.1483 | 0.1833 | **0.5253** | 0.2137 | 0.2106 |
| | FMNSIT | 3 | **0.6357** | 0.3657 | 0.3100 | **0.6333** | 0.4667 | 0.3333 | **0.6339** | 0.4286 | 0.3138 |
| | | 5 | **0.7967** | 0.3860 | 0.3231 | **0.6202** | 0.4200 | 0.3600 | **0.6729** | 0.4099 | 0.3426 |
| | | 7 | **0.8010** | 0.4167 | 0.3133 | **0.6000** | 0.3149 | 0.4286 | **0.6803** | 0.3262 | 0.3529 |
| | | 10 | **0.8533** | 0.4157 | 0.3256 | **0.4000** | 0.3017 | 0.3000 | **0.5390** | 0.3543 | 0.3113 |
| | | 12 | **0.8560** | 0.3676 | 0.2954 | **0.4167** | 0.2167 | 0.2500 | **0.5530** | 0.2833 | 0.2527 |

TABLE III
AVERAGE PERFORMANCE OF IDENTIFYING THE ATTACKERS WITH DIFFERENT NUMBER OF ATTACKERS ($N_a$) ON TWO DATASETS IN **CBA** AND **DBA**.

| Attack Scheme | Dataset | $N_a$ | Precision | | | Recall | | | F1-Score | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ours | FAA-DL | Krum | ours | FAA-DL | Krum | ours | FAA-DL | Krum |
| CBA | MNSIT | 3 | **0.6917** | 0.4630 | 0.2222 | **0.8333** | 0.5283 | 0.5833 | **0.7470** | 0.5061 | 0.3743 |
| | | 5 | **0.6933** | 0.5413 | 0.1667 | **0.6000** | 0.5200 | 0.5128 | **0.6378** | 0.5370 | 0.3246 |
| | | 7 | **0.6360** | 0.5260 | 0.1574 | **0.5220** | 0.5067 | 0.5072 | **0.5358** | 0.5178 | 0.3203 |
| | | 10 | **0.5267** | 0.4933 | 0.1709 | **0.4800** | 0.3933 | 0.4333 | **0.4943** | 0.4163 | 0.3273 |
| | FMNSIT | 3 | **0.6333** | 0.5935 | 0.1347 | **0.6333** | 0.4815 | 0.4040 | **0.6333** | 0.5149 | 0.2020 |
| | | 5 | **0.6433** | 0.5889 | 0.1458 | **0.6000** | 0.4889 | 0.4375 | **0.6196** | 0.5270 | 0.2187 |
| | | 7 | **0.7667** | 0.6167 | 0.1319 | **0.5714** | 0.4000 | 0.3958 | **0.6025** | 0.4571 | 0.1979 |
| | | 10 | **0.7933** | 0.4211 | 0.1347 | **0.5400** | 0.4000 | 0.4040 | **0.5868** | 0.4016 | 0.2020 |
| DBA | MNSIT | 3 | **0.6200** | 0.4606 | 0.1944 | **0.7325** | 0.4848 | 0.2286 | **0.6747** | 0.4726 | 0.2917 |
| | | 5 | **0.5533** | 0.4010 | 0.1709 | **0.5600** | 0.4167 | 0.2692 | **0.5500** | 0.3984 | 0.2564 |
| | | 7 | **0.6933** | 0.3806 | 0.1836 | **0.4286** | 0.3611 | 0.2255 | **0.4771** | 0.3484 | 0.2754 |
| | | 10 | **0.5400** | 0.3359 | 0.1778 | **0.4160** | 0.4103 | 0.2500 | **0.4764** | 0.3260 | 0.2667 |
| | FMNSIT | 3 | **0.6200** | 0.4342 | 0.1049 | **0.6312** | 0.4652 | 0.3148 | **0.6275** | 0.4428 | 0.1574 |
| | | 5 | **0.7933** | 0.4597 | 0.1250 | **0.5200** | 0.4444 | 0.3750 | **0.5752** | 0.4395 | 0.1875 |
| | | 7 | **0.6000** | 0.3622 | 0.1079 | **0.4571** | 0.3333 | 0.3238 | **0.4696** | 0.3284 | 0.1619 |
| | | 10 | **0.4600** | 0.3714 | 0.1010 | **0.4100** | 0.3286 | 0.3030 | **0.4327** | 0.3622 | 0.1515 |

