Privacy-Preserving Federated Learning With Malicious Clients and Honest-but-Curious Servers

Junqing Le, *Member, IEEE*, Di Zhang, Xinyu Lei, *Member, IEEE*, Long Jiao,
Kai Zeng, *Member, IEEE*, and Xiaofeng Liao, *Fellow, IEEE*

***Abstract*— Federated learning (FL) enables multiple clients to jointly train a global learning model while keeping their training data locally, thereby protecting clients' privacy. However, there still exist some security issues in FL, e.g., the honest-but-curious servers may mine privacy from clients' model updates, and the malicious clients may launch poisoning attacks to disturb or break global model training. Moreover, most previous works focus on the security issues of FL in the presence of only honest-but-curious servers or only malicious clients. In this paper, we consider a stronger and more practical threat model in FL, where the honest-but-curious servers and malicious clients coexist, named as the non-fully trusted model. In the non-fully trusted FL, privacy protection schemes for honest-but-curious servers are executed to ensure that all model updates are indistinguishable, which makes malicious model updates difficult to detect. Toward this end, we present an Adaptive Privacy-Preserving FL (Ada-PPFL) scheme with Differential Privacy (DP) as the underlying technology, to simultaneously protect clients' privacy and eliminate the adverse effects of malicious clients on model training. Specifically, we propose an adaptive DP strategy to achieve strong client-level privacy protection while minimizing the impact on the prediction accuracy of the global model. In addition, we introduce DPAD, an algorithm specifically designed to precisely detect malicious model updates, even in cases where the updates are protected by DP measures. Finally, the theoretical analysis and experimental results further illustrate that the proposed Ada-PPFL enables client-level privacy protection with 35% DP-noise savings, and maintains similar prediction accuracy to models without malicious attacks.**

***Index Terms*— Poisoning attacks, privacy protection, federated learning, differential privacy, anomaly detection.**

Manuscript received 5 March 2022; revised 22 March 2023 and 4 June 2023; accepted 10 July 2023. Date of publication 17 July 2023; date of current version 25 July 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61932006 and Grant 62202071; in part by the National Key Research and Development Program of China under Grant 2018AAA0100101; in part by the China Postdoctoral Science Foundation under Grant 2022M710518 and Grant 2022M710520; in part by the Natural Science Foundation of Chongqing, China, under Grant CSTB2022NSCQ-MSX0358 and Grant CSTB2022NSCQ-MSX1217; and in part by the Natural Science Foundation under Grant CNS-2153393. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Thorsten Strufe. *(Corresponding author: Xiaofeng Liao.)*

Junqing Le, Di Zhang, and Xiaofeng Liao are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, and the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: junqingle@cqu.edu.cn; dizhang@cqu.edu.cn; xfliao@cqu.edu.cn).

Xinyu Lei is with the Department of Computer Science, Michigan Technological University, Houghton, MI 49931 USA (e-mail: xinyulei@mtu.edu).

Long Jiao and Kai Zeng are with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA (e-mail: ljiao@gmu.edu; kzeng2@gmu.edu).

Digital Object Identifier 10.1109/TIFS.2023.3295949

I. Introduction

DEEP learning has demonstrated good performance in many fields, such as image classification [1], [2], pattern recognition [3] and language processing [4]. To train a neural network, traditionally centralized learning requires collecting a large amount of data from clients. However, this approach may lead to serious leakage of clients' privacy, because client's sensitive data is directly accessible to the collector. To solve this issue, federated learning [5], [6] (FL) has been proposed. In FL, multiple clients can collaborate to train a global learning model while keeping their training data locally. It only requires clients to upload their model updates (i.e., gradients or weights) to a central server, and the original data of clients is completely invisible from others, thereby protecting privacy.

Unfortunately, there still exist many security issues in FL. As the uploaded model updates may "memorize" the information about the training data [7], the attackers (i.e., honest-but-curious servers) can compromise clients' privacy through these model updates [8]. In addition, the malicious clients can launch untargeted poisoning attacks (i.e., sign-flipping attack and additive noise attack) [9], [10] and targeted poisoning attacks [11], [12] to prevent model convergence, corrupt the global model and cause misclassification.

