参考该篇论文格式写

Covert Model Poisoning Against Federated Learning: Algorithm Design and Optimization

Kang Wei, *Student Member, IEEE,* Jun Li, *Senior Member, IEEE,* Ming Ding, *Senior Member, IEEE,* Chuan Ma, Yo-Seb Jeon, *Member, IEEE,* and H. Vincent Poor, *Life Fellow, IEEE*

Abstract—Federated learning (FL), as a type of distributed machine learning frameworks, is vulnerable to external attacks on FL models during parameters transmissions. An attacker in FL may control a number of participant clients, and purposely craft the uploaded model parameters to manipulate system outputs, namely, model poisoning (MP). In this paper, we aim to propose effective MP algorithms to combat state-of-the-art defensive aggregation mechanisms (e.g., Krum and Trimmed mean) implemented at the server without being noticed, i.e., covert MP (CMP). Specifically, we first formulate the MP as an optimization problem by minimizing the Euclidean distance between the manipulated model and designated one, constrained by a defensive aggregation rule. Then, we develop CMP algorithms against different defensive mechanisms based on the solutions of their corresponding optimization problems. Furthermore, to reduce the optimization complexity, we propose low complexity CMP algorithms with a slight performance degradation. In the case that the attacker does not know the defensive aggregation mechanism, we design a blind CMP algorithm, in which the manipulated model will be adjusted properly according to the aggregated model generated by the unknown defensive aggregation. Our experimental results demonstrate that the proposed CMP algorithms are effective and substantially outperform existing attack mechanisms.

Index Terms—Federated learning, model poisoning attack, robust aggregation

第一段，引入联邦学习
第二段，引入模型投毒攻击
第三段，讲对应的防御算法
第四段：突出本文工作

1 Introduction

With the development of Internet of Things (IoT), various end devices, such as sensors and smart phones, generate huge amounts of data and send them to cloud servers for processing [1]. Big data-driven artificial intelligence (AI) has been widely applied in many aspects of modern society [2], [3]. As a result, data privacy and confidentiality have become more and more concerned as they usually contain clients' sensitive information [4], [5], [6]. Federated learning (FL), emerging as a promising distributed machine learning paradigm [7], is capable of pushing model training to end devices without exposing their private training data. Therefore, in recent years, a wide range of privacy-sensitive applications are developed along with FL, such as mobile keyboard prediction [8] and visual object detection for safety [9], etc.

Although FL can help preserve clients' privacy, it is possibly trained across a fleet of unreliable devices with private and uninspectable datasets [10], [11], [12], [13], [14], compared with distributed datacenter learning and centralized learning schemes. Therefore, a new attack framework on federated training systems has be explored [15], i.e. model poisoning attack. Model poisoning takes advantage of the observation that a participant client in FL can directly influence parameters of the joint model. Therefore, in model poisoning, the attacker may take over a number of clients and manipulates the local model parameters sent from these clients to the server during the learning process [16], [17]. For example, with model poisoning, a competitor can degrade performance of a FL model or achieve its own goals by a particular trojan trigger [17], [18]. In addition, the client compromised by an attacker can also incorporate the evasion of potential defenses into its loss function during training process.

本地受损客户端的损失函数被修改

Therefore, defence mechanisms have drawn more and more attentions. Detection methods based on model validation are proposed to capture anomalous models uploaded by the clients, and reduce the weighting of these models when performing aggregation [19], [20], [21]. However, these detection methods, relying on auxiliary validation dataset, will increase the risk of the privacy leakage and become impractical for real-time training due to high complexity. As a type of alternative defence mechanisms, robust aggregation rules (e.g., Krum [22] and Trimmed mean [23]) take the advantage of low complexity and no additional privacy concerns. To be specific, in Krum [22], for each client's model, the server will calculate the sum of its Euclidean distances to the models of other clients, and select the one which has the minimum sum. In Trimmed mean [23], for the parameters embedded in a designated position of local models, the server will remove a number of the largest and smallest values before aggregation. It can be noted that both Krum and Trimmed mean can effectively mitigate the impact of unreasonable models with low complexity.

选取代表性的几个鲁棒聚合算法。

- Kang Wei, Jun Li and Chuan Ma are with School of Electrical and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. Jun Li is also with the School of Computer Science and Robotics, National Research Tomsk Polytechnic University, Tomsk, 634050, RUSSIA. E-mail: {kang.wei, jun.li, chuan.ma}@njust.edu.cn.
- Ming Ding is with Data61, CSIRO, Sydney, Australia. E-mail: ming.ding@data61.csiro.au.
- Yo-Seb Jeon is with Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, South Korea. E-mail: yoseb.jeon@postech.ac.kr.
- H. Vincent Poor is with Department of Electrical Engineering, Princeton University, NJ, USA. E-mail: poor@princeton.edu.

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

该模型投毒攻击可以降低全局模型性能，或者达到攻击者的某种目的。

In this paper, we are interested in proposing model poisoning attacks against state-of-the-art robust aggregation rules implemented at the server. The proposed attacking algorithms will stealthily cheat the server to adopt compromised models from the manipulated clients, termed as covert model poisoning (CMP). It is expected that the designed CMP will destroy the original FL model for performance degradation or achieve the attacker's purposes. To the best of the authors' knowledge, this the first piece of work in FL that systematically studies on model poisoning.

The main contributions can be summarized as follows:

- We formulate the model poisoning as an optimization problem by minimizing the Euclidean distance between the manipulated model and designated one, constrained by a defensive aggregation rule. Then, we develop CMP algorithms against different defensive aggregation rules according to the solutions of their corresponding optimization problems.
- We also propose a low complexity CMP algorithm for Krum with a slight performance degradation. In this algorithm, we reduce the searching dimension of the optimization problem when comparing the summations of Euclidean distances among the clients.
- In the case that the attacker does not know the defence mechanism, we design a blind CMP algorithm, in which the manipulated model will be adjusted properly according to the aggregated model.
- We conduct extensive experiments on real-word datasets, i.e., MNIST, CIFAR and House pricing dataset. The experimental results demonstrate that the proposed CMP algorithms are more effective than existing attack mechanisms, such as Arjun's attack and label flipping attack. More specifically, our original CMP can achieve a high rate of attacker's accuracy ($\approx 90\%$). For instance, the aggregated model can be manipulated successfully by the CMP under the Krum, and then wrongly identify a given digit 9 as 8 in MNIST. Meanwhile, our CMP with approximated constraint achieves a rate of 87% in terms of the attacker's accuracy and a 73% complexity reduction relative to the original CMP. Furthermore, the proposed blind CMP algorithm can also achieve a rate of 76% in terms of the attacker's accuracy.

The rest of this paper is organized as follows. Section 2 introduces the related background. A detailed description of the proposed algorithms is given in Section 3. Section 4 introduces the experimental setup. In Section 5, the experimental results are presented and discussed. Finally, this paper is concluded in Section 6. A summary notations can be seen in Tab. 1.

2 Preliminaies

In this section, we will present preliminaries and related background knowledge on FL and model poisoning attack.

2.1 Federated Learning

As a kind of decentralized training frameworks [24], FL can preserve clients' private information by its unique distribution learning mechanism. In details, all participants $\mathcal{C}_i$, $\forall i \in \boldsymbol{\mathcal{U}}$, $\boldsymbol{\mathcal{U}} = \{1, 2, \ldots, U\}$ only need to share the same learning objective and model structure, where the central server sends the current global model parameters to all clients in each communication round.

TABLE 1
Summary of Main Notations

Notation	Description
$\boldsymbol{\mathcal{U}}$	The set of all clients
U	Total number of all clients
$\boldsymbol{\mathcal{M}}$	The set of compromised clients
M	The number of compromised clients
$\boldsymbol{\mathcal{B}}$	The set of benign clients
B	The number of benign clients
$\mathcal{C}_i$	The i-th client
$\mathcal{D}_i$	The dataset held by the client $\mathcal{C}_i$
$\boldsymbol{\mathcal{D}}$	The set of all clients' datasets
$\lvert \cdot \rvert$	The cardinality of a set
t	The index of the t-th communication round
T	The number of communication rounds
$\boldsymbol{\theta}$	The vector of model parameters
$\boldsymbol{\theta}^t$	Global parameters aggregated from local parameters at the t-th communication round
$\boldsymbol{\theta}_i^t$	Local training parameters of the i-th client
$\widehat{\boldsymbol{\theta}}_i^t$	Local training parameters after attack's crafting
$\boldsymbol{\theta}^*$	The optimal parameters that minimize $F(\boldsymbol{\theta})$
$F(\boldsymbol{\theta})$	Global loss function
$F_i(\boldsymbol{\theta})$	Local loss function from the i-th client

Then, each client uploads the model parameters after the local training procedure based on the shared global model and local datasets $\mathcal{D}_i$, $\forall i \in \boldsymbol{\mathcal{U}}$, and then all uploaded models will be averaged by the server as the current global model, which is expressed as

$$\boldsymbol{\theta}^t = \sum_{i \in \boldsymbol{\mathcal{U}}} p_i \boldsymbol{\theta}_i^t, \tag{1}$$

where $\boldsymbol{\theta}^t$ is the global model at the t-th communication round, $\boldsymbol{\theta}_i^t$ is the uploaded model of i-th client at the t-th communication round, $p_i = |\mathcal{D}_i|/|\boldsymbol{\mathcal{D}}|$ and $\boldsymbol{\mathcal{D}} = \sum_{i \in \boldsymbol{\mathcal{U}}} \cup \mathcal{D}_i$. At the server, the goal is to learn a model over data that resides at the U associated clients. Formally, this FL task can be expressed as

