High-Confidence Computing 1 (2021) 100002

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.elsevier.com/locate/hcc

Towards multi-party targeted model poisoning attacks against federated learning systems

Zheyi Chen, Pu Tian, Weixian Liao*, Wei Yu

Department of Computer and Information Sciences, Towson University, MD 21252 USA

ARTICLE INFO

Keywords:
Adversarial federated learning
Perfect knowledge
Limited knowledge
Boosting strategy
High-confidence computing

ABSTRACT

The federated learning framework builds a deep learning model collaboratively by a group of connected devices via only sharing local parameter updates to the central parameter server. Nonetheless, the lack of transparency in the local data resource makes it prone to adversarial federated attacks, which have shown increasing ability to reduce learning performance. Existing research efforts either focus on the single-party attack with impractical perfect knowledge setting and limited stealthy ability or the random attack that has no control on attack effects. In this paper, we investigate a new multi-party adversarial attack with the imperfect knowledge of the target system. Controlled by an adversary, a number of compromised devices collaboratively launch targeted model poisoning attacks, intending to misclassify the targeted samples while maintaining stealthy under different detection strategies. Specifically, the compromised devices jointly minimize the loss function of model training in different scenarios. To overcome the update scaling problem, we develop a new boosting strategy by introducing two stealthy metrics. Via experimental results, we show that under both perfect knowledge and limited knowledge settings, the multi-party attack is capable of successfully evading detection strategies while guaranteeing the convergence. We also demonstrate that the learned model achieves the high accuracy on the targeted samples, which confirms the significant impact of the multi-party attack on federated learning systems.

1. Introduction

Recently, the proliferation of widely deployed smart devices gives rise to the increasing interest in accessing, extracting and sharing insightful knowledge by mining and learning from massive amounts of data collected from different places in a distributed manner [9,17,24–26,33,42]. In the realm of machine learning, based on learning parameters generated by distributed on-device datasets, federated learning is capable of establishing machine learning models without requiring the direct share of privacy-sensitive data [53]. It is significantly different from the centralized model training that requires the direct share of data from different places [45]. Iteratively, a central parameter server randomly selects a group of local learners[1], aggregates the parameter updates provided by selected local learners, and improves the global model training progressively [1,39]. During the training process, the local training data always resides in individual devices, allowing federated learning to train the model indirectly on privacy-sensitive data, including users' messages and patients' medical records, among others. Consequently, federated learning becomes a promising distributed learning paradigm by taking advantage of locally available computing ability with local private data.

Nonetheless, federated learning appears vulnerable to adversarial attacks. While training via the collaboration of a large number of learning nodes, it is hard for the central parameter server to design effective mechanisms to filter adversarial participants or malicious parameter updates during training process, because it is challenging to distinguish a well-manipulated model from a benign model, which are both trained on locally inaccessible data. Particularly, the central parameter server only has access to hundreds or thousands of similar local parameter updates generated by distributed learning nodes in a deep learning model, making the detection of malicious updates much challenging. Therefore, adversarial federated training models on learning nodes have introduced a new attack venue because learning nodes can be easily compromised and manipulated [3]. For example, compared to attacking a cloud server directly, the learning nodes with their local data can be more easily compromised and controlled by an external adversary [10]. When this occurs, compromised learning nodes can jointly submit malicious model updates that are trained on both benign and malicious

* Corresponding author.
E-mail addresses: zchen12@students.towson.edu (Z. Chen), ptian1@students.towson.edu (P. Tian), wliao@towson.edu (W. Liao), wyu@towson.edu (W. Yu).

[1] Note that local learner, learning node, learning agent, and client are interchangeable in this paper.

https://doi.org/10.1016/j.hcc.2021.100002
Received 9 December 2020; Accepted 31 January 2021

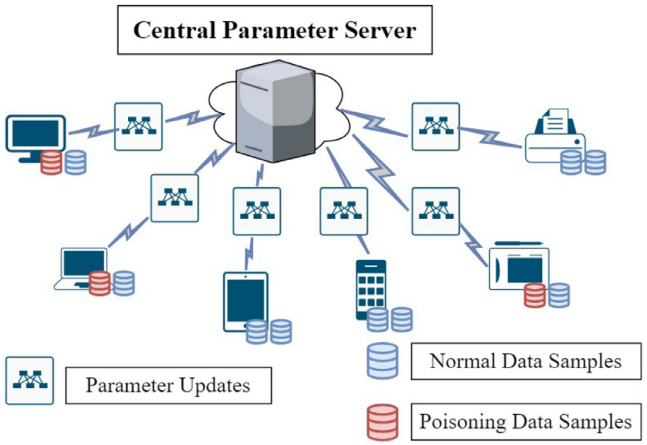

Fig. 1. An overview of adversarial federated learning model

data and intentionally mislead the global training model on the central parameter server.

All the above-mentioned characteristics of federated learning bring an emerging attack interface, called data poisoning attack. With such an attack interface, the adversary could compromise a group of local learners and then manipulate their local data samples to establish the targeted backdoor to affect the accuracy of the global learning model generated by federated learning. Moreover, the poisoning data samples would not incur much computation cost as the poisoning data samples could be directly created by using simple label flipping methods. In this paper, we investigate a challenging and timely-important adversarial federated learning problem, where in the training process, a group of learning nodes controlled by the adversary collaboratively launch a multi-party targeted model poisoning attack.

Fig. 1 illustrates an overview of system model. In particular, the central parameter server assigns a federated learning task to multiple learning nodes. Learning nodes report their local learning updates in each iteration, but never share their local data samples to the parameter server directly. As a multi-party attack, we assume that a number of learning nodes are compromised and inject poisoning data to manipulate the local data with adversarial labels during learning process. The single-node attack with the full information of targeted system (i.e., perfect knowledge) can achieve relatively remarkable performance [6]. Nonetheless, prefect knowledge requires that the adversary is capable of accessing any information over federated learning systems. Note that it is often impractical and unrealistic to assume that the adversary who has the prefect knowledge over a large-scale distributed system. Compared to the attack with prefect knowledge, the attack with limited knowledge is much closer to the real-world attacking practice. As an example, it is hard for the adversary to detect the data distribution over entire training dataset in order to mimic the benign training dataset.

Therefore, we consider the attack with either prefect knowledge or limited knowledge in our study. Comparably, a multi-party attack with limited knowledge is launched by multiple malicious learning nodes controlled by the adversary collaboratively. The intuition is that jointly learning to target a model will increase the possibility of a successful attack and reduce the possibility of being detected by existing anomaly detection mechanisms. The goals of attacks are to not only misclassify the global model with targeted samples (backdoor), but also ensure that the global model achieves convergence and high accuracy in its benign task.

To this end, we design the boosting strategy to achieve the model replacement in the targeted poisoning attack. Different from previous works, we study how to ensure that multiple malicious learning nodes can jointly launch the targeted poisoning attack in the federated learning process. Note that some potential detection mechanisms are provisioned in the federated learning process so that the adversary's malicious behavior can be limited or bounded [4,18]. In this sense, we formulate a joint optimization problem, in which multiple malicious learning nodes leverage the boosting strategies collaboratively so that the targeted poisoning attack can be achieved. The stealthy factors in the formulated problem include the clustering accuracy, cosine similarity and $\mathcal{L}_2$ norm, which enable malicious learning nodes to collaboratively choose diverse stealth metrics, avoid the parameter scaling problem, and assist in the poisoned parameter updates provided by the malicious learning nodes to circumvent the detection mechanisms. In comparison with the single-node attack, our investigated multi-party attack could collaboratively dominate the training process even if rigorous and multiple defense mechanisms are in place. Our investigated problem is a complex learning and joint optimization problem. To solve this problem, we use a gradient-based method to compute local updates. Then, the malicious learning nodes jointly find the boosting strategies in each iteration. Also, we conduct the convergence analysis of our investigated attack scheme and show that the convergence property of the training process is guaranteed.