from the global model at the cost of $3 \sim 10\%$ BA loss. Specially, we observe that, in DBA, there are some backdoor remnants, and occurs a significant degradation of BA. The reason is that DBA uses multiple backdoor fragments to attack, but grouped together to be effective. Different from the CBA using one same global trigger, eliminating the backdoor in DBA needs to collect all kinds of local triggers. Our method can find most of them and reduce the ASR below 20%.

Moreover, we use Grad-CAM [39] to provide a visualization of the model remedy. As shown in Fig. 7, the area with red color represents the features that are more important for the model decision. We find that our method can weak the models' attention on the backdoor trigger, and make the model perform more similarly to the clean model. *We conclude that our method can achieve a larger drop in attack success rate by trading off a small decrease in benign accuracy.*

### *D. Ablation study*

In order to illustrate the effectiveness of each component in 1st step, we conduct the ablation studies on MNIST under the CBA scheme. We consider the different compositions of three similarity metrics and algorithms. The settings and results are shown in Table. V. We notice that each component (similarity metric or algorithm) in our design is effective since the incor-

TABLE IV
BENIGN ACCURACY (BA) AND ATTACK SUCCESS RATE (ASR) BEFORE AND AFTER MITIGATING BACKDOOR ATTACKS.

| Dataset | Models | BA | ASR |
|---|---|---|---|
| MNIST | Clean Model | 99.67% | 0.45% |
| | Backdoored Model (CBA) | 98.51% | 97.57% |
| | Repaired Model (CBA) | 96.90% | **5.57%** |
| | Backdoored Model (DBA) | 98.92% | 98.42% |
| | Repaired Model (DBA) | 95.46% | **17.98%** |
| FMNIST | Clean Model | 89.56% | 0.32% |
| | Backdoored Model (CBA) | 87.49% | 99.50% |
| | Repaired Model (CBA) | 82.17% | **8.03%** |
| | Backdoored Model (DBA) | 85.30% | 99.80% |
| | Repaired Model (DBA) | 79.85% | **18.89%** |

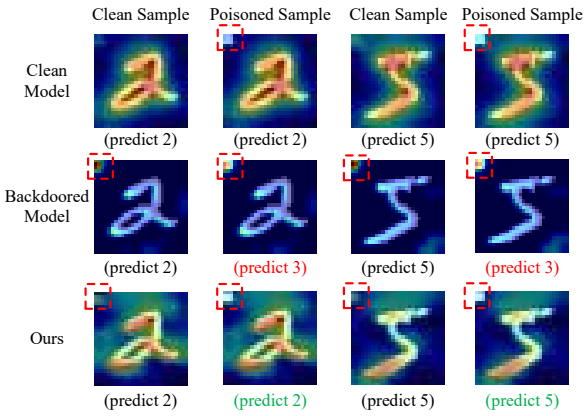



Fig. 7. The visualization of the models' attention before and after eliminating the backdoor by using Grad-CAM in two MNIST samples.

poration of each component can achieve higher performance than the counterpart built without the component.

**Similarity.** We use all algorithms and leave them unchanged to analyze the effectiveness of similarity metrics. From the table, the $JSD$ has a larger impact on the performance than the others, which means the probabilistic similarity fluctuations can better reflect the attack.

**Algorithms.** We use all the similarity metrics and compare the effectiveness of each algorithm. The three algorithms alone are not effective, but their compositions can achieve significant performance improvement. Moreover, there is an interesting point to note. *The metrics are more important than algorithms in the performance increment*, the decoupling of anyone metric can cause larger performance degradation compared to that of any algorithm. It is indicated that the similarity estimation is more critical in identifying attack rounds.

