Covert Model Poisoning Against Federated Learning: Algorithm Design and Optimization
Kuang Wei, Member, IEEE, Jun Li, Senior Member, IEEE, Ming Ding, Senior Member, IEEE, Chuan Ma, Member, IEEE, Yo-Seb Jeon, Member, IEEE, and H. Vincent Poor, Life Fellow, IEEE

Abstract—Federated learning (FL), as a type of distributed machine learning, is vulnerable to external attacks during parameter transmissions between learning agents and a model aggregator. In particular, malicious participant clients in FL can purposefully craft their uploaded model parameters to manipulate system outputs, which is known as a model poisoning (MP) attack. In this paper, we propose effective MP algorithms to attack the classical defensive aggregation Krum at the aggregator. The proposed algorithms are designed to evade detection, 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 Krum. Then, we develop CMP algorithms against Krum based on the solutions of this optimization problem. Furthermore, to reduce the optimization complexity, we propose low complexity CMP algorithms having only a slight performance degradation. Our experimental results demonstrate that the proposed CMP algorithms are effective and can substantially outperform existing attack mechanisms, such as Arjun’s attack and the label flipping attack. More specifically, our original CMP can achieve a high rate of the attacker’s accuracy (∼90%). For example, in our experiments using the MNIST dataset, the proposed CMP attacking algorithm against Krum can successfully manipulate the aggregated model to incorrectly classify a given digit as a different one (e.g., 9 as 8). Meanwhile, our CMP algorithm with an approximated constraint can achieve a rate of 87% in terms of the attacker’s accuracy (attacker-desired results), with a 73% complexity reduction compared to the original CMP.

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

I. INTRODUCTION

With the development of Internet of Things (IoT), various end devices, such as sensors and smart phones, can generate very large amounts of data and send them to cloud servers for processing [1], [2], [3], [4], [5]. Big-data-driven artificial intelligence (AI) has been widely applied in many aspects of modern society. As a result, data privacy and confidentiality have become of increasing concern as such data often contain clients’ sensitive information [6], [7], [8], [9]. Federated learning (FL), emerging as a promising distributed machine learning paradigm [10], is capable of pushing model training to end devices without directly exposing their private training data. In FL, end-user clients collaboratively train a common model in an iterative process by sharing models trained individually on local datasets with an aggregator, or server, that redistributes the model back to the clients for further training. Therefore, in recent years, a wide range of privacy-sensitive applications have been developed along with FL, such as Emoji prediction [11], visual object detection for safety [12], etc.

Although FL can help preserve clients’ privacy, it is possibly trained across a fleet of unreliable clients with private and uninspectable datasets, compared with distributed datacenter learning and centralized learning schemes. Therefore, a new attack framework on federated training systems has been explored [13], [14], [15], i.e., model poisoning attacks. Model poisoning takes advantage of the observation that a client participating in FL can directly influence parameters of the joint model. 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]. For example, with model poisoning, a competitor can degrade the performance of an FL model or achieve its own goals by a particular Trojan trigger [16], [17]. In addition, a client compromised by an attacker (or malicious client) can also incorporate the evasion of potential defenses into its loss function during training.

Because of these vulnerabilities, defense mechanisms for FL have drawn considerable attention. Detection methods based

Manuscript received 21 September 2021; revised 14 March 2023; accepted 26 April 2023. Date of publication 8 May 2023; date of current version 16 May 2024.

This work was supported in part by the National Natural Science Foundation of China under Grants 61872184 and 62002170, in part by the Fundamental Research Funds for the Central Universities under Grant 30921013104, in part by the Future Network Grant of Provincial Education Board in Jiangsu, in part by the Research Initiation Project of Zhejiang Lab under Grant 2022P0D0AC02, in part by the Youth Foundation Project of Zhejiang Lab under Grant K2023PD0AA01, in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant 2022R1C1C101010074, and in part by the U.S National Science Foundation under Grant CNS-2128448. (Corresponding author: Jun Li)

Kuang Wei and Jun Li are with the School of Electrical and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: kang.wei@njjust.edu.cn; jun.li@njjust.edu.cn).

Chuan Ma is with the Zhejiang Lab, Hangzhou 310027, China, and also with the Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing, Jiangsu 211189, China (e-mail: chuan.ma@njjust.edu.cn).

Ming Ding is with the Data61, CSIRO, Sydney, NSW 2015, Australia (e-mail: ming.ding@data61.csiro.au).

Yo-Seb Jeon is with the Department of Electrical Engineering, POSTECH, Pohang, Gyeongbuk 37673, South Korea, and also with the Institute for Convergence Research and Education in Advanced Technology, Yonsei University, Incheon 21983, South Korea (e-mail: yoseb.jeon@postech.ac.kr).

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TDSC.2023.3274119, provided by the authors.

Digital Object Identifier 10.1109/TDSC.2023.3274119

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
on model validation have been proposed to capture anomalous models uploaded by clients, and reduce the weighting of these models when performing aggregation [18], [19]. However, these detection methods, relying on an auxiliary validation dataset, will increase the risk of privacy leakage and may be impractical for real-time training due to high complexity. As an alternative type of defense mechanisms, robust aggregation rules offer low complexity and no additional privacy concerns [20], [21]. To be specific, in Krum [20], 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 that has the minimum sum. In comparison with the detection methods relying on auxiliary validation datasets, Krum can effectively mitigate the impact of unreasonable models or privacy with low complexity and will not increase the risk of the privacy leakage. In addition, we can note that Krum has already been widely adopted as the baselines in many works [22], [23].

In this paper, we are interested in proposing model poisoning attacks against the above robust aggregation rule (Krum) implemented at the server. The proposed attack approach, termed covert model poisoning (CMP), will stealthily induce the server to adopt compromised models from malicious clients. It is expected that the designed CMP will degrade the original FL model or otherwise achieve the attacker’s purposes (here, and throughout, the attacker refers to the set of clients colluding in an attack).

