Data Poisoning Attacks on Federated Machine Learning

Gan Sun, *Member, IEEE*, Yang Cong, *Senior Member, IEEE*, Jiahua Dong, Qiang Wang, Lingjuan Lyu, and Ji Liu

***Abstract*—Federated machine learning which enables resource-constrained node devices (e.g., Internet of Things (IoT) devices and smartphones) to establish a knowledge-shared model while keeping the raw data local, could provide privacy preservation, and economic benefit by designing an effective communication protocol. However, this communication protocol can be adopted by attackers to launch data poisoning attacks for different nodes, which has been shown as a big threat to most machine learning models. Therefore, we in this article intend to study the model vulnerability of federated machine learning, and even on IoT systems. To be specific, we here attempt to attacking a popular federated multitask learning framework, which uses a general multitask learning framework to handle statistical challenges in the federated learning setting. The problem of calculating optimal poisoning attacks on federated multitask learning is formulated as a bilevel program, which is adaptive to the arbitrary selection of *target* nodes and *source attacking* nodes. We then propose a novel systems-aware optimization method, called as attack on federated learning (AT^{2}FL), to efficiently derive the implicit gradients for poisoned data, and further attain optimal attack strategies in the federated machine learning. This is an earlier work, to our knowledge, that explores attacking federated machine learning via data poisoning. Finally, experiments on several real-world data sets demonstrate that when the attackers directly poison the *target* nodes or indirectly poison the related nodes via using the communication protocol, the federated multitask learning model is sensitive to both poisoning attacks.**

***Index Terms*—Bilevel optimization, data poisoning, federated machine learning, multitask learning.**

Manuscript received June 9, 2021; revised September 21, 2021 and October 19, 2021; accepted November 4, 2021. Date of publication November 17, 2021; date of current version June 23, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62003336 and Grant 62073205; in part by the National Postdoctoral Innovative Talents Support Program under Grant BX20200353; in part by the State Key Laboratory of Robotics under Grant 2022-Z06; and in part by the Nature Foundation of Liaoning Province of China under Grant 2020-KF-11-01. *(Gan Sun and Jiahua Dong contributed equally to this work.) (Corresponding author: Yang Cong.)*

Gan Sun and Yang Cong are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, and also with the Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China (e-mail: sungan1412@gmail.com; congyang81@gmail.com).

Jiahua Dong is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, also with the Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: dongjiahua@sia.cn).

Qiang Wang was with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, also with Shenyang University, Shenyang 110044, China (e-mail: wangqiang@sia.cn).

Lingjuan Lyu is with Sony AI, Tokyo 108-0075, Japan (e-mail: lingjuanlvsmile@gmail.com).

Ji Liu is with the Beijing Kuaishou Technology Company, Ltd., Beijing 100005, China (e-mail: jiliu@kwai.com).

Digital Object Identifier 10.1109/JIOT.2021.3128646

I. Introduction

MACHINE learning has been widely applied into a broad array of applications, e.g., spam filtering [35], lesions segmentation [11], natural gas price prediction [1], and Internet of Things (IoT) devices [10], [16], [33]. Among these applications, the reliability or security of the machine learning system has been a great concern, including adversaries [15], [34]. For example, for a product recommendation system [28], researchers can either rely on public E-commerce platforms, e.g., Taobao or Amazon Mechanical Turk, or collect training data by private teams. Unfortunately, both of these above systems have the opportunity of being injected corrupted or poisoned data by attackers, which could be a security risk to the physical objects or the IoT systems. To improve the robustness and reliability of existing machine learning and IoT systems, it is critical to study how well machine learning performs under the poisoning attacks.

For the attack strategy on existing machine learning methods, it can be partitioned into two categories: 1) causative attacks and 2) exploratory attacks [3]. Causative attacks methods affect machine learning models via controlling over training data, whereas exploratory attacks methods could take use of misclassifications without affecting the training phase. However, more previous researches on poisoning attacks focus on the scenarios that training samples are collected in a centralized location, or the training samples are sent to a centralized location via a distributed data collection network, e.g., autoregressive models [1], support vector machines (SVMs) [5], and collaborative filtering [20]. There exist scarce works studying poisoning attacks on federated machine learning [18], [30], [32], where the training data are distributed across multiple IoT devices (e.g., users' mobile devices: phones/tablets), and may be privacy sensitivity. To further improve its robustness, in this article, our work explores how to attack the federated learning system via data poisoning.

For federated machine learning [26], [27], [31], [32], its main idea is to build a knowledge-shared machine learning models while guaranteeing data privacy, where the raw data are distributed on multiple local devices. Even though most recent progressions have been achieved on tackling

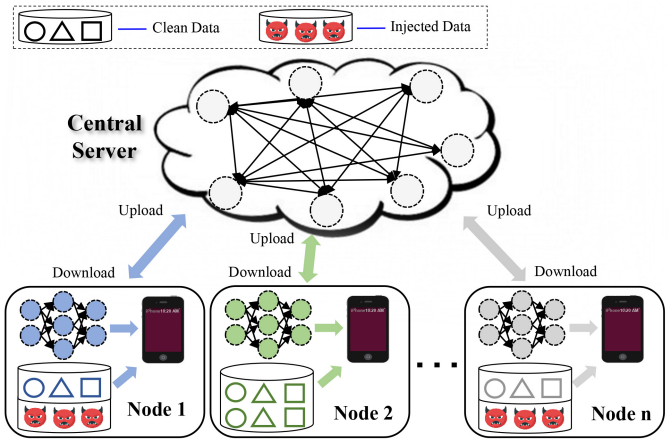


Fig. 1. Demonstration of our data poisoning attack model on federated machine learning, where different colors denote different nodes (devices), and there are n nodes in this federated learning system. Some nodes are injected by corrupted/poisoned data, while the models among some nodes could be trained with clean data.

the statistical challenges (i.e., the data collected across this network are in a *non-IID* manner, and the raw data generated in each node are in a unique distribution) or improving privacy-preservation [17], the attempt that makes federated learning more reliability under poisoning attacks, is still scarce. For example, consider several different E-commerce companies in a same region, the target is to utilize user and product information (e.g., user's browsing, collecting, and purchasing history), and establish a product purchase prediction model. The attackers can dominate a prescribed number of user accounts and inject the poisoning data in a direct manner. Furthermore, due to the communication protocol existing amongst different companies, this protocol could also open a feasible door for the attacker to indirectly affect the unaccessible *target* nodes. This scenario is also not well explored by existing poisoning methods whose training data are collected in a centralized location.

Motivated by the aforementioned analyses, we attempt to analyze optimal poisoning attacks on federated machine learning. More specifically, as shown in Fig. 1, we here concentrate on the recently proposed federated multitask learning framework [24], a federated learning framework which captures node relationships among multiple nodes to tackle the statistical challenges in the federated setting. Our work is to formulate the optimal data poisoning attack strategy on the federated multitask learning model as a general bilevel optimization problem, which can be adaptive to any selection of *target* nodes and *source attacking* nodes. However, current optimization methods for this bilevel problem are not suited to tackle the systems challenges (e.g., stragglers and high communication cost issues) that exist in the federated learning system. As a key component of this work, we thus design a novel optimization method, attack on federated learning (AT^{2}FL), to derive the implicit gradients for computing poisoned data in the *source attacking* nodes. Furthermore, the obtained gradient can be effectively used to compute the optimal attack strategies. Finally, the effectiveness of the proposed optimal attack strategy against random baselines is empirically justified on several real-world applications. The experiment results strongly support our proposed model when attacking federated machine learning based on the communication protocol.

The contribution of this work is given in threefold.

1) We propose a bilevel optimization framework to compute optimal poisoning attacks on federated machine learning. This is an earlier attempt, to our best knowledge, that explores the vulnerability of federated machine learning from the perspective of data poisoning.
2) We derive an effective optimization method, i.e., AT^{2}FL, to solve the optimal data attack problem, which can address systems challenges existing in federated machine learning.
3) We justify the empirical performance of our optimal attack strategy, and our proposed AT^{2}FL algorithm with several real-world benchmarks. The presented experiment results indicate that the communication protocol among multiple nodes could open a door for attacker to attack federated machine learning.