## VII. DISCUSSION

**Assumptions on the defender.** In our method, we assume that the defender can access to all the global models and record all the intermediate training results. It is essential for the server to store these training results for future authentication, audit [40], and profit allocation [41], although this may require a high storage overhead. Besides, a poisoned sample can be easily distinguished from the misclassified samples since it always shows a target manner on the backdoored model.

TABLE V
ABLATION STUDY OF DIFFERENT METRICS AND ALGORITHMS IN IDENTIFYING ATTACK ROUNDS.

| Similarity | | | Algorithms | | | Performance | | |
|---|---|---|---|---|---|---|---|---|
| CS | D | JSD | OC | IF | LOF | Precision | Recall | F1-Score |
| ✓ | | | ✓ | ✓ | ✓ | 0.5095 | 0.5214 | 0.5118 |
| | ✓ | | ✓ | ✓ | ✓ | 0.5833 | 0.5143 | 0.5420 |
| | | ✓ | ✓ | ✓ | ✓ | 0.6643 | 0.5000 | 0.5515 |
| ✓ | ✓ | | ✓ | ✓ | ✓ | 0.6405 | 0.5500 | 0.6258 |
| ✓ | | ✓ | ✓ | ✓ | ✓ | 0.6667 | 0.5786 | 0.6370 |
| | ✓ | ✓ | ✓ | ✓ | ✓ | 0.6767 | 0.5429 | 0.6253 |
| ✓ | ✓ | ✓ | ✓ | | | 0.5238 | 0.4571 | 0.5069 |
| ✓ | ✓ | ✓ | | ✓ | | 0.5714 | 0.4286 | 0.5495 |
| ✓ | ✓ | ✓ | | | ✓ | 0.5911 | 0.5071 | 0.5741 |
| ✓ | ✓ | ✓ | ✓ | ✓ | | 0.7500 | 0.5714 | 0.6503 |
| ✓ | ✓ | ✓ | ✓ | | ✓ | 0.7738 | 0.5729 | 0.6888 |
| ✓ | ✓ | ✓ | | ✓ | ✓ | 0.7262 | 0.5674 | 0.6352 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **0.8452** | **0.6071** | **0.7636** |

**Co-occurrences of the attackers.** Our method can identify the attackers based on the co-occurrence difference between benign participants and attackers. However, if all the attackers have the same frequency as benign participants, our method might not be effective. In our consideration, high frequency can eliminate the conflicts between attackers and inject the backdoor into the global model with as fewer shots as possible. We present the attack process in different frequencies of attackers in Appendix B1.

## VIII. CONCLUSION

In this work, we propose a framework to mitigate stealthy backdoor attacks in typical FL processes. We design a novel on-off multi-shot attack form to mimic the attack strategy of smart attackers and present a sequence of observations on its stealthiness against SoTA defenses. In view of such tricky threats, we provide a remedy based on jointly identifying attack round, attackers, and mitigating the influence via removing certain malicious ingredients. Experimental results show that our method is effective in each step. Moreover, we present the ablation study on the first step and evaluate the importance of each component.

## ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2020YFC2003400), the National Natural Science Foundation of China (Nos. 62072465, 62172155, 62102425, 62102429), the Science and Technology Innovation Program of Hunan Province (No. 2021RC2071), and the Natural Science Foundation of Hunan Province (No. 2022JJ40564).

## REFERENCES


[1] T. Zhou, Z. Cai, and F. Liu, "The crowd wisdom for location privacy of crowdsensing photos: Spear or shield?" in *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, 2021, pp. 1–23.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of AISTATS*, 2017, pp. 1273–1282.

[3] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[4] C. Ying, H. Jin, X. Wang, and Y. Luo, "Double insurance: Incentivized federated learning with differential privacy in mobile crowdsensing," in *Proc. of SRDS*, 2020, pp. 81–90.

[5] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *Journal of Biomedical Informatics*, vol. 99, p. 103291, 2019.

[6] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Proc. of NeurIPS*, vol. 33, pp. 16 070–16 084, 2020.

[7] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. of AISTATS*. PMLR, 2020, pp. 2938–2948.