The main contributions of this paper can be summarized as follows:

- We formulate model poisoning as an optimization problem by minimizing the Euclidean distance between the manipulated model and a designated one (beneficial to the malicious clients), constrained by a defensive aggregation rule. Then, we develop CMP algorithms against the Krum robust aggregation rule 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 search dimension of the optimization problem compared to the summations of Euclidean distances among the clients.
- We conduct extensive experiments on real-word datasets, i.e., MNIST, CIFAR and House Pricing datasets. The experimental results demonstrate that the proposed CMP algorithms are more effective than existing attack mechanisms, such as Arjun’s attack and the label flipping attack. More specifically, our original CMP can achieve a high rate of attacker’s accuracy (≈ 90%). For instance, the aggregated model can be manipulated successfully by the CMP under 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.

The rest of this paper is organized as follows. Section II introduces relevant background. A detailed description of the proposed algorithms is given in Section III. Further analysis and limitations of our proposed algorithms are presented in Section IV. Section V introduces the experimental setup. In Section VI, the experimental results are presented and discussed. The related work is introduced in Section VII. Finally, this paper is concluded in Section VIII.

II. PRELIMINARIES

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

A. Federated Learning

As a kind of decentralized training frameworks [11], FL can preserve clients’ private information by its unique distribution learning mechanism. In details, all participants \(C_i, \forall i \in U, U = \{1, 2, \ldots, U\}\) 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.

Then, each client uploads the model parameters after the local training procedure based on the shared global model and local datasets \(D_i, \forall i \in U\), and then all uploaded models will be averaged by the server as the current global model, which is expressed as

\[
\theta_i = \sum_{i \in U} p_i \theta_i^j,
\]

where \(\theta_i\) is the global model at the \(t\)-th communication round, \(\theta_j^i\) is the uploaded model of \(i\)-th client at the \(t\)-th communication round, \(p_i = |D_i|/|D|, \quad D = \sum_{i \in U} |D_i|\) and \(|\cdot|\) denotes the size of the set. 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

\[
\Theta = \arg \min_{\theta \in \Theta} F(D, \theta),
\]

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

B. Model Poisoning Attacks

In FL, model poisoning attacks are a natural and powerful attack class [10], where malicious clients can directly manipulate updates to the central server. Fig. 1 shows a high level of model poisoning attack compared to the data poisoning attack. We will focus on settings where some number of clients collude with each other and craft the uploaded model parameters to achieve their goal in common. This can result in convergence to suboptimal models, or even lead to divergence. If malicious clients have access to the benign clients (updates and datasets), they may be able to craft their outputs to have similar variances and magnitudes as the correct model updates, making them difficult to detect.

**Attacker’s Goal:** The goal of malicious clients 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 [16]. For instance, the malicious clients 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 [13]).

**Participant Collusion:** An important axis to evaluate in the context of specific federated settings is the capability of participant collusion. Intuitively, the malicious clients may be more effective if they are able to coordinate their poisoned updates than if they acted individually. Collusion may not happen in ‘real time’ (within-update), but rather across model updates (cross-update collusion).

Clearly, defining and describing these capabilities and goals are necessary for the community. In Table I, we provide a detailed characteristics description of malicious clients’ capabilities and goals that can be studied.

### III. PROPOSED COVERT MODEL POISONING

In this section, we will propose CMP algorithms against the Krum aggregation rule with the aim of targeted and untargeted attacks.

#### A. Problem Formulation for CMP

In the model poisoning attack, the malicious clients manipulate their uploaded models to influence the behavior of the learning algorithm according to predefined goals. We assume that there exists $M$ malicious clients, and they will directly alter the outputs (local training models) to bias the learned model towards to the attacker’s objective function $F_A(\cdot)$ (a purpose of all malicious clients in common). This attacker’s objective function is constructed to achieve the attacker’s goal and is not the legitimate clients’ objective function $F(\cdot)$, e.g., mislead a spam filter to pass certain types of spam emails. We also define $B$ as the number of benign clients and we have $M + B = U$. We define $\mathcal{M}$ as the set of these malicious clients, $\mathcal{B}$ as the set of benign clients, and $\mathcal{U}$ as the set of all clients, where $\mathcal{M} \subseteq \mathcal{U}$ and $\mathcal{B} = \mathcal{U} / \mathcal{M}$. In details, in each communication round, each benign client computes a local parameter vector $\theta_i, \forall i \in \mathcal{B}$, but each malicious client provides an unreliable parameter vector $\hat{\theta}_i, \forall i' \in \mathcal{M}$. 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 $\hat{\theta}$ can be expressed as

$$\hat{\theta} = \frac{1}{B} \sum_{i \in \mathcal{B}} p_i \theta_i + \frac{1}{M} \sum_{i' \in \mathcal{M}} p_{i'} \hat{\theta}_{i'},$$  

(3)

where $p_i$ and $p_{i'}$ are the aggregated weights as same as (1). Due to the existence of the unreliable parameter vectors from malicious clients, the performance of the aggregated model $\hat{\theta}$ may be bad.

However, the goal of the malicious clients 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

$$\hat{\theta}_M^* = \arg \min_{\hat{\theta}} F_A(\hat{\theta}),$$  

s.t. $\hat{\theta} = A(\hat{\theta}_{i'}; \theta_i), \forall i' \in \mathcal{M}, i \in \mathcal{B}$,  

(4)

where $A$ represents the aggregation rule and $\hat{\theta}_M^* = \{\hat{\theta}_{i'} | i' \in \mathcal{M}\}$ represents the optimal poisoning models.
B. CMP for Krum Aggregation

Existing robust aggregations, such as Krum [20], usually select one or several trust models of U local models and then perform average process using them. In details, for each local model $\theta_i$, the server with Krum computes the $U - M - 2$ local models that are the closest to $\theta_i$, with respect to Euclidean distance. Moreover, the server computes the sum of the distances between $\theta_i$ and its closest $U - M - 2$ local models. Krum selects the local model with the smallest sum of distances as the global model. When $M < U/2$, Krum has theoretical guarantees for the convergence for certain objective functions.