The remainder of this work is organized as follows. The first section provides a brief review of some related works. Section III introduces the formulation of poisoning attacks on federated multitask learning. Then, how to efficiently optimize the proposed model via attacking alternating minimization strategy is proposed in Section IV. We report the experimental results and discuss the potential directions in the last two sections.

II. Related Work

Since our work mainly draws from data poisoning attacks and federated machine learning, we first give a brief overview on data poisoning attacks, followed by federated machine learning.

A. Data Poisoning Attacks

For the *data poisoning attacks*, it has become an urgent research field in adversarial machine learning [15], in which the target is against machine learning algorithms [4], [15]. The earlier attempt for this field is to investigate the poisoning attacks on SVMs [5]. After calculating a gradient based on the characteristics of the optimal solution for SVM, a gradient ascent strategy can be used as an attack based on the obtained gradient. Furthermore, poisoning attack is investigated on many popular machine learning models, containing autoregressive model [1], matrix factorization based collaborative filtering [20], and neural networks for graph data [37]. In addition to single-task learning models, perhaps [36] is the most relevant work to ours in the data poisoning attacks field, which provides the first study on one much challenging problem, i.e., the vulnerability of multitask learning [12], [25]. It develops a stochastic gradient ascent-based algorithm for solving the optimal attack problem. However, the motivations between [36] with our work are significantly different as follows.

1) The data sample in [36] are put together, which is different from the data isolated islands scenario in federated machine learning. In other words, the raw data set for building federated learning models are distributed

amongst multiple nodes/devices, while data fusion and collection is forbidden in this scenario.

2) The proposed algorithm in [36] is based on the optimization method of current multitask learning methods, which is not suitable to address the systems challenges (e.g., stragglers, high communication cost, etc) in the federated learning setting. How to tackle these challenges and launch data poisoning attacks is a key component of this work.

B. Federated Machine Learning

For the *federated machine learning*, its main purpose is to update classifier fast for modern massive data sets, and the training data it can handle are with the following properties [19]: 1) *Non-IID*: the raw data on each node/device are drawn from an unique distribution and 2) *Unbalanced*: the number of training samples for different nodes/devices may vary by orders of magnitude. Based on these data distribution characteristics, most existing federated learning [32] models can be divided into:

1) horizontal (sample-based) federated learning, i.e., a same feature space is shared among different nodes/devices while the sample space is not. One of the representative works is a multitask-based federated learning system [24], which is proposed to preserve data privacy while allowing multiple nodes to complete separate task learning. Moreover, Caldas *et al.* [6] discussed a novel federated multitask learning system by employing a family of nonlinear models, and explore the robustness capability for the systems challenges; Corinzia *et al.* [8] introduced an algorithm for federated multitask learning models, where the federated network of the server and the clients are treated as a star-shaped Bayesian network;
2) vertical (feature-based) federated learning, i.e., a same sample ID space is shared among different nodes while the feature space is different. For the vertically partitioned data, several machine learning methods have been proposed to preserve data privacy, e.g., secure linear regression [13] and gradient descent methods [14];
3) federated transfer learning, i.e., both feature space and sample space are different among two data sets. For this setting, most popular transfer learning methods can be utilized to provide solutions for the entire sample and feature space with the federated setting. As for the applications, Ma *et al.* [22] proposed a Federated Data Cleaning model for edge intelligence with IoT applications; Liu *et al.* [21] proposed a federated-autonomous deep learning model to balance global model training for healthcare data.

As the first attempt, Bagdasaryan *et al.* [2] develops a new model-replacement methodolody that exploits these vulnerabilities and demonstrates its efficacy on federated learning tasks. Moreover, Xie *et al.* [29] presented a distributed backdoor attack framework, which can be attached via decomposing a global trigger pattern into separate local patterns. However, these methods above focus on the federated averaging framework, i.e., the aggregating model updates in the central server, which cannot be effectively utilized on popular horizontal federated learning: federated multitask learning. We in this article try to fill the data poisoning gap by investigating poisoning attack against well-known federated multitask learning.

TABLE I
NOTATIONS FOR ALL THE USED VARIABLES

Variables	Interpretation
D_ℓ	Clean data for the ℓ-th node
D_t	Clean data for the t-th *target* node
$\hat{D}_\ell$	Injected data for the ℓ-th node
$\hat{D}_s$	Injected data for the s-th *source attacking* node
$\mathcal{N}_{\rm tar}$	A set of *target* nodes
$\mathcal{N}_{\rm sou}$	A set of *source attacking* nodes
$\mathcal{H}$	Upper level function in Eq. (3)
W	Weight matrix
Ω	Model relationship among nodes

III. PROPOSED FORMULATION

In this section, we introduce our proposed poisoning attacks strategies on federated machine learning in details. Therefore, we first present some preliminaries (such as notations given in Table II) about federated multitask learning, which is a general multitask learning formulation in the context of federated learning. Then, we provide the mathematical form of our poisoning attack formulation, followed by how to optimize our proposed model.

A. Federated Multitask Learning

In the setting of federated machine learning, the target is to learn a shared model with data which has been collected or generated by m distributed nodes, where the local data for each node/device ($\ell \in [m]$) are generated via a distinct distribution. Different from most prior federated learning formulations, it is natural to fit multiple local data sets via separate models, $\{w_1, \ldots, w_m\}$, and the node correlations can be modeled through a multitask learning framework. In this article, we thus focus on an effective horizontal (sample based) federated learning model, i.e., federated multitask learning [24]. More specifically, the federated multitask learning model in [24] via incorporating node relationships is formulated as

$$\min_{W,\Omega} \sum_{\ell=1}^{m}\sum_{i=1}^{n_\ell} \mathcal{L}_\ell\left(w_\ell^\top x_\ell^i, y_\ell^i\right) + \lambda_1 \mathrm{tr}(W\Omega W^\top) + \lambda_2 \|W\|_F^2 \quad (1)$$

where (x_ℓ^i, y_ℓ^i) for the ℓth node denotes the ith sample, n_ℓ for the ℓth node is the number of clean samples, $W = [w_1, \ldots, w_m] \in \mathbb{R}^{d\times m}$ denotes a matrix with the ℓth column corresponding to the weight vector for the ℓth node. Matrix $\Omega \in \mathbb{R}^{m\times m}$ describes relationships among different nodes, which is a known a-priori (e.g., $\boldsymbol{I}_{m\times m} - (1/m)\mathbf{1}\mathbf{1}^\top$) in the initialization phase. $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are the parameters to control corresponding regularization terms.

From the equation above, we can notice that matrix Ω can be calculated centrally since it is not dependent on the raw local data. One major contribution of [24] is an efficient distributed optimization method for the variable W. Furthermore, the update of W can be achieved by the extending distributed primal-dual optimization method [23]. Let $n \coloneqq \sum_{\ell=1}^{m} n_\ell$ and

$X \coloneqq \text{Diag}(X_1, \ldots, X_m) \in \mathbb{R}^{md\times n}$. Given the fixed variable Ω, and defined dual variable $\alpha_\ell \in \mathbb{R}^{n_\ell}$ for each corresponding node, the dual formulation for optimization problem in (1) can be defined by

$$\min_{\alpha,W,\Omega} \sum_{\ell=1}^{m}\sum_{i=1}^{n_\ell} \mathcal{L}_\ell^*\left(-\alpha_\ell^i\right) + \lambda_1 \mathcal{R}^*(X\alpha) \tag{2}$$

where $\mathcal{L}_\ell^*$ and $\mathcal{R}^*$ are the corresponding conjugate dual functions of $\mathcal{L}_\ell$ and $\text{tr}(W\Omega W^\top) + \lambda_2/\lambda_1\|W\|_F^2$, and α_ℓ^i in $\alpha \in \mathbb{R}^n$ is dual variable of the ith data point (x_ℓ^i, y_ℓ^i) for the ℓth node. Meanwhile, we further denote $D_\ell = \{(X_\ell, \alpha_\ell, y_\ell)|X_\ell \in \mathbb{R}^{d\times n_\ell}, \alpha_\ell \in \mathbb{R}^{n_\ell}, y_\ell \in \mathbb{R}^{n_\ell}\}$ as the clean data in the node ℓ in this work.