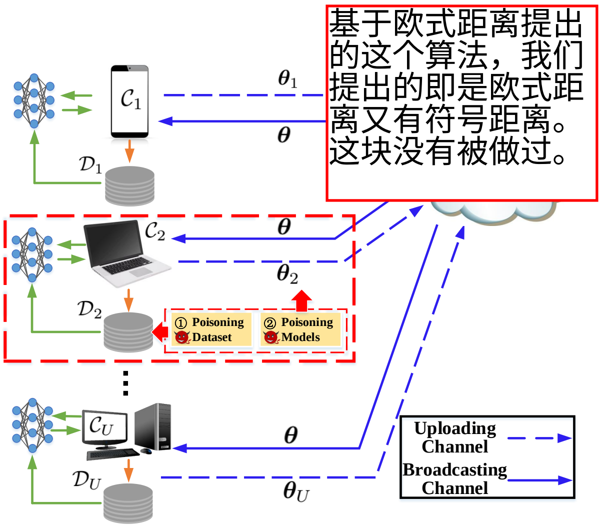
优化损失函数最小化

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} F(\boldsymbol{\mathcal{D}}, \boldsymbol{\theta}), \tag{2}$$

where $\boldsymbol{\Theta}$ represents the domain of legal models, $F(\boldsymbol{\mathcal{D}}, \boldsymbol{\theta}) = \sum_{i \in \boldsymbol{\mathcal{U}}} p_i F(\mathcal{D}_i, \boldsymbol{\theta})$ and $F(\mathcal{D}_i, \cdot)$ is the local objective function of the i-th client.

2.2 Model Poisoning Attack

In FL, model poisoning attack is a natural and powerful attack class [16], where an attacker can directly manipulate updates to the central server. Fig. 1 shows a high level of model poisoning attack compared with the data poisoning attack. We will focus on settings where an attacker controls some number of clients and craft the uploaded model parameters. This can result in convergence to sub-optimal models, or even lead to divergence. If the attacker has access to the model or non-compromise clients (updates and datasets), they may be able to craft their outputs to



基于欧式距离提出的这个算法，我们提出的即是欧式距离又有符号距离。这块没有被做过。

Fig. 1. Data poisoning attacks vs. model poisoning attacks.

have similar variances and magnitudes as the correct model updates, making them difficult to detect.

Attacker's Goal: The goal of an attacker can be classified into two categories: untargeted attacks and targeted attacks. Of particular importance to untargeted model makes the learnt model unusable and eventually lead to denial-of-service attacks [17]. For instance, an attacker may perform such attacks to its competitor's FL system. In targeted poisoning attacks, the learnt model produces attacker-desired predictions for particular testing examples, e.g., predicting spams as non-spams and predicting attacker-desired labels for testing examples with a particular trojan trigger (these attacks are also known as backdoor/trojan attacks [15]).

Attacker's Background Knowledge: In this work, we assume that the attacker may know the global model, local training datasets and local models on the compromised clients. Therefore, we characterize the attacker's background knowledge in three scenarios:

- **Full Knowledge Background.** In the case of full knowledge background, the attacker knows the local training datasets and local models of all the clients as well as the aggregation rule. In particular, the attacker could know the aggregation rule in various scenarios. For instance, the service provider may make the aggregation rule public in order to increase transparency and trust of the FL system.
- **Partial Knowledge Background.** In the case of partial knowledge background, besides the aggregation rule, the attacker only knows the global model, local training datasets and local models of the compromised clients.
- **No Knowledge Background.** In the case of no knowledge background, the attacker does not know the aggregation rule. However, since the attacker controls the compromised clients, it knows their local training datasets and local models.

Participant Collusion: An important axis to evaluate in the context of specific federated settings is the capability of participant collusion. In training-time attacks, there may be various attackers compromising various numbers of clients. Intuitively, the attackers may be more effective if they are able to coordinate their poisoned updates than if they each acted individually. Collusion may not happen in 'real time' (within-update), but rather across model updates (cross-update collusion).

3 Proposed Covert Model Poisoning

In this section, we will propose CMP algorithms with the aim of targeted attacks under various knowledge backgrounds.

3.1 Problem Formulation for CMP

总的优化目标是使得聚合参数偏向投毒参数

In the model poisoning attack, the attacker may directly control a number of compromised clients and manipulate their uploaded models to influence the behavior of the learning algorithm according to predefined goals. We assume that there exists M clients being compromised by an attacker, and it will directly alter the outputs of these clients to bias the learned model towards to the objective $F_A(\cdot)$. We also define B as the number of benign clients and we have $M + B = U$. We define $\boldsymbol{\mathcal{M}}$ as the set of these compromised clients, $\boldsymbol{\mathcal{B}}$ as the set of benign clients, and $\boldsymbol{\mathcal{U}}$ as the set of all clients, where $\boldsymbol{\mathcal{M}} \subseteq \boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{U}}/\boldsymbol{\mathcal{M}}$. In details, in each communication round, each benign client computes a local parameter vector $\boldsymbol{\theta}_i$, $\forall i \in \boldsymbol{\mathcal{B}}$, but each compromised client provides an unreliable parameter vector $\widehat{\boldsymbol{\theta}}_{i'}$, $\forall i' \in \boldsymbol{\mathcal{M}}$. With a specific aggregation rule and all uploaded models, the server can update the global model. A traditional aggregation rule is to average the local model parameters as the global model parameters. For example, considering the mean aggregation rule, the aggregated model parameter $\widehat{\boldsymbol{\theta}}$ can be expressed as

$$\widehat{\boldsymbol{\theta}} = \sum_{i\in\boldsymbol{\mathcal{B}}} p_i \boldsymbol{\theta}_i + \sum_{i'\in\boldsymbol{\mathcal{M}}} p_{i'} \widehat{\boldsymbol{\theta}}_{i'}, \tag{3}$$

Due to the existence of the unreliable parameter vectors from compromised clients, the performance of the aggregated model $\widehat{\boldsymbol{\theta}}^t$ may be bad. However, the goal of the attacker is usually to find a set of M local poisoning models that minimizes the objective function $F_A(\cdot)$ when they are uploaded to the server. Hence, we can formulate the attacker's objective of each communication round as the following optimization problem:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^*_{\boldsymbol{\mathcal{M}}} &= \mathop{\arg\min}_{\widehat{\boldsymbol{\theta}}_{i'} \subseteq \boldsymbol{\Theta}, i' \in \boldsymbol{\mathcal{M}}} F_A\left(\widehat{\boldsymbol{\theta}}\right), \\ \text{s.t.} \quad & \widehat{\boldsymbol{\theta}} = \mathcal{A}(\widehat{\boldsymbol{\theta}}_{i'}; \boldsymbol{\theta}_i), \forall i' \in \boldsymbol{\mathcal{M}}, i \in \boldsymbol{\mathcal{B}}, \end{aligned} \tag{4}$$

where $\mathcal{A}$ represents the aggregation rule and $\widehat{\boldsymbol{\theta}}^*_{\boldsymbol{\mathcal{M}}} \triangleq \{\widehat{\boldsymbol{\theta}}^*_{i'} | i' \in \boldsymbol{\mathcal{M}}\}$ represents the optimal poisoning models.

3.2 CMP for Mean Aggregation

In the previous work, the attacker's goal is to make the aggregated model minimize the attacker's objective function $F_A(\cdot)$, not the legitimate clients' objective function $F(\cdot)$, e.g., mislead a spam filter to pass certain types of spam emails. If the server adopts the mean aggregation rule as Eq. (1), substituting the constrain Eq. (3) into the optimization objective, and then we can obtain the following optimization problem:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^*_{\boldsymbol{\mathcal{M}}} = & \mathop{\arg\min}_{\widehat{\boldsymbol{\theta}}_{i'}\subseteq\boldsymbol{\Theta}, i'\in\boldsymbol{\mathcal{M}}} \; F_A\left(\widehat{\boldsymbol{\theta}}\right), \\ & \text{s.t. } \widehat{\boldsymbol{\theta}} = \sum_{i\in\boldsymbol{\mathcal{B}}} p_i\boldsymbol{\theta}_i + \sum_{i'\in\boldsymbol{\mathcal{M}}} p_{i'}\widehat{\boldsymbol{\theta}}_{i'}, \end{aligned} \tag{5}$$

where $\boldsymbol{\Theta}$ is the feasible domain of the FL training models. The attacker's objective function $F_A(\widehat{\boldsymbol{\theta}})$ has been typically computed on a specific target model $\widehat{\boldsymbol{\theta}}^*$. In this example, we may define $F_A(\widehat{\boldsymbol{\theta}}) = \|\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}^*\|^2$ with an appropriate norm. If the attacker has the full knowledge of this system, we know that the solution of this optimization is available by solving the optimization problem in Eq. (5) directly.

However, if the attacker has the partial knowledge as described in Section 2.2, the optimal solution can be obtained in the following theorem.