In recent years, numerous researchers have studied the aforementioned security issues of FL and proposed various defense solutions such as the Secure Multiparty Computation (SMC)-based schemes [13], the Homomorphic Encryption (HE)-based schemes [14], the Differential Privacy (DP)-based schemes [15], [16], [17], the byzantine-resilient aggregation rules [18], [19], [20], the anomaly detection based schemes [21], and the hybrid schemes [22], [23].

Nevertheless, most previous works focus on the security issues of FL in the presence of only honest-but-curious servers or only malicious clients. The investigation of a stronger and more practical threat model in FL, where both honest-but-curious servers and malicious clients coexist, has received limited attention in comparison. In this threat model, it is assumed that any parties (i.e., servers and clients) may not be fully trusted. Specifically, the servers may be honest-but-curious, which honestly follows the aggregation protocol, but may launch active or passive attacks to get client's privacy from the model updates. The local clients may be malicious, which may not follow the protocol and try to break the model training by sending incorrect values to servers (such as poisoning attacks). The previous FL schemes only focused

1556-6021 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

on honest-but-curious servers or malicious clients can be considered as two special cases in the non-fully trusted FL.

In the non-fully trusted FL setting, the conflict arises because the privacy-preserving scheme adopted to defend against honest-but-curious server aims to ensure the indistinguishability of model updates, but the defenses against malicious clients' attacks by removing outliers are based on the differences between model updates. To clarify, high-performance mechanisms for detecting malicious clients often encounter the issue of inadequate privacy protection for client data. On the contrary, solutions designed to effectively defend against privacy breaches by honest-but-curious server pose challenges in terms of detecting malicious clients.

When it comes to resolving the conflict between detecting malicious clients and protecting client data privacy, the previous solutions have exhibited the following drawbacks. 1) SMC-based schemes adopt a secure aggregation protocol to mask the client-provided model updates before uploading to the server. However, the masked updates can completely discard the characteristics of the original model updates, making it difficult to identify and eliminate the malicious model updates (i.e., the maliciously manipulated model updates from malicious clients). 2) HE-based schemes allow the server to aggregate the clients' model updates over the encrypted domain without decryption. However, the detection for malicious model updates is also difficult in the encrypted domain. To address this issue, an anomaly detection based on the ciphertext domain has been proposed in [22], but the benign clients' information may be easily visible to malicious clients because of the use of the same decryption key. 3) In DP-based schemes, noise introduced into model updates accumulates during training, which dramatically degrades the accuracy of model training and increases the difficulty of detecting malicious model updates. 4) The schemes based on byzantine-resilient aggregation rules can partially mitigate the impacts caused by malicious clients, but they cannot completely eliminate them. 5) An anomaly detection method [21] for robust FL is proposed to eliminate these impact from the malicious clients. However, this approach requires prior training based on a public and malicious client-free dataset to obtain a detection model, and this detection model is not efficient to detect the DP-protected model updates. 6) The scheme in [23] uses a combination of SMC and byzantine-resilient aggregation rules to deal with the privacy leakage and poisoning attacks are also presented in the non-fully trusted FL. However, significant computation and communication overheads are required to achieve desirable results in practical applications.

To protect clients' privacy and obtain a global model with high performance in the non-fully trusted FL, an Adaptive Privacy-Preserving FL (Ada-PPFL) scheme based on DP and anomaly detection is proposed in this paper. In non-fully trusted FL, the honest-but-curious server may recover the private data from a specific client precisely based on the client's accessible model updates, resulting in client-level privacy leakage [17], [24], [25]. For privacy protection, Ada-PPFL uses DP to protect the model updates (satisfying (ε, δ)-DP). Besides, to eliminate the impact of malicious model updates, an anomaly detection method is designed. In the anomaly detection, the similar model updates are clustered into the same class, and the updates with large differences (from the majority of model updates) are treated as anomalies. Furthermore, the design of Ada-PPFL faces two major technical challenges, outlined as follows.