B. Poisoning Attacks on Federated Multitask Learning

We in this section first introduce the problem setting of the data poisoning attack on federated machine learning. Based on real-world scenarios, we then provide three kinds of attacks, followed by a bilevel formulation to compute optimal attacks.

Suppose that the attempt of attackers is to degrade the model performance of a set of *target* nodes $\mathcal{N}_{\text{tar}} \subset m$ by injecting corrupted/poisoned data to a set of *source attacking* nodes $\mathcal{N}_{\text{sou}} \subset m$. Based on the dual problem in (2), we denote $\hat{D}_\ell = \{(\hat{X}_\ell, \hat{\alpha}_\ell, \hat{y}_\ell)|\hat{X}_\ell \in \mathbb{R}^{d\times\hat{n}_\ell}, \hat{\alpha}_\ell \in \mathbb{R}^{\hat{n}_\ell}, \hat{y}_\ell \in \mathbb{R}^{\hat{n}_\ell}\}$ as the set of malicious data injected to the node ℓ, where $\hat{n}_\ell$ denotes the number of injected samples for the node ℓ. To be specific, $\hat{D}_\ell$ will be $\emptyset$, i.e., $\hat{n}_\ell = 0$, if $\ell \notin \mathcal{N}_{\text{sou}}$. Based on real-world federated learning scenarios, as shown in Fig. 2, we here define the following three kinds of attacks.

1) *Direct Attack:* $\mathcal{N}_{\text{tar}} = \mathcal{N}_{\text{sou}}$. All the *target* nodes can be directly injected poisoned data by an attacker, since a door will be opened when collecting training data. Take human activity recognition as an example, the activities of mobile phone users in a same cell network can be recognized via their individual sensors, image, or text data. An attacker can attack the target mobile phones directly by providing counterfeit sensor data into the target phones (nodes).
2) *Indirect Attack:* $\mathcal{N}_{\text{tar}} \cap \mathcal{N}_{\text{sou}} = \emptyset$. Any of the target nodes cannot be directly injected poisoned data by an attacker, due to the improving security. However, due to the communication protocol existing amongst multiple mobile phones, the attacker can inject poisoned data samples to other related mobile phones and affect the target nodes in an indirect way.
3) *Hybrid Attack:* An attack style which integrates both direct attack and indirect attack ways, i.e., the attacker can simultaneously inject poisoned data samples into both *target* nodes and *source attacking* nodes.

To deteriorate the model performance of *target* nodes maximally, a bilevel optimization problem can be adopted to formulate the optimal attack problem by following [5]:

$$\begin{aligned} \max_{\left\{\hat{D}_s|s\in\mathcal{N}_{\text{sou}}\right\}} \quad & \sum_{\{t|t\in\mathcal{N}_{\text{tar}}\}} \mathcal{L}_t(D_t, w_t) \\ \text{s.t.,} \quad & \min_{\alpha,W,\Omega} \sum_{\ell=1}^{m} \frac{1}{n_\ell + \hat{n}_\ell} \mathcal{L}_\ell^*\left(D_\ell \cup \hat{D}_\ell\right) + \lambda_1 \mathcal{R}^*(X\alpha) \end{aligned} \tag{3}$$

where $\hat{D}_s$ denotes the injected data for the sth *source attacking* node and D_t indicates the clean data for the tth *target* node. Intuitively, the variables in the upper-level problem are the injected data points $\hat{D}_s$, and we denote this upper-level problem as a $\mathcal{H}$ in this article. The lower-level problem for the (3) is a federated multitask learning problem, where the training set is composed of both clean and injected data points. Therefore, the lower-level problem in (3) above can be considered as a constraint for the upper-level problem.

IV. Attack on Federated Learning

This section proposes an effective algorithm for computing optimal attack strategies, i.e., AT^{2}FL. Specifically, we follow the setting of most data poisoning attack strategies (e.g., [5] and [36]), and design a projected stochastic gradient ascent algorithm. This algorithm can increase the empirical loss of target nodes maximally, and further damage their classification or regression performances. Moreover, this article intends to calculate the gradients by utilizing the optimality conditions, since the closed-form relation between the injected data and empirical loss does not exist.

A. Attacking Alternating Minimization

Due to the nonconvexity property, the bilevel problems are usually hard to be solved. Although the upper-level problem in our bilevel formulation (3) is a simple primal problem, the lower-level problem has the nonlinear and nonconvex characteristics. To solve this problem effectively, the idea here is to update the injected data iteratively in the direction of maximizing the function $\mathcal{H}$ of target nodes. Therefore, to reduce the computational complexity in optimizing the optimal attack problem, we need to optimize over the features of injected data $(\hat{x}_s^i, \hat{\alpha}_s^i)$ by fixing the labels of injected data, where $s \in \mathcal{N}_{\text{sou}}$. The update rules for injected data $\hat{x}_s^i$ can then be expressed as

$$\left(\hat{x}_s^i\right)^k \leftarrow \text{Proj}_{\mathbb{X}}\left(\left(\hat{x}_s^i\right)^{k-1} + \eta\nabla_{\left(\hat{x}_s^i\right)^{k-1}}\mathcal{H}\right) \quad \forall s \in \mathcal{N}_{\text{sou}} \tag{4}$$

where $\eta > 0$ denotes the step size, variable k is the kth iteration. Symbol $\mathbb{X}$ denotes the feasible region for the injected data, which can be given via the first restriction in the upper-level problem $\mathcal{H}$. To be more specific, $\text{Proj}_{\mathbb{X}}(x)$ can be defined as x if $\|x\|_2 \le r$; $xr/\|x\|_2$, otherwise, i.e., $\mathbb{X}$ can be considered as an ℓ_2-norm ball by following [9]. Accordingly, the corresponding dual variable $\hat{\alpha}_s^i$ can be updated gradually as the $\hat{x}_s^i$ comes as following:

$$\left(\hat{\alpha}_s^i\right)^k \leftarrow \left(\hat{\alpha}_s^i\right)^{k-1} + \Delta\left(\hat{\alpha}_s^i\right) \quad \forall s \in \mathcal{N}_{\text{sou}} \tag{5}$$

where $\Delta(\hat{\alpha}_s^i)$ denotes the gradient information for the dual variable $\hat{\alpha}_s^i$.

B. Gradients Computation

One of the major issue in updating $(\hat{x}_s^i, \hat{\alpha}_s^i)^k$ for the (4) is how to calculate $\nabla_{(\hat{x}_s^i)^{k-1}}\mathcal{H}$. To attain this, we adopt the chain rule, and compute the gradient of each t-th target node in $\mathcal{H}$, i.e., $\nabla_{\hat{x}_s^i}\mathcal{L}_t(D_t, w_t)$, which could obtain the following equation:

$$\nabla_{\hat{x}_s^i}\mathcal{L}_t(D_t, w_t) = \nabla_{w_t}\mathcal{L}_t(D_t, w_t) \cdot \nabla_{\hat{x}_s^i} w_t \quad \forall t \in \mathcal{N}_{\text{tar}}. \tag{6}$$