Theorem 1 *With a certain target $F_A(\widehat{\boldsymbol{\theta}})$ and M compromised clients under the mean aggregation rule in FL, the crafted model can be calculated by*

$$\widehat{\boldsymbol{\theta}}_i = \frac{1}{\sum\limits_{i\in\boldsymbol{\mathcal{M}}} p_i}\left(\widehat{\boldsymbol{\theta}}^* + \left(\frac{2}{\sum\limits_{i\in\boldsymbol{\mathcal{M}}} p_i} - 1\right)\sum_{i\in\boldsymbol{\mathcal{M}}} p_i\boldsymbol{\theta}_i\right), \forall i \in \boldsymbol{\mathcal{M}}, \tag{6}$$

and the loss function value can be expressed as

$$F_A\left(\widehat{\boldsymbol{\theta}}\right) = \left(\frac{2}{\sum\limits_{i\in\boldsymbol{\mathcal{M}}} p_i} - 1\right)^2 \left\|\sum_{i\in\boldsymbol{\mathcal{M}}} p_i\boldsymbol{\theta}_i\right\|^2. \tag{7}$$

Proof: See Appendix A. □

From **Theorem** 1, we can obtain a solution for the attacker by estimating the local models of benign clients. This estimation method will also be used in the following algorithms. If the sever applies a robust aggregation rule (e.g., Krum [22] and trimmed mean [23]), the optimization of the term $\widehat{\boldsymbol{\theta}}^*_{\boldsymbol{\mathcal{M}}}$ in Eq. (4) will be more complicated. In the following subsection, we will propose attacking algorithms against the Krum aggregation.

3.3 CMP for Krum Aggregation with Full and Partial Knowledge Background

Krum [22] selects one of the U local models that is the most similar to other models as the global model. The intuition is that even if the selected local model is from a compromised client, its impact may be constrained since it is similar to other local models possibly from benign clients. In details, for each local model $\boldsymbol{\theta}_i$, the server computes the $U-M-2$ local models that are the closest to $\boldsymbol{\theta}_i$ with respect to Euclidean distance. Moreover, the server computes the sum of the distances between $\boldsymbol{\theta}_i$ and its closest $U-M-2$ local models. Krum selects the local model with the smallest sum of distances as the global model. When $M < \frac{U-2}{2}$, Krum has theoretical guarantees for the convergence for certain objective functions.

Without a loss of generality, we assume the first M clients are compromised. Our directed deviation goal is to craft local models $\widehat{\boldsymbol{\theta}}_1, \ldots, \widehat{\boldsymbol{\theta}}_M$ of the compromised clients. Recall that Krum selects one local model as the global model in each communication round. Suppose $\widehat{\boldsymbol{\theta}}$ is the selected local model in the current communication round. Our goal is to craft the M compromised local models such that the local model selected by Krum has the optimal solution to minimize $F_A(\widehat{\boldsymbol{\theta}})$. Therefore, we can make Krum select a certain crafted local model (e.g., $\widehat{\boldsymbol{\theta}}_1$ without a loss of generality) via crafting the M compromised local models. Then, we aim to solve the optimization problem in Eq. (8) by $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}_1$. Therefore, under the Krum rule $\mathcal{A}_{\text{krum}}$, our optimization problem can be expressed as

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^*_{\boldsymbol{\mathcal{M}}} = & \mathop{\arg\min}_{\widehat{\boldsymbol{\theta}}_{i'}\subseteq\boldsymbol{\Theta}, i'\in\boldsymbol{\mathcal{M}}} \; F_A\left(\widehat{\boldsymbol{\theta}}\right), \\ & \text{s.t. } \widehat{\boldsymbol{\theta}} = \mathcal{A}_{\text{krum}}(\widehat{\boldsymbol{\theta}}_{i'}; \boldsymbol{\theta}_i), \forall i' \in \boldsymbol{\mathcal{M}}, i \in \boldsymbol{\mathcal{B}}. \end{aligned} \tag{8}$$

We know that the output of Krum rule is an integer programming problem. This optimization requires solving a bilevel problem in which the outer optimization amounts to minimize the attacker's objective and the known dataset by the attacker, while the inner optimization corresponds to the aggregation rule on all received models. Since solving this problem is highly complex, previous work [25] has exploited gradient-based optimization, along with the idea of implicit differentiation. Under these conditions, it is possible to apply a gradient descent strategy to obtain a (possibly) local minimum of the optimization problem of Eq. (8) in an iterative manner.

Our proposed CMP algorithm against Krum is given as **Algorithm** 1. First, we define the known dataset and local models by the attacker as $\boldsymbol{\mathcal{D}}_{\text{att}}$ and $\boldsymbol{\theta}_{\text{att}}$, respectively. In the case of full knowledge background, we can obtain that $\boldsymbol{\mathcal{D}}_{\text{att}} = \boldsymbol{\mathcal{D}}_{\boldsymbol{\mathcal{U}}}$ and $\boldsymbol{\theta}_{\text{att}} = \boldsymbol{\theta}_{\boldsymbol{\mathcal{U}}}$, where $\boldsymbol{\theta}_{\boldsymbol{\mathcal{U}}} \triangleq \{\boldsymbol{\theta}_i | i \in \boldsymbol{\mathcal{U}}\}$ and $\boldsymbol{\mathcal{D}}_{\boldsymbol{\mathcal{U}}} \triangleq \{\mathcal{D}_i | i \in \boldsymbol{\mathcal{U}}\}$. Correspondingly, in the case of partial knowledge background, we have $\boldsymbol{\mathcal{D}}_{\text{att}} = \boldsymbol{\mathcal{D}}_{\boldsymbol{\mathcal{M}}}$, $\boldsymbol{\theta}_{\text{att}} = \boldsymbol{\theta}_{\boldsymbol{\mathcal{M}}}$, where $\boldsymbol{\theta}_{\boldsymbol{\mathcal{M}}} \triangleq \{\boldsymbol{\theta}_i | i \in \boldsymbol{\mathcal{M}}\}$ and $\boldsymbol{\mathcal{D}}_{\boldsymbol{\mathcal{M}}} \triangleq \{\mathcal{D}_i | i \in \boldsymbol{\mathcal{M}}\}$. With the different degrees of the knowledge background, the algorithm optimizes all compromised models $\boldsymbol{\theta}_{\boldsymbol{\mathcal{M}}}$ in each communication round, by updating their feature vectors according to a given direction obtained by the gradient descent strategy. Therefore, at the t-th communication round, the update rule can be expressed as:

$$\widehat{\boldsymbol{\theta}}^t_{1,k+1} = \Pi_{\boldsymbol{\Theta}}\left(\widehat{\boldsymbol{\theta}}^t_{1,k} - \eta_k \nabla_{\widehat{\boldsymbol{\theta}}_1} F_A(\widehat{\boldsymbol{\theta}}^t_{1,k})\right), \tag{9}$$

where k represents the iteration when optimizing Eq. (8) and $\Pi_{\boldsymbol{\Theta}}(\cdot)$ is a projection operator to project $\boldsymbol{\theta}$ onto the feasible domain $\boldsymbol{\Theta}$, to handle bounded feature values. Note that this update step should also enforce $\widehat{\boldsymbol{\theta}}^t_{1,k+1}$ to lie within the feasible domain $\boldsymbol{\Theta}$, which can be typically achieved through the robust aggregation rule. Then, in order to achieve the goal of participant collusion, this algorithm will obtain the updated $\widehat{\boldsymbol{\theta}}^t_{i,k+1}$, $i = 2, 3, \ldots, M$ by adding slight noises (Gaussian noises with zero mean and σ standard deviation) to $\widehat{\boldsymbol{\theta}}^t_{1,k+1}$. Each noise vector should be clipped by the clipping threshold ε. After updating the crafted models, this algorithm will check whether Krum selects $\widehat{\boldsymbol{\theta}}^t_{1,k+1}$ as the global model. if not, then we decrease the step size η with a decay parameter λ. We repeat this process until Krum selects $\widehat{\boldsymbol{\theta}}^t_{1,k+1}$ or η is smaller than a certain threshold ς.

Algorithm 1 Original Covert Model Poisoning

```
Require: $\mathcal{D}_{\text{att}} = \mathcal{D}_{\mathcal{U}}$ (full knowledge), $\mathcal{D}_{\text{att}} = \mathcal{D}_{\mathcal{M}}$ (partial knowledge), $\sigma$, $\eta_0$, $\varepsilon$ and $\lambda$.
t ← 0 (communication round counter)
while $t < T$ do
    The compromised clients craft models as follows:
    $\boldsymbol{\theta}^t_{\text{att}} = \boldsymbol{\theta}^t_{\mathcal{U}}$ (if full knowledge) and
    $\boldsymbol{\theta}^t_{\text{att}} = \boldsymbol{\theta}^t_{\mathcal{M}}$ (if partial knowledge)
    $\widehat{\boldsymbol{\theta}}^t_{1,k} \leftarrow \boldsymbol{\theta}^t$ and $k \leftarrow 0$ (iteration counter)
    repeat
        $\widehat{\boldsymbol{\theta}}^t_{1,k+1} \leftarrow \Pi_{\boldsymbol{\Theta}}\left(\widehat{\boldsymbol{\theta}}^t_{1,k} - \eta_k \nabla_{\widehat{\boldsymbol{\theta}}_1} F_A(\widehat{\boldsymbol{\theta}}^t_{1,k})\right)$
        for $i = 2, 3, \ldots, M$ do
            $\boldsymbol{n}_i \leftarrow \mathcal{N}(0, \sigma)$
            $\widehat{\boldsymbol{\theta}}^t_{i,k+1} \leftarrow \widehat{\boldsymbol{\theta}}^t_{1,k+1} + \varepsilon \boldsymbol{n}_i / \|\boldsymbol{n}_i\|$
        end for
        $\widehat{\boldsymbol{\theta}}^t_{k+1} \leftarrow \mathcal{A}_{\text{krum}}(\widehat{\boldsymbol{\theta}}^t_{i'}; \boldsymbol{\theta}^t_i), \forall i' \in \mathcal{M}, i \in \mathcal{B}$
        if $\widehat{\boldsymbol{\theta}}^t_{1,k+1} \neq \widehat{\boldsymbol{\theta}}^t_{k+1}$ then
            $\eta_{k+1} \leftarrow \lambda \eta_k$
            $\widehat{\boldsymbol{\theta}}^t_{1,k+1} \leftarrow \widehat{\boldsymbol{\theta}}^t_{1,k}$
        end if
        $k \leftarrow k + 1$
    until $\eta_k < \varsigma$
end while
```