To validate the effectiveness of the investigated new attack, in the experiments, we first craft the targeted poisoning data samples on both training and testing phases and set up the different scenarios. In real-world federated learning systems, both non-independent and identical distributed (non-i.i.d.) data are widely considered. The non-i.i.d. data distribution brings benefits to the adversary. This is because the non-i.i.d. data distribution makes the parameters of trained model become much different from a statistical view. Such a difference could reduce the effectiveness of detection mechanisms on the central parameter server. Thus, we choose i.i.d. data distribution to further limit the malicious behavior of adversary.

Moreover, we consider the different knowledge levels in the experiments. The experimental results show that the targeted model poisoning attack launched by a group of malicious nodes can successfully inject the back-door into the global model. In comparison with the single-party attack, our investigated attack achieves better performance on stealthy metrics. Furthermore, we show the difference between the single-node attack and the multi-party attack with a statistical view, which validates the feasibility of multi-party attack and calls for the attention in developing new detection and defense strategies for making federated learning systems secure.

We summarize our main contributions in this paper as follows:

- We formulate a joint optimization problem to assign our targeted poisoning attack over multiple malicious learning nodes. In the investigated problem, multiple malicious learning nodes collaboratively conduct the model replacement by determining local malicious updates and boosting strategies in each iteration.
- The proposed attack scheme includes diverse stealthy metrics to circumvent different detection mechanisms on the central parameter server. We also constrain the adversary with either perfect or limited knowledge level.
- Extensive evaluation results validate the capability and necessity of the multi-party attack under risk-sensitive settings in federated learning systems.

The remainder of this paper is organized as follows: In Section 2, we discuss the related work and highlight the key differences between state-of-the-art works and our work. In Section 3, we introduce the preliminary background of federated learning framework and detection mechanisms used by the system operator. In Section 4, we formulate a multi-party targeted model poisoning attack problem, introduce our designed proposed boosting strategy, and present the analysis of convergence. In Section 5, we conduct substantial experiments and show the effectiveness of our attack scheme compared to some representative baseline schemes. In Section 6, we discuss some open issues. Finally, we conclude the paper in Section 7.

2. Related Work

The federated learning decouples the need of storing data in a centralized manner and enables the distributed nodes to collaboratively learn a shared learning model without sharing their data directly [32]. Local learning nodes only exchanges parameters with the central parameter server that carries out parameter aggregation. As data is not shared with the central parameter service directly, the federated learning not only offers better privacy protection, but also reduces the amounts of data transmission over the network. Nonetheless, the federated learning is vulnerable to adversarial attacks because the parameter server has the loose control of local learning nodes. After a number of learning nodes are compromised by the adversary, they may either be injected with false data (i.e., data poisoning) or replaced with malicious model parameters (i.e., model poisoning). Some attacks have been studied in existing literature [13,15,50,51]. It is worth noting that false data injection attacks have been shown great impact on a variety of IoT-based smart-world systems, including the smart grid, smart transportation system, and smart manufacturing system, and among others [27–29,43,44,46].

Poisoning attacks aim at misclassifying the trained model by injecting attacker-specific samples [1,20,30,39]. In such an attack, the adversary first generates the well-crafted poisoning samples and then injected into the training dataset. By doing this, the adversarial learning model could capture the specific features for misclassification purposes. Nonetheless, such an attack might be difficult to achieve under a centralized training environment because most data centers fully check the collected data before it is used by machine learning algorithms. Existing research efforts have shown that Well-trained machine learning models are prone to adversarial attacks [19,22,34,36]. Similar to the data poisoning, an adversarial example is a well-crafted sample produced by adversaries and delicately designed to fool the machine learning model. On the application layer, the model vendor may not realize these tiny/marginal differences between the original sample and adversarial example, which could significantly reduce the robustness of machine learning models. For example, Jia et al. [19] investigated the attack that only could add a short sentence into the original paragraph and then make the machine model for reading comprehension system to completely fail to answer questions about the paragraph.

There are some research efforts on poisoning attacks on federated learning systems. For example, Biggio et al. [7] studied the poisoning attack against support vector machines, where attack is launched during the testing phase. Likewise, Bhagoji et al. [6] studied the attack scheme against federated learning and explored the targeted poisoning attack that was launched by a single, non-collusive malicious learning node. It is worth noting that the difference between their work and our work is the formulated problem, where a group of malicious learning nodes jointly launch the targeted poisoning attack in our study. Moreover, our experimental results show that multiple malicious learning nodes are capable of bypassing rigorous detection mechanisms. Additionally, Bagdasaryan et al. [3] studied both single and multiple malicious learning nodes to launch the targeted poisoning attack, which are performed at the convergence time of training phase. Yet, our investigated attack commits to making the targeted poisoning attack be performed in each possible iteration even when multiple detection mechanisms are in place.

Another line of research focuses on the Byzantine attack, in which an adversary injects corrupted data or controls the node to send arbitrary uploads to the central server [48]. When a poisoned local update is received by the server and applied in the global aggregation, it degrades the performance dramatically. Byzantine-resilience gradient aggregation is a security technique, developed as an alternative aggregation mechanism, to guarantee the convergence of the distributed machine learning with Byzantine nodes [2,8,47]. Nonetheless, some assumptions of above works in the Byzantine-resilience gradient aggregation conflict with the real-world federated learning practice. First, most Byzantine-resilience works assumed the i.i.d. data distribution. Second, the loss function in our work is non-convex. Third, multiple malicious nodes are compromised by the adversary, implying that these malicious learning nodes share the system information, not independently.

There are some other research efforts on applying federal learning to improve communication efficiency [21], deal with the heterogeneity [35] and the privacy protection of federated learning systems [12,31], among others. For example, Choudhury et al. [12] introduced a privacy-aware federated learning framework with two levels of privacy protection and evaluated the effectiveness of their proposed framework on healthcare dataset. Likewise, Ma et al. [31] studied the differential privacy against the data poisoning attack from both attack and defense perspectives. Their study demonstrated that the differential privacy could resist a number of poisoning samples in the training process and provided quantitative bounds on how much an adversary could change the distribution. Note that the differential privacy might be a promising solution to overcome data poisoning attacks, but these existing works do not consider our investigated attack scenario in this paper, such as launching both data and model poisoning attack via multiple malicious learning nodes.

Some research efforts have been conducted on designing defensive schemes. For example, in [40,41], authors enhanced the robustness to poisonous attacks by adding an accuracy check and filtering procedure at the server with a predefined distribution identical dataset. At the server side, the designed Zeno (++) scheme asked for sample data to obtain the anomaly score. Further, Zhao et al. [52] proposed the poisoning defense generative adversarial network (PDGAN) to defend against the attack. In their study, the GAN structure was used to reconstruct the training data based on received updates and each node's accuracy was audited with the generated data. Likewise, Wu et al. [38] proposed a geometric scheme to reduce the variance and increase the robustness to adversarial attacks. In addition, the existing works [3,6] showed that the basic model replacement could still successfully achieve attack objective under Byzantine-resilience gradient aggregation. The rationale behind this is the attack objective, which aims to achieve an acceptable accuracy over the normal testing dataset, as well as the targeted dataset. Nonetheless, these defensive approaches usually require the upper limit of compromised learning nodes. As the number of malicious nodes increases, the successful attack rate becomes higher. Thus, our investigated multiple-party attack is expected to achieve a better chance to bypass the defensive schemes.

3. Preliminary

In this section, we introduce the preliminary background for federated learning systems and some detection mechanisms used by the central parameter server.