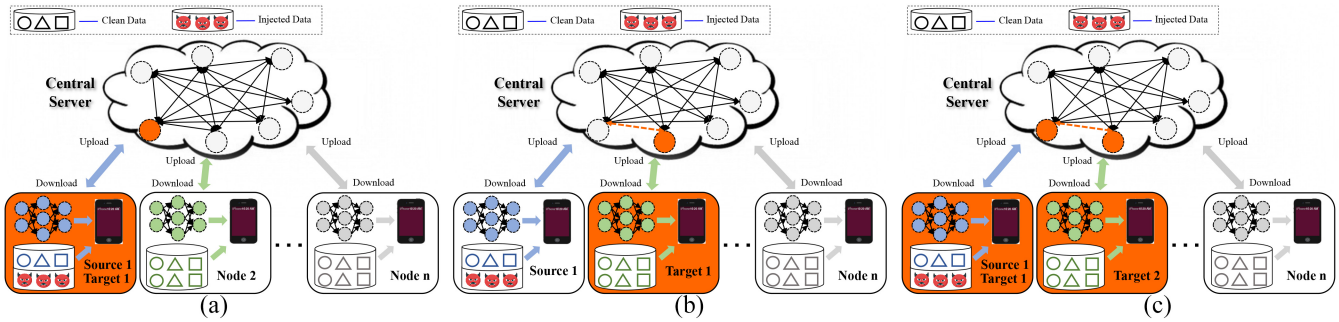


Fig. 2. Demonstration of three different data poisoning strategies: (a) direct attack (Left), (b) indirect attack (Middle), (c) and hybrid attack (Right), where the nodes marked with **orange** color are attacked.

From the equation above, it can be easily to find that the first term in the right side can be easily attained since it only depends on the loss function $\mathcal{L}_t(\cdot)$. However, the second term in the right side is dependent on the optimality conditions of the lower-level problem for (3). In the following section, we concentrate on two well-known loss functions: 1) least-square loss (regression problems) and 2) hinge loss (classification problems), and provide the details of computing the gradient $\nabla_{\hat{x}_s^i} w_t$. Please check the definitions in the Appendix.

Based on the loss functions above, we first fix variable Ω to remove the restriction of the lower-level problem, and attain the dual subproblem as follows:

$$\min_{\alpha, W} \sum_{\ell=1}^{m} \frac{1}{n_\ell + \hat{n}_\ell} \mathcal{L}_\ell^*\Big(D_\ell \cup \hat{D}_\ell\Big) + \lambda_1 \mathcal{R}^*(X\alpha). \tag{7}$$

Since the optimization problem of the lower-level problem can be treated as a constraint to the upper-level problem, Ω in the problem of (7) can be considered as a constant-value matrix when calculating the gradients. Additionally, since $\mathcal{R}^*(X\alpha)$ is continuous differentiable, W and α could be connected via defining the next formulation by following [23]:

$$w_\ell(\alpha) = \nabla \mathcal{R}^*(X\alpha). \tag{8}$$

Therefore, the key component on the rest is how to update the dual variable α and further compute the corresponding gradients in (6).

Update Dual Variable α: Consider the maximal increase of the dual objective with a least-square loss function or a hinge loss function, where we only allow to change each element of α. We can reformulate (7) as the following constrained optimization problem for the ℓth node:

$$\min_{\alpha} \sum_{\ell=1}^{m} \frac{1}{n_\ell + \hat{n}_\ell} \Big(\mathcal{L}_\ell^*\big(-\alpha_\ell^i\big) + \mathcal{L}_\ell^*\Big(-\hat{\alpha}_\ell^{i'}\Big)\Big) + \lambda_1 \mathcal{R}^*\big(X\big[\alpha_\ell; \hat{\alpha}_\ell\big]\big). \tag{9}$$

To solve (9) across distributed nodes, several data-local subproblems can be defined as follows. Furthermore, this dual problem can be well-formulated via a careful quadratic approximation, which can be separately computed among the nodes. At every step k, two samples (i.e., i in $\{1, \ldots, n_\ell\}$ and i' in $\{1, \ldots, \hat{n}_\ell\}$) are chosen uniformly at random from original clean data and injected data, respectively, and the updates of both α_ℓ^i and $\hat{\alpha}_\ell^{i'}$ in node ℓ can be computed as

$$\begin{aligned} \big(\alpha_\ell^i\big)^k &= \big(\alpha_\ell^i\big)^{k-1} + \Delta\alpha_\ell^i \\ \Big(\hat{\alpha}_\ell^{i'}\Big)^k &= \Big(\hat{\alpha}_\ell^{i'}\Big)^{k-1} + \Delta\hat{\alpha}_\ell^{i'} \end{aligned} \tag{10}$$

where both $\Delta\alpha_\ell^i$ and $\Delta\hat{\alpha}_\ell^{i'}$ are the chosen stepsizes to obtain maximal ascent of the dual objective in (9) when all variables are fixed. In order to attain maximal dual ascent for (10), one has to optimize

$$\begin{aligned} \Delta\alpha_\ell^i &= \arg\min_{a \in \mathbb{R}} \mathcal{L}_\ell^*\big(-\big(\alpha_\ell^i + a\big)\big) + a\langle w_\ell(\alpha_\ell), x_\ell^i\rangle + \frac{\lambda_1}{2}\big\|x_\ell^i a\big\|_{\mathrm{M}_\ell}^2 \\ \Delta\hat{\alpha}_\ell^{i'} &= \arg\min_{\hat{a} \in \mathbb{R}} \mathcal{L}_\ell^*\Big(-\Big(\hat{\alpha}_\ell^{i'} + \hat{a}\Big)\Big) + \hat{a}\langle w_\ell\big(\hat{\alpha}_\ell\big), x_\ell^{i'}\rangle + \frac{\lambda_1}{2}\Big\|x_\ell^{i'}\hat{a}\Big\|_{\mathrm{M}_\ell}^2 \end{aligned} \tag{11}$$

where $M_\ell \in \mathbb{R}^{d\times d}$ is the ℓth diagonal block of a symmetric positive definite matrix M. $M^{-1} = \bar{\Omega} + \lambda_2/\lambda_1 I_{md\times md}$, where $\bar{\Omega} := \Omega \otimes \mathrm{I}_{d\times d} \in \mathbb{R}^{md\times md}$. The approximation subproblem in the above (11) which could mitigate the systems challenges (i.e., stragglers or computational costs) has been proved in [24]. Furthermore, $\Delta\alpha_\ell^i$ can be calculated in a closed-form for the least-square loss function, i.e.,

$$\Delta\alpha_\ell^i = \frac{y_\ell^i - \big(x_\ell^i\big)^\top x_\ell^i \alpha_\ell^i - 0.5\big(\alpha_\ell^i\big)^{k-1}}{0.5 + \lambda_1 \big\|x_\ell^i\big\|_{\mathrm{M}_\ell}^2}. \tag{12}$$

Meanwhile, $\Delta\hat{\alpha}_\ell^{i'}$ can be computed in a same manner. Furthermore, we substitute the hinge loss into the optimization problem in (7), and can obtain

$$\Delta\alpha_\ell^i = y_\ell^i \max\left(0, \min\left(1, \frac{1 - \big(x_\ell^i\big)^\top x_\ell^i \big(\alpha_\ell^i\big)^{k-1} y_\ell^i}{\lambda_1 \big\|x_\ell^i\big\|_{\mathrm{M}_\ell}^2} + \big(\alpha_\ell^i\big)^{k-1} y_\ell^i\right)\right) - \alpha_\ell^i. \tag{13}$$

Update Gradient: Given (12), (13) with (8), we can compute the gradient in (4). To be more specific, for the least-square loss, we can calculate the gradient of each injected data $\hat{x}_s^i$ with its associated *target* node t

$$\begin{aligned} &\nabla_{(\hat{x}_s^i)} \mathcal{L}_t\Big((w_t)^\top x_t^j, y_t^j\Big) \\ &= 2\Big((w_t)^\top x_t^j - y_t^j\Big) x_t^j \cdot \nabla_{\hat{x}_s^i} w_t \\ &= 2\Big((w_t)^\top x_t^j - y_t^j\Big) x_t^j \cdot \Delta\hat{\alpha}_s^i \Omega(t, s) \end{aligned} \tag{14}$$