[8] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. of ICML*. PMLR, 2019, pp. 634–643.

[9] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *Proc. of ICLR*. ICLR, 2019.

[10] Z. Yin, Y. Yuan, P. Guo, and P. Zhou, "Backdoor attacks on federated learning with lottery ticket hypothesis," *arXiv preprint arXiv:2109.10512*, 2021.

[11] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proc. of USENIX Security*, 2020, pp. 1605–1622.

[12] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. of ICML*. PMLR, 2018, pp. 5650–5659.

[13] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Proc. of NeurIPS*, vol. 30, 2017.

[14] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *Proc. of ICML*. PMLR, 2019, pp. 6893–6901.

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

[16] H. Zeng, T. Zhou, Y. Guo, Z. Cai, and F. Liu, "FedCav: Contribution-aware model aggregation on distributed heterogeneous data in federated learning," in *Proc. of ICPP*, 2021, pp. 1–10.

[17] T. D. Nguyen, P. Rieger, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, A.-R. Sadeghi, T. Schneider *et al.*, "FLGUARD: Secure and private federated learning," *arXiv preprint arXiv:2101.02281*, 2021.

[18] S. Shi, C. Hu, D. Wang, Y. Zhu, and Z. Han, "Federated anomaly analytics for local model poisoning attack," *IEEE Journal on Selected Areas in Communications (JSAC)*, 2021.

[19] Y. Wang, D. Zhai, Y. Zhan, and Y. Xia, "Rflbat: A robust federated learning algorithm against backdoor attack," *arXiv preprint arXiv:2201.03772*, 2022.

[20] Q. Li, X. Wei, H. Lin, Y. Liu, T. Chen, and X. Ma, "Inspecting the running process of horizontal federated learning via visual analytics," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2021.

[21] 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.

[22] S. Goldwasser, M. P. Kim, V. Vaikuntanathan, and O. Zamir, "Planting undetectable backdoors in machine learning models," *arXiv preprint arXiv:2204.06974*, 2022.

[23] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[24] B. G. Tekgul, Y. Xia, S. Marchal, and N. Asokan, "Waffle: Watermarking in federated learning," in *Proc. of SRDS*, 2021, pp. 310–320.

[25] S. Awan, B. Luo, and F. Li, "CONTRA: Defending against poisoning attacks in federated learning," in *Proc. of ESORICS*. Springer, 2021, pp. 455–475.

[26] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *Proc. of RAID*, 2020, pp. 301–316.

[27] Q. Li, Q. Shen, Y. Ming, P. Xu, Y. Wang, X. Ma, and H. Qu, "A visual analytics approach for understanding egocentric intimacy network evolution and impact propagation in mmorpgs," in *Proc. of PacificVis*. IEEE, 2017, pp. 31–40.

[28] Q. Li, K. S. Njotoprawiro, H. Haleem, Q. Chen, C. Yi, and X. Ma, "Embeddingvis: A visual analytics approach to comparative network embedding inspection," in *Proc. of VAST*. IEEE, 2018, pp. 48–59.

[29] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," in *Proc. of USENIX Security*, 2021, pp. 1505–1521.

[30] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. of SP*. IEEE, 2019, pp. 707–723.

[31] Y. Liu, M. Fan, C. Chen, X. Liu, Z. Ma, L. Wang, and J. Ma, "Backdoor defense with machine unlearning," in *Proc. of INFOCOM*, 2022.

[32] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention-distillation: Erasing backdoor triggers from deep neural networks," in *Proc. of ICLR*, 2020.

[33] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Proc. of NeurIPS*, vol. 12, 1999.

[34] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proc. of SIGMOD*, 2000, pp. 93–104.

[35] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. of ICDM*. IEEE, 2008, pp. 413–422.

[36] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu, "Feddc: Federated learning with non-iid data via local drift decoupling and correction," in *Proc. of CVPR*, 2022.

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

[38] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[39] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. of ICCV*, 2017, pp. 618–626.