3.4 CMP for Krum Aggregation with Low Complexity

The aforementioned algorithm is essentially a standard gradient-ascent algorithm with the integer constraint. The key challenge of solving the optimization problem is that the constraint of the optimization problem is highly nonlinear and the search space of the local models $\widehat{\boldsymbol{\theta}}_1, \ldots, \widehat{\boldsymbol{\theta}}_M$ is large. We know that there are $M-1$ compromised clients assisting $\widehat{\boldsymbol{\theta}}_1$. Because $M < \frac{U-2}{2}$, then we know $M - 1 < \frac{U}{2} - 2 < \frac{U}{2} - 1 < U - M - 2$. Therefore, it is necessary for the crafted model $\widehat{\boldsymbol{\theta}}_1$ to be close to $U - 2M - 1$ benign clients' with respect to Euclidean distance. Consider the differences among $\widehat{\boldsymbol{\theta}}_i$, $\forall i \in \mathcal{M}$, we set an enough small value ε, where $\|\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_j\| \leq \varepsilon$, $\forall i, j \in \mathcal{M}$. The sum of the Euclidean distances of the crafted model $\widehat{\boldsymbol{\theta}}_1$ can be expressed as $\min \sum_{j \in \mathcal{S}'} \|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\| + (M-1) \cdot \varepsilon$, where $\mathcal{S}' \subseteq \mathcal{B}$ is the subset of $\mathcal{B}$ and $|\mathcal{S}'| = U - 2M - 1$. Intuitively, we can utilize the sum of the Euclidean distances to simplify the Krum rule. Thus, we make the following approximation:

$$\min_{\substack{i \in \mathcal{U}, \mathcal{S} \subseteq \mathcal{U}/i, \\ |\mathcal{S}| = U - M - 2}} \sum_{j \in \mathcal{S}} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\| \lessapprox \min_{\substack{i \in \mathcal{B}, \mathcal{S} \subseteq \mathcal{B}/i, \\ |\mathcal{S}| = U - M - 2}} \sum_{j \in \mathcal{S}} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|, \tag{10}$$

where $\mathcal{S} \subseteq \mathcal{U}/i$ is the subset of $\mathcal{U}/i$. Our approximation represents suboptimal solutions to the optimization problem, which means that the attacks based on this approximation may have suboptimal performance. After this approximation, we can simplify the aforementioned optimization problem as

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^*_1 = & \arg\min_{\boldsymbol{\theta}_1 \subseteq \boldsymbol{\Theta}} F_A\left(\widehat{\boldsymbol{\theta}}^*, \widehat{\boldsymbol{\theta}}_1\right), \\ & \text{s.t. } \min \sum_{j \in \mathcal{S}'} \|\widehat{\boldsymbol{\theta}}_1 - \boldsymbol{\theta}_j\| + (M-1) \cdot \varepsilon - E \leq 0, \\ & \quad \mathcal{S}' \subseteq \mathcal{B}, |\mathcal{S}'| = U - 2M - 1, \end{aligned} \tag{11}$$

where

$$E = \min_{\substack{i \in \mathcal{B}, \mathcal{S} \subseteq \mathcal{B}/i, \\ |\mathcal{S}| = U - M - 2}} \sum_{j \in \mathcal{S}} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|. \tag{12}$$

In order to simplify the constraints of $\mathcal{S}'$, we can transform Eq. (11) into a mixed optimization problem as

转化为混合优化方法

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^*_1 = & \arg\min F_A\left(\widehat{\boldsymbol{\theta}}_1\right), \\ & \text{s.t. } \min_{\alpha} \sum_{j \in \mathcal{B}} \alpha_j \|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\| \\ & \quad + (M-1) \cdot \varepsilon - E \leq 0, \\ & \sum_{j \in \mathcal{B}} \alpha_j = U - 2M - 1, \alpha_j \in \{0, 1\}. \end{aligned} \tag{13}$$

The bilevel optimization problem in (13) is NP hard in general. Specifically, we require the attack space $\boldsymbol{\Theta}$ to be differentiable (e.g. the attacker can change the local models in $\boldsymbol{\Theta}$ for aggregation). We know that the objective $F_A(\cdot)$ of the attacker is usually convex. In the following, we will present an efficient solution for a broad class of local model poisoning attacks.

We can note that without the integer constraint, the Lagrangian method offers an effective solution for this optimization problem. Therefore, we decompose this problem into two problems $\mathbf{P}_1$ and $\mathbf{P}_2$. The problem $\mathbf{P}_1$ can be expressed as

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^\star_1 = & \arg\min_{\widehat{\boldsymbol{\theta}}_1 \subseteq \boldsymbol{\Theta}} F_A\left(\widehat{\boldsymbol{\theta}}_1\right), \\ & \text{s.t. } \sum_{j \in \mathcal{B}} \alpha_j \|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\| + (M-1) \cdot \varepsilon - E \leq 0. \end{aligned} \tag{14}$$

The solution of $\mathbf{P}_1$ can be expressed by $\widehat{\boldsymbol{\theta}}^\star_1 = h(\alpha)$. Therefore, the optimization problem $\mathbf{P}_2$ can be given by:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^*_1 = & \arg\min_{\alpha} F_A\left(h(\alpha)\right) \\ & \text{s.t. } \sum_{j \in \mathcal{B}} \alpha_j = U - 2M - 1 \\ & \quad \alpha_j(\alpha_j - 1) = 0, \forall j \in \mathcal{B}. \end{aligned} \tag{15}$$

Specifically, the Lagrangian function of the problem $\mathbf{P}_1$ can be written as

$$\begin{aligned} \mathcal{L}(\alpha, \lambda) = F_A\left(\widehat{\boldsymbol{\theta}}_1\right) + \lambda \cdot & \Bigg[\sum_{j \in \mathcal{B}} \alpha_j \|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\| \\ & + (M-1) \cdot \varepsilon - E\Bigg], \end{aligned} \tag{16}$$

Based on the Karush-Kuhn-Tucker (KKT) conditions, the model $\widehat{\boldsymbol{\theta}}^\star_1$ and the optimal Lagrangian multiplier λ should satisfy the following equation set:

$$\begin{cases} \dfrac{\partial F_A\left(\widehat{\boldsymbol{\theta}}_1\right)}{\partial \widehat{\boldsymbol{\theta}}_1} + \lambda \cdot \displaystyle\sum_{j \in \mathcal{B}} \dfrac{\alpha_j(\widehat{\boldsymbol{\theta}}^\star_1 - \boldsymbol{\theta}_j)}{\|\widehat{\boldsymbol{\theta}}^\star_1 - \boldsymbol{\theta}_j\|} = 0, \lambda > 0 \\ \displaystyle\sum_{j \in \mathcal{B}} \alpha_j \|\widehat{\boldsymbol{\theta}}^\star_1 - \boldsymbol{\theta}_j\| + (M-1) \cdot \varepsilon - E = 0. \end{cases} \tag{17}$$

Based on the equation set (17), we can obtain the model $\widehat{\boldsymbol{\theta}}^\star_1$ for the problem $\mathbf{P}_1$. Conventional nonlinear optimization often has high local searching ability but low global searching ability. The genetic algorithm uses survival of the fittest as a method to achieve a good solution for its optimization problem. Combining the genetic algorithm, we can escape from local minima and obtain satisfied results. By this approximation, we can obtain low complexity CMP algorithm based on **Algorithm** 1.

Furthermore, we can find that how to choose an initial model is crucial for the proposed algorithm Therefore, we derive an initial model for the optimization problem in **Algorithm** 1. Formally, we have the following theorem.

Theorem 2 *With the attacker's objective function* $F_A(\widehat{\boldsymbol{\theta}}^*, \widehat{\boldsymbol{\theta}}_1) = \|\widehat{\boldsymbol{\theta}}^* - \widehat{\boldsymbol{\theta}}_1\|^2$, $\widehat{\boldsymbol{\theta}}_{\text{init}}$ *is properly an initial model for CMP algorithm in each communication round as follows:*

$$\widehat{\boldsymbol{\theta}}_{\text{init}} = \Xi + \left(\widehat{\boldsymbol{\theta}}^* - \Xi\right) \left(\|\Xi\|^2 - \frac{1}{B}\sum_{j\in\boldsymbol{\mathcal{B}}}\|\boldsymbol{\theta}_j\|^2 + \Lambda\right)^{\frac{1}{2}} / \left\|\widehat{\boldsymbol{\theta}}^* - \Xi\right\|, \tag{18}$$

where $\Lambda = -\frac{U-2M-2}{2} - (M-1)\varepsilon + E$, $\Xi = \frac{1}{B}\sum_{j\in\boldsymbol{\mathcal{B}}}\boldsymbol{\theta}_j$, *and* $\widehat{\boldsymbol{\theta}}^*$ *is the adversarial objective model.*

Proof: See Appendix B. □

From **Theorem** 2, we can note that if Λ is larger, the result will be closer to $\widehat{\boldsymbol{\theta}}^*$. The intuition is that a larger Λ means a loose constraint and leads to a better solution for this optimization problem.