Algorithm 1 AT^{2}FL

Input: Nodes $\mathcal{N}_{\text{tar}}, \mathcal{N}_{\text{sou}}$, attacker budget $\hat{n}_s$;

Randomly initialize $\forall \ell \in \mathcal{N}_{\text{sou}}$,

$$\hat{D}_\ell^0 = \{(\hat{X}_\ell^0, \hat{\alpha}_\ell^0, \hat{y}_\ell^0) | \hat{X}_\ell^0 \in \mathbb{R}^{d \times \hat{n}_\ell}, \hat{\alpha}_\ell^0 \in \mathbb{R}^{\hat{n}_\ell}, \hat{y}_\ell^0 \in \mathbb{R}^{\hat{n}_\ell}\} \quad (15)$$

Initialize $\hat{D}_\ell = \hat{D}_\ell^0, \ \forall \ell \in \mathcal{N}_{\text{sou}}$ and matrix Ω^0;
for $k = 0, 1, \ldots$ **do**
  Set subproblem learning rate η;
  for all nodes $\ell = 1, 2, \ldots, m$ in parallel **do**
    Compute the approximate solution $\Delta\alpha_\ell$ via Eq. (12) or Eq. (13);
    Update local variables $\alpha_\ell \leftarrow \alpha_\ell + \Delta\alpha_\ell$;
    if node $\ell \in \mathcal{N}_{\text{sou}}$ **then**
      Compute the approximate solution $\Delta\hat{\alpha}_\ell$ via Eq. (12) or Eq. (13);
      Update local variables $\hat{\alpha}_\ell \leftarrow \hat{\alpha}_\ell + \Delta\hat{\alpha}_\ell$;
    end if
  end for
  Update variables W^k and Ω^k based on the latest α;
  for all source nodes $s = 1, 2, \ldots, \mathcal{N}_{\text{sou}}$ in parallel **do**
    #$\mathcal{N}_{\text{tar}} = \mathcal{N}_{\text{sou}}$: Direct Attack #
    #$\mathcal{N}_{\text{tar}} \cap \mathcal{N}_{\text{sou}} = \emptyset$: Indirect Attack #
    #Otherwise: Hybrid Attack#
    Update $\hat{x}_s^i$ based on the Eq. (4);
  end for
  $\hat{D}_\ell = \hat{D}_\ell^k, \ \forall \ell \in \mathcal{N}_{\text{sou}}$;
end for

where j denotes the jth sample for t-th *target* node. Similarly, for the hinge loss, we can have

$$\begin{aligned} &\nabla_{(\hat{x}_s^i)} \mathcal{L}_t\left((w_t)^\top x_t^j, y_t^j\right) \\ &= y_t^j x_t^j \cdot \nabla_{\hat{x}_s^i} w_t \\ &= y_t^j x_t^j \cdot \Delta\alpha_s^i \Omega(t, s). \end{aligned} \quad (16)$$

Finally, we initialize the data for source nodes by (15), and the whole optimization procedure of attacking on the federated learning can be summarized in Algorithm 1.

V. Experiments

In this section, we empirically justify the performance of the proposed poisoning attack strategies, and its convergence analysis. More specifically, we first introduce several adopted data sets, followed by the experimental results. Then, some analyses about our proposed model are reported.

A. Real Data Sets

We adopt the following benchmark data sets for our experiments, containing three classification data sets and one regression data set.

EndAD (Endoscopic Image Abnormality Detection): This data set is collected from 544 healthy volunteers and 519 volunteers with various lesions, consisting of cancer, gastritis, bleeding, and ulcer. With the help of these volunteers, we obtain 9769 lesion images and 9768 healthy images with the resolution 489 × 409. Some examples are shown in Fig. 3. In the implementation, a 128-dimensional deep feature is extracted via the VGG model trained in [7]. To simulate the federated learning procedure, we split the lesion images on a per-disease basis, while splitting the healthy images randomly. Then, we obtain a federated learning data set with six clinics, where each clinic can be regarded as one node, i.e., six nodes.

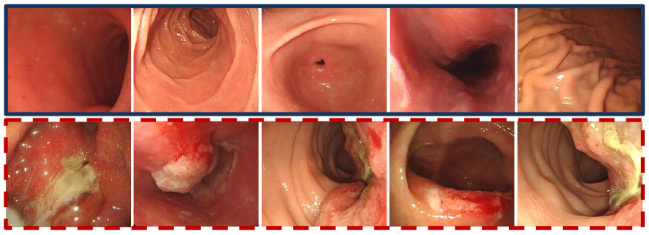


Fig. 3. Sample images from **EndAD** data set, where the first and the second rows are healthy images and lesion images, respectively.

Human Activity Recognition[1]*:* This classification data set consists of mobile phone gyroscope and accelerometer data, which are collected from 30 individuals, performing one of six activities: 1) walking; 2) walking upstairs; 3) walking downstairs; 4) sitting; 5) lying down; and 6) standing. The provided feature vectors of time is 561, whose variables are generated from the frequency domain. In this experiment, we model each individual as a separate task and aim to predict between sitting and other activities (e.g., lying down or walking). Therefore, we have 30 nodes in total, where each node corresponds to an individual.

Landmine: This data set is built to detect whether an area has a land mine using radar images, which can be modeled as a binary classification task. Each object in this collected data set can be described via a 9-D feature vector (three correlation-based features, four moment-based features, one energy-ratio feature, and one spatial variance feature) and the corresponding binary response (1 for landmine and -1 for clutter). The number of total samples for this data set is 14 820, which can be partitioned into 29 different geographical regions, i.e., the node number of this landmine data set is 29.

Parkinson Data[2]*:* This data set is collected with 16 biomedical features for patients, and aims to predict Parkinson's disease symptom score. The number of total patients is 42, while the number of observation for the parkinson data set is 5875. By treating the symptom score prediction for each patient as a regression task, we can attain 42 regression tasks and the number of samples for each task varies between 101 and 168. Additionally, the data set's output is a score consisting of Total and Motor, we thus have two regression data sets: 1) *parkinson-total* and 2) *parkinson-motor* in this experiment.

[1]https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

[2]https://archive.ics.uci.edu/ml/datasets/parkinsons+telemonitoring

TABLE II
COMPARISONS BETWEEN OUR MODEL AND STATE-OF-THE-ARTS IN TERMS OF CLASSIFICATION ERROR AND RMSE ON FIVE DATA SETS: MEAN AND STANDARD ERRORS OVER TEN RANDOM RUNS. MODELS WITH THE BEST PERFORMANCE ARE BOLDED

	Metrics	Non attacks	Random direct attacks	Random indirect attacks	Random hybird attacks	Direct attacks	Indirect attacks	Hybird attacks
EndAD	Error(%)	6.881±0.52	7.659±1.14	6.888±0.45	7.154±0.16	**28.588±3.74**	7.324±0.62	16.190±2.26
Human Activity	Error(%)	2.586±0.84	3.275±0.71	2.894±0.83	3.172±0.69	**29.422±2.96**	3.438±0.34	17.829±2.75
Landmine	Error(%)	5.682±0.28	5.975±0.36	5.735±0.36	5.819±0.22	**13.648±0.54**	7.428±0.39	9.579±0.27
	Avg. Error (%)	5.049±0.55	5.636±0.74	5.172±0.55	5.382±0.36	**23.886±2.41**	6.069±0.45	14.533±1.76
Parkinson-Total	RMSE	6.302±0.45	13.651±2.10	6.633±0.75	11.145±1.83	**44.939±3.21**	7.763±0.82	21.990±3.17
Parkinson-Moter	RMSE	4.125±0.50	11.472±2.51	5.046±1.14	9.422±1.81	**32.992±3.78**	6.866±1.21	16.956±3.78
	Avg. RMSE(%)	5.213±0.48	12.562±2.31	5.839±0.95	10.284±1.82	**38.966±3.49**	7.314±1.02	19.473±3.48