3.1. Federated Learning Systems

Without loss of generality, in the federated learning system, we assume there are a number local data and computing resource provided by smart IoT devices as training nodes and one aggregator (i.e., the central parameter server). These training nodes are assigned to train the same machine learning task with locally collected data. We assume that there are N distributed nodes in the set $I = \{1, 2, \ldots, n, \ldots, N\}$ ($n \in [1, N]$), where node n collects local dataset, denoted as $\mathcal{D}_n = \left\{ d_1, d_2, \ldots, d_j, \ldots, d_{D_n} \right\}$. Here, D_n represents the number of data samples. The goal of a federated learning task is to obtain an optimal weight vector (i.e., $\mathbf{w}_G^*$), which minimizes the global loss function $F_G(\cdot)$ as follows:

$$\mathbf{w_G}^* = arg\ min\ F_G(\mathbf{w}). \tag{1}$$

To be specific, in each iteration t, a number of distributed learning nodes are selected to participate in the global aggregation. Each node n trains the received global weight vector $\mathbf{w}_G^{t-1}$ with E_n epochs to minimize the local loss function $F_n(.)$ over its own dataset $\mathcal{D}_n$. At the end of the

iteration t, each node has a local weight vector $\mathbf{w}_n^t$. Then, the local update $\boldsymbol{\delta}_n^t$ is sent back to the central parameter server, where $\boldsymbol{\delta}_n^t$ is computed as $\boldsymbol{\delta}_n^t = \mathbf{w}_n^t - \mathbf{w}_G^{t-1}$. On the central parameter server side, all received local updates are aggregated based on the weight averaging method, and we have

$$\mathbf{w}_G^t = \mathbf{w}_G^{t-1} + \eta \sum_{n=1}^{P} \frac{D_n}{D_P} \boldsymbol{\delta}_n^t. \tag{2}$$

Here, P ($P \subseteq I$) represents the nodes that participate in the global aggregation in each round and η is the learning rate on the central server. $\boldsymbol{\delta}_n^t$ is the n's local update at t's iteration. Once all nodes in the node set I participate in the global aggregation, the federated learning process is synchronous. We set $P = I$, if the federated learning process is synchronous.

3.2. Detection Mechanisms

In our study, we aim to study the malicious behavior of adversary when different detection mechanisms are in place. Particularly, we consider the potential detection mechanisms provisioned by the central parameter server, including accuracy checking and statistical metrics checking. For example, it could be a method similar to Reject on Negative Impact (RONI) [4] or TRIM [18]. In RONI, the central parameter server tests the impact of each local update and then block partial local updates, which show significant negative impact on the global learning model. TRIM aims to select a subset of all received local updates, which could minimize the loss function on the central parameter server. Note that it is hard to identify the most effective detection mechanisms, which could be used to verify the efficiency and effectiveness of the investigated attacks. Thus, we consider the following generic detection mechanisms that could limit the malicious behavior of the adversary in this paper.

Clustering Accuracy Checking Mechanism: In this mechanism, the server checks the validation accuracy of $\mathbf{w}_i^t = \mathbf{w}_G^{t-1} + \delta_i^t$. The model is obtained by adding the update from note i to the current state of the global model. If the resulting model has a validation accuracy that is much lower than that of the model obtained by aggregating all the other updates, $\mathbf{w}_{G\setminus i}^t = \mathbf{w}_G^{t-1} + \delta_{I\setminus i}^t$, the server can flag the update as being anomalous. With the adversary' view, it implies

$$F_{G\setminus m}^t(\mathcal{D}_{test}, \mathbf{w}_{G\setminus m}^t) - F_m^t(\mathcal{D}_{test}, \mathbf{w}_m^t) \leq \gamma^t, \tag{3}$$

where $\mathcal{D}_{test}$ represents testing datasets on the central parameter server, m represents malicious nodes in each iteration, and γ^t represents a predefined threshold to filter the received updates.

Statistical Checking Mechanism: With statistical checking mechanism, some potential statistical metrics can be considered by the central parameter server to measure the difference between given updates. Knowing that the update provided by each candidate is a weight vector with the same size, the central parameter server is supposed to measure its orientation and magnitude. To be specific, we assume that the central parameter exploits the $\mathcal{L}_2$ norm and the cosine similarity to measure the difference between two provided weight vectors. First, a suitable distance metric ($\mathcal{L}_2$ norm) is capable of measuring the magnitude of the update provided by each candidate, making the distance between the malicious update and the update provided by any other be limited in a given range. Similarly, the cosine similarity is a metric that is commonly used to measure the similarity of orientation.

4. A Multi-party Targeted Model Poisoning Attack

In this section, we introduce our proposed multi-party targeted model poisoning attack in detail. In particular, we first introduce our scheme and then conduct convergence analysis. We list the key notations in Table 1.

Table 1
List of key notations

N	Total number of nodes
D_n	Number of data samples on node n
$\mathbf{w}_G$	Global learning model
$\mathbf{w}_n$	Local learning model on node n
$\mathbf{w}^*$	Optimized weight parameter
t	Iteration
$\boldsymbol{\delta}_n$	Local update from node n
λ	Scaling factor used by boosting strategy
$F_n()$	Loss function on node n
$\mathcal{D}_{tar}$	Targeted data samples (pairs)
$S_{cos}(a, b)$	Cosine similarity between vector a and b
M	Malicious node set

4.1. Boosting-based Joint Optimization

Recall that compared to attack against a cloud server directly, the learning nodes with their local data can be compromised and completely controlled by an external adversary. We consider multi-party targeted attack in this paper. On one hand, it makes sense to consider the multi-party attack, because it is easy to see in real-world practice where the adversary could control multiple devices in the federated learning system. On the other hand, the multi-party attack is stronger and dangerous than the single-party attack. Compared to the single-party attack, the investigated multi-party attack is capable of not only being more stealthy, but also avoiding the single attack failure problem (i.e., single node will totally fail to carry out the attack task if that node is detected by the central parameter server).

To be more specific, the malicious learning task could be implemented by the multiple malicious learning nodes. Therefore, we assume that multiple learning nodes are compromised by the adversary so that the compromised nodes could collude with each other, controlled by the adversary. To launch an effective attack, we first introduce the attack goal that is to make the trained model misclassify with the targeted samples on the testing data. Traditional poisoning attacks mainly focus on the label flipping attack, in which the adversary compromises a node to inject poisoning data samples or tamper the local data samples with wrong labels [7]. Nonetheless, the existing works show that the pure label flipping attack could not achieve a good performance, even adversaries have already poisoned a large fraction of data samples [3]. It is worth noting that one challenge to launch an effective attack over a federated learning system is to overcome a scaling factor ($\alpha_n = \frac{D_n}{D_N}$), where the scaling factor represents the weight of each update in the aggregation. For example, the adversary seeks a fit method to make the malicious updates dominate the training process even with a number of benign nodes. The adversary therefore boosts the updates with a factor to improve the performance of the targeted poisoning attack, namely boosting strategy.

We now introduce how to obtain a malicious update with the boosting strategy. From the adversary's view, the targeted poisoning attack in the federated learning system is a learning problem, where the difference is the weight vectors that are trained by poisoning data. Thus, the malicious learning nodes train the received weight vector $\mathbf{w}_G^{t-1}$ to obtain the updated weight vector $\mathbf{w}_m^t$. Then, the update $\tilde{\delta}_m^t$ is computed by $\mathbf{w}_m^t - \mathbf{w}_G^{t-1}$. Nonetheless, the original update $\tilde{\delta}_m^t$ may not achieve a successful targeted poisoning attack, because updates provided by normal nodes with clean data lead to the scaling issue. To overcome this issue, before the original malicious update $\tilde{\delta}_m^t$ is sent to the central parameter server, we design the boosting strategy that makes these updates amplified by λ_m. An aggressive booting strategy may achieve a better performance on the targeted samples and make multiple malicious nodes learning dominate the training process, but it also increases the risk to be detected by the central parameter server.

Thus, the targeted poisoning attack aims to exploit multiple learning nodes, making the learned model converge to a point with a good performance and avoiding being detected by the detection mechanisms. To be specific, both its normal task and the targeted objective can not only achieve a high accuracy over the testing dataset, but also the proposed attack scheme satisfies stealthy requirements. Overall, we show that there are three terms in Equation (4), which consider explicit boosting and two stealthy metrics.

$$\begin{aligned} \min_{\lambda,\delta} \sum_{m=1}^{M} & \lambda_m F_m\left(\mathcal{D}_{tar}, \mathbf{w}_G^t\right) + F_m\left(\mathcal{D}_m, \mathbf{w}_m^t\right) \\ & + p\left(1 - S_{\cos}\left(\delta_m^t, \overline{\delta}_{I\setminus m}^t\right)\right), \\ & s.t. \quad \mathcal{L}_2\left(\delta_m^t, \delta_n^t\right) \le C, \quad \forall n \in I \setminus M. \end{aligned} \tag{4}$$

Here, p represents a distance factor, $S_{cos}(\cdot)$ represents the cosine similarity between two given vectors, C represents a predefined threshold in the attack scenario, and $\mathcal{D}_{tar}$ represents the target samples (pairs) selected by the adversary, respectively. Finally, from the adversary's perspective, we translate the multi-party targeted poisoning attack to the formulated problem, allowing multiple malicious nodes cooperate to realize diverse stealthy metrics.