3.5 CMP with No Knowledge Background

In this subsection, we consider the CMP with no knowledge background (CMP-NKB) based on the aforementioned algorithm. When the attacker does not know the aggregation rule, this attacker can only adjust the crafted models via the feedback of the aggregation rule.

Algorithm 2 outlines our proposed CMP-NKB. At the t-th communication round, the server broadcasts the aggregated global model $\boldsymbol{\theta}^t$ to all clients. The benign clients respectively train the parameters by using local databases with preset termination conditions. After completing the local training, the i-th client, $\forall i$, will upload the local parameters $\boldsymbol{\theta}_i^t$ to the server for aggregation. However, different from these benign clients, the attacker aims to craft effective models to evade the robust aggregation rule instead of local training. In the CMP-NKB, when the global model has been received at the t-th communication round, the attacker may calculate the distance between the $\boldsymbol{\theta}^t$ and $\widehat{\boldsymbol{\theta}}_1^t$. Using this distance and threshold ξ, we can obtain the result whether our crafted models evade the robust aggregation rule successfully. If the result is positive, the attacker will enhance the poisoning degree by increasing the step size of the label flipping training. Meanwhile, the attacker need to weaken the poisoning degree by decreasing the step size.

The poisoning problem under the robust aggregation can be viewed as a global game between two players, i.e., a learner and an attacker. The game captures the interactions on a network of a FL training model including $U-M$ benign clients and a centralized server, and an attacker with M compromised clients. However, if we treat each compromised client as an independent attacker, then the global game can be treated as a multi-agent problem, which can be valuable future work and solved by the multi-agent reinforcement learning algorithms.

未来的挑战，多个恶意客户端参数不同。进行优化问题。

Algorithm 2 Original Covert Model Poisoning with No Knowledge Background (CMP-NKB)

```
Require: $\boldsymbol{\mathcal{D}}_{\text{att}} = \boldsymbol{\mathcal{D}}_{\boldsymbol{\mathcal{M}}}$, $\eta^0$ and $\lambda$.
$t \leftarrow 0$ (communication round counter)
while $t < T$ do
    The server broadcasts the aggregated model $\boldsymbol{\theta}^t$
    The compromised clients craft models as follows:
    if $\|\widehat{\boldsymbol{\theta}}_1^t - \boldsymbol{\theta}^t\| \leq \xi$ then
        $\eta^{t+1} \leftarrow \lambda\eta^t$
    else
        $\eta^{t+1} \leftarrow \eta^t/\lambda$
    end if
    $\widehat{\boldsymbol{\theta}}_1^t \leftarrow \Pi_{\Theta}\left(\boldsymbol{\theta}^t - \eta^t\nabla_{\boldsymbol{\theta}^t}F_A(\boldsymbol{\theta}^t; \boldsymbol{\mathcal{D}}_{\text{att}})\right)$
    for $i = 2, 3, \ldots, M$ do
        $\boldsymbol{n}_i \leftarrow \mathcal{N}(0, \sigma)$
        $\widehat{\boldsymbol{\theta}}_i^t \leftarrow \widehat{\boldsymbol{\theta}}_i^t + \varepsilon\boldsymbol{n}_i/\|\boldsymbol{n}_i\|$
    end for
    All clients upload the local models to the server
    The server aggregate uploaded models by
    a certain aggregation rule
    $t \leftarrow t + 1$
end while
```

注：放在我的方法中

4 Experimental Setup

In this section, we implemented our attacks using Pytorch. We trained all of the models on a server equipped with three Tesla P100 PCIe and each with 16 GB of memory. We evaluate the effectiveness of our proposed attack methods using multiple datasets and learning models in different scenarios, e.g., the impact of different parameters and known vs. different aggregation rules.

4.1 Datasets

Our experiments use three real datasets:

1) **House pricing dataset.** House pricing dataset is used to predict house sale prices as a function of predictor variables such as square footage, number of rooms, and location [26]. In total, it includes 1460 houses and 81 features. We preprocess by onehot encoding all categorical features and normalize numerical features, resulting in 275 total features;

2) **MNIST.** MNIST is a dataset of handwritten digits consists of 60000 training examples and 10000 testing examples [27] formatted as 28×28 size gray scale images;

3) **CIFAR-**10. CIFAR-10 consists of 60000 color images in 10 object classes such as deer, airplane, and dog with 6000 images included per class. The complete dataset is predivided into 50000 training images and 10000 test images.

We normalize each numerical dimension to $[0, 1]$. In addition, we also map labels to numbers such that the distance between two points with different labels is no smaller than the distance between points with the same label. In training and testing SVM, we map one label to '1' and the rest to '-1'. Each data point has a unit weight.

4.2 Machine Learning Models

Our experiments evaluate four supervised learning models including linear regression, support vector machine (SVM), multi-layer perceptron (MLP) and conventional neural network (CNN). 1) Linear regression (LR), which is performed with SGD on the House pricing dataset. We use the normalized cost as the following loss function. The considered aggregation rules have theoretical guarantees for the error rate of LR classifier. we conduct the house pricing prediction task by LR; 2) SVM, which is trained on the IPUMS-US dataset. In this model, the hinge loss function is applied. We conduct experiments using the SVM classifier to predict whether the digit is even or odd; 3) MLP, which is conducted on the standard MNIST dataset. For MNIST, We use a simple feedforward deep neural network with ReLU units and softmax of 10 classes (corresponding to the 10 digits) with the cross-entropy loss; 4) CNN, which has 2 convolutional layers with dropout and is applied for the CIFAR-10 dataset. For CIFAR-10, we also use softmax of 10 classes with the cross-entropy loss.

Our machine learning architecture does not necessarily achieve the smallest error rates for the considered datasets, as our goal is not to search for the best learning architecture. Our goal is to show that our attack methods can increase the testing error rates of the machine learning classifiers or bias the learned model towards the attack's objective.

4.3 Benchmarks

Furthermore, we compare existing attacks with our proposed methods, which are detailedly described as follows. 1) Gaussian attack. Specifically, for each compromised client, we sample a noise vector from the Gaussian distribution and add it on the parameter of the local model on the compromised client. We use this Gaussian attack to show that crafting compromised local models randomly can not effectively attack the Byzantine-robust aggregation rules; 2) Label flipping attack. This is a data poisoning attack that does not require knowledge of the training data distribution. On each compromised worker device, this attack flips the label of each training instance; 3) Fang's Full knowledge attack or partial knowledge attack [17]. These attacks manipulate the local model parameters on compromised worker devices during the learning process against the Byzantine robust FL. 4) Arjun' attack [28]. This attack considered the non-colluding malicious clients and designed an alternating minimization strategy, which alternately optimizes for the training loss and the adversarial objective.

4.4 Performance Metrics

For the House Pricing prediction, we evaluate the performance by the normalized cost (test loss). We use this dataset for the experiment of untargeted attack and a larger value of test loss means a better attack performance. For the MNIST, if we consider the case of untargeted attack, we will use the error rate of the FL model to evaluate our CMP. Specifically, if our CMP can achieve a high error rate, it can attack the FL system effectively. We also apply the MNIST and CIFAR-10 in the case of targeted attack. In this scenario, we will use attacker's accuracy (predicting the attacker-desired labels by testing data) to evaluate our CMP. Moreover, if the Krum is adopted in the FL system, we denote Successful Attacking Rate as the rate that compromised models are selected by the Krum in each communication.

4.5 Parameter Setting

We describe parameter settings for the FL and our attack methods. We record each experiment for 50 trials and report the average results. In our FL system, the number of total clients is set to 50, the number of compromised client is set from 0 to 10 and the degree of non-independent-and-identically-distributed (non-i.i.d.) is set in the range of $[0, 1]$. The targeted attack flips the labels of the MNIST or CIFAR-10 dataset with attacker-desired labels, which are shown in Tab. 2.

Inspired by the above optimization for the targeted attack, we can craft local models of compromised clients achieve untargeted attacks via solving the similar optimization, which only differs form the objective function. We can use the original objective function of the benign client and maximize it as the objective. As same as the targeted attack, we can utilize the same assumptions and approximations, and then solve it by the gradient ascent strategy.

5 Performance Evaluation

In this section, we start by presenting our results for the stand-alone scenario, followed by our results for the FL scenario. We perform the original CMP under full and partial knowledge backgrounds, denoted by CMP-FKB-Orgcontr and CMP-PKB-Orgcontr in our experiments, respectively. Correspondingly, the CMP attack with low complexity under full and partial knowledge backgrounds are named as CMP-FKB-Simcontr and CMP-PKB-Simcontr, respectively.

In Section 5.1 and Section 5.2, we consider the untargeted attack, where the attacker aim to destroy the FL model. In Section 5.3 and Section 5.4, we attack the MLP and CNN with MNIST or CIFAR-10 datasets with the proposed targeted attacking algorithms, respectively, where original labels and attacker-desired labels are shown in Tab. 2.