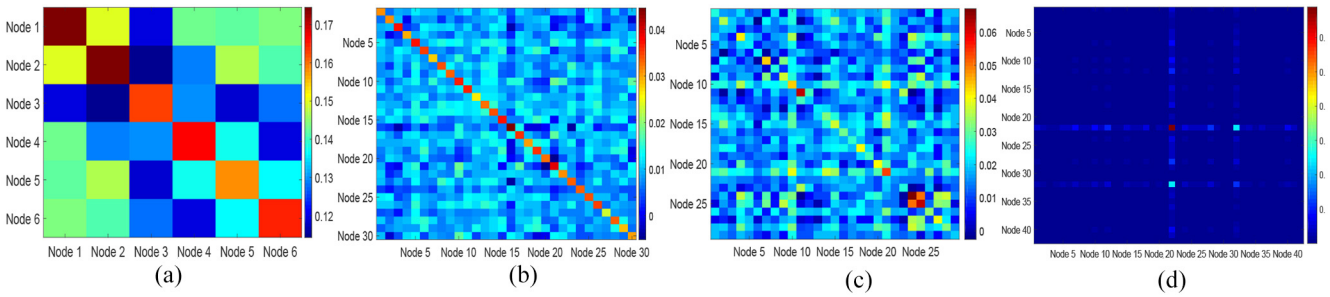


Fig. 4. Correlation matrices Ω of different data sets: (a) End_AD data set, (b) human activity data set, (c) landmine data set, and (d) parkinson data set, where the first half of nodes and the rest of nodes denote the $\mathcal{N}_{\text{tar}}$ and $\mathcal{N}_{\text{sou}}$ for the indirect attack scenario, respectively. The darker color indicates the higher correlation for each data set, and vice versa.

For the evaluation, classification error is adopted to evaluate the classification performance. For the regression problems, RMSE (i.e., root mean squared error) is a reasonable evaluation criterion. The smaller the Error and RMSE value is, the better the performance of the model could be, i.e., the weaker the attack strategy will be.

B. Evaluating Attack Strategies

In our following experiments, we justify the performance of our poisoning attack strategies via comparing with random direct/inditect/hybrid attacks on different data sets. For each data set, we randomly choose half of nodes as $\mathcal{N}_{\text{tar}}$, while selecting the rest of nodes as free nodes. For example, the number of $\mathcal{N}_{\text{tar}}$ for Human Activity data set is 15, and the number of $\mathcal{N}_{\text{tar}}$ for EndAD data set is 3, respectively. Moreover, the details of evaluated attack strategies are as follows.

1) *Direct Attacks:* All the source attacking nodes $\mathcal{N}_{\text{sou}}$ are set as the same as the target nodes, i.e., $\mathcal{N}_{\text{sou}} = \mathcal{N}_{\text{tar}}$.
2) *Indirect Attacks:* All the source attacking nodes $\mathcal{N}_{\text{sou}}$ are from the rest of nodes, where the number of $\mathcal{N}_{\text{sou}}$ is the same as that of target nodes $\mathcal{N}_{\text{tar}}$.
3) *Hybrid Attacks:* All the source attacking nodes $\mathcal{N}_{\text{sou}}$ are selected from all the nodes randomly, where the number of $\mathcal{N}_{\text{sou}}$ is the same as that of $\mathcal{N}_{\text{tar}}$.
4) *Random Direct/Indirect/Hybrid Attacks:* The attack strategies are the same as that of Direct/Indirect/Hybrid attacks. However, the injected data samples for the source nodes are chosen in a random manner.

For the used parameters in our model, the step size η_1 in (4) is set as 100, both λ_1 and λ_2 are set as 0.001 among all the experiments for a fair comparison. Furthermore, the number of injected data samples for each source attacking node is set as 20% of the clean data. The experimental results presented in Table II are averaged over ten random repetitions. From the given results, we can have the following conclusions.

1) All the attack strategies (e.g., direct attacks, random attacks, etc) have the impacts on all the data sets, which verifies the vulnerability of federated machine learning. Among all the attack strategies, notice that the direct attacks can significantly damage the classification or regression on all the data sets. For example, the performances of EndAD and Human Activity data sets can lead to 21.707% and 26.836% deterioration in terms of classification error, respectively. This observation indicates direct attacks are the big threats to the federated machine learning system.
2) When comparing with the random attack strategies, our proposed attack strategies can obtain better deterioration performance among most cases for the federated machine learning problems, which justifies that the learning attack strategies can work better than just launching random attack strategies.
3) For the indirect attack strategy, we can notice its performances are not as good as direct and hybrid attack strategies, e.g., Human Activity Recognition data set. This is because that the indirect attack can be successfully launched via effectively using the communication protocol among different nodes, where the communication protocol is bridged by the model relationship matrix Ω in this article. To verify this observation, we also present the corresponding correlation matrix of each used data set in Fig. 4. As illustrated in Fig. 4, the nodes from $\mathcal{N}_{\text{tar}}$ and $\mathcal{N}_{\text{sou}}$ have highly correlated in Landmine data set, i.e., Fig. 4(c). However, the node correlations in the Parkinson data set [i.e., Fig. 4(d)] are not close. This observation indicates the reason why indirect attack strategy performs better on the Landmine data set when comparing with other used data sets.

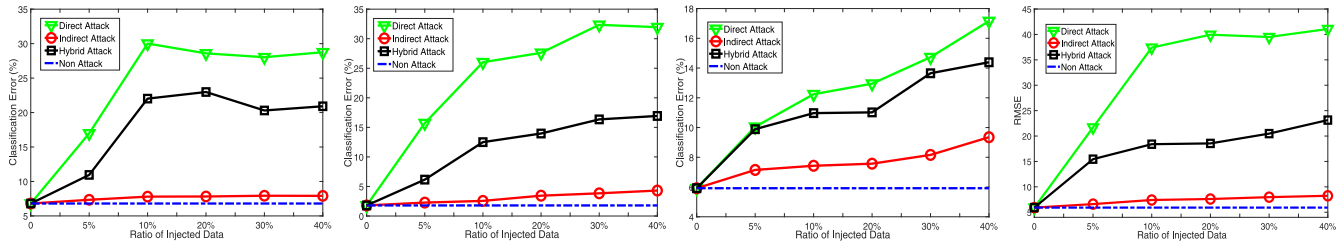


Fig. 5. Effect of ratio of injected data on (a) EndAD, (b) human activity, (c) landmine, and (d) parkinson-total data sets, where different lines denote different types of attacking strategies.

C. Sensitivity Study and Convergence Analysis

This section first conducts sensitivity studies on how the ratio of injected data and step size η affect different attack strategies. Then, we also provide the convergence analysis about our proposed AT^{2}FL.

1) Effect of Ratio of Injected Data: To investigate the effect of ratio of injected poisoned data, we use all the data sets in this section. For each data set, we take the same attack setting as that in Table II, e.g., for the direct attack, the $\mathcal{N}_{\text{tar}}$ and $\mathcal{N}_{\text{sou}}$ are selected by randomly selecting half of nodes as $\mathcal{N}_{\text{tar}}$, and the rest of nodes to form $\mathcal{N}_{\text{sou}}$. For the injected data in each source attacking node, we tune it in a range $\{0, 5\%, 10\%, 20\%, 30\%, 40\%\}$ of the clean data, and present the results in Fig. 5. From the provided results in Fig. 5, we can find that: 1) for all the used data sets, the performances under different attack strategies are decreased with the increasing of the injected data. This observation demonstrates the effectiveness of attack strategies computed by our proposed AT^{2}FL; 2) obviously, the performances of direct attack are better than that of indirect attack by varying the ratio of injected data. This is because that it can directly involve the data of $\mathcal{N}_{\text{tar}}$; 3) indirect attack strategy almost has no effect on EndAD data set since the values of correlation matrix Ω for this data set are relatively low, and each node can learn good classifier without the help of other nodes; and 4) different from other data sets, the classification errors slightly decrease after the ratio of injected data is 10%. It is because the classification error is computed on the test data while the formulation in (3) aims to maximize the loss on the training data. Meanwhile, the classification error on the test data obtains the upper bound when the ratio of injected data is 10%.