Based on the formulated problem, we now discuss the impact of the constraints and the different knowledge levels. As the federated learning framework works in a distributed environment, this nature makes the adversary not have access to a large number of local samples. We then translate this nature to be a mathematical constraint in our work, where we have a number of targeted poisoning data samples D_{tar} on each malicious learning node. We notice that the formulated problem is a complex learning problem, which also includes a vector $\boldsymbol{\lambda}$ to denote the boosting strategy for a group of malicious learning nodes.

To solve this problem, we assume that each learning node has limited computing resources. Thus, the epoch E_i and batch size B_i are constant values. Due to the inherent complexity of Equation (4) (i.e., the most of loss function in the machine leaning field are non-convex), we empirically set up a discrete space for λ_m to control the boosting strategy on each malicious learning node. Also, the malicious learning nodes use a gradient-based method to find local updates over the poisoning datasets. Therefore, the adversary manipulates a group of compromised learning nodes, changing the boosting strategies in each training iteration to inject the targeted poisoning features into the global training model. In this sense, the proposed attack scheme may not be the optimal solution, but it clearly shows that the group of malicious learning nodes could achieve the attack goal in the federated learning system.

We then explain the impact of the different knowledge level on the federated learning system. In the worst-case scenario, the adversary has the full knowledge of the targeted system, namely the attack with perfect knowledge, including the entire training dataset, weight vectors provided by other local learners, parameter aggregation algorithm, and other system information. In contrast, the adversary could launch the poisoning attack with limited knowledge where this condition is satisfied in many real-world scenarios. To be specific, the adversary's malicious behavior is constrained; for example, only a partial dataset is exposed to the adversary.

4.2. Convergence Analysis

We now a brief convergence analysis for the proposed attack schema to show that the convergence could still be guaranteed in the training phase. We make two assumptions about the loss function $\mathbf{w}$: (i) *convex*, $F(\alpha\mathbf{w} + (1-\alpha)\mathbf{w}') \le \alpha F(\mathbf{w}) + (1-\alpha)F(\mathbf{w}')$ for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$ and $\alpha \in [0,1]$; (ii) *β-smooth*, $||F(\mathbf{w}) - F(\mathbf{w}')|| \le \beta||\mathbf{w} - \mathbf{w}'||$ for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$. With the previous assumptions, we now discuss the convergence of the loss function in Theorem 1.

Theorem 1. *For the learning rate $\eta \le \frac{1}{\beta}$, we can have $||F(\mathbf{w}^{t+1}) - \mathbf{w}^*||^2 \le ||F(\mathbf{w}^t) - \mathbf{w}^*||^2$, where $F(\mathbf{w}^t)$ denotes global function at the t^{th} iteration and $\mathbf{w}$ is the weight parameter.*

Proof. The key to prove Theorem 1 is to show that the loss function at the $(i+1)^{th}$ iteration is closer than its previous iteration t. To express the relationship between $F(\mathbf{w}^{t+1})$ and $F(\mathbf{w}^t)$, we first use the stochastic gradient descent (SGD) rule. Then, we rearrange the equation and obtain the expansion using the squared norm. By applying the *β-smooth* property, we accomplish our proof listed below.

$$\begin{aligned} & ||F(\mathbf{w}^{t+1}) - \mathbf{w}^*||^2 \\ & = ||F(\mathbf{w}^t) - \eta\nabla F(\mathbf{w}^t) - \mathbf{w}^*||^2 \\ & = ||(F(\mathbf{w}^t) - \mathbf{w}^*) - \eta\nabla F(\mathbf{w}^t)||^2 \\ & = ||(F(\mathbf{w}^t) - \mathbf{w}^*)||^2 - 2\eta\nabla F(\mathbf{w}^t)^T(F(\mathbf{w}^t) - \mathbf{w}^*) + \eta^2||\nabla F(\mathbf{w}^t)||^2 \\ & < ||(F(\mathbf{w}^t) - \mathbf{w}^*)||^2 - \eta\frac{||\nabla F(\mathbf{w}^t)||^2}{\beta} + \eta^2||\nabla F(\mathbf{w}^t)||^2 \\ & = ||(F(\mathbf{w}^t) - \mathbf{w}^*)||^2 - \eta(\tfrac{1}{\beta} - \eta)||\nabla F(\mathbf{w}^t)||^2. \end{aligned}$$

Then, if $\eta \le \frac{1}{\beta}$ satisfies, we have that,$||F(\mathbf{w}^{t+1}) - \mathbf{w}^*||^2 \le ||F(\mathbf{w}^t) - \mathbf{w}^*||^2$ and thus completes the proof. □

In our work, we aim to mislead the model to targeted incorrect labels. The loss function is trained to converge with injected false data samples. In other words, we could guarantee that the loss function $F(\mathbf{w})$ decreases as the iteration t grows when the loss function is *convex* and β-smooth, as shown in Theorem 1.

5. Performance Evaluation

In this section, we conduct experiments to evaluate the performance of our proposed attack scheme under different scenarios, such as perfect knowledge (PK) vs. limited knowledge (LK) level, along with i.i.d. data distribution, and others. In the following, we first present the evaluation methodology and then describe evaluation results.

5.1. Evaluation Methodology

Dataset: In this paper, we use the MNIST 10-digit handwritten dataset [23] (including a training set of 60,000 samples and a testing set of 10,000 samples) to validate the efficacy of our investigated attack scheme. To simulate the attack, we choose 1048 samples in the training set to create the target poisoning data. In this paper, we flip the sample label '5' to '7' as our target. A targeted poisoning testing dataset (800 samples) is created, where we consider '5' is misclassified to '7' as a correct classification. In practice, the adversary could choose arbitrary samples by injecting crafted poisoning data as their targets [23].

Note that the adversary could choose arbitrary data samples or inject any well-crafted poisoning data to targeted backdoors. In this study, we focus on a generic federated learning framework that can be applicable to IoT systems, where a number of IoT devices could contribute their local computing resources and data to train the machine learning model and update the training model parameters to the aggregator. To this end, we use MNIST 10-digital handwriting dataset that is a representative dataset in machine learning field, as an example to demonstrate our idea. It is worth noting the automatic handwriting recognition system can be considered a form of computer vision-based IoT system, in which a number of digital handwriting recognition devices can be deployed as IoT sensors to capture the samples of handwriting from different locations. As future research, we plan to consider additional IoT scenarios and explore relevant IoT datasets to validate the efficacy of our approach further.

Deep learning model: To demonstrate the efficacy of our approach, we consider to deploy the Convolutional Neural Network (CNN) model on all nodes. In the deployed CNN mode, there are two 5x5 convolutional layers (32 channels in the first layer and 64 channels in the second layer). After each layer, a 2x2 maximum pooling layer, a fully connected layer with 512 units of the *ReLu* activation function, and a final soft-max output layer (1,663,370 parameters) are used [37].

Knowledge level: The more an adversary knows the system, the higher the impact he or she could compromise the system through attacks. In this paper, we conduct the experiments under different knowledge levels. First, the attack is launched with prefect knowledge, where

weight vectors provided by other candidates and the number of data samples on the other nodes is accessible by the adversary. By contrast, we also evaluate the attack with limited knowledge, where some of updates from normal nodes are accessible by the adversary. Note that it is easy to implement such an attack with limited knowledge. Consider that an adversary could compromise a number of nodes in the network: the adversary first injects the poisoning data on these compromised nodes. Then, the poisoning attack is launched through these compromised nodes with the poisoning data, while the adversary could receive the updates, which are provided by all compromised learning nodes.