5.1 House Pricing Prediction

In the first scenario, we evaluate the performance of the house pricing prediction task by LR with the normalized cost. Figs. 2 and 3 show the loss function value and attack success rate using our untargeted attack methods under various percentages of compromised clients compared with existing works, respectively.

The results in Fig. 2 show that our attacks are effective and substantially outperform existing attacks. If the attacker has the full knowledge, our proposed attack can damage this LR model completely while the robust aggregation rule is existing. However, our proposed attack algorithm will have less effect compared with the full knowledge attack. We can also find that when the percentage of compromised clients is larger, our proposed attack will have a deeper effect, which is in line with the intuition. In Fig. 3, we show the attack success rate of our proposed attacks against the Krum rule compared with existing works. Our proposed

TABLE 2
Description of the targeted attack (original labels and attacker-desired labels).

MNIST/CIFAR-10										
Original Labels	0/Airplane	1/Automobile	2/Bird	3/Cat	4/Deer	5/Dog	6/Frog	7/Horse	8/Ship	9/Truck
Attacker-desired Labels	9/Truck	0/Airplane	1/Automobile	2/Bird	3/Cat	4/Deer	5/Dog	6/Frog	7/Horse	8/Ship

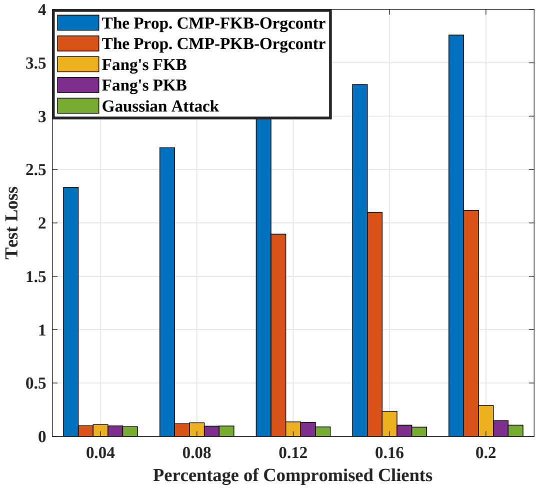


Fig. 2. The loss function of our proposed untargeted attacking algorithms i.e. the prop. CMP-PKB and the prop. CMP-FKB, against Krum aggregation compared with existing works under various percentages of compromised clients with $p = 0.5$ and $T = 30$.

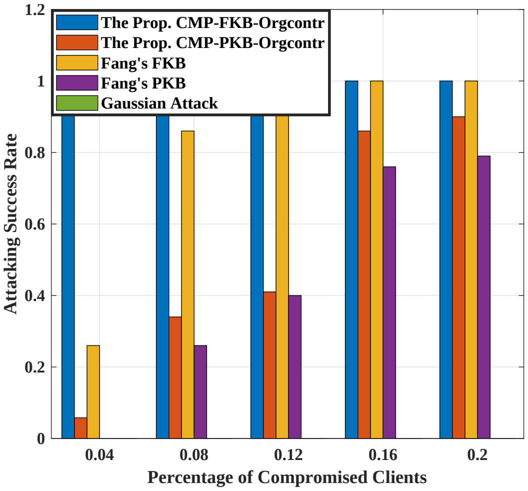


Fig. 3. The attack success rate of our proposed untargeted attacking algorithms i.e. the prop. CMP-PKB and the prop. CMP-FKB, against Krum aggregation compared with existing works under various percentages of compromised clients with $p = 0.5$ and $T = 30$.

attacks outperform other methods in both full knowledge and partial knowledge attacks.

Furthermore, Our proposed attack algorithms increase the error rates significantly as we compromise more clients and Gaussian attacks have no notable impact on the error rates. In Fig. 3, we can note that the attack success rate under the partial knowledge background increases with the percentage of compromised clients.

5.2 Parity Classifier using MNIST

In this subsection, we conduct experiments using the SVM classifier to predict whether the digit is even or odd. In Tab. 3, we show the error rate and the attack success rate of our proposed untargeted attacking methods against Krum aggregation compared with existing works, respectively.

First, these results show that our attacks are effective and substantially outperform existing attacks, i.e., our attacks result in higher error rates. For instance, when the degree of non-i.i.d. is set to 0, our CMP-FKB attack increases the error rate from 0.287 to 0.425 (around 48.08% relative increase) compared with Fang's FKB. Furthermore, our CMP-PKB attack also increases the error rate as well as the attack success rate compared with Fang's PKB under different degrees of non-i.i.d.. Finally, we can note that when the degree of non-i.i.d. is small, the attacker under the partial knowledge background has a large value of the attack success rate. The intuition is that if the degree of non-i.i.d. is small, the divergence of different local models will be small and the estimation of the unknown models by the compromised models will be accurate.

5.3 Handwriting Digits Recognition

In this subsection, we conduct our experiments using MNIST by the MLP classifier to classify the handwriting digits. We show the attack accuracy and the attack success rate of our proposed targeted attacking methods against mean, Krum and Trimmed mean aggregation rules compared with existing works, respectively.

In Tab. 4, we show the attacker's accuracy of our proposed CMP-FKB against mean aggregation compared with existing works under various degrees of non-i.i.d. with $U = 50$, $M = 10$ and $T = 30$. We can note that the proposed CMP algorithms are more effective than existing attacks and achieve a high attacker' accuracy (above $\%85$).

In Tab. 5, we show the comparison of attacker's accuracy between our proposed targeted attack algorithms, i.e. The Prop. CMP-FKB-Simcontr, The Prop. CMP-PKB-Orgcontr, The Prop. CMP-PKB-Simcontr and The Prop. CMP-PKB-Simcontr, and existing algorithms on MLP model and MNIST dataset, under various numbers of communication rounds with $p = 0.5$, $U = 50$ and $M = 10$. These results show that the proposed CMP algorithms are effective and substantially outperform existing attacks, such as Arjun's attack and label flipping attack. Considering Krum, with the original constraint, our proposed algorithms can usually achieve a high attacker's accuracy, especially for a large T. For instance, when $T = 50$, our full knowledge attack with the original constraint increases the attack accuracy from 0.052 to 0.904 as well as our partial knowledge attack increases it to 0.546. Meanwhile, using the approximate

TABLE 3
The comparison of test accuracy between our proposed untargeted attack algorithms, i.e. the prop. CMP-PKB, the prop. CMP-FKB and existing algorithms on SVM model and MNIST dataset under various degrees of non-i.i.d. with $U = 50$, $M = 10$ and $T = 30$.

The Degree of Non-i.i.d.	Approach	Error Rate (%)	Successful Attacking Rate (%)	Time Cost (second)
$p = 0$	The Prop. CMP-FKB-Orgcontr	42.498	100	3.357
	The Prop. CMP-PKB-Orgcontr	22.201	92.0	1.268
	Fang's FKB	28.714	100	0.077
	Fang's PKB	16.863	82.6	0.115
	Gaussian Attack	15.510	0.00	0.00073
$p = 0.5$	The Prop. CMP-FKB-Orgcontr	43.254	100	3.338
	The Prop. CMP-PKB-Orgcontr	22.058	98.0	1.561
	Fang's FKB	28.985	100	0.0737
	Fang's PKB	18.121	76.8	0.0531
	Gaussian Attack	15.783	0.00	0.00072
$p = 1.0$	The Prop. CMP-FKB-Orgcontr	39.161	100	3.469
	The Prop. CMP-PKB-Orgcontr	33.306	79.8	3.784
	Fang's FKB	31.510	100	0.067
	Fang's PKB	23.306	58.6	0.010
	Gaussian Attack	18.930	0.00	0.00073

TABLE 4
The attacker's accuracy of our proposed CMP-FKB against mean aggregation compared with existing works under various degrees of non-i.i.d. with $U = 50$, $M = 10$ and $T = 30$.

The Degree of Non-i.i.d.	Approach	Attacker's Accuracy (%)
$p = 0$	The Prop. CMP-FKB	89.38
	Arjun's Attack	12.50
	Label Flipping	4.16
$p = 0.5$	The Prop. CMP-FKB	86.91
	Arjun's Attack	11.71
	Label Flipping	3.77
$p = 1.0$	The Prop. CMP-FKB	90.29
	Arjun's Attack	8.78
	Label Flipping	3.84

constraint, our proposed algorithm will have a small time cost but a low attacker's accuracy.

In Tab. 6, we show the attacker's accuracy of our proposed CMP-NKB against Krum aggregation compared with existing works under various percentages of compromised clients with $p = 0.5$ and $T = 300$. We can note that the proposed algorithm also can achieve a high attacker's accuracy, although this attacker only know the information of compromised clients (the datasets of compromised clients and global model parameters).

In order to show the effectiveness of proposed attacks, we also use the Trimmed mean aggregation [25] in the FL system. In Tab. 7, we show the attacker's accuracy of CMP-NKB against Trimmed mean aggregation compared with existing works on MLP model and MNIST dataset with $p = 0.5$ and $T = 30$. We can note that CMP-NKB can also achieve a high attacker's accuracy compared with existing attacks, such as Arjun's attack and label flipping attack.

5.4 Visual Results

In this subsection, we apply several interpretability techniques, and then provide insights into the internal feature representations of the neural network under the proposed CMP-NKB. In detail, we use a suite of these techniques to try and discriminate between the behavior of a benign global model and one that has been trained to satisfy the adversarial objective of misclassifying a single example. Fig. 4 compares the outputs of MLP based FL with the interpretability technique corresponding to the digit 9 for malicious models at different communication rounds. We also show the results by CNN based FL with CIFAR-10 dataset in Fig. 5.