TABLE I

THE COMPARISON (✓: SUPPORT, ×: NO SUPPORT)

Schemes Performance	[13]	[14]	[15–21]	[22]	[23]	Ada-PPFL
Non-fully trusted scenario	×	×	×	✓	✓	✓
Low computation cost	✓	×	✓	×	×	✓
High prediction accuracy	✓	✓	×	✓	×	✓
Strong privacy protection	✓	✓	✓	×	✓	✓

1) *The first challenge is to achieve the strong client-level privacy protection while ensuring high prediction accuracy of the global model:* Privacy protection based on DP is computationally efficient, and it is provably secure theoretically. Additionally, model updates that are protected by DP still retain certain features that can aid in the detection of malicious clients. Nevertheless, it is important to strike a balance when adding DP-noise, as excessive noise can lead to a significant decrease in prediction accuracy, while insufficient noise fails to provide adequate privacy protection. To address this challenge, we propose an adaptive DP strategy. In this strategy, after each client generates new DP-protected local model updates in each round, it adaptively selects to use some new local model updates and discard others (the discarded ones are replaced by the model updates of the prior round). The developed selection criterion guarantees that only those model updates that can contribute to high prediction accuracy of the global model are included in the aggregation process.

2) *The second challenge is to precisely detect malicious model updates:* If the updates are protected, the malicious ones are hard to detect precisely since the protection process might significantly reduce the detection precision. To tackle this challenge, we devise a DP-tolerant Anomaly Detection (named as DPAD) algorithm to detect the malicious model updates. The DPAD design is inspired by DBSCAN algorithm proposed in [26]. Specifically, DPAD uses spatial density information to learn a cluster over two-dimensional (2D) space. The model updates outside the cluster boundary are evaluated as anomalies. Consider there are two clusters obtained on the original model updates and DP-protected model updates, after DP-protection, any local subspace density is changed slightly.

The comparisons with the previous schemes [13], [14], [15], [16], [17], [18], [19], [20], [21] are summarized in TABLE I. The comparison results show that the proposed Ada-PPFL can deal with the non-fully trusted scenario of FL, and shows superiority in computation cost, prediction accuracy and privacy protection. The main contributions of this paper can be summarized as follows.

- We propose an Ada-PPFL scheme with DP as the underlying technology for the non-fully trusted FL. The proposed Ada-PPFL can eliminate the adverse effects of malicious clients on model training and prevent honest-but-curious servers from stealing clients' privacy.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

- We propose an adaptive DP strategy to achieve a strong client-level privacy protection, while minimally compromising the prediction accuracy of the global model.
- We introduce an innovative approach termed DPAD that demonstrates remarkable efficacy in precisely detecting malicious model updates, even if all model updates are DP-protected.

The remainder of this paper is organized as follows. In Section II, some related works are reviewed. The preliminaries about FL, DP and DBSCAN are introduced in Section III. Section IV describes the system model, threat model, assumptions, and goals of Ada-PPFL. Section V presents the detailed design of Ada-PPFL. The convergence and security analyses of Ada-PPFL are presented in Section VI. Performance evaluations are conducted in Section VII, followed by the conclusions in Section VIII.

II. Related Work

Federate Learning: FL is an efficient and secure scheme for distributed network, originally proposed by Jakub Konecny *et al*. [5] in 2015. In the setting of federated learning (FL), the raw client data remains on the local devices and is never transmitted to the central server. Only the model updates, such as model gradients or weights, are uploaded to the server for aggregation. The two most commonly used FL models are Federated Stochastic Gradient Descent (FedSGD) [15] and Federated Averaging (FedAvg) [27]. In FedSGD, the interaction between the server and the client is the model gradients. FedAvg is a communication-efficient FL approach that uploads model weights instead of gradients.

Attacks in FL: The adversaries, such as honest-but-curious servers and malicious clients, can launch attacks targeting clients' privacy or model training. In FL, an honest-but-curious server can control everything (including gradients and weights) sent to the server in all rounds, and may attempt to exploit this access to extract sensitive information from the clients, thereby compromising their privacy. For example, the honest-but-curious server can successfully reconstruct the private data of the clients or infer the private features of the training data through model inversion attack [28] and inference attacks [29], [30]. For the malicious clients, they can undermine the model performance of FL by poisoning local data and model parameters. Examples of such attacks include label-flipping attacks [9], noise attacks [9], [10] and backdoor attacks [11], [12].