Baseline schemes for comparison: To further demonstrate the effectiveness of our investigated attack scheme, we consider the following two baseline schemes in our experiments: (i) *Single naive attack with $\mathcal{L}_2$ norm constraint (SNAL2):* In this scheme, we only set up one malicious learning node in the entire training process. Then, this malicious learning node launches the attack with $\mathcal{L}_2$ *norm* constraint. This baseline scheme is to show that the multiple malicious learning nodes have the ability of achieving a better performance on the adversary's main target with the same distance constraint [3]. (ii) *Single attack node without any distance constraint (SAnoDC):* Similar to the previous one, we relax the distance constraint in this baseline scheme, in which the malicious learning node could choose any booting strategy to launch the attack. This baseline scheme is based on the one in [6]. The stealthy performance of this baseline scheme is collected, once this baseline scheme achieves a similar performance with our scheme [3].

Performance metrics: We consider the following performance metrics for accuracy: (i) *Accuracy (global)*: It is defined as the ratio of the number of correct predictions and the total number of predictions on testing dataset. (ii) *Accuracy (targeted sample)*: It has the same definition as global accuracy, but it works on the targeted testing dataset only.

To measure the stealthy behavior of the investigated attack scheme, we also consider the following stealthy metrics: (i) *Accuracy checking (stealthy metric):* With this metric, the aggregator compares the accuracy on testing dataset for the weight vector from the selected learning node i and the weight vector aggregated by others. (ii) *Cosine similarity (stealthy metric):* With this metric, we show the value of cosine similarity between the the weight vector from the malicious learning node and the weight vector aggregated from other nodes. (iii) $\mathcal{L}_2$ *norm (stealthy metric):* With the training process, we record the maximal euclidean distance between malicious learning nodes and normal learning nodes.

In this paper, we investigate the federated learning performance with the number of learning nodes $N = 10$ and the number of malicious nodes $M = 2$. We empirically set a discrete space as the boosting strategy for malicious learning nodes. In most real-world scenarios, the federated learning system trains the global model over non-i.i.d. data distribution, in which all received updates make the central parameter server be a challenge to find an advisable tolerance. Nonetheless, non-i.i.d. data distribution provides the significant advantage to the adversary. For example, the central parameter server is supposed to predetermine a larger threshold on accuracy checking and statistical checking. Thus, we set up the i.i.d. data distribution in this paper, where the original dataset is shuffled, and then partitioned into 10 clients (each node has 6000 training samples). Moreover, we conduct other groups of federated learning process to show the advantage of multi-party attack with statistical views. In these groups, the central parameter server trains the global learning model with 20 local nodes, where we also set up the ratio of malicious node as 0.1 (10 nodes and 1 malicious node), 0.2, 0.3 and 0.4, respectively. The global accuracy of these learning models converges to an acceptable point and the attack tasks achieve a relative high accuracy (over 90% accuracy).

5.2. Evaluation Results

We first conduct our experiments under the attack with prefect knowledge and i.i.d. data distribution. The experiments begin with a synchronous fashion, in which the central parameter server aggregates

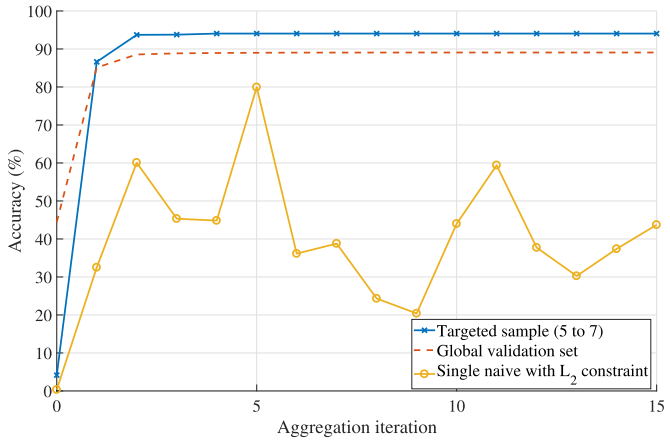

Fig. 2. Performance comparison for targeted and global accuracy with perfect knowledge

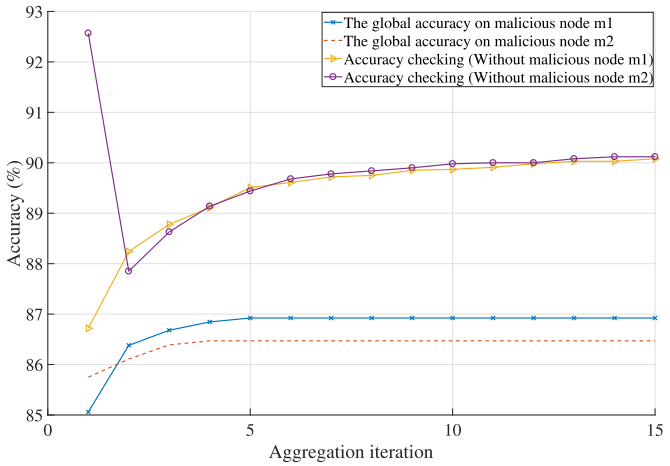

Fig. 3. Performance comparison for perfect knowledge with accuracy checking

all received updates. The aggregated global weight vector is tested over both the normal testing dataset and the targeted dataset. In addition to the performance on the targeted samples, the adversary also concerns the stealthy performance in order to avoid being detected. The reason is that the trained model could be worthless, if the malicious nodes are detected by the central parameter server.

Fig. 2 shows that the proposed attack achieves around 93% accuracy on the main task. We can also observe that the poisoned global model converges to a good point (around 90%) on the normal testing dataset. The results for baseline scheme SNAL2 in both Fig. 2 and Fig. 4 show that the trained global model cannot converge to a good point over the targeted testing dataset. This is because distance constraint makes single malicious node fail to poison the global model and keep stealthy simultaneously. In contrast, we limit the level of knowledge on the adversary where we randomly provide the update information from three normal nodes. The rationale behind the achieved performance is that our investigated attacking scheme successfully overcomes the normal updates provided by other candidates.

Fig. 3 represents the clustering accuracy checking under the attack with prefect knowledge, where we test the global accuracy over the aggregated model without two malicious node independently. In Fig. 3, each aggregated model without one malicious learning node achieves better accuracy over the normal testing data, but the update provided by each malicious learning node also achieves a relative acceptable performance over the same testing dataset. It is acceptable by the central parameter server, because the nature data distribution may contribute

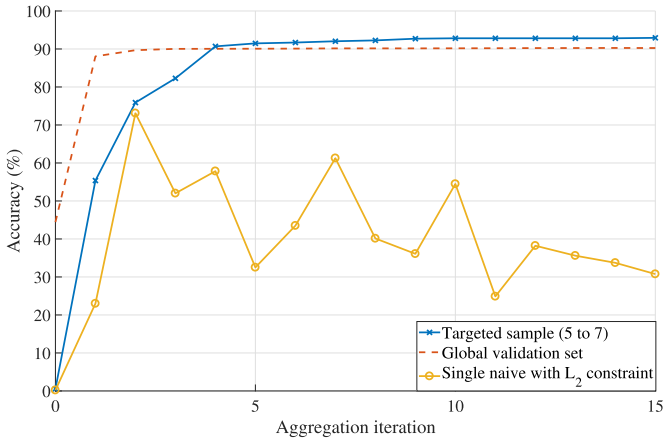

Fig. 4. Performance comparison for targeted and global accuracy with limited knowledge

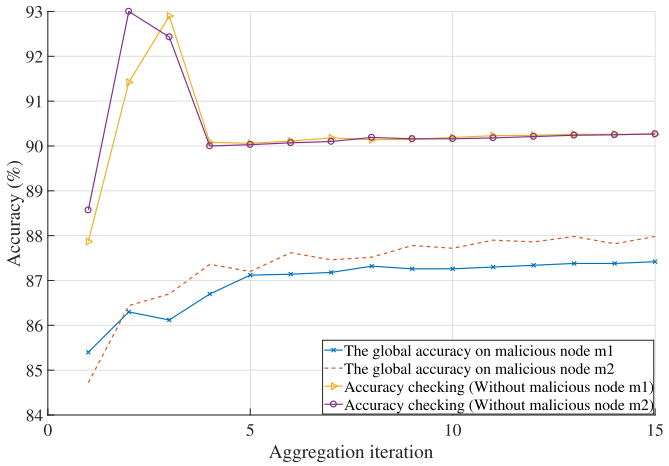

Fig. 5. Performance comparison for limited knowledge with accuracy checking

these performance variation between any other two nodes. Moreover, the risk of false detection, flagging a normal node as malicious one, is increased, if the central parameter server sets up a small predefined tolerance. For completeness, we show the experimental results under the limited knowledge case with i.i.d. data distribution. As shown in Fig. 4, the adversary achieves around 90% accuracy on the targeted testing dataset. The performance of the proposed attack scheme degrades around 3% under the attack with limited knowledge, because the adversary lacks of adequate information to accurately estimate the global aggregated update. Similar to Fig. 3, Fig. 5 shows that both malicious learning nodes (m_1 and m_2) have an acceptable stealthy performance on the clustering accuracy checking. These results confirm that the adversary could significantly improve the stealthy performance under both the perfect and limited knowledge cases by adding the normal training data in the training process.