We will consider two types of practical knowledge backgrounds. In the case of full knowledge background, the malicious clients know the local training datasets and local models [25] of all the clients as well as the aggregation rule [16]. In particular, malicious clients could know the aggregation rule in various scenarios. For instance, the service provider may make the aggregation rule public to increase transparency and trust of the FL system. In addition, malicious clients can obtain access to the local training datasets and local models of benign clients via the Trojan Horse programs [26]. We can notice that when considering the full knowledge background, the attacker knows the local training datasets and uploaded models of benign clients as well as the aggregation. The attacker cannot change these training datasets and uploaded models of benign clients to manipulate system outputs. It can only craft uploaded model parameters of malicious clients to manipulate global models. Thus, it is not trivial to find the optimum solution for crafted models due to the large-scale machine learning model parameters and the complex aggregation rule. In the case of partial knowledge background, besides the aggregation rule, malicious clients only knows their own global model, local training datasets and local models [24], [27]. We have to emphasize that it is meaningful to study the full knowledge scenario because this study represents the worst case similar. Further, we also use a shadow aggregator, designed by some one robust aggregation in the current communication round. Suppose $\hat{\theta}$ is the aggregated model by some one robust aggregation in the current communication round. Our goal is to craft the M local models of malicious clients such that the local models aggregated by the robust aggregation has the optimal solution to minimize $F_A(\hat{\theta})$. Therefore, under the Krum aggregation rule $A_{krum}$, our optimization problem can be expressed as

$$\hat{\theta}_M = \arg\min_{\hat{\theta} \in \theta \in M} F_A(\hat{\theta}),$$

s.t. $\hat{\theta} = A_{krum}(\hat{\theta}_i; \theta_i), \forall i' \in M, i \in B$. (5)

We know that the output of robust aggregations is an integer programming problem due to the selection operation. 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 malicious clients, while the inner optimization corresponds to the aggregation rule on all received models. Since solving this problem is highly complex, previous work [28] 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 (5) in an iterative manner.

Our proposed CMP algorithm against Krum is given as Algorithm 1. The core idea of this algorithm is to make $\hat{\theta}_{1,k+1}$ be selected successfully by Krum in addition to minimizing the objective $F_A(\cdot)$, since the rule of Krum is to select one of local models as the global model at the next communication round. We use the gradient descent method to minimize the objective $F_A(\cdot)$. In order to make the crafted model be selected successfully by Krum, we craft local models of other compromised clients (except for the first client) by adding random and small noise to $\hat{\theta}_{1,k+1}$. In this way, all compromised clients’ models will look similar. Further, we also use a shadow aggregator, designed by available local models and datasets, to check whether the crafted model is selected, and then adjust the learning rate of the gradient descent.

First, we define the known dataset and local models by the malicious clients as $D_{\text{att}}$ and $\theta_{\text{att}}$, respectively. In the case of full knowledge background, we can obtain that $D_{\text{att}} = D_U$ and $\theta_{\text{att}} = \theta_U$, where $\theta_U \triangleq \{\theta_i | i \in U\}$ and $D_U \triangleq \{D_i | i \in U\}$. Correspondingly, in the case of partial knowledge background, we have $D_{\text{att}} = D_M$, $\theta_{\text{att}} = \theta_M$, where $\theta_M \triangleq \{\theta_i | i \in M\}$ and $D_M \triangleq \{D_i | i \in M\}$. With different degrees of the knowledge background, the algorithm optimizes all local models $\theta_M$ of malicious clients 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
Algorithm 1: Original Covert Model Poisoning.

Require: The attack dataset $D_{\text{att}} = D_{\text{t}}$ (full knowledge) or $D_{\text{att}} = D_{\mathcal{M}}$ (partial knowledge), the global model $\theta^*$, the artificial noise variance $\sigma^2$, the noise norm threshold $\varepsilon$ (very small), the initial learning rate $\eta_0$, the decay rate for the learning rate $\lambda$ ($\lambda < 1$) and the threshold $\varsigma$.

1: $t \leftarrow 0$ (communication round counter)
2: while $t < T$ do
3: The malicious clients craft uploaded models as follows:
4: $\theta_{\text{att}}^{t} = \theta_{t}^{i}$ (if full knowledge) and
5: $\theta_{M}^{t} = \theta_{M}^{i}$ (if partial knowledge)
6: for $k \leftarrow 0$ (iteration counter)
7: repeat
8: $\hat{\theta}_{t,k+1}^{i} \leftarrow \Pi_{\Theta} (\hat{\theta}_{t,k}^{i} - \eta_{k} \nabla_{\hat{\theta}_{1}} F_{A}(\hat{\theta}_{t,k}^{i}))$
9: for $i = 2, 3, \ldots, M$ do
10: $n_{i} \leftarrow \mathcal{N}(0, \sigma^2)$
11: $\hat{\theta}_{t,k+1}^{i} \leftarrow \hat{\theta}_{t,k+1}^{i} + n_{i} \cdot \max\{1, \varepsilon/\|n_{i}\|\}$
12: end for
13: $\hat{\theta}_{M,k+1}^{i} = \{\hat{\theta}_{M,k+1}^{i} | \forall i \in \mathcal{M}\}$
14: $\hat{\theta}_{k+1}^{i} \leftarrow \Pi_{\Theta_{\text{krum}}} (\hat{\theta}_{M,k+1}^{i}; \theta_{\text{att}}^{t})$
15: if $\hat{\theta}_{k+1}^{i} \neq \hat{\theta}_{t,k+1}^{i}$ then
16: $\eta_{k+1} \leftarrow \lambda \eta_{k}$
17: $\hat{\theta}_{t,k+1}^{i} \leftarrow \hat{\theta}_{t,k}^{i}$
18: end if
19: $k \leftarrow k + 1$
20: until $\eta_{k} < \varsigma$
21: Output $\hat{\theta}_{M,k}^{i}$ as the crafted models
22: end while