In Fig. 4, we can note that the visual results of MLP based FL model corresponding to the digit 9 using our CMP-NKB against Krum aggregation are similar with the digit 0. Especially, with the increasing numbers of communication rounds, the visual results trend to be closer to the digit 0 instead of 9, which means that our CMP-NKB can achieve an outstanding attacking performance. Besides, we also adopt the CNN based FL model with CIFAR-10 dataset and show the visual results in 5. We can also note that the proposed algorithm can successfully obtain the attacker's goal and make the FL model beneficial to the attacker, i.e. mislead the model to classify the horse as the frog.

6 Conclusions

In this paper, we have performed the systematic study of model poisoning attacks for FL models against defensive aggregation rules, i.e., Krum and Trimmed mean. We have formulated the model poisoning as an optimization problem by minimizing the Euclidean distance between the manipulated model and designated one, constrained by a defensive aggregation rule. Then, we have developed CMP algorithms against different defensive aggregation rules according to the solutions of their corresponding optimization problems. We have also proposed a low complexity CMP algorithm for Krum with a slight performance degradation. In the case that the attacker does not know the defence mechanism, we have designed a blind CMP algorithm, in which the manipulated model will be adjusted properly according to the aggregated model. Finally, We have conducted extensive experiments on real-word datasets, i.e., MNIST, CIFAR and House pricing dataset. The experimental results have demonstrated that the proposed CMP algorithms are more effective than existing attack mechanisms, such as Arjun's attack and label flipping attack.

TABLE 5
The comparison of attacker's accuracy between the proposed targeted attack algorithms, i.e. the Prop. CMP-FKB-Simcontr, the Prop. CMP-PKB-Orgcontr, the Prop. CMP-PKB-Simcontr, the Prop. CMP-PKB-Simcontr, and existing algorithms on MLP model and MNIST dataset, under various numbers of communication rounds with $p = 0.5$, $U = 50$ and $M = 10$.

The Number of Communication Rounds	Approach	Attacker's Accuracy (%)	Successful Attacking Rate (%)	Time Cost (Second)
$T = 30$	The Prop. CMP-FKB-Orgcontr	87.8	33.3	63.90
	The Prop. CMP-FKB-Simcontr	71.9	5.6	17.19
	The Prop. CMP-PKB-Orgcontr	36.6	4.0	14.01
	The Prop. CMP-PKB-Simcontr	10.4	10.0	6.76
	Arjun's Attack	9.6	0.00	0.015
	Label Flipping	9.2	0.00	0.00
$T = 50$	The Prop. CMP-FKB-Orgcontr	90.4	33.3	63.90
	The Prop. CMP-FKB-Simcontr	87.0	10.0	17.19
	The Prop. CMP-PKB-Orgcontr	54.6	4.6	14.01
	The Prop. CMP-PKB-Simcontr	5.0	10.0	6.76
	Arjun's Attack	5.2	0.00	0.015
	Label Flipping	4.7	0.00	0.00
$T = 70$	The Prop. CMP-FKB-Orgcontr	86.7	100	63.90
	The Prop. CMP-FKB-Simcontr	87.3	33.3	17.19
	The Prop. CMP-PKB-Orgcontr	70.0	6.4	14.01
	The Prop. CMP-PKB-Simcontr	3.2	100	6.76
	Arjun's Attack	3.3	0.00	0.015
	Label Flipping	6.3	0.00	0.00

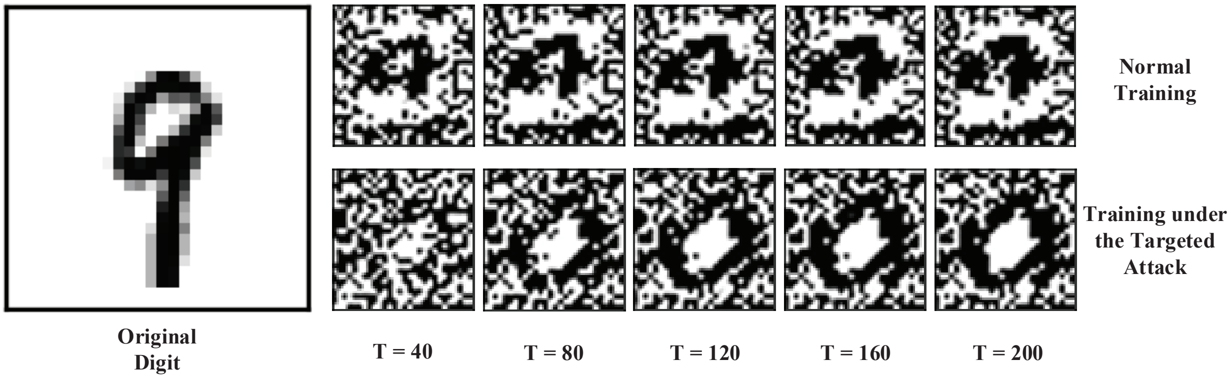


Fig. 4. The visual results of MLP based FL with the interpretability technique corresponding to the digit 9 under our proposed CMP-NKB against Krum aggregation at different communication rounds. Basically speaking, the subfigures show a “typical” 9 understood by the machine model with normal training and targeted attack.

TABLE 6
The attacker's accuracy of the proposed CMP-NKB against Krum aggregation compared with existing works on MLP model and MNIST dataset under various percentages of compromised clients with $p = 0.5$, $U = 50$ and $T = 300$.

Percentage of Compromised Clients	Approach	Attacker's Accuracy (%)
0.20	The Prop. CMP-NKB	75.72
	Arjun's Attack	3.58
	Label Flipping	2.02
0.16	The Prop. CMP-NKB	40.43
	Arjun's Attack	1.30
	Label Flipping	2.15
0.12	The Prop. CMP-NKB	11.59
	Arjun's Attack	2.6
	Label Flipping	5.14
0.08	The Prop. CMP-NKB	12.63
	Arjun's Attack	2.93
	Label Flipping	2.08
0.04	The Prop. CMP-NKB	9.24
	Arjun's Attack	2.80
	Label Flipping	2.21

TABLE 7
The attacker's accuracy of the proposed CMP-NKB against Trimmed mean aggregation compared with existing works on MLP model and MNIST dataset under various degrees of non-i.i.d. with $U = 50$, $M = 10$ and $T = 300$.

The Degree of Non-i.i.d.	Approach	Attacker's Accuracy (%)
$p = 0$	The Prop. CMP-NKB	34.56
	Arjun's Attack	3.67
	Label Flipping	4.72
$p = 0.5$	The Prop. CMP-NKB	64.82
	Arjun's Attack	8.71
	Label Flipping	5.40
$p = 1.0$	The Prop. CMP-NKB	70.23
	Arjun's Attack	10.52
	Label Flipping	5.89

REFERENCES

[1] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, “IoT-based big data storage systems in cloud computing: Perspectives and challenges,” *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, 2017.

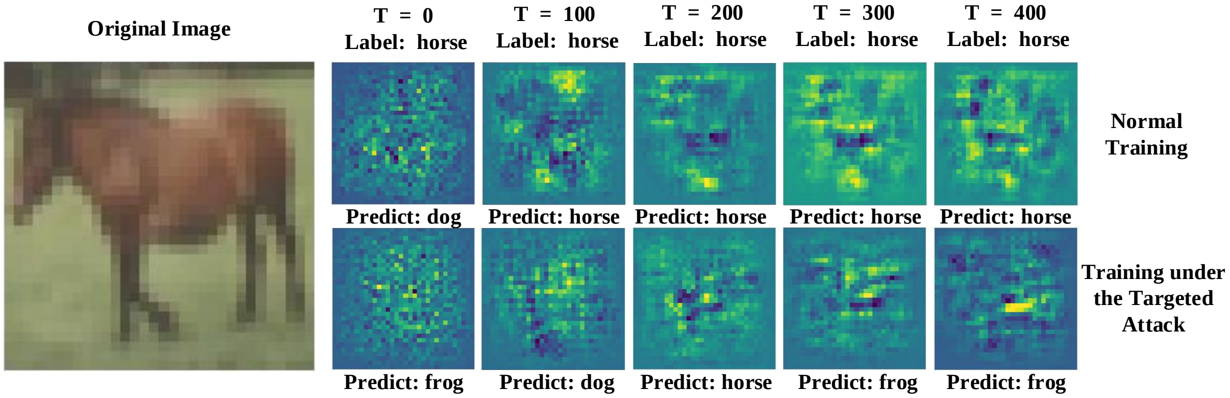


Fig. 5. The visual results of CNN based FL with the interpretability technique corresponding to the image 'horse' under our CMP-NKB against Krum aggregation at different communication rounds. Basically speaking, the subfigures show a "typical" horse image understood by the machine model with normal training and targeted attack.

[2] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 2018.

[3] D. C. Nguyen, P. Cheng, M. Ding, D. Lopez-Perez, P. N. Pathirana, J. Li, A. Seneviratne, Y. Li, and H. V. Poor, "Enabling AI in future wireless networks: A data life cycle perspective," *IEEE Commun. Surveys Tuts.*, Early Access 2020.

[4] C. Ma, J. Li, M. Ding, B. Liu, K. Wei, J. Weng, and H. V. Poor, "RDP-GAN: A rényi-differential privacy based generative adversarial network," *arXiv*, 2020. [Online]. Available: http://arxiv.org/abs/2007.02056

[5] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preservation in location-based services: A novel metric and attack model," *IEEE Trans. Mobile Comput.*, Early Access 2020.