To further explain the advance of the proposed attack scheme, we use baseline scheme SAnoDC to confirm more stealthy performance of the proposed attack scheme. In the baseline scheme SAnoDC, we set up one malicious node to launch the attack. Then, the two statistical metrics, including the cosine similarity and the $\mathcal{L}_2$ norm, are collected in Table 2 and Table 3, respectively. Note that the baseline scheme achieves a similar performance on the targeted dataset. In Table 2, we compute the cosine similarity between the update of each malicious and the update aggregated by other node without that malicious node. Compared to the baseline scheme, our proposed attack scheme has a higher cosine similarity, confirming that the multiple malicious learning nodes could successfully fool the central parameter server. The rationale behind these results is that the regular detection mechanisms such as clustering accuracy and cosine similarity have no ability to detect multiple malicious learning nodes in the training process. To be specific, these detection mechanisms fail to flag malicious nodes, once the malicious can inject the malicious feature into both checked objective and contrast one.

In Table 3, we collect the maximal $\mathcal{L}_2$ norm between the update of each malicious node and the update provided by each normal node. Our proposed attack scheme shows a lower $\mathcal{L}_2$ norm, which represents the updates provided by our attack scheme to mimic the normal update on the magnitude. This is because our proposed attack scheme is able to implement the model replacement by multiple malicious learning nodes. It also implies that the stealthy performance is related to the attack resources (e.g., the number of malicious learning nodes and the number of poisoning samples) on the adversary side.

We notice another regular setup in the federated learning process where the central parameter server randomly chooses a fraction of candidates to participate in the global aggregation. In this case, even multiple malicious nodes cannot guarantee to be selected in each iteration; however, the adversary still concerns about the performance of the targeted poisoning attack. We therefore conduct the experiment to simulate the aforementioned scenario, in which the single round attack is launched in the round $t = 1$, as shown in Fig. 10. We see that the proposed attack scheme achieves a significant higher performance over the targeted testing dataset immediately and then the performance is degraded by other learning nodes in next rounds. In other words, the proposed attack scheme can be launched at any given time.

Finally, we provide a statistical view to show the difference in update distributions between the single-node attack and the multi-node attack. From Fig. 6 to Fig. 9, we increase the ratio of malicious learning nodes (10%, 20%, 30%, and 40%) in the federated learning process, in which each group of learning process achieves a good convergence on global accuracy and a high accuracy on the adversary's desired targets. Note that the attack tasks achieve with over 90% accuracy. We then sample the average of normal updates and malicious updates to draw the histograms in iteration $t = 5$, $t = 10$ and $t = 15$ for each federated learning group. The results show that update in the multi-party attack makes itself be similar to the normal updates. Particularly, the results in Fig. 9 confirm that the multi-party malicious learning nodes could mimic the normal updates as well as successfully achieve the malicious objective. All these results validate the necessity of our attack scheme in which the multi-party attack could successfully inject back-doors in the federated training system while remain stealthy.

6. Discussion

Our experimental results have confirmed that our proposed multi-party attack can inject the targeted backdoor into the global learning model in federated learning over several different scenarios. In the following, we briefly discuss several open issues as future research directions.

Multiple Attack Objectives: In this paper, we focus on the targeted poisoning attack over the federated learning framework. We find that the federated learning framework also raises a new attack interface, in which the adversary may have multiple attack objectives. For instance, the adversary intends to degrade the global performance (accuracy) of the learning model [14]. In such a case, the adversary prevents the central server from reaching a good global accuracy over the training process. In such an attack, the central parameter server pays attention to the attack behaviors, while performing active defense strategies to alleviate the attack effect (e.g., blocking or rejecting the updates provided by subsets of local learners that looks suspicious). To this end, the adversary can evolve his or her attack strategies to circumvent or mislead the parameter server. To be specific, we consider the new attack scheme that could compute the perturbation range on each dimension in local datasets. Then, by crafting malicious values in the perturbation range,

Table 2
Stealthy performance (Cosine Similarity) under different cases

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Cos sim (PK m_1)	0.60	0.38	0.41	0.40	0.41	0.39	0.52	0.41	0.46	0.41	0.44	0.68	0.42	0.37	0.42
Cos sim (PK m_2)	0.59	0.37	0.41	0.43	0.41	0.45	0.51	0.46	0.47	0.43	0.42	0.68	0.43	0.47	0.53
Cos sim (SAnoDC)	0.56	0.31	0.35	0.29	0.26	0.24	0.24	0.25	0.21	0.29	0.26	0.28	0.18	0.12	0.18
Cos sim (LK m_1)	0.42	0.34	0.39	0.41	0.33	0.31	0.29	0.28	0.32	0.31	0.34	0.30	0.28	0.27	0.3
Cos sim (LK m_2)	0.41	0.33	0.39	0.42	0.33	0.32	0.29	0.29	0.34	0.31	0.36	0.30	0.28	0.28	0.32

Table 3
Stealthy performance (Maximal $\mathcal{L}_2$ norm) under different cases

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Max $\mathcal{L}_2$ (PK)	13.4	12.2	13.9	13.5	13.9	11.4	11.6	10.4	11.0	8.2	8.5	7.7	7.9	8.5	7.6
Max $\mathcal{L}_2$ (SAnoDC)	38.2	40.0	39.0	27.8	23.2	23.7	19.7	21.4	18.0	15.3	15.7	15.5	15.5	18.3	13.2
Max $\mathcal{L}_2$ (LK)	12.4	11.6	10.3	8.4	7.8	7.1	6.7	7.1	6.6	5.5	6.3	5.8	6.3	5.9	6.0

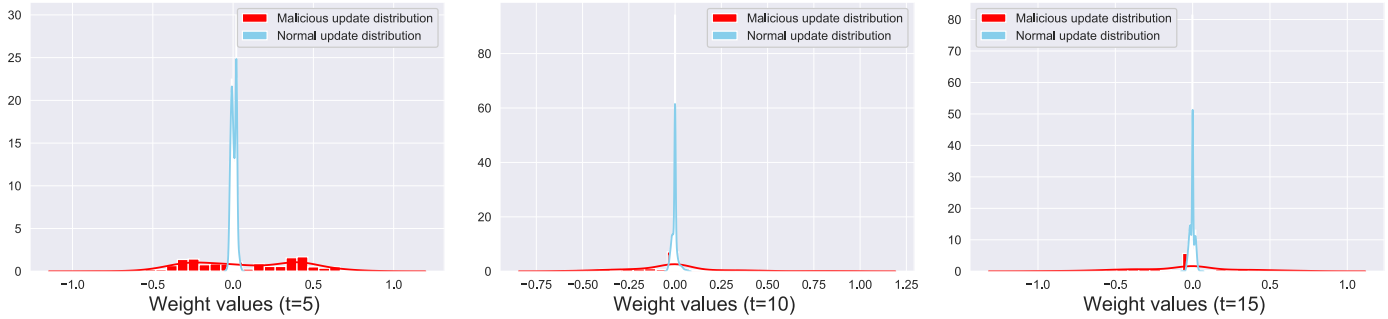

Fig. 6. Comparison of visualized weight update distributions between normal updates and malicious updates (ratio = 0.1)