2) Effect of Number of Target Nodes: In order to study how the number of target nodes and the ratio of injected data affect the performance of our proposed attack strategies, we utilize Human Activity Recognition and evaluate two representative attack strategies in this section. By fixing other parameters and varying the number of target nodes in the range $\{3, 6, 9, 12, 15\}$, the ratio of injected data in range $\{0, 5\%, 10\%, 20\%, 30\%\}$, we provide the corresponding 20th iteration performances of different attack strategies in Fig. 6. For the presented experiment, the best classification performances are obtained when the ratios of injected data are set as 0. When the ratios of injected data are set from 0 to 5%, the model performances can be decreased significantly, indicating that the attack strategies with less injected data could also deteriorate the model performances. As the ratio of injected data increases, the classification errors of corresponding models are incrementally increasing in a fluctuate way. Moreover, the classification error will sightly increase when the number of target nodes is small, while the classification performance will tend to be small fluctuation with target nodes number increasing. The possible reason is that the test classification performances among all the nodes converge to be a fixed point as the number of target nodes increasing.

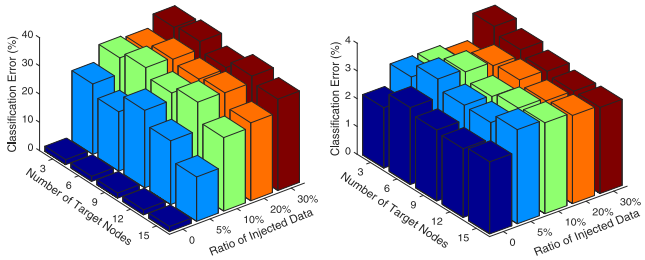


Fig. 6. Effect of number of target nodes and ratio of injected data on the human activity recognition data set: (a) direct attack and (b) indirect attack.

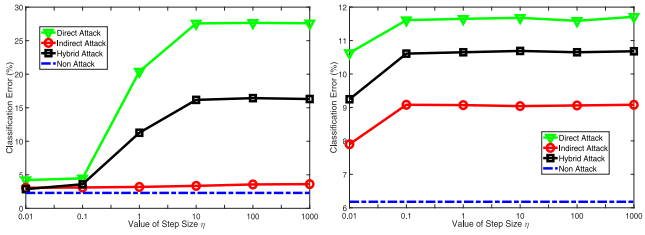


Fig. 7. Effect of step size η of (4) on (a) human activity and (b) landmine data sets, where different lines denote different attacking strategies.

3) Effect of Step Size η: In order to study how the value of step size η affects the performance of our proposed attack strategies, we adopt Human Activity Recognition and Landmine data sets in this section. By fixing other parameters (e.g., λ_1 and λ_2) and varying the value of step size η of (4) in range $\{0.01, 0.1, 1, 10, 100, 1000\}$, we present the corresponding 20th iteration performances of different attack strategies in Fig. 7. Notice that the performances of different attack strategies are improved with the increasing of the step size η. This is because the gradient in (4) with small step size will be

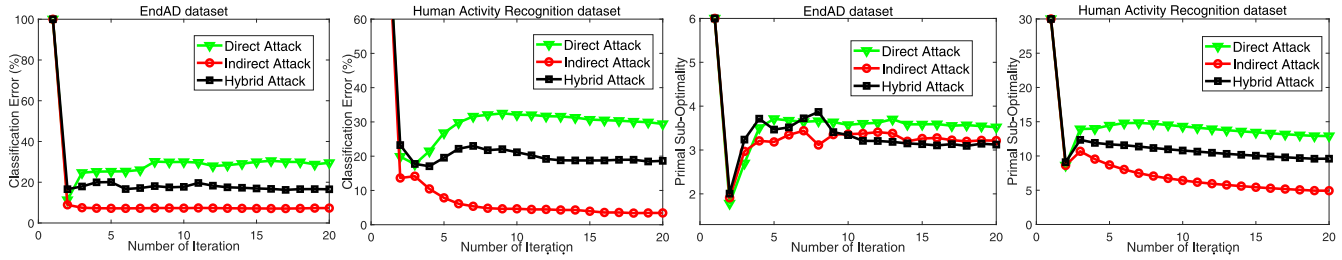


Fig. 8. Convergence curves of our proposed AT^{2}FL algorithm, where the classification errors are in (a) EndAD and (b) human activity data sets, and the primal suboptimality performances of the lower-level problem for (3) are in (c) EndAD and (d) human activity data sets.

treated as noise information, which will just have little impact on the classification performance. However, these attack strategies also outperform the nonattack strategy. Furthermore, the performances of different attack strategies tend to a fixed point with the increasing of step size η. This observation indicates that our AT^{2}FL can effectively converge to a local optimum when the step size is enough.

4) Convergence of Proposed AT^{2}FL: To study the convergence of our proposed AT^{2}FL algorithm, we adopt the EndAD and Human Activity Recognition data sets in this section. Specifically, under different attack strategies, we present the primal suboptimality values of the lower-level problem for (3) and classification error (%) of the nodes in $\mathcal{N}_{\text{tar}}$ in Fig. 8. From the presented curves in Fig. 8, we can notice that the original values of primal suboptimality and classification error are higher. This is because that the weight matrix W in (3) is just initialized with the injected data points. Then, our proposed AT^{2}FL algorithm can converge to a local optima after a few iterations for all the three kinds of attacks on EndAD and Human Activity Recognition data sets. This observation indicates the effectiveness of our proposed AT^{2}FL algorithm for launching the poisoning attack.

VI. Conclusion

In this article, we take an earlier attempt on how to effectively launch data poisoning attacks on federated machine learning. Benefitting from the communication protocol, we propose a bilevel data poisoning attacks formulation by following general data poisoning attacks framework, where it can include three different kinds of attacks. As a key contribution of this work, we design an AT^{2}FL to address the system challenges (e.g., high communication cost) existing in federated setting, and further compute optimal attack strategies. Extensive experiments demonstrate that the attack strategies computed by AT^{2}FL can significantly damage performances of real-world applications. From the study in this article, we find that the communication protocol in federated learning can be used to effectively launch indirect attacks, e.g., when two nodes have a strong correlation. Except for the horizontal federated learning in this work, we will consider the data poisoning attacks study on vertical (feature based) federated learning and federated transfer learning in the future.

Appendix
Definition of Least-Square and Hinge Losses

For the regression problem in this article, we adopt least-square loss $\mathcal{L}_\ell(w_\ell^\top x_\ell^i) = (w_\ell^\top x_\ell^i - y_\ell^i)^2$, and dual formulation is

$$\mathcal{L}_\ell^*(-\alpha_\ell^i) = -y_\ell^i \alpha_\ell^i + (\alpha_\ell^i)^2/4. \tag{17}$$

For the classification problems, we adopt hinge loss: $\mathcal{L}_\ell(w_\ell^\top x_\ell^i) = \max(0, 1 - w_\ell^\top x_\ell^i y_\ell^i)$, and the dual problem is

$$\mathcal{L}_\ell^*(-\alpha_\ell) = \begin{cases} -y_\ell^i \alpha_\ell^i, & 0 \le y_\ell^i \alpha_\ell^i \le 1, \\ \infty, & \text{otherwise} \end{cases} \tag{18}$$

where α_ℓ^i is the corresponding dual variable for the ℓth node in the regression or classification problems.