[6] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[7] W. Shiqiang, T. Tiffany, S. Theodoros, L. Kin, K., M. Christian, H. Ting, and C. Kevin, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.

[8] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv*, vol. abs/1811.03604, 2018. [Online]. Available: http://arxiv.org/abs/1811.03604

[9] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019.

[10] C. Ma, J. Li, M. Ding, H. H. Yang, F. Shu, T. Q. S. Quek, and H. V. Poor, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, 2020.

[11] Z. Wang *et al.*, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Paris, France, Apr. 2019, pp. 2512–2520.

[12] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 332–349.

[13] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 691–706.

[14] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 108, Online, Aug. 2020, pp. 2938–2948.

[16] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv*, Dec. 2019. [Online]. Available: https://arxiv.org/abs/1912.04977

[17] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proc. USENIX Security Symposium (USENIX Security)*, Boston, MA, USA, Aug. 2020, pp. 1605–1622.

[18] L. Muñoz González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proc. ACM Workshop on Artificial Intelligence and Security (AISec)*, Dallas, Texas, USA, Nov. 2017, pp. 27–38.

[19] Z. Lingchen, W. Qian, Z. Qin, Z. Yan, and C. Yanjiao, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1486–1500, 2020.

[20] L. Zhao, S. Hu, Q. Wang, J. Jiang, S. Chao, X. Luo, and P. Hu, "Shielding collaborative learning: Mitigating poisoning attacks through client-side detection," *IEEE Trans Depend. Sec. Comput.*, Early Access 2020.

[21] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, and S. Yu, "Pdgan: A novel poisoning defense method in federated learning using generative adversarial network," in *Proc. Algorithms and Architectures for Parallel Processing (ICA3PP)*, Melbourne, Australia, Nov. 2019, pp. 595–609.

[22] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Long Beach, California, USA, Dec. 2017, pp. 119–129.

[23] Y. Dong, C. Yudong, R. Kannan, and B. Peter, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. International Conference on Machine Learning (ICML)*, Stockholm, Sweden, Jul. 2018, pp. 5650–5659.

[24] H. B. Mcmahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv*, Feb. 2016. [Online]. Available: https://arxiv.org/abs/1602.05629

[25] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. 2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, Jul. 2018, pp. 19–35.

[26] Kaggle, "House Prices: Advanced Regression Techniques," May 2017. [Online]. Available: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

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

[28] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing Federated Learning through an Adversarial Lens," in *Proc. International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, Jun. 2019, pp. 634–643.

Appendix A
Proof of Theorem 1

In the mean aggregation, we know that

$$\widehat{\boldsymbol{\theta}} = \sum_{i\in\boldsymbol{\mathcal{B}}} p_i\boldsymbol{\theta}_i + \sum_{i'\in\boldsymbol{\mathcal{M}}} p_{i'}\widehat{\boldsymbol{\theta}}_{i'}. \tag{19}$$

Substituting (19) into $F_A(\widehat{\boldsymbol{\theta}})$, we have

$$F_A(\widehat{\boldsymbol{\theta}}) = \left\|\widehat{\boldsymbol{\theta}}^{*} - \sum_{i\in\boldsymbol{\mathcal{B}}} p_i\boldsymbol{\theta}_i + \sum_{i'\in\boldsymbol{\mathcal{M}}} p_{i'}\widehat{\boldsymbol{\theta}}_{i'}\right\|^2. \tag{20}$$

In order to minimize $F_A(\widehat{\boldsymbol{\theta}})$, we can obtain

$$\sum_{i'\in\boldsymbol{\mathcal{M}}} p_{i'}\widehat{\boldsymbol{\theta}}_{i'} = \widehat{\boldsymbol{\theta}}^{*} - \sum_{i\in\boldsymbol{\mathcal{B}}} p_i\boldsymbol{\theta}_i. \tag{21}$$

However, the local models of benign clients are unknown for the attacker under the partial knowledge background. Therefore, The compromised models can be utilized to estimate the unknown models and we have

$$\begin{aligned}\sum_{i\in\boldsymbol{\mathcal{M}}} p_i\widehat{\boldsymbol{\theta}}_i &= \widehat{\boldsymbol{\theta}}^{*} - \frac{\sum_{i\in\boldsymbol{\mathcal{U}}} p_i - \sum_{i\in\boldsymbol{\mathcal{M}}} p_i}{\sum_{i\in\boldsymbol{\mathcal{M}}} p_i}\sum_{i\in\boldsymbol{\mathcal{M}}} p_i\boldsymbol{\theta}_i\\ &= \widehat{\boldsymbol{\theta}}^{*} + \left(\frac{2}{\sum_{i\in\boldsymbol{\mathcal{M}}} p_i} - 1\right)\sum_{i\in\boldsymbol{\mathcal{M}}} p_i\boldsymbol{\theta}_i.\end{aligned} \tag{22}$$

Note that, we can set

$$\widehat{\boldsymbol{\theta}}_i = \frac{1}{\sum_{i\in\boldsymbol{\mathcal{M}}} p_i}\left(\widehat{\boldsymbol{\theta}}^{*} + \left(\frac{2}{\sum_{i\in\boldsymbol{\mathcal{M}}} p_i} - 1\right)\sum_{i\in\boldsymbol{\mathcal{M}}} p_i\boldsymbol{\theta}_i\right), \tag{23}$$

which is a solution of (22). Hence, we have

$$F_A\left(\widehat{\boldsymbol{\theta}}\right) = \left(\frac{2}{\sum_{i\in\boldsymbol{\mathcal{M}}} p_i} - 1\right)^2\left\|\sum_{i\in\boldsymbol{\mathcal{M}}} p_i\boldsymbol{\theta}_i\right\|^2. \tag{24}$$

This completes the proof. □

Appendix B
Proof of Theorem 2

Considering (13), we can further have

$$\min_{\alpha}\sum_{j\in\boldsymbol{\mathcal{B}}}\alpha_j\|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\| \le \min_{\alpha}\sum_{j\in\boldsymbol{\mathcal{B}}}\frac{\alpha_j^2 + \|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\|^2}{2}, \tag{25}$$

and

$$\begin{aligned}&\min_{\alpha}\sum_{j\in\boldsymbol{\mathcal{B}}}\frac{\alpha_j^2 + \|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\|^2}{2}\\ &= \frac{U-2M-2}{2} + \sum_{j\in\boldsymbol{\mathcal{B}}}\frac{\|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\|^2}{2}.\end{aligned} \tag{26}$$

Then, we can adjust the optimization (13) with a stronger constraint as

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_1^{*} = \;& \arg\min_{\widehat{\boldsymbol{\theta}}_1\subseteq\boldsymbol{\Theta}} F_A(\widehat{\boldsymbol{\theta}}_1),\\ \text{s.t.}\;& \sum_{j\in\boldsymbol{\mathcal{B}}}\frac{\|\boldsymbol{\theta}_j - \widehat{\boldsymbol{\theta}}_1\|^2}{2} + \frac{U-2M-2}{2} + (M-1)\cdot\varepsilon\\ & - E \le 0.\end{aligned} \tag{27}$$

With an auxiliary model $\widehat{\boldsymbol{\theta}}^{*}$, we can obtain an initial solution of (13) using $F_A(\widehat{\boldsymbol{\theta}}^{*}, \widehat{\boldsymbol{\theta}}_1) = \|\widehat{\boldsymbol{\theta}}_1 - \widehat{\boldsymbol{\theta}}^{*}\|^2$. Fortunately, the Lagrangian method offers an effective solution for (13). Based on the KKT conditions, we can obtain the following solution for $\widehat{\boldsymbol{\theta}}_1^{*}$:

$$\begin{aligned}&\left\|\widehat{\boldsymbol{\theta}}_1 - \frac{1}{B}\sum_{j\in\boldsymbol{\mathcal{B}}}\boldsymbol{\theta}_j\right\|^2 - \frac{1}{B^2}\left\|\sum_{j\in\boldsymbol{\mathcal{B}}}\boldsymbol{\theta}_j\right\|^2 + \frac{1}{B}\sum_{j\in\boldsymbol{\mathcal{B}}}\|\boldsymbol{\theta}_j\|^2\\ &+\frac{2}{B}\left(\frac{U-2M-2}{2} + (M-1)\cdot\varepsilon - E\right) = 0.\end{aligned} \tag{28}$$

Therefore, we can obtain the initial model as

$$\begin{aligned}&\widehat{\boldsymbol{\theta}}_{\text{init}} = \frac{1}{B}\Xi + \left(\widehat{\boldsymbol{\theta}}^{*} - \frac{1}{B}\Xi\right)\\ &\left(\frac{1}{B^2}\|\Xi\|^2 - \frac{1}{B}\sum_{j\in\boldsymbol{\mathcal{B}}}\|\boldsymbol{\theta}_j\|^2 - \Lambda\right)^{\frac{1}{2}} \Big/ \left\|\widehat{\boldsymbol{\theta}}^{*} - \frac{1}{B}\Xi\right\|,\end{aligned} \tag{29}$$

where $\Lambda = \frac{U-2M-2+2(M-1)\varepsilon-2E}{B}$ and $\Xi = \sum_{j\in\boldsymbol{\mathcal{B}}}\boldsymbol{\theta}_j$. This completes the proof. □