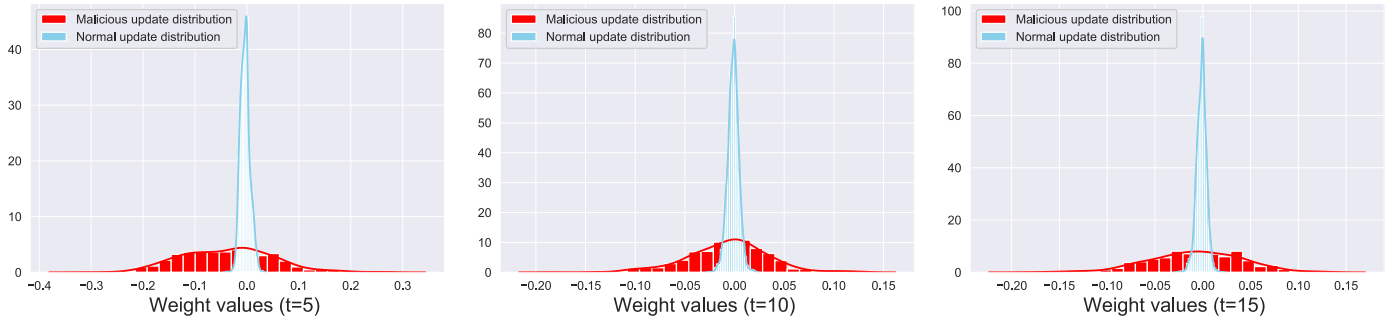

Fig. 7. Comparison of visualized weight update distributions between normal updates and malicious updates (ratio = 0.2)

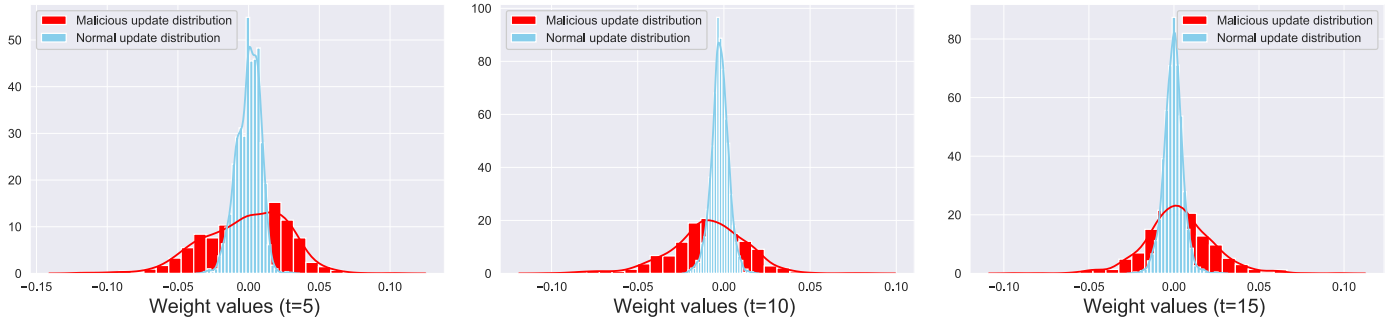

Fig. 8. Comparison of visualized weight update distributions between normal updates and malicious updates (ratio = 0.3)

the adversary could successfully achieve the attack objectives (e.g., preventing convergence and targeted backdoors).

Defense Approaches: Anomaly detection methods based on statistical inference have been applied to detect unknown threats to cybersecurity in a variety of systems [16,49]. Nonetheless, some existing attack schemes demonstrated the intelligence to circumvent the existing defense approaches [5]. To tackle these attacks over the federated learning system, we first consider the trustworthiness as a promising solution. To be specific, we shall systematically investigate and design schemes that enable the trusted evaluation of all components in the federated

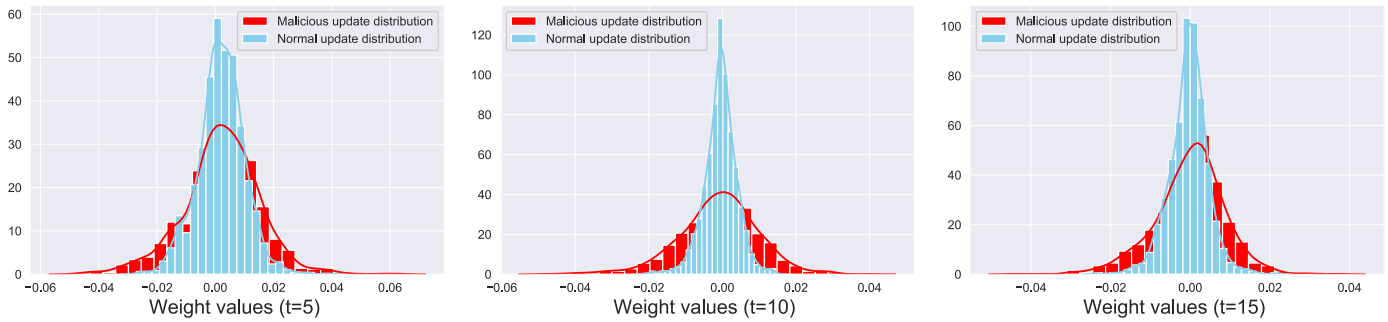

Fig. 9. Comparison of visualized weight update distributions between normal updates and malicious updates (ratio = 0.4)

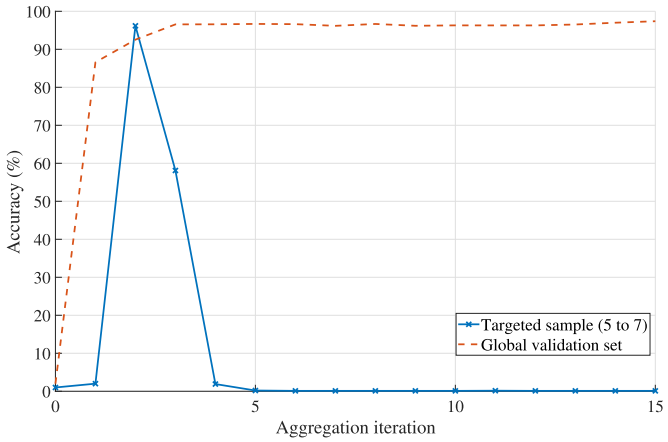

Fig. 10. The accuracy performance of a single round attack

learning system as well as the transmitted and received information. By leveraging this, the central parameter server (i.e., learning aggregator) can have an overall trust score of individual learning nodes and then making decision (e.g., accepting or rejecting the updates) in the training process. As another defense, clustering-based schemes could be leveraged to improve the security and reliability of the federated learning process [11]. For example, we shall consider developing clustering-based schemes on each dimension of the received update that makes the server block the perturbation range targeted by malicious learning nodes.