[40] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[41] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *Proc. of Big Data*. IEEE, 2019, pp. 2577–2586.


## APPENDIX

### A. HFLens on local models

Here, we report the visualization of all local models projections by using t-SNE. We generate the model feature vectors of the local models as described in § III-C2, then perform t-SNE on vectors to generate the 2D projections (shown in Fig. 8). Each point in the 2D space represents the projection of the local models with the linked black curve representing the performance evolution trend. The red points representing the poisoned local models are mixed with the green points and are hard to be categorized as outliers. Besides, the projections of the local models can also verify that the local models may diverge in the training process when the data is non-IID. *We conclude that the method based on t-SNE is ineffective in*

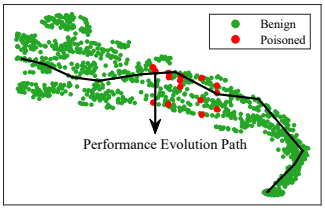



Fig. 8. Visualization on the 2D projection of all the local models in each round using t-SNE.

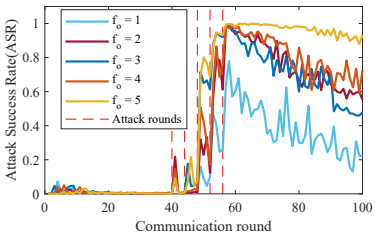



Fig. 9. Impact of the co-occurrences of the attackers in 5 shots.

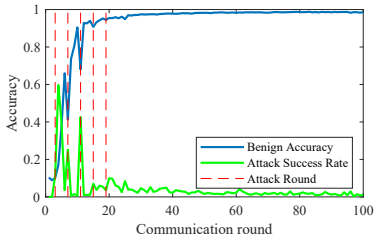
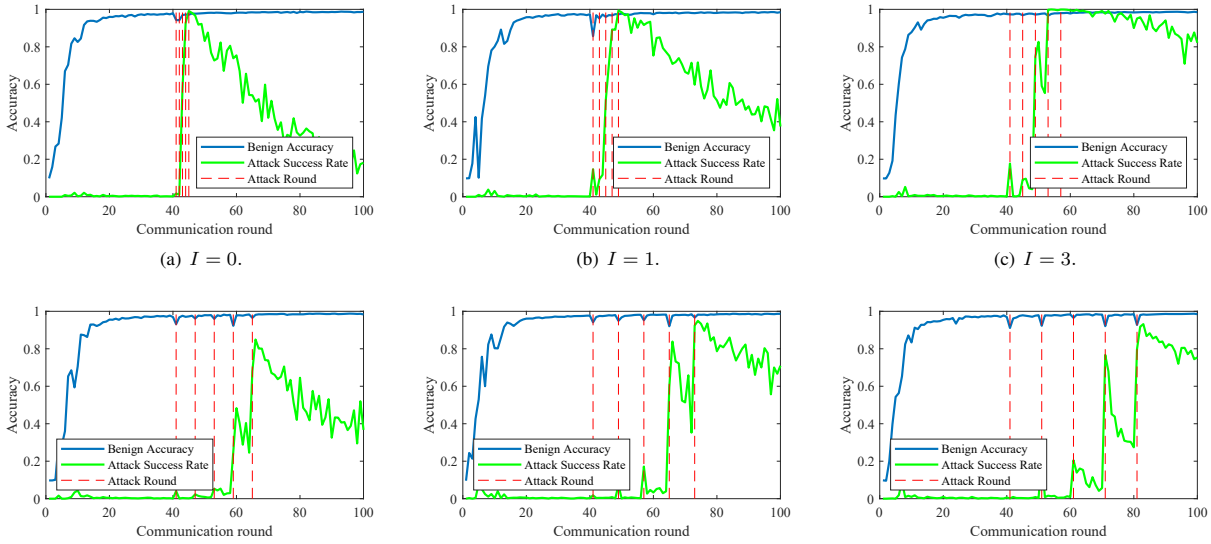



Fig. 10. The training process of the attack rounds in non-convergence phase.




(a) $I = 0$. (b) $I = 1$. (c) $I = 3$.