Defenses in FL: Recently, researchers have focused on addressing adversarial attacks, and many outstanding defense strategies have been proposed. For the honest-but-curious servers, the strategies based on Secure Multiparty Computation (SMC) [13], Homomorphic Encryption (HE) [14], and Differential Privacy (DP) [15], [16], [17] are the most popular and efficient solutions. Besides, some solutions have been developed to mitigate the impact of poisoning attacks on model training. The byzantine-resilient aggregation rules, including Krum [18], GeoMed [19] and Trimmed mean [20], can mitigate the impact of byzantine nodes (i.e., malicious nodes/clients) by selecting a representative client model update to estimate the true center model updates. The works in [9] and [10] have adopted variance-reduced stochastic gradient descent (SGD) and additional regularization term to achieve a similar defense. In [21], the central server utilizes a powerful detection model to identify and remove malicious model updates. In addition, the work in [31] applied trusted execution environment (TEE) to guarantee integrity and privacy, and in [32] introduced a defense against Sybil-based poisoning. In practice, the adversaries in FL may include both adversary servers and malicious clients. However, the defense strategies that can address this case have not been investigated in depth.

III. Preliminaries

In this section, some related preliminaries about federated learning, differential privacy and density-based spatial clustering of applications with noise are introduced.

Federated Learning: FL is a machine learning setting where multiple decentralized clients collaborate to train a model under the coordination of a central server. In FL, the raw datasets are kept locally on the clients' devices [33]. In general, the processes of FL can be decoupled into multiple training rounds. In each round, the server first sends the initial model or global model updates to clients. Then each client trains the model locally using his/her data, and finally uploads the trained model updates to the server for aggregation.

Assume that there are N clients and the set of samples is $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, where the set u_i contains the samples stored at client i. Let (x_k, y_k) be a sample k of client i, then the samples of client i can be rewritten as $u_i = (X_i, Y_i) = \{(x_k, y_k)\}_{k\in[1,n_i]}$, where $n_i = |u_i|$ is the sample number of client i. The loss function on the sample (x_k, y_k) with model parameters (i.e., model weights) ω is typically defined as $f_k(\omega) = \ell(\omega; x_k, y_k)$. In FL, the goal of model training is to find a weight ω that minimizes the loss. The distributed optimization model can be expressed as

$$\min_{\omega} F(\omega) = \sum_{i=1}^{N} \frac{n_i}{n} F_i(\omega),$$

where the local objective $F_i(\omega) = \frac{1}{n_i} \sum_{k\in[1,n_i]} f_k(\omega)$ and $n = |\mathcal{U}|$ is the number of total samples.

Differential Privacy (DP): DP [34] provides a strong criterion of privacy preservation for the algorithms on aggregate datasets. The DP can be formally defined as follows.

Definition 1: Differential Privacy: Let $\mathcal{D}$ be a collection of datasets. A mechanism $\mathcal{M}$: $\mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-DP, if for any two adjacent datasets $d, d' \in \mathcal{D}$ and any outputs $O \subseteq \mathcal{R}$ it holds that

$$\Pr[\mathcal{M}(d) \in O] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in O] + \delta,$$

where $\varepsilon > 0$ is the privacy budget and decides the privacy level, i.e., the smaller ε, the stronger privacy guarantee.

In the context of FL, d and d' can be defined as two different types of adjacent datasets, i.e., sample-adjacent datasets [17], [35], [36] and client-adjacent datasets [17], [37], [38].

Sample-adjacent datasets: Let d and d' be two datasets of training samples. If d and d' differ in only one sample, they are sample-adjacent.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

Client-adjacent datasets: Let d and d' be two datasets of training samples, where each sample is associated with a client. Then d and d' are adjacent if the samples of only one client in d are different from the samples of any client in d'.

For example, let $d = \{a_1, a_2, a_3, a_4, a_