7. Final Remarks

In this paper, we have studied the problem of a collaborative attack on federated learning systems. The data privacy protection, which is enabled by sharing only model parameter updates, makes federated learning inevitably vulnerable to attacks. We have investigated a new multi-party model poisoning scheme, which enables multiple collusive attack entities (i.e., a group of compromised learning nodes) to mislead the training process to some targeted malicious labels. In particular, a group of compromised learning nodes collaboratively solve a joint minimization problem, in which a malicious training model is learned iteratively on both benign and malicious data while being limited to diverse knowledge levels for parameter updates by other learning nodes. To further improve the effectiveness of the investigated attack scheme, we have developed a boosting strategy that allows multiple malicious learning nodes to choose diverse stealth metrics and overcomes the update scaling problem. The experimental results show that under different knowledge levels and data distributions, our investigated multi-party targeted model attack can be carried out successfully and outperforms existing attacks schemes, which confirms the feasibility of multi-party attack on federated learning systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Alfeld, X. Zhu, P. Barford, Data poisoning attacks against autoregressive models, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[2] D. Alistarh, Z. Allen-Zhu, J. Li, Byzantine stochastic gradient descent, in: Advances in Neural Information Processing Systems, 2018, pp. 4613–4623.
[3] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, arXiv preprint arXiv:1807.00459 (2018).
[4] M. Barreno, B. Nelson, A.D. Joseph, J.D. Tygar, The security of machine learning, Machine Learning 81 (2) (2010) 121–148.
[5] G. Baruch, M. Baruch, Y. Goldberg, A little is enough: Circumventing defenses for distributed learning, in: Advances in Neural Information Processing Systems, 2019, pp. 8632–8642.
[6] A.N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: International Conference on Machine Learning, 2019, pp. 634–643.
[7] B. Biggio, B. Nelson, P. Laskov, Poisoning attacks against support vector machines, in: Proceedings of the 29th International Coference on International Conference on Machine Learning, Omnipress, 2012, pp. 1467–1474.
[8] P. Blanchard, R. Guerraoui, J. Stainer, et al., Machine learning with adversaries: Byzantine tolerant gradient descent, in: Advances in Neural Information Processing Systems, 2017, pp. 119–129.
[9] Z. Cai, Z. He, Trading private range counting over big iot data, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 144–153, doi:10.1109/ICDCS.2019.00023.
[10] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv preprint arXiv:1712.05526 (2017).
[11] Z. Chen, P. Tian, W. Liao, W. Yu, Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning, IEEE Transactions on Network Science and Engineering (2020) 1.
[12] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, A. Das, Differential privacy-enabled federated learning for sensitive health data, 2019.
[13] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, L. Song, Adversarial attack on graph structured data, arXiv preprint arXiv:1806.02371 (2018).
[14] E.M. El Mhamdi, R. Guerraoui, S. Rouault, The hidden vulnerability of distributed learning in Byzantium, in: Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, 80, PMLR, 2018, pp. 3521–3530.
[15] J. Gao, J. Lanchantin, M.L. Soffa, Y. Qi, Black-box generation of adversarial text sequences to evade deep learning classifiers, in: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, 2018, pp. 50–56.
[16] X. Gu, L. Akoglu, A. Rinaldo, Statistical analysis of nearest neighbor methods for anomaly detection, in: Advances in Neural Information Processing Systems, 2019, pp. 10921–10931.
[17] W.G. Hatcher, W. Yu, A survey of deep learning: Platforms, applications and emerging research trends, IEEE Access 6 (2018) 24411–24432, doi:10.1109/ACCESS.2018.2830661.
[18] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, B. Li, Manipulating machine learning: Poisoning attacks and countermeasures for regression learning, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, pp. 19–35.
[19] R. Jia, P. Liang, Adversarial examples for evaluating reading comprehension systems, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2021–2031.
[20] W. Jiang, H. Li, S. Liu, X. Luo, R. Lu, Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles, IEEE Transactions on Vehicular Technology (2020).
[21] J. Konecny, H.B. McMahan, F.X. Yu, P. Richtarik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, NIPS Workshop on Private Multi-Party Machine Learning, 2016.

[22] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: Artificial Intelligence Safety and Security, Chapman and Hall/CRC, 2018, pp. 99–112.
[23] Y. LeCun, C. Cortes, C.J. Burges, The MNIST database of handwritten digits, 2009, 2009.
[24] H. Li, K. Ota, M. Dong, Learning iot in edge: Deep learning for the internet of things with edge computing, IEEE network 32 (1) (2018) 96–101.
[25] F. Liang, C. Qian, W.G. Hatcher, W. Yu, Search engine for the internet of things: Lessons from web search, vision, and opportunities, IEEE Access 7 (2019) 104673–104691, doi:10.1109/ACCESS.2019.2931659.
[26] F. Liang, W. Yu, D. An, Q. Yang, X. Fu, W. Zhao, A survey on big data market: Pricing, trading and protection, IEEE Access 6 (2018) 15132–15154, doi:10.1109/ACCESS.2018.2806881.
[27] J. Lin, W. Yu, N. Zhang, X. Yang, L. Ge, Data integrity attacks against dynamic route guidance in transportation-based cyber-physical systems: Modeling, analysis, and defense, IEEE Transactions on Vehicular Technology 67 (9) (2018) 8738–8753.
[28] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, IEEE Internet of Things Journal 4 (5) (2017) 1125–1142.
[29] X. Liu, C. Qian, W.G. Hatcher, H. Xu, W. Liao, W. Yu, Secure internet of things (iot)-based smart-world critical infrastructures: Survey, case study and research opportunities, IEEE Access 7 (2019) 79523–79544.
[30] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, X. Zhang, Trojaning attack on neural networks, 2018, doi:10.14722/ndss.2018.23300.
[31] Y. Ma, X. Zhu, J. Hsu, Data poisoning against differentially-private learners: Attacks and defenses, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 4732–4738.
[32] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, 54, 2017, pp. 1273–1282.
[33] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for iot big data and streaming analytics: A survey, IEEE Communications Surveys Tutorials 20 (4) (2018) 2923–2960, doi:10.1109/COMST.2018.2844341.
[34] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E.C. Lupu, F. Roli, Towards poisoning of deep learning algorithms with back-gradient optimization, in: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 27–38.
[35] J. Pang, Y. Huang, Z. Xie, Q. Han, Z. Cai, Realizing the heterogeneity: A self-organized federated learning framework for iot, IEEE Internet of Things Journal (2020) 1, doi:10.1109/JIOT.2020.3007662.
[36] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2016, pp. 372–387.
[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.
[38] Z. Wu, Q. Ling, T. Chen, G.B. Giannakis, Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks, arXiv preprint arXiv:1912.12716 (2019).
[39] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, F. Roli, Is feature selection secure against training data poisoning? in: International Conference on Machine Learning, 2015, pp. 1689–1698.
[40] C. Xie, O. Koyejo, I. Gupta, Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance, arXiv preprint arXiv:1805.10032 (2018).
[41] C. Xie, O. Koyejo, I. Gupta, Zeno++: Robust fully asynchronous {sgd}, 2020.
[42] H. Xu, W. Yu, D. Griffith, N. Golmie, A survey on industrial internet of things: A cyber-physical systems perspective, IEEE Access 6 (2018) 78238–78259.
[43] H. Xu, W. Yu, X. Liu, D. Griffith, N. Golmie, On data integrity attacks against industrial internet of things, in: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 2020, pp. 21–28, doi:10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00020.
[44] Q. Yang, L. Chang, W. Yu, On false data injection attacks against kalman filtering in power system dynamic state estimation, Security and Communication Networks 9 (9) (2016) 833–849, doi:10.1002/sec.835.
[45] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (2) (2019) 12.
[46] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, W. Zhao, On false data-injection attacks against power system state estimation: Modeling and countermeasures, IEEE Transactions on Parallel and Distributed Systems 25 (3) (2014) 717–729.
[47] D. Yin, Y. Chen, K. Ramchandran, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in: International Conference on Machine Learning, 2018, pp. 5636–5645.
[48] D. Yin, Y. Chen, K. Ramchandran, P. Bartlett, Defending against saddle point attack in byzantine-robust distributed learning, arXiv preprint arXiv:1806.05358 (2018).
[49] W. Yu, D. Griffith, L. Ge, S. Bhattarai, N. Golmie, An integrated detection system against false data injection attacks in the smart grid, Security and Communication Networks 8 (2) (2015) 91–109, doi:10.1002/sec.957.
[50] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: Attacks and defenses for deep learning, IEEE transactions on neural networks and learning systems 30 (9) (2019) 2805–2824.
[51] X. Zeng, C. Liu, Y.-S. Wang, W. Qiu, L. Xie, Y.-W. Tai, C.-K. Tang, A.L. Yuille, Adversarial attacks beyond the image space, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4302–4311.
[52] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, S. Yu, Pdgan: A novel poisoning defense method in federated learning using generative adversarial network, in: International Conference on Algorithms and Architectures for Parallel Processing, Springer, 2019, pp. 595–609.
[53] X. Zheng, Z. Cai, Privacy-preserved data sharing towards multiple parties in industrial iots, IEEE Journal on Selected Areas in Communications 38 (5) (2020) 968–979, doi:10.1109/JSAC.2020.2980802.