(d) $I = 5$. (e) $I = 7$. (f) $I = 9$.

Fig. 11. Impact of different shot interval ($I$) in the performance of OMBA.

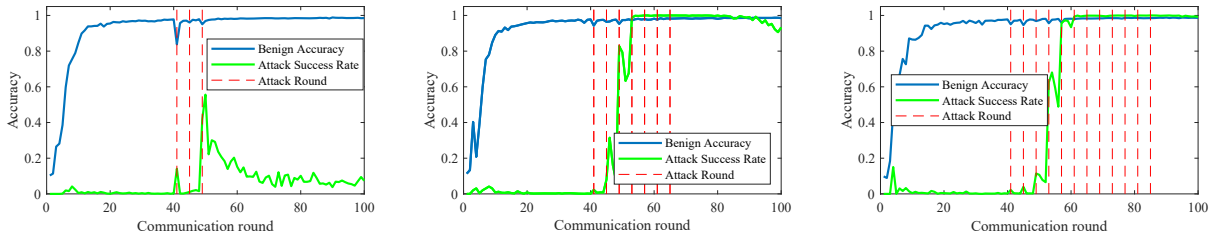
*distinguishing whether the local model is from the benign participants or the attackers.*

### B. More experiments in OMBA

*1) Impact of the co-occurrence:* We explore the performance of the backdoor injection by varying different co-occurrence. For a fair comparison, we set that there are $N_s = 5$ shots in round $40^{th}, 44^{th}, 48^{th}, 52^{nd}$, and $56^{th}$. And each attack round contains 3 attackers colluding to upload the poisoned local updates. We note the variable $f_o$ as the occurrence frequency of each participant. For example, $f_o = 1$ means that an attacker can only appear once in 5 attack rounds, and it needs $\frac{3N_s}{f_o} = 15$ different attackers to complete the attack. For different $f_o$, the speed of the backdoor injection is shown in Fig. 9. With the increasing $f_o$, the backdoor can be injected into the global model with fewer shots. Moreover, the injected backdoor can not easily be weakened by the benign local updates in the following training rounds. The reason is that there are some conflicts between different attackers, which makes the feature of the injected backdoor not intense as benign samples, and easily forgotten by the global model. *In summary, setting a low $f_o$ in OMBA is less effective since it needs more attackers but obtains a weaker backdoored global model.*

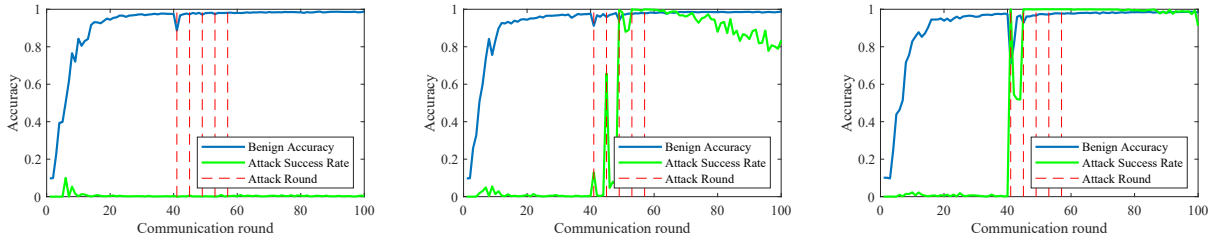
*2) Impact of shot interval (I):* We explore the impact of different $I$ in OMBA. We set there are $N_s = 5$ shots and $N_a = 3$, each attacker would perform 5 shots starting from round $40^{th}$. Fig. 11 shows the results. A small $I$ indicates a more intensive attack, the ASR is rapidly increasing but a sudden drop in benign accuracy according to Fig. 11(a), 11(b). A large $I$ (see Fig. 11(d), 11(e), and 11(f)) needs more shots to obtain an obvious ASR increment for a long interval may cause the backdoor gradually weakened by benign participants. We conclude that a proper $I$ can strike a compromise between stealthiness and effectiveness.