communication round, the update rule can be expressed as

$$\hat{\theta}_{t,k+1}^{i} = \Pi_{\Theta} \left( \hat{\theta}_{t,k}^{i} - \eta_{k} \nabla_{\hat{\theta}_{1}} F_{A}(\hat{\theta}_{t,k}^{i}) \right),$$

where $\hat{\theta}_{t,k}^{i}$ is the poisoned model of the $i$-th client in the $k$-th iteration at the $t$-th communication round, $\Pi_{\Theta}$ represents the projection operator to project $\theta$ onto the feasible domain $\Theta$, to handle bounded feature values. Note that this update step should also enforce $\hat{\theta}_{t,k+1}^{i}$ to lie within the feasible domain $\Theta$, which can be typically achieved through the robust aggregation. Then, in order to achieve the goal of participant collusion, this algorithm will obtain the updated $\hat{\theta}_{t,k+1}^{i}$, $i = 2, 3, \ldots, M$ by adding slight noises (Gaussian noises with zero mean and $\sigma$ standard deviation) to $\hat{\theta}_{t,k+1}^{i}$. Each noise vector should be clipped by the clipping threshold $\varepsilon$. After updating the crafted models, this algorithm will check whether Krum selects $\hat{\theta}_{t,k+1}^{i}$ as the global model. If not, then we decrease the step size $\eta$ with a decay parameter $\lambda$. We repeat this process until Krum selects $\hat{\theta}_{t,k+1}^{i}$ or $\eta$ is smaller than a certain threshold $\varsigma$.

We can see that Algorithm 1 will craft attacking models to the best of known knowledge, and adopt the stochastic gradient descent (SGD) method based on the available data to train attacking models with the robust aggregation rule. Hence, the complexity of the proposed algorithm can be divided into two parts, i.e., SGD and robust aggregation. We further assume that the computational complexity of the robust aggregation is $I$.

C. CMP for Krum Aggregation With Low Complexity

In this subsection, we will propose a low complexity attack scheme for krum aggregation based on the aforementioned optimization problem by addressing the constraint with some beneficial approximations.

The aforementioned algorithm with krum aggregation 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 $\theta_{1}, \ldots, \theta_{M}$ is large. We know that there are $M - 1$ malicious clients assisting $\hat{\theta}_{1}$. Because $M < \frac{U-2}{\tau}$, then $\min M < 1 < \frac{U}{\tau} - 2 < \frac{U}{\tau} - 1 < U - M - 2$. Therefore, it is necessary for the crafted model $\hat{\theta}_{1}$ to be close to $U - 2M - 1$ benign clients’ with respect to euclidean distance. Consider the differences among $\hat{\theta}_{1}, \forall i \in \mathcal{M}$, we set an enough small value $\varepsilon$, where $\|\hat{\theta}_{1} - \theta_j\| \leq \varepsilon$, $\forall i, j \in \mathcal{M}$. The sum of the Euclidean distances of the crafted model $\hat{\theta}_{1}$ can be expressed as

$$\min_{\subseteq \mathcal{B} j \in \mathcal{S}} \sum_{j \in \mathcal{S}} \|\hat{\theta}_{i} - \theta_j\| \approx \min_{\subseteq \mathcal{B} j \in \mathcal{S}} \sum_{j \in \mathcal{S}} \|\hat{\theta}_{i} - \theta_j\|.$$

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
\[
\hat{\theta}_1^* = \arg \min_{\theta_1 \in \Theta} F_A(\hat{\theta}_1),
\]
s.t. \( \min_{\mathcal{S}'} \sum_{i \in \mathcal{S}'} \|\hat{\theta}_1 - \theta_i\| + (M - 1) \cdot \varepsilon \leq 0, \)
\[
\mathcal{S}' \subseteq \mathcal{B}, |\mathcal{S}'| = U - 2M - 1,
\]
where
\[
E = \min_{\mathcal{S} \subseteq \mathcal{B}} \sum_{i \in \mathcal{S}} \|\theta_i - \theta_j\|. \tag{9}
\]
In order to simplify the constraints of \( \mathcal{S}' \), we can transform (8) into a mixed optimization problem as
\[
\hat{\theta}_1^* = \arg \min_{\theta_1 \in \Theta} F_A(\hat{\theta}_1),
\]
s.t. \( \min_{\alpha} \sum_{i \in \mathcal{B}} \alpha_i \|\theta_i - \hat{\theta}_1\|\)
\[
+ (M - 1) \cdot \varepsilon - E \leq 0,
\]
\[
\sum_{i \in \mathcal{B}} \alpha_i = U - 2M - 1, \alpha_i \in \{0, 1\}. \tag{10}
\]
The bilevel optimization problem in (10) is NP hard in general. Specifically, we require the attack space \( \Theta \) to be differentiable (e.g. the malicious clients can change the local models in \( \Theta \) 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 function of the problem \( \mathcal{P}_1 \) and \( \mathcal{P}_2 \). The problem \( \mathcal{P}_1 \) can be expressed as
\[
\hat{\theta}_1^* = \arg \min_{\theta_1 \in \Theta} F_A(\hat{\theta}_1),
\]
s.t. \( \sum_{i \in \mathcal{B}} \alpha_i \|\theta_i - \hat{\theta}_1\| + (M - 1) \cdot \varepsilon - E \leq 0. \tag{11}
\]
The solution of \( \mathcal{P}_1 \) can be expressed by \( \hat{\theta}_1^* = h(\alpha) \). Therefore, the optimization problem \( \mathcal{P}_2 \) can be given by
\[
\hat{\theta}_1^* = \arg \min_{\alpha} F_A(h(\alpha))
\]
s.t. \( \sum_{i \in \mathcal{B}} \alpha_i = U - 2M - 1 \)
\[
\alpha_i(\alpha_i - 1) = 0, \forall i \in \mathcal{B}. \tag{12}
\]
Specifically, the Lagrangian function of the problem \( \mathcal{P}_1 \) can be written as
\[
L(\alpha, \lambda) = F_A(\hat{\theta}_1) + \lambda \cdot \sum_{i \in \mathcal{B}} \alpha_i \|\theta_i - \hat{\theta}_1\|
\]
\[
+ (M - 1) \cdot \varepsilon - E \tag{13}
\]
Based on the Karush-Kuhn-Tucker (KKT) conditions, the model \( \hat{\theta}_1^* \) and the optimal Lagrangian multiplier \( \lambda \) should satisfy the following equation set
\[
\begin{cases}
\frac{\partial F_A(\hat{\theta}_1)}{\partial \hat{\theta}_1} + \lambda \cdot \sum_{i \in \mathcal{B}} \alpha_i(\theta_i - \hat{\theta}_1) = 0, \lambda > 0 \\
\sum_{i \in \mathcal{B}} \alpha_i \|\theta_i - \hat{\theta}_1\| + (M - 1) \cdot \varepsilon - E = 0.
\end{cases} \tag{14}
\]