References

[1] S. Alfeld, X. Zhu, and P. Barford, "Data poisoning attacks against autoregressive models," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1452–1458.
[2] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 2938–2948.
[3] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010.
[4] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf. Comput. Commun. Security*, 2006, pp. 16–25.
[5] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1467–1474.
[6] S. Caldas, V. Smith, and A. Talwalkar, "Federated kernelized multi-task learning," in *Proc. Sysml*, 2018, pp. 1–3.
[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," 2014, *arXiv:1405.3531*.
[8] L. Corinzia, A. Beuret, and J. M. Buhmann, "Variational federated multi-task learning," 2019, *arXiv:1906.06268*.
[9] I. Daubechies, M. Fornasier, and I. Loris, "Accelerated projected gradient method for linear inverse problems with sparsity constraints," *J. Fourier Anal. Appl.*, vol 14, nos. 5–6, pp. 764–792, 2008.
[10] H. Deng, Z. Qin, L. Sha, and H. Yin, "A flexible privacy-preserving data sharing scheme in cloud-assisted IoT," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11601–11611, Dec. 2020.
[11] J. Dong, Y. Cong, G. Sun, B. Zhong, and X. Xu, "What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 4022–4031.
[12] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2004, pp. 109–117.
[13] A. Gascón *et al.*, "Secure linear regression on vertically partitioned datasets," IACR Cryptol. ePrint Arch., Lyon, France, Rep. 2016/892, 2016.

[14] S. Han, W. K. Ng, L. Wan, and V. C. S. Lee, "Privacy-preserving gradient-descent methods," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 6, pp. 884–899, Jun. 2010.
[15] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. 4th ACM Workshop Security Artif. Intell.*, 2011, pp. 43–58.
[16] L. Jiang, R. Tan, X. Lou, and G. Lin, "On lightweight privacy-preserving collaborative learning for Internet-of-Things objects," in *Proc. Int. Conf. Internet Things Design Implement.*, 2019, pp. 70–81.
[17] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nat. Mach. Intell.*, vol. 2, pp. 305–311, Jun. 2020.
[18] J. Konečnỳ, B. McMahan, and D. Ramage, "Federated optimization:Distributed optimization beyond the datacenter," 2015, *arXiv:1511.03575*.
[19] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.
[20] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2016, pp. 1885–1893.
[21] D. Liu, T. Miller, R. Sayeed, and K. D. Mandl, "FADL:Federated-autonomous deep learning for distributed electronic health record," 2018, *arXiv:1811.11400*.
[22] L. Ma, Q. Pei, L. Zhou, H. Zhu, L. Wang, and Y. Ji, "Federated data cleaning: Collaborative and privacy-preserving data cleaning for edge intelligence," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6757–6770, Apr. 2021.
[23] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 64–72.
[24] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 4424–4434.
[25] G. Sun, Y. Cong, D. Hou, H. Fan, X. Xu, and H. Yu, "Joint household characteristic prediction via smart meter data," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1834–1844, Mar. 2019.
[26] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing class imbalance in federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 10165–10173.
[27] S. Wang *et al.*, "Federated learning for task and resource allocation in wireless high altitude balloon networks," *IEEE Internet Things J.*, early access, May 13, 2021, doi: 10.1109/JIOT.2021.3080078.
[28] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Coupled multi-layer attentions for co-extraction of aspect and opinion terms," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 3316–3322.
[29] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[30] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 911–926, 2019.
[31] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *J. Healthc. Informat. Res.*, vol. 5, no. 1, pp. 1–19, 2021.
[32] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, p. 12, 2019.
[33] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, "When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultra-dense network," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2238–2251, Feb. 2021.
[34] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.
[35] M. Zhao, B. An, and C. Kiekintveld, "Optimizing personalized email filtering thresholds to mitigate sequential spear phishing attacks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 658–665.
[36] M. Zhao, B. An, Y. Yu, S. Liu, and S. J. Pan, "Data poisoning attacks on multi-task relationship learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 2628–2635.
[37] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2018, pp. 2847–2856.

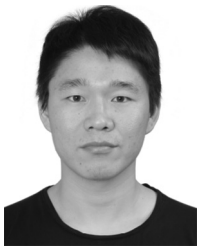
Gan Sun (Member, IEEE) received the B.S. degree from Shandong Agricultural University, Tai'an, China, in 2013, and the Ph.D. degree from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2020.

He has been visiting Northeastern University, Shenyang, China, from April 2018 to May 2019, Massachusetts Institute of Technology, Cambridge, MA, USA, from June 2019 to November 2019. He is an Associate Professor with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. He also has some top-tier conference papers accepted at CVPR, ICCV, ECCV, NeurIPS, AAAI, and IJCAI, and some top-tier journal papers accepted at IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and *Pattern Recognition*. His current research interests include lifelong machine learning, multitask learning, medical data analysis, domain adaptation, deep learning, and 3-D computer vision.

Yang Cong (Senior Member, IEEE) received the B.Sc. degree from Northeast University, Shenyang, China, in 2004, and the Ph.D. degree from the State Key Laboratory of Robotics, Chinese Academy of Sciences, Shenyang, in 2009.

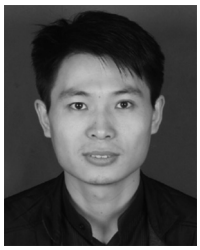
He is a Full Professor with Chinese Academy of Sciences, Shenyang. He was a Research Fellow of the National University of Singapore, Singapore, and Nanyang Technological University, Singapore, from 2009 to 2011, and a Visiting Scholar with the University of Rochester, Rochester, NY, USA. He has authored over 90 technical papers. His current research interests include image processing, compute vision, machine learning, multimedia, medical imaging, data mining, and robot navigation.

Prof. Cong has served on the editorial board of the *Journal of Multimedia*.



Jiahua Dong received the B.S. degree from Jilin University, Changchun, China, in 2017. He is currently pursuing the Ph.D. degrees with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and the University of Chinese Academy of Sciences, Beijing, China.

His current research interests include biomedical image processing, domain adaptation, and transfer learning.



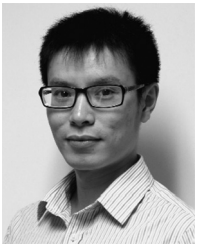
Qiang Wang received the M.S. degree from Tianjin Normal University, Tianjin, China, in 2008, and the Doctoral degree in pattern recognition and intelligent systems from the University of Chinese Academy of Sciences, Beijing, China, in 2020.

He is a Research Associate with the Key Laboratory of Manufacturing Industrial Integrated, Shenyang University, Shenyang, China. His research focuses on deep learning, feature selection, and image restoration.

Lingjuan Lyu received the Ph.D. degree from the University of Melbourne, Parkville, VIC, Australia, in 2018.

She is currently a Senior Research Scientist and a Team Leader with Sony AI, Tokyo, Japan. She was an Expert Researcher with Ant Group, Hangzhou, China, a Research Fellow with the National University of Singapore, Singapore, and a Research Fellow (Level B3, same level as a Lecturer/Assistant Professor) of the Australian National University, Canberra, ACT, Australia. Her current research interests span distributed machine learning, privacy, robustness, fairness, and edge intelligence.

Dr. Lyu was a winner of the IBM Fellowship Program (50 winners Worldwide) and contributed to various professional activities.



Ji Liu received the B.S. degree from the University of Science and Technology of China, Hefei, China, the master's degree from Arizona State University, Tempe, AZ, USA, and the Ph.D. degree from the University of Wisconsin–Madison, Madison, WI, USA.

He is currently a Researcher with Beijing Kuaishou Technology, Beijing, China. He was an Assistant Professor of Computer Science, Electrical Computer Engineering and the Goergen Institute for Data Science, University of Rochester, Rochester, NY, USA. He founded the Machine Learning and Optimization Group with UR and published more than 60 papers in top conferences and journals. His research interests cover a broad scope of machine learning, optimization, and their applications in other areas, such as healthcare, bioinformatics, computer vision, game AI, and many other data analytics related areas. His recent research focus is on reinforcement learning, asynchronous parallel algorithms, decentralized algorithms, sparse learning (compressed sensing) theory and algorithm, healthcare, and bioinformatics.

Dr. Liu won the award of Best Paper Honorable Mention at SIGKDD 2010, the Award of Facebook Best Student Paper Award at UAI 2015, the IBM Faculty Award 2017, and the MIT TR35 Award China.