*3) Impact of the attack timing:* In this part, we explore a reasonable time for attackers to inject the backdoor. The attack time can roughly divide into two phases: the non-convergence phase and the convergent phase. Fig. 10 shows the training process of the attackers trying to inject the backdoor into




(a) $N_s = 3$ (b) $N_s = 7$ (c) $N_s = 12$

Fig. 12. Impact of different number of attack rounds ($N_s$) in the performance of OMBA.




(a) $N_a = 1$ (b) $N_a = 5$ (c) $N_a = 10$

Fig. 13. Impact of different number of attackers ($N_a$) in the performance of OMBA.

the global model by adopting OMBA in the non-convergence phase. We can observe that the backdoor injection fails since the global model is in a dilemma between the benign and poisoned local updates in the non-convergence phase.

*4) Impact of $N_s$:* We present the performance of OMBA in different $N_s$. More shots will make the backdoor injected into the global model more solid and not easily forgotten. Fig. 12 shows the results of different $N_s$, fewer shots might cause the backdoor not be successfully injected, and too many shots are unnecessary since the backdoor has been embedded into the global model. We conclude that the 'smart' attacker might perform as fewer shots as possible to embed the backdoor into the global model.

*5) Impact of $N_a$:* We explore the impact of different $N_a$ in OMBA. More attackers colluding in an attack round will make the backdoor injected into the global model more quickly. However, too many attackers are unrealistic and may cause an obvious benign accuracy drop according to our exploration shown in Fig. 13.

### C. Proof of Theorem 1

In this part, we show the proof of the Theorem 1.

*Proof.* Based on Equ. 6, we use $\Delta\overline{w}^t$ for the expectation of the local updates from the benign participants in round $t^{th}$. The 'healthy' model trained by the rest benign participants is:

$$w^T_{healthy} = w^0 + \sum_{t=1}^{T} \underbrace{\frac{1}{|\mathbf{S}^t| - |\mathbf{A}^t|} \sum_{k \in \mathbf{S}^t \setminus \mathbf{A}^t} \Delta w^t_k}_{\Delta\overline{w}^t}. \tag{10}$$

We assume that the initialization model $w^0 = 0$, we can bound the difference between $w^T_{healthy}$ and $\widehat{w}$:

$$\begin{aligned} &|w^T_{healthy} - \widehat{w}| \\ &= \left| (1-\rho) w^0 + \sum_{t \in \mathbf{T}_a} \left( \frac{1}{|\mathbf{S}^t| - |\mathbf{A}^t|} - \frac{\rho}{|\mathbf{S}^t|} \right) \sum_{k \in \mathbf{S}^t \setminus \mathbf{A}^t} \Delta w^t_k \right| \\ &= \left| (1-\rho) w^0 + \sum_{t \in \mathbf{T}_a} \left[ 1 - \rho(1 - \frac{|\mathbf{A}^t|}{|\mathbf{S}^t|}) \right] \Delta\overline{w}^t \right| \\ &= \left| \sum_{t}^{\mathbf{T}_a} \left[ 1 - \rho(1 - \frac{|\mathbf{A}^t|}{|\mathbf{S}^t|}) \right] \Delta\overline{w}^t \right| \end{aligned} \tag{11}$$

Let $\Delta\overline{w} = \sup \Delta\overline{w}^t$, we can get the upper bound:

$$|w^T_{healthy} - \widehat{w}| \leq \sum_{t \in \mathbf{T}_a} \Delta\overline{w}^t \leq |\mathbf{T}_a| \Delta\overline{w}. \tag{12}$$

If $\rho = \frac{|\mathbf{S}^t|}{|\mathbf{S}^t| - |\mathbf{A}^t|}$, we can get the lower bound:

$$|w^T_{healthy} - \widehat{w}| \geq 0. \tag{13}$$

Hence, we bound the difference between the repaired model and the 'healthy' one, and finish the proof of Theorem 1.

$$0 \leq |w^T_{healthy} - \widehat{w}| \leq |\mathbf{T}_a| \Delta\overline{w}. \tag{14}$$

□