Further analysis and limitations
where $\Theta$ is the feasible domain of the FL training models. The attacker’s objective function $F_A(\hat{\Theta})$ has been typically computed on a specific target model $\hat{\Theta}$. In this example, we may define $F_A(\hat{\Theta}) = ||\hat{\Theta} - \hat{\theta}||^2$ with an appropriate norm. If the malicious clients have the full knowledge of this system, we know that the solution of this optimization is available by solving the optimization problem in (16) directly.

However, if the malicious clients have the partial knowledge as described in Section II-B, the optimal solution can be obtained in the following theorem.

**Theorem 2:** With a certain target $F_A(\hat{\Theta})$ and $M$ malicious clients under the mean aggregation rule in FL, the crafted model can be calculated by

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

(17)

and the loss function value can be expressed as

$$F_A(\hat{\Theta}) = \left( \frac{2}{\sum_{i \in \mathcal{M}} p_i} - 1 \right) \frac{\sum_{i \in \mathcal{M}} p_i \hat{\theta}_i}{\sum_{i \in \mathcal{M}} p_i \hat{\theta}_i}^2 .$$

(18)

**Proof:** See Appendix B, available online.

From Theorem 2, we can obtain a solution for the malicious clients by estimating the local models of benign clients. This estimation method has also been used in the proposed attacking algorithms against Krum aggregation. We can note that the attacking performance is determined by the non-independent identically distributed (non-i.i.d.) degree of data, which can be shown in the experimental results.

**B. CMP for Advanced Robust Aggregations**

As Krum [20], more and more robust aggregation rules have been proposed, such as trimmed mean [21], AFA [29], DnC [30] and so on, which also select one or several trust models and then perform average process using them. The standard of how to select the trust model is becoming more complicated, and no limited to relying on the European distance between these local models. However, if we can explore the judgement standard of the trust model, malicious clients can finish the participant collusion by this standard. In the future, we also plan to focus on enhancing our proposed CMP to explore the vulnerabilities of these advanced aggregation rules.

**V. EXPERIMENTAL SETUP**

In this section, we implemented our attacks using Pytorh. 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.

**A. Datasets**

Our experiments use four real datasets:

- The 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 [31]. In total, it includes 1,460 houses and 81 features. We preprocess by onehot encoding all categorical features and normalize numerical features, resulting in 275 total features;
- MNIST is a dataset of handwritten digits consists of 60,000 training examples and 10,000 testing examples [32] formatted as 28 × 28 size gray scale images;
- CIFAR-10 consists of 60,000 color images in 10 object classes such as deer, airplane, and dog, included per class [33]. The complete dataset is pre-divided into 50,000 training images and 10,000 test images;
- Flowers consists of 5241 color images in 5 object classes i.e., daisy, dandelion, rose, sunflower and tulip, included per class. The complete dataset is pre-divided into 4,317 training images and 924 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. To generate distributed datasets, we use two schemes: (i) a balanced scheme, where the samples are uniformly distributed across $U$ clients with the same number, and (ii) non-i.i.d-ness, where the samples are non-i.i.d.-ness with a degree factor $p$ and distributed across $U$ nodes.

**B. Machine Learning Models**

Our experiments evaluate four supervised learning models including linear regression (LR), support vector machine (SVM), multi-layer perceptron (MLP) and convolutional neural network (CNN).

- LR is performed with stochastic gradient descent (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;
- SVM is trained on the MNIST 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;
- MLP is conducted on the standard MNIST dataset. This is a simple feedforward neural network, in which there are one hidden layer with ReLU units and softmax of 10 classes (corresponding to the 10 digits) with the cross-entropy loss.
- CNN has two convolutional layers and two linear layers with dropout is applied for the CIFAR-10 and Flowers datasets. We use softmax of 10 and 5 classes with the cross-entropy loss for the CIFAR-10 and Flowers datasets, respectively.

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.
in our experiments, respectively. Correspondingly, and for Krum to guarantee the showing that our attacks are effective and (In the first scenario, we evaluate the performance of the aggregation rule, in order to showcase the effectiveness of the proposed attack algorithms. Inspired by the above optimization for the targeted attack, we can craft local models of malicious clients achieve untargeted attacks via solving the similar optimization, which only differs from 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 method.

VI. 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-Orgconstr and CMP-PKB-Orgconstr in our experiments, respectively. Correspondingly, the CMP attack with low complexity under full and partial knowledge backgrounds are named as CMP-FKB-Simconstr and CMP-PKB-Simconstr, respectively.

In Section VI-A, we consider the untargeted attack, where the malicious clients aim to destroy the FL model. In Section VI-B, we flip the labels of the MNIST or CIFAR-10 dataset.

A. Untargeted Attack

1) House Pricing Prediction: In the first scenario, we evaluate the performance of the house pricing prediction task by LR with the normalized cost against krum. Figs. 3 and 4 show the loss function value and attack success rate using our untargeted attack methods under various percentages of malicious clients compared with existing works, respectively. The results in Fig. 3 show that our attacks are effective and substantially outperform existing attacks. If malicious clients have 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 malicious clients is larger, our proposed attack will have a deeper effect, which is in line with the intuition. In Fig. 4, we show the attack success rate of our proposed

| Parameter | Description | Value |
|-----------|-------------|-------|
| $U$       | Number of clients | 50    |
| $M$       | Number of malicious clients | [0, 10] |
| $p$       | Degree of Non-IID | [0, 1] |
| $\epsilon$ | Distance parameter for Krum attacks | 0.01 |

C. Benchmarks

Furthermore, we compare existing attacks with our proposed methods, which are detailedly described as follows. 1) Gaussian attack. Specifically, for each malicious client, we sample a noise vector from the Gaussian distribution and add it on the parameter of the local model on the malicious client [34]. We use this Gaussian attack to show that crafting local models of malicious clients randomly can not effectively attack the Byzantine-robust aggregation rules; 2) Label flipping attack. This is a data poisoning attack [35] that does not require knowledge of the training data distribution. On each malicious client, this attack flips the label of each training instance; 3) Fang’s Full knowledge attack or partial knowledge attack [16]. These attacks manipulate the local model parameters on malicious clients during the learning process against the Byzantine robust FL. 4) Arjun’ attack [24]. 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. However, this attack is only effectively for the IID setting.

D. Performance Metrics

In this subsection, we introduce performance metrics which have been used in this paper.

1) Test loss: 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.

2) Error rate: For the MNIST and untargeted attack, we will use the error rate of the FL model to evaluate our CMP, i.e., $1 - \mathcal{V}$, where $\mathcal{V}$ is the test accuracy with the correct labels and different from the attacker’s accuracy in the below. Specifically, if our CMP can achieve a high error rate, it can attack the FL system effectively.

3) Attacker’s accuracy. We also apply the MNIST and CIFAR-10 in the case of targeted attack. In this scenario, we will use attacker’s accuracy (calculate the accuracy between predicting results using testing data and the attacker’s labels) to evaluate our CMP.

4) Successful attacking rate. If the Krum is adopted in the FL system, we denote successful attacking rate as the rate that local models of malicious clients are selected by the Krum in each communication.

E. Parameter Setting

We describe parameter settings for the FL and our attack methods. Table II summarizes the setting for key parameters. We record each experiment for 10 trials and report the average results. The number of malicious clients is set from 0 to 12 and the degree of Non-IID is set in the range of [0,1]. For the targeted attack, we flip the labels of the MNIST or CIFAR-10 dataset, which are shown in Table III. Although the server in FL systems will not know the number of malicious clients in the real world. In the experiments, we adopt the parameter $(U - M - 2)$ and $(U - M - 1)$ (a robust case) for Krum to guarantee the performance of the aggregation rule, in order to showcase the effectiveness of the proposed attack algorithms.

Inspired by the above optimization for the targeted attack, we can craft local models of malicious clients achieve untargeted attacks via solving the similar optimization, which only differs from 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.
attacks against the Krum rule compared with existing works, i.e., our proposed attacks have higher probabilities to make local models of malicious clients be selected by the Krum in each communication. Our proposed 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. 4, we can note that the attack success rate under the partial knowledge background increases with the percentage of malicious clients.

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 Table IV, 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.-ness 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.-ness. Finally, we can note that when the degree of non-i.i.d.-ness is small, the malicious clients under the partial knowledge background have a large value of the attack success rate. The intuition is that if the degree of non-i.i.d.-ness is small, the divergence of different local models will be small and the estimation of the unknown models by the malicious models will be accurate.

B. Targeted Attack

1) Handwriting Digits and Flowers 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, DnC and Trimmed mean aggregation rules compared with existing works, respectively.
In Table V, we show the attacker’s accuracy of our proposed CMP-FKB-Orgcontr against mean aggregation compared with existing works under various degrees of non-i.i.d.-ness with $U = 50$, $M = 10$ and $T = 30$. Note that the proposed CMP algorithms are more effective than existing attacks and achieve a high attacker’ accuracy (above 85%).

In Table VI, 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 attacker’s accuracy from 0.052 to 0.904 as well as our partial knowledge attack increases it to 0.546. Meanwhile, using the approximate constraint, our proposed algorithm will have a small time cost but a low attacker’s accuracy. In Table VI, we can note that a simplified solution with the partial knowledge background (CMP-PKB-Simcontr) may achieve a higher successful attacking rate than the original solution (CMP-PKB-Orgcontr). The reason is that, with the partial knowledge background, the attacker needs to craft its uploaded model parameters based on partial training datasets. Hence, the attacker cannot guarantee a successful attacking rate due to the partial training datasets, while using the original solution. We also evaluate our algorithm on the Flowers dataset, as shown in Table VII. It can be noted that the proposed CMP algorithm can achieve a better attacking performance than the baselines, especially for the CMP-FKB-Orgcontr algorithm.

In Table VII, we show the comparison of attacker’s accuracy between our proposed targeted attack algorithms and existing algorithms on CNN model and the Flowers dataset against Krum (using $(U - M - 1)$ as the Krum parameter, with the various dataset distributions). In Table VIII, we show the attacker’s accuracy of our proposed algorithms under various numbers of communication rounds with $p = 0.5$, $U = 50$ and $M = 10$. These results show that the proposed CMP algorithms are more effective than existing attacks and achieve a high attacker’s accuracy (above 85%). However, when the degree of non-i.i.d.-ness is set to 0, the proposed CMP-FKB-Orgcontr can obtain a high attacker’s accuracy (above 90%). However, when the degree of non-i.i.d.-ness is set to 0, the performance of the proposed algorithms will decrease noticeably.

2) Attacking on Other Aggregation Rules: Furthermore, we also conduct additional experiments on other aggregation rules, i.e., DnC and Trimmed Mean, to show the effectiveness of the proposed attack algorithms, as shown in Table IX. For DnC and Trimmed Mean rules, we split the known dataset into two parts to train the crafted models and validate the attacker’s accuracy. Different from attacking on Krum in Algorithm 1, we decide whether to adjust the learning rate based on the increment of the validation accuracy, instead of whether the
malicious client has been selected. From the experimental results in Tab. 9, we can see that the proposed CMP-FKB-Orgcontr and CMP-PKB-Orgcontr can achieve high attacker’s accuracies for the MNIST dataset, i.e., above 60% and 50%, respectively. In addition, for the CIFAR-10 and Flowers datasets, the proposed algorithms outperform baselines for both partial and full knowledge background scenarios. Specifically, for CIFAR-10, the proposed CMP-PKB-Orgcontr increases the attacker’s accuracy from about 20% to above 30% compared to baselines, and the proposed CMP-FKB-Orgcontr increases it to above 35%. For the Flowers dataset, the proposed CMP-PKB-Orgcontr and CMP-FKB-Orgcontr can also improve the attacker’s accuracy greatly compared to baselines (more than 4% improvement in accuracy).

3) Data Visualization of the MP Attack: 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-PKB-Orgcontr (one of our proposed CMP algorithms). 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. 5 compares the outputs of MLP based FL with the interpretability technique corresponding to the digit 1 for malicious models at different communication rounds, respectively. We also show the results by CNN based FL with CIFAR-10 dataset in Fig. 6.

In Fig. 5, we can note that the visual results of MLP based FL model corresponding to the digit 1 using our CMP-PKB-Orgcontr algorithm against Krum aggregation are similar with the digit 2. Especially, with the increasing numbers of communication rounds, the visual results trend to be closer to the digit 2 instead of 1, which means that our CMP-PKB-Orgcontr 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 Fig. 6. We can also note that the proposed algorithm can successfully obtain the attacker’s goal and make the FL model beneficial to the malicious clients, i.e. mislead the model to classify the horse as the frog.

VII. RELATED WORKS

In this section, we investigate various adversarial attacks and defensive mechanisms in FL, which are active areas of research.

A. Adversarial Attacks in Federated Learning

The security of machine learning has received considerable attention in different communities [10], [35], [36], [37], [38], [39]. Although the data is not explicitly exposed in the original format in distributed learning frameworks, e.g., FL [14], [15], different types of attacks against distributed machine learning algorithms have been designed and analyzed including poisoning attacks (e.g., [16], [19], [24]), and privacy attacks (e.g., [40], [41], [42], [43]). For example, poisoning attackers can control a subset of clients and manipulate their outputs sent to the server [16]. In this way, attacker can dominate the cluster and change the judgment boundary of the global model, or make the global model deviate from the right direction. The work in [44] presented a new attack paradigm, in which a malicious opponent may interfere with or backdoor the process of distributed learning by applying limited...
changes to the reported parameters. The work in [13] proposed a new model-replacement method that demonstrated its efficacy on poisoning models of standard FL tasks. In addition, Bagdasaryan et al. in [13] also developed and evaluated a generic constrain-and-scale technique that incorporates the evasion of defenses into the attacker’s loss function during training. The work in [24] explored the threat of model poisoning attacks on FL initiated by a single, non-colluding malicious client where the adversarial objective is to cause the model to misclassify a set of chosen inputs with high confidence. In [16], Fang et al. formulate the local model poisoning attack as optimization problems, and then apply this attack to four recent Byzantine-robust FL methods. Although existing works have developed some novel model poisoning methods to destroy the FL training model, in this work we have considered the coordinated poisoned updates of all malicious clients and achieve the attacker’s purposes instead of destroying that will be more effective than they acted individually.

B. Defense Against Poisoning Attacks

With the development of different types of attacks, how to design defensive algorithms against attacks from malicious attackers in distributed learning frameworks has become crucial. For detecting poisoned updates in the collaborative learning [22], the results of client-side cross-validation were applied for adjusting the weights of the updates when performing aggregation, where each update is evaluated over other clients’ local data. The work in [18] considered the existence of unreliable participants and used the auxiliary validation data to compute a utility score for each participant to reduce the impact of these participants. The work in [29] proposed a robust aggregation rule, called adaptive federated averaging, that detects and discards bad or malicious local model updates based on a hidden Markov model. To tackle adversarial attacks in FL aggregation process, the work in [34] presented a novel aggregation algorithm with the residual-based reweighting method, in which the weights for average of all local models are estimated robustly. The work in [19] proposed a novel poisoning defense method in FL, in which a participant whose accuracy is lower than a predefined threshold will be identified as an attacker and corresponding model parameters will be removed from the training procedure in this iteration.

VIII. CONCLUSION

In this paper, we have studied model poisoning attacks on FL models against the existing defensive aggregation rule, i.e., Krum. We have formulated the model poisoning as an optimization problem by minimizing the Euclidean distance between the manipulated model and a 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. Finally, we have conducted extensive experiments on real-world datasets, i.e., MNIST, CIFAR and House Pricing datasets. The experimental results have demonstrated that the proposed CMP algorithms are more effective than the existing attack methods, such as Arjun’s attack and the label flipping attack. An interesting problem for future work is to investigate defense algorithms. In the future, we plan to investigate the defensive algorithms based on the model interpretability instead of the statistical properties of various models.

REFERENCES

[1] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” IEEE Internet Things J., vol. 3, no. 6, pp. 854–864, Dec. 2016.
[2] W. Yu et al., “A survey on the edge computing for the Internet of Things,” IEEE Access, vol. 6, pp. 6900–6919, 2017.
[3] S. Yan, S. V. Hanly, and I. B. Collings, “Optimal transmit power and flying location for UAV covert wireless communications,” IEEE J. Sel. Areas Commun., vol. 39, no. 11, pp. 3321–3333, Nov. 2021.
[4] S. Yan, Y. Cong, S. V. Hanly, and X. Zhou, “Gaussian signalling for covert communications,” IEEE Trans. Wireless Commun., vol. 18, no. 7, pp. 3542–3553, Jul. 2019.
[5] S. Yan, B. He, X. Zhou, Y. Cong, and A. L. Swindlehurst, “Delay-tolerant covert communications with either fixed or random transmit power,” IEEE Trans. Inf. Forensics Secur., vol. 14, no. 1, pp. 129–140, Jan. 2019.
[6] M. Huai, J. Sun, R. Cai, L. Yao, and A. Zhang, “Malicious attacks against deep reinforcement learning interpretations,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, CA, USA, 2020, pp. 472–482.
[7] A. Li, Y. Duan, H. Yang, Y. Chen, and J. Yang, “TIPRDC: Task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, CA, USA, 2020, pp. 824–832.
[8] K. Wei et al., “Low-latency federated learning over wireless channels with differential privacy,” IEEE J. Sel. Areas Commun., vol. 40, no. 1, pp. 290–307, Jan. 2022.
[9] J. Li et al., “Blockchain assisted decentralized federated learning (BLADE-FL): Performance analysis and resource allocation,” IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 10, pp. 2401–2415, Oct. 2022.
Kang Wei (Student Member, IEEE) received the BS degree in information engineering from Xidian University, Xian, China, in 2014, and the PhD degree from the Nanjing University of Science and Technology, supervised by prof. Jun Li. He is currently a postdoctoral fellow with The Hong Kong Polytechnic University. He has won the 2022 IEEE Signal Processing Society Best Paper Award and 2022 Wiley China Open Science Author of the Year. He has served as reviewer for several prestigious journals and conferences. He mainly focuses on privacy protection and optimization techniques for edge intelligence, including federated learning, differential privacy and network resource allocation.

Jun Li (Senior Member, IEEE) received the PhD degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, P. R. China, in 2009. From January 2009 to June 2009, he worked in the Department of Research and Innovation, Alcatel Lucent Shanghai Bell as a research scientist. From June 2009 to April 2012, he was a postdoctoral fellow with the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. From April 2012 to June 2015, he was a research fellow with the School of Electrical Engineering and Telecommunications, the University of Sydney, Australia. Since 2015, he is a professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a visiting professor with Princeton University from 2018 to 2019. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultra-dense wireless networks, mobile edge computing, network privacy and security, and industrial Internet of things. He has co-authored more than 200 papers in IEEE journals and conferences, and holds 1 US patent and more than 10 Chinese patents in these areas. He is serving as an editor of IEEE Transactions on Wireless Communication and TPC member for several flagship IEEE conferences.
Ming Ding (Senior Member, IEEE) received the BS and MS degrees (with first-class Hons.) in electronics engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the Doctor of Philosophy (PhD) degree in signal and information processing from SJTU, in 2004, 2007, and 2011, respectively. From April 2007 to September 2014, he worked with Sharp Laboratories of China in Shanghai, China as a researcher/senior researcher/principal researcher. Currently, he is a principal research scientist with Data61, CSIRO, in Sydney, NSW, Australia. His research interests include information technology, data privacy and security, and machine learning and AI. He has authored more than 200 papers in IEEE journals and conferences, all in recognized venues, and around 20 3GPP standardization contributions, as well as two books, i.e., “Multi-point Cooperative Communication Systems: Theory and Applications” (Springer, 2013) and “Fundamentals of Ultra-Dense Wireless Networks” (Cambridge University Press, 2022). Also, he holds 21 US patents and has co-invented another 100+ patents on 4G/5G technologies. Currently, he is an editor of IEEE Transactions on Wireless Communications and IEEE Communications Surveys and Tutorials. Besides, he has served as a guest editor/co-chair/co-tutor/TPC member for multiple IEEE top-tier journals/conferences and received several awards for his research work and professional services, including the prestigious IEEE Signal Processing Society Best Paper Award in 2022.

Chuan Ma (Member, IEEE) received the BS degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013, and PhD degree from the University of Sydney, Australia, in 2018. From 2018 to 2022, he worked as a lecturer with the Nanjing University of Science and Technology, and now he is a Principal investigator with Zhejiang Lab. He has published more than 40 journal and conference papers, including a best paper in WCNC 2018, and a best paper award in IEEE Signal Processing Society 2022. His research interests include stochastic geometry, wireless caching networks and distributed machine learning, and now focuses on the Big Data analysis and privacy-preserving.

Yo-Seb Jeon (Member, IEEE) received the BS (Top Hons.) and the PhD degrees from the Department of Electrical Engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2012 and 2016, respectively. From September 2016 to August 2018, he was a postdoctoral research associate with POSTECH. From September 2018 to January 2020, he was a postdoctoral research fellow with the Department of Electrical Engineering at Princeton University. Since 2020, he has been on the faculty at POSTECH, where he is currently an Assistant Professor with the Department of Electrical Engineering. His research interests include the areas of wireless communications, machine learning, and signal processing. He was a recipient of the TJ PARK Graduate Fellowship from POSTECH (2012–2014). He was recognized as an Exemplary Reviewer of IEEE Transactions on Communications, in 2021.

H. Vincent Poor (Life Fellow, IEEE) received the PhD degree in EECS from Princeton University, in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University professor. During 2006 to 2016, he served as the dean of Princeton’s School of Engineering and Applied Science. He has also held visiting appointments with several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the recent book Machine Learning and Wireless Communications. (Cambridge University Press, 2022). He is a member of the National Academy of Engineering and the National Academy of Sciences and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.