

Data Poisoning Attacks on Federated Machine Learning

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

Abstract—Federated machine learning which enables resource-constrained node devices (*e.g.*, mobile phones and IoT devices) to learn a shared model while keeping the training data local, can provide privacy, security and economic benefits by designing an effective communication protocol. However, the communication protocol amongst different nodes could be exploited by attackers to launch data poisoning attacks, which has been demonstrated as a big threat to most machine learning models. In this paper, we attempt to explore the vulnerability of federated machine learning. More specifically, we focus on attacking a federated multi-task learning framework, which is a federated learning framework via adopting a general multi-task learning framework to handle statistical challenges. We formulate the problem of computing optimal poisoning attacks on federated multi-task learning as a bilevel program that is adaptive to arbitrary choice of *target* nodes and *source attacking* nodes. Then we propose a novel systems-aware optimization method, AT²FL, which is efficiency to derive the implicit gradients for poisoned data, and further compute optimal attack strategies in the federated machine learning. Our work is an earlier study that considers issues of data poisoning attack for federated learning. To the end, experimental results on real-world datasets show that federated multi-task learning model is very sensitive to poisoning attacks, when the attackers either directly poison the *target* nodes or indirectly poison the related nodes by exploiting the communication protocol.

Index Terms—Federated Machine Learning, Data Poisoning.

I. INTRODUCTION

Mchine learning has been widely-applied into a broad array of applications, *e.g.*, spam filtering [20] and natural gas price prediction [1]. Among these applications, the reliability or security of the machine learning system has been a great concern, including adversaries [10], [19]. For example, for product recommendation system [16], researchers can either rely on public crowd-sourcing platform, *e.g.*, Amazon Mechanical Turk or Taobao, or private teams to collect training datasets. However, both of these above methods have the opportunity of being injected corrupted or poisoned data by attackers. To improve the robustness of real-world machine learning systems, it is critical to study how well machine learning performs under the poisoning attacks.

For the attack strategy on machine learning methods, it can be divided into two categories: causative attacks and

G. Sun, Y. Cong, J. Dong and Q. Wang are with State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China, Email: sungan1412@gmail.com, congyang81@gmail.com, dongjiahua@sia.cn, wangqiang@sia.cn.

J. Dong is also with University of Chinese Academy of Sciences, Beijing.
J. Liu is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: jliu@cs.rochester.edu).

Manuscript received April 19, 2005; revised August 26, 2015.

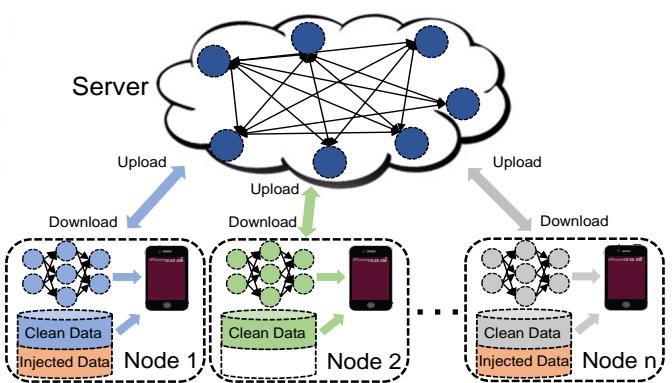


Fig. 1. The demonstration of our data poisoning attack model on federated machine learning, where different colors denote different nodes, and there are n nodes in this federated learning system. Some nodes are injected by corrupted/poisoned data, and some nodes are only with clean data.

exploratory attacks [3], where exploratory attacks influence learning via controlling over training data, and exploratory attacks can take use of misclassifications without affecting training. However, 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 network, *e.g.*, support vector machines [5], autoregressive models [1] and collaborative filtering [13]. However, none of the current works study poisoning attacks on federated machine learning [11], [17], [18], where the training data are distributed across multiple devices (*e.g.*, users' mobile devices: phones/tablets), and may be privacy sensitive. To further improve its robustness, in this paper, our work explores how to attack the federated machine learning.

For the federated machine learning, its main idea is to build machine learning models based on data sets that are distributed across multiple devices while preventing data leakage. Although recent improvements have been focusing on overcoming the statistical challenges (*i.e.*, the data collected across the network are in a *non-IID* manner, where the data on each node are generated by a distinct distribution) or improving privacy-preserving, the attempt that makes federated learning more reliable under poisoning attacks, is still scarce. For example, consider several different e-commerce companies in a same region, and the target is to establish a prediction model for product purchase based on user and product information, *e.g.*, user's browsing and purchasing history. The attackers can control 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 also opens a door for the attacker to indirectly affect the unaccessible *target* nodes, which is also not addressed by existing poisoning methods whose training data are collected in a centralized location.

Motivated by the aforementioned analysis, we attempt to analyze optimal poisoning attacks on federated machine learning. More specifically, as shown in Figure 1, we focus on the recently-proposed federated multi-task learning framework [15], 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 poisoning attack strategy on federated multi-task learning model as a general bilevel optimization problem, which can be adaptive to any choice 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.*, high communication cost, stragglers) that exist in federated learning. As a key component of this work, we thus design a novel optimization method, ATTack on Federated Learning (AT²FL), 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, we empirically evaluate the effectiveness of the proposed optimal attack strategy against random baselines on several real-world datasets. The experiment results strongly support our proposed model when attacking federated machine learning based on the communication protocol.

The novelty of our proposed model is threefold:

- We propose a bilevel optimization framework to compute optimal poisoning attacks on federated machine learning. To our best knowledge, this is an earlier attempt to explore the vulnerability of federated machine learning from the perspective of data poisoning.
- We derive an effective optimization method, *i.e.*, ATTack on Federated Learning (AT²FL), to solve the optimal attack problem, which can address systems challenges associated with federated machine learning.
- We demonstrate the empirical performance of our optimal attack strategy, and our proposed AT²FL algorithm with several real-world datasets. The experiment results indicate that the communication protocol among multiple nodes opens a door for attacker to attack federated machine learning.

II. RELATED WORK

Our work mainly draws from data poisoning attacks and federated machine learning, so we give a brief review on these two topics.

For the **data poisoning attacks**, it has become an urgent research field in the adversarial machine learning, in which the target is against machine learning algorithms [4], [10]. The earlier attempt that investigates the poisoning attacks on support vector machines (SVM) [5], where the adopted attack uses a gradient ascent strategy in which the gradient is obtained based on properties of the SVM's optimal solution.

Furthermore, poisoning attack is investigated on many machine learning models, including autoregressive models [1], matrix factorization based collaborative filtering [13] and neural networks for graph data [22]. In addition to single task learning models, perhaps [21] is the most relevant work to ours in the context of data poisoning attacks, which provides the first study on one much challenging problem, *i.e.*, the vulnerability of multi-task learning. However, the motivations for [21] and our work are significantly different as follows:

- The data sample in [21] are put together, which is different from the scenario in federated machine learning, *i.e.*, machine learning models are built based on datasets that are distributed across multiple nodes/devices while preventing data leakage.
- The proposed algorithm in [21] is based on optimization method of current multi-task learning methods, which is not suited to handle the systems challenges in federated learning, including high communication cost, etc. Handling these challenges in the setting of data poisoning attacks is a key component of this work.

For the **federated machine learning**, its main purpose is to update classifier fast for modern massive datasets, and the training data it can handle are with the following properties [12]: 1) *Non-IID*: data on each node/device may be drawn from a different distribution; 2) *Unbalanced*: the number of training samples for different nodes/devices may vary by orders of magnitude. Based on the distribution characteristics of the data, federated learning [18] can be categorized into: 1) horizontal (sample-based) federated learning, *i.e.*, datasets share the same feature space but different in samples. The representative work is a multi-task style federated learning system [15], which is proposed to allow multiple nodes to complete separate tasks while preserving security and sharing knowledge; 2) vertical (feature-based) federated learning, *i.e.*, two datasets share the same sample ID space but differ in feature space. Several privacy-preserving machine learning methods have been presented for vertically partitioned data, *e.g.*, secure linear regression [8], gradient descent methods [9]; 3) federated transfer learning, *i.e.*, two datasets differ not only in samples but also in feature space. For this case, traditional transfer learning techniques can be adopted into provide solutions for the entire sample and feature space with the federated setting. As an first attempt, [2] develops a new model-replacement methodology that exploits these vulnerabilities and demonstrated its efficacy on federated learning tasks. However, its objective is to retain high accuracy on the backdoor subtask after attacking. On the contrast, in this work, we try to fill data poisoning gap by investigating poisoning attack against horizontal federated machine learning.

III. THE PROPOSED FORMULATION

In this section, we introduce the details of our proposed poisoning attacks strategies on federated machine learning. Therefore, we firstly present some preliminaries about federated multi-task learning, which is a general multi-task learning formulation in federated setting. Then we provide the mathematical form of our poisoning attack formulation, followed by how to optimize our proposed model.

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

A. Federated Machine Learning

In the setting of federated machine learning, the target is to learn a model over data that resides on, and has been generated by, m distributed nodes. Each node/device, $\ell \in [m]$, may generate local data via a distinct distribution, and so it is natural to fit separate models, $\{w_1, \dots, w_m\}$, to the distributed data-one for each local dataset. In this paper, we focus on an effective horizontal (sample-based) federated learning model, *i.e.*, federated multi-task learning [15], which is proposed via incorporating with a general multi-task learning framework. More specifically, the federated multi-task learning model can be formulated as:

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

where (x_ℓ^i, y_ℓ^i) is the i -th sample for the ℓ -th node, n_ℓ is the number of clean samples for the ℓ -th node, $W = [w_1, \dots, w_m] \in \mathbb{R}^{d \times m}$ is a matrix whose ℓ -th column is the weight vector for the ℓ -th node and matrix $\Omega \in \mathbb{R}^{m \times m}$ models relationships among nodes. $\lambda_1 > 0$ and $\lambda_2 > 0$ are the parameters to control regularization terms.

From the equation above, we can notice that solving for matrix Ω is not dependent on the data and therefore can be computed centrally; one major contribution of [15] is an efficient distributed optimization method for the W step. Furthermore, the update of W can be achieved by extending works on distributed primal-dual optimization [14]. Let $n := \sum_{\ell=1}^m n_\ell$ and $X := \text{Diag}(X_1, \dots, X_m) \in \mathbb{R}^{md \times n}$. With the fixed variable Ω , the dual formulation of optimization problem in Eq. (1), defined with respect to dual variables $\alpha_\ell \in \mathbb{R}^{n_\ell}$, is given by:

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

where \mathcal{L}_ℓ^* and \mathcal{R}^* are the conjugate dual functions of \mathcal{L}_ℓ and $\text{tr}(W \Omega W^\top) + \lambda_2 / \lambda_1 \|W\|_F^2$, respectively, and α_ℓ^i in $\alpha \in \mathbb{R}^n$ is the dual variable of the i -th data point (x_ℓ^i, y_ℓ^i) for the ℓ -th node. Meanwhile, we 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 Multi-task Learning

In this section, we firstly introduce the problem setting of the data poisoning attack on federated machine learning. Then three

kinds of attacks based on real-world scenarios are provided, followed by a bilevel formulation to compute optimal attacks.

Suppose that the attacker aims to degrade the 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 Eq. (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}_ℓ denotes the number of injected samples for the node ℓ . More specifically, \hat{D}_ℓ will be \emptyset , *i.e.*, $\hat{n}_\ell = 0$, if $\ell \notin \mathcal{N}_{\text{sou}}$. We also define the following three kinds of attacks based on real-world federated learning scenarios.

- **Direct attack:** $\mathcal{N}_{\text{tar}} = \mathcal{N}_{\text{sou}}$. Attacker can directly inject data to all the *target* nodes since a door will be opened when collecting data. For example, consider learning or recognizing the activities of mobile phone users in a cell network based on their individual sensor, text or image data. Attackers can directly attack the target mobile phones (nodes) by providing counterfeit sensor data into the target phones (nodes).
- **Indirect attack:** $\mathcal{N}_{\text{tar}} \cap \mathcal{N}_{\text{sou}} = \emptyset$. Attacker cannot directly inject data to any of the target nodes. However, due to the communication protocol existing amongst multiple mobile phones, the attacker can inject poisoned data to other mobile phones and affect the target nodes in an indirect way.
- **Hybrid attack:** An attack style which combines direct attack and indirect attack, *i.e.*, the attacker can inject poisoned data samples into target nodes and source attacking nodes simultaneously.

To maximally degrade the performance of *target* nodes, we then formulate the optimal attack problem as the following bilevel optimization problem by following [5]:

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

where \hat{D}_s denotes the injected data for the s -th source attacking node. Intuitively, the variables in the upper level problem are the data points \hat{D}_s , and we denote the upper level problem as a \mathcal{H} in this paper. The lower level problem in the Eq. (3) is a federated multi-task learning problem with training set consisting of both clean and injected data points. Therefore, the lower level problem can be considered as the constraint of the upper level problem.

IV. ATTACK ON FEDERATED LEARNING (AT²FL)

This section proposes an effective algorithm for computing optimal attack strategies, *i.e.*, AT²FL. More specifically, we follow the setting of most data poisoning attack strategies (*e.g.*, [5], [21]), and design a projected stochastic gradient ascent based algorithm that efficiently maximally increases the empirical loss of target nodes, and further damages their classification/regression performance. Notice that there is no closed-form relation between the empirical loss and the injected data, so we intend to compute the gradients by exploiting the optimality conditions of the subproblem.

A. Attacking Alternating Minimization

Bilevel problems are usually hard to optimize due to their non-convexity property. In our bilevel formulation Eq. (3), although the upper level problem is a simple primal problem, the lower level problem is highly non-linear and non-convex. To effectively solve this problem, the idea is to iteratively update the injected data in the direction of maximizing the function \mathcal{H} of target nodes. Therefore, in order to reduce the complexity of 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. Then the update rules for injected data \hat{x}_s^i can be written as:

$$(\hat{x}_s^i)^k \leftarrow \text{Proj}_{\mathbb{X}}((\hat{x}_s^i)^{k-1} + \eta_1 \nabla_{(\hat{x}_s^i)^{k-1}} \mathcal{H}), \quad (4)$$

where $\eta_1 > 0$ is the step size, k denotes the k -th iteration, and \mathbb{X} represents the feasible region of the injected data, which is specified by the first constraint in the upper level problem \mathcal{H} . More specifically, $\text{Proj}_{\mathbb{X}}(x)$ is x if $\|x\|_2 \leq r$; $xr/\|x\|_2$, otherwise, i.e., \mathbb{X} can be considered as an ℓ_2 -norm ball by following [7]. Accordingly, the corresponding dual variable $\hat{\alpha}_s^i$ can be updated gradually as the \hat{x}_s^i comes as following:

$$(\hat{\alpha}_s^i)^k \leftarrow (\hat{\alpha}_s^i)^{k-1} + \Delta(\hat{\alpha}_s^i). \quad (5)$$

B. Gradients Computation

To compute the gradient of the t -th target node in Eq. (4), i.e., $\nabla_{\hat{x}_s^i} \mathcal{L}_t(D_t, w_t)$, we adopt the chain rule and then obtain the following equation:

$$\nabla_{\hat{x}_s^i} \mathcal{L}_t(D_t, w_t) = \nabla_{w_t} \mathcal{L}_t(D_t, w_t) \cdot \nabla_{\hat{x}_s^i} w_t. \quad (6)$$

From the equation above, we can notice that the first term on the right side can be easily computed since it depends only on the loss function $\mathcal{L}_t(\cdot)$. However, the second term on the right side depends on the optimality conditions of lower level problem of Eq. (3). In the following, we show how to compute the gradient $\nabla_{\hat{x}_s^i} w_t$ with respect to two commonly-adopted loss functions: least-square loss (regression problems) and hinge loss (classification problems). Please check the definitions in the **Appendix A**.

Based on the loss functions above, we first fix variable Ω to eliminate the constraints of the lower level problem, and obtain the following dual subproblem:

$$\min_{\alpha, W} \sum_{\ell=1}^m \frac{1}{n_\ell + \hat{n}_\ell} \mathcal{L}_\ell^*(D_\ell \cup \hat{D}_\ell) + \lambda_1 \mathcal{R}^*(X\alpha). \quad (8)$$

Since the optimality of the lower level problem can be considered as a constraint to the upper level problem, we can treat Ω in problem of Eq. (8) as a constant-value matrix when computing the gradients. Additionally, since $\mathcal{R}^*(X\alpha)$ is continuous differentiable, W and α could be connected via defining the next formulation by following [14]:

$$w_\ell(\alpha) = \nabla \mathcal{R}^*(X\alpha). \quad (9)$$

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

Algorithm 1 ATTack on Federated Learning (AT²FL)

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

- 1: Randomly initialize $\forall \ell \in \mathcal{N}_{\text{sou}}$,
 - $\hat{D}_\ell^0 = \{(\hat{X}_\ell^0, \hat{\alpha}_\ell^0, \hat{y}_\ell^0) | \hat{X}_\ell^0 \in \mathbb{R}^{d \times \hat{n}_\ell}, \hat{\alpha}_\ell^0 \in \mathbb{R}^{\hat{n}_\ell}, \hat{y}_\ell^0 \in \mathbb{R}^{\hat{n}_\ell}\} \quad (7)$
 - 2: Initialize $\hat{D}_\ell = \hat{D}_\ell^0, \forall \ell \in \mathcal{N}_{\text{sou}}$ and matrix Ω^0 ;
 - 3: **for** $k = 0, 1, \dots$ **do**
 - 4: Set subproblem learning rate η ;
 - 5: **for** all nodes $\ell = 1, 2, \dots, m$ in parallel **do**
 - 6: Compute the approximate solution $\Delta\alpha_\ell$ via Eq. (13) or Eq. (14);
 - 7: Update local variables $\alpha_\ell \leftarrow \alpha_\ell + \Delta\alpha_\ell$;
 - 8: **if** $\ell \in \mathcal{N}_{\text{sou}}$ **then**
 - 9: Compute the approximate solution $\Delta\hat{\alpha}_\ell$ via Eq. (13) or Eq. (14);
 - 10: Update local variables $\hat{\alpha}_\ell \leftarrow \hat{\alpha}_\ell + \Delta\hat{\alpha}_\ell$;
 - 11: **end if**
 - 12: **end for**
 - 13: Update W^k and Ω^k based on the latest α ;
 - 14: **for** all source nodes $s = 1, 2, \dots, \mathcal{N}_{\text{sou}}$ in parallel **do**
 - 15: Update \hat{x}_s^i based on the Eq. (4);
 - 16: **end for**
 - 17: $\hat{D}_\ell = \hat{D}_\ell^k, \forall \ell \in \mathcal{N}_{\text{sou}}$;
 - 18: **end for**
-

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 Eq. (8) as the following constrained optimization problem for the ℓ -th node:

$$\min_{\alpha} \sum_{\ell=1}^m \frac{1}{n_\ell + \hat{n}_\ell} (\mathcal{L}_\ell^*(-\alpha_\ell^i) + \mathcal{L}_\ell^*(-\hat{\alpha}_\ell^{i'})) + \lambda_1 \mathcal{R}^*(X[\alpha_\ell; \hat{\alpha}_\ell]). \quad (10)$$

To solve Eq. (10) across distributed nodes, we can define the following data-local subproblems, which are formulated via a careful quadratic approximation of this dual problem to separate computation across the nodes. Moreover, at every step k , two samples (i.e., i in $\{1, \dots, n_\ell\}$ and i' in $\{1, \dots, \hat{n}_\ell\}$) 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:

$$\begin{aligned} (\alpha_\ell^i)^k &= (\alpha_\ell^i)^{k-1} + \Delta\alpha_\ell^i, \\ (\hat{\alpha}_\ell^{i'})^k &= (\hat{\alpha}_\ell^{i'})^{k-1} + \Delta\hat{\alpha}_\ell^{i'}, \end{aligned} \quad (11)$$

where both $\Delta\alpha_\ell^i$ and $\Delta\hat{\alpha}_\ell^{i'}$ are the stepsizes chosen to achieve maximal ascent of the dual objective in Eq. (10) when all variables are fixed. To achieve maximal dual ascent, one has to optimise:

$$\begin{aligned} \Delta\alpha_\ell^i &= \arg \min_{a \in \mathbb{R}} \mathcal{L}_\ell^*(-(\alpha_\ell^i + a)) + a \langle w_\ell(\alpha_\ell), x_\ell^i \rangle + \frac{\lambda_1}{2} \|x_\ell^i a\|_{M_\ell}^2 \\ \Delta\hat{\alpha}_\ell^{i'} &= \arg \min_{\hat{a} \in \mathbb{R}} \mathcal{L}_\ell^*(-(\hat{\alpha}_\ell^{i'} + \hat{a})) + \hat{a} \langle w_\ell(\hat{\alpha}_\ell), x_\ell^{i'} \rangle + \frac{\lambda_1}{2} \|x_\ell^{i'} \hat{a}\|_{M_\ell}^2 \end{aligned} \quad (12)$$

where $M_\ell \in \mathbb{R}^{d \times d}$ is the ℓ -th diagonal block of a symmetric positive definite matrix M . Furthermore, $M^{-1} =$

$\bar{\Omega} + \lambda_2/\lambda_1 I_{md \times md}$, where $\bar{\Omega} := \Omega \otimes I_{d \times d} \in \mathbb{R}^{md \times md}$. Therefore, $\Delta\alpha_\ell^i$ can be computed in a closed-form for the least-square loss function, i.e.,

$$\Delta\alpha_\ell^i = \frac{y_\ell^i - (x_\ell^i)^\top x_\ell^i \alpha_\ell^i - 0.5(\alpha_\ell^i)^{k-1}}{0.5 + \lambda_1 \|x_\ell^i\|_{M_\ell}^2}. \quad (13)$$

Meanwhile, $\Delta\hat{\alpha}_\ell^{i'}$ can be computed in a same manner. Furthermore, we substitute the hinge loss into the optimization problem in Eq. (8), and can obtain:

$$\begin{aligned} \Delta\alpha_\ell^i = & y_\ell^i \max \left(0, \min \left(1, \frac{1 - (x_\ell^i)^\top x_\ell^i (\alpha_\ell^i)^{k-1} y_\ell^i}{\lambda_1 \|x_\ell^i\|_{M_\ell}^2} + (\alpha_\ell^i)^{k-1} y_\ell^i \right) \right) \\ & - (\alpha_\ell^i). \end{aligned} \quad (14)$$

Update Gradient: Given Eq. (13), Eq. (14) with Eq. (9), we can compute the gradient in Eq. (4). For the least-square loss, we can the gradient of each injected data \hat{x}_s^i with its associated target node t :

$$\begin{aligned} & \nabla_{(\hat{x}_s^i)} \mathcal{L}_t((w_t)^\top x_t^j, y_t^j) \\ &= 2((w_t)^\top x_t^j - y_t^j) x_t^j \cdot \nabla_{\hat{x}_s^i} w_t \\ &= 2((w_t)^\top x_t^j - y_t^j) x_t^j \cdot \Delta\hat{\alpha}_s^i \Omega(t, s), \end{aligned} \quad (15)$$

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

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

Finally, the whole optimization procedure of attack on federated learning can be summarized in **Algorithm 1**.

V. EXPERIMENTS

In this section, we evaluate the performance of the proposed poisoning attack strategies, and its convergence. More specifically, we firstly introduce several adopted datasets, followed by the experimental results. Then some analyses about our proposed model are reported.

A. Real Datasets

We adopt the following benchmark datasets for our experiments, containing three classification datasets and one regression dataset:

EndAD (Endoscopic image abnormality detection): this dataset is collected from 544 healthy volunteers and 519 volunteers with various lesions, including gastritis, cancer, 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. 2. In the implementation, we extract a 128-dimensional deep feature via the VGG model trained in [6]. 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 dataset with 6 clinics, where each clinic can be regarded as one node, i.e., 6 nodes.



Fig. 2. Sample images from **EndAD** dataset, where the first and the second rows are healthy images and lesion images, respectively.

Human Activity Recognition¹: this classification dataset consists of mobile phone accelerometer and gyroscope data, which are collected from 30 individuals, performing one of six activities: {walking, walking-upstairs, walking-downstairs, sitting, standing, lying down}. The provided feature vectors of time is 561, whose variables are generated from 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 dataset intends to detect whether or not a land mine is presented in an area based on radar images. Moreover, it is modeled as a binary classification problem. Each object in a given dataset is represented by a 9-dimensional feature vector (four-moment based features, three correlation-based features, one energy-ratio feature, one spatial variance feature) and the corresponding binary label (1 for landmine and -1 for clutter). This dataset consists of a total of 14,820 samples divided into 29 different geographical regions, i.e., the total node number is 29.

Parkinson Data² is used to predict Parkinson's disease symptom score for patients based on 16 biomedical features. The Parkinson dataset contains 5,875 observations for 42 patients, and predicting the symptom score for each patient is treated as a regression task, leading to 42 regression tasks with the number of instances for each task ranging from 101 to 168. Additionally, the dataset's output is a score consisting of Total and Motor, we thus have two regression datasets: **Parkinson-Total** and **Parkinson-Motor** in this experiment.

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

B. Evaluating Attack Strategies

In our experiments, we evaluate the performance of our proposed poisoning attack strategies with random/direct/indirect/hybrid attacks on different datasets. For each dataset, we randomly choose half of nodes as N_{tar} , while selecting the rest of nodes as free nodes. For example, the number of N_{tar} for Human Activity dataset is 15, and the

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

²<https://archive.ics.uci.edu/ml/datasets/parkinsons+telemointoring>

TABLE II

COMPARISONS BETWEEN OUR MODEL AND STATE-OF-THE-ARTS IN TERMS OF CLASSIFICATION ERROR AND RMSE ON FIVE DATASETS: MEAN AND STANDARD ERRORS OVER 10 RANDOM RUNS. MODELS WITH THE BEST PERFORMANCE ARE BOLDED.

	Metrics	Non attacks	Random direct attacks	Random indirect attacks	Random hybrid attacks	Direct attacks	Indirect attacks	Hybrid attacks
EndAD	Error(%)	6.881 \pm 0.52	7.659 \pm 1.14	6.888 \pm 0.45	7.154 \pm 0.16	28.588\pm3.74	7.324 \pm 0.62	16.190 \pm 2.26
Human Activity	Error(%)	2.586 \pm 0.84	3.275 \pm 0.71	2.894 \pm 0.83	3.172 \pm 0.69	29.422\pm2.96	3.438 \pm 0.34	17.829 \pm 2.75
Landmine	Error(%)	5.682 \pm 0.28	5.975 \pm 0.36	5.735 \pm 0.36	5.819 \pm 0.22	13.648\pm0.54	7.428 \pm 0.39	9.579 \pm 0.27
	Avg. Error (%)	5.049 \pm 0.55	5.636 \pm 0.74	5.172 \pm 0.55	5.382 \pm 0.36	23.886\pm2.41	6.069 \pm 0.45	14.533 \pm 1.76
Parkinson-Total	RMSE	6.302 \pm 0.45	13.651 \pm 2.10	6.633 \pm 0.75	11.145 \pm 1.83	44.939\pm3.21	7.763 \pm 0.82	21.990 \pm 3.17
Parkinson-Moter	RMSE	4.125 \pm 0.50	11.472 \pm 2.51	5.046 \pm 1.14	9.422 \pm 1.81	32.992\pm3.78	6.866 \pm 1.21	16.956 \pm 3.78
	Avg. RMSE(%)	5.213 \pm 0.48	12.562 \pm 2.31	5.839 \pm 0.95	10.284 \pm 1.82	38.966\pm3.49	7.314 \pm 1.02	19.473 \pm 3.48

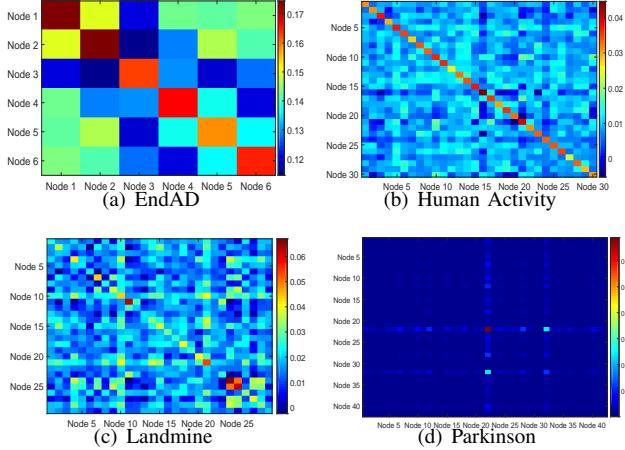


Fig. 3. The correlation matrices Ω of different datasets: End_AD dataset (a), Human Activity dataset (b), Landmine dataset (c) and Parkinson dataset (d), where the first half of nodes and the rest of nodes denote the \mathcal{N}_{tar} and \mathcal{N}_{sou} for the indirect attack scenario, respectively. The darker color indicates the higher correlation for each dataset, and vice versa.

number of \mathcal{N}_{tar} for EndAD dataset is 3, respectively. Moreover, the details of evaluated attack strategies are:

- Direct attacks: all the source attacking nodes \mathcal{N}_{sou} are set as the same as the target nodes, *i.e.*, $\mathcal{N}_{sou} = \mathcal{N}_{tar}$.
- Indirect attacks: all the source attacking nodes \mathcal{N}_{sou} are from the rest of nodes, where the number of \mathcal{N}_{sou} is same as that of \mathcal{N}_{tar} .
- Hybrid attacks: all the source attacking nodes \mathcal{N}_{sou} are randomly selected from all the nodes, where the number of \mathcal{N}_{sou} is same as that of \mathcal{N}_{tar} .
- Random Direct/Indirect/Hybrid attacks: the attack strategy are same as that of Direct/Indirect/Hybrid attacks. However, the injected data samples are randomly chosen.

For the used parameters in our model, the step size η_1 in Eq. (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 averaged over ten random repetitions are presented in Table. II. From the presented results, we have the following observations:

- All the attack strategies (*e.g.*, direct attacks, random attacks, etc) have the impact on all the datasets, 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 datasets. For example, the perfor-

mances of EndAD and Human Activity datasets 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.

- 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.
- For the indirect attack strategy, we can notice its performances are not good as direct and hybrid attack strategies, *e.g.*, Human Activity Recognition dataset. 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 paper. To verify this observation, we also present the corresponding correlation matrix of each used dataset in Figure 3. As illustrated in Figure 3, the nodes from \mathcal{N}_{tar} and \mathcal{N}_{sou} have highly correlated in Landmine dataset, *i.e.*, Figure 3(c). However, the node correlations in the Parkinson dataset (*i.e.*, Figure 3(d)) are not close. This observation indicates the reason why indirect attack strategy performs better on the Landmine dataset when comparing with other used datasets.

C. Sensitivity Study and Convergence Analysis

This subsection 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 AT²FL.

- 1) *Effect of Ratio of Injected Data:* To investigate the effect of ratio of injected poisoned data, we use all the datasets in this subsection. For each dataset, we take the same attack setting as that in Table II, *e.g.*, for the direct attack, the \mathcal{N}_{tar} and \mathcal{N}_{sou} are selected by randomly selecting half of nodes as \mathcal{N}_{tar} , and the rest of nodes to form \mathcal{N}_{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 Figure 4. From the provided results in Figure 4, we can find that: 1) For all the used datasets, 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 AT²FL; 2) Obviously, the performances of direct

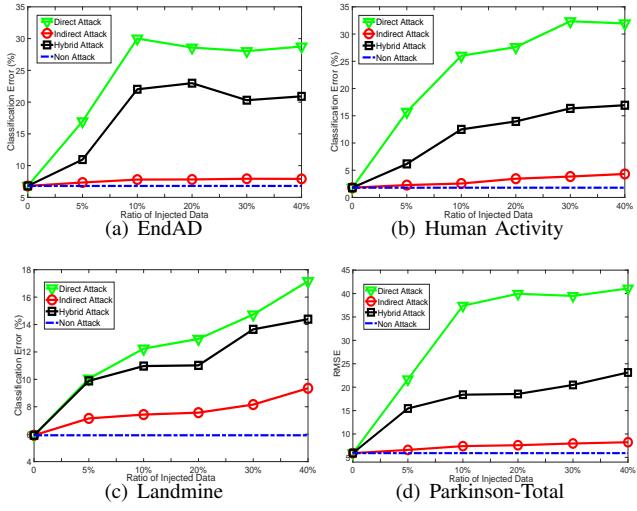


Fig. 4. The effect of ratio of injected data on EndAD (a), Human Activity (b), Landmine (c) and Parkinson-Total (d) datasets, where different lines denote different types of attacking strategies.

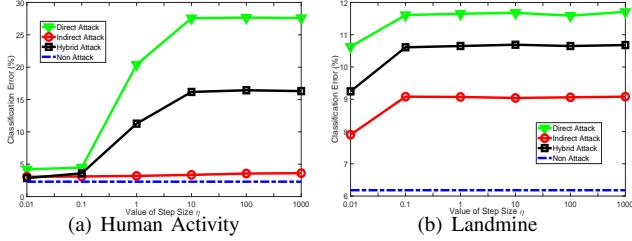


Fig. 5. The effect of step size η of Eq. (4) on Human Activity (a) and Landmine (b) datasets, where different lines denote different attacking strategies.

attack are better than that of indirect attack by varying the ratio of injected data. This because that it can directly involve the data of \mathcal{N}_{tar} ; 3) Indirect attack almost has no effect on EndAD dataset since the values of correlation matrix Ω for this dataset are relatively low, and each node can learn good classifier without the help of other nodes; 4) Different from other datasets, 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 Eq. (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 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 datasets in this subsection. By fixing other parameters (e.g., λ_1 and λ_2) and varying the value of step size η of Eq. (4) in range $\{0.01, 0.1, 1, 10, 100, 1000\}$, we present the corresponding 20-th iteration performances of different attack strategies in Figure 5. Notice that the performances of different attack strategies are improved with the increasing of the step size η . This is because the gradient in Eq. (4) with small step size will be treated as noise information, which will just have little impact on the classification performance. However, these attack strategies also outperforms the non-attack strategy. Furthermore,

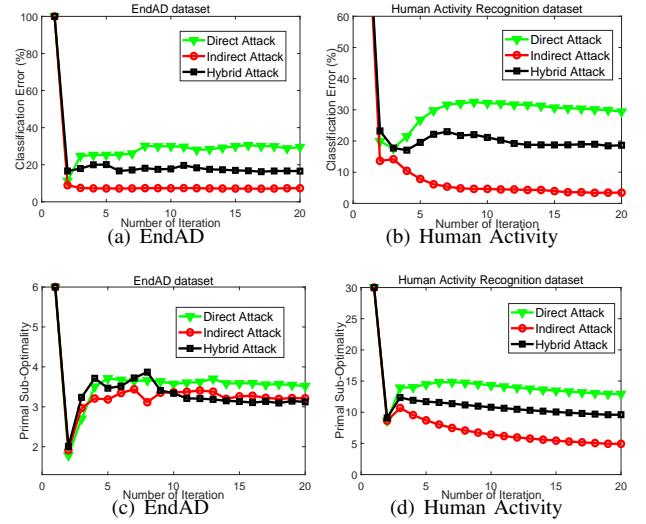


Fig. 6. The convergence curves of our proposed AT²FL algorithm, where the classification Errors are in (a) EndAD and (b) Human Activity datasets, and the primal sub-optimality performances of the lower level problem for Eq. (3) are in (c) EndAD and (d) Human Activity datasets.

the performance of different attack strategies tend to a fixed point with the increasing of step size η . This observation indicates that our AT²FL can effectively converge to a local optimum when the step size is enough.

3) *Convergence of Proposed AT²FL:* To study the convergence of our proposed AT²FL algorithm, we adopt the EndAD and Human Activity Recognition datasets in this subsection. Specifically, under different attack strategies, we present the primal sub-optimality of the lower level problem for Eq. (3) and classification Error (%) of the nodes in \mathcal{N}_{tar} in Figure 6. From the presented curves in Figure 6, we can notice that the original values of primal sub-optimality and classification error are higher. This is because that the weight matrix W in Eq. (3) is just initialized with the injected data points. Then our proposed AT²FL algorithm can converge to a local optima after a few iterations for all the three kinds of attacks on EndAD and Human Activity Recognition datasets. This observation indicates the effectiveness of our proposed AT²FL algorithm.

VI. CONCLUSION

In this paper, we take an earlier attempt on how to effectively launch data poisoning attacks on federated machine learning. Benefiting 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 a ATTack on Federated Learning (AT²FL) 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 AT²FL can significantly damage performances of real-world applications. From the study in this paper, we find that the communication protocol in federated learning can be used to effectively launch indirect attacks, e.g., when two nodes have 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 A

DEFINITION OF LEAST-SQUARE AND HINGE LOSSES

For the regression problem in this paper, 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:

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

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:

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

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

REFERENCES

- [1] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*, 2018.
- [3] Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.
- [4] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25. ACM, 2006.
- [5] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.
- [6] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.
- [7] Ingrid Daubechies, Massimo Fornasier, and Ignace Loris. Accelerated projected gradient method for linear inverse problems with sparsity constraints. *Journal of Fourier analysis and applications*, 14(5-6):764–792, 2008.
- [8] Adrià Gascón, Philipp Schoppmann, and Mariana Raykova. Secure linear regression on vertically partitioned datasets.
- [9] Shuguo Han, Wee Keong Ng, Li Wan, and Vincent CS Lee. Privacy-preserving gradient-descent methods. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):884–899, 2009.
- [10] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58. ACM, 2011.
- [11] Jakub Konečný, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.
- [12] Jakub Konečný, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.
- [13] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems*, pages 1885–1893, 2016.
- [14] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International conference on machine learning*, pages 64–72, 2014.
- [15] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.
- [16] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [17] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. Verifynet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security*, 15:911–926, 2019.
- [18] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12, 2019.
- [19] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 2019.
- [20] Mengchen Zhao, Bo An, and Christopher Kiekintveld. Optimizing personalized email filtering thresholds to mitigate sequential spear phishing attacks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [21] Mengchen Zhao, Bo An, Yaodong Yu, Sulin Liu, and Sinno Jialin Pan. Data poisoning attacks on multi-task relationship learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [22] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856. ACM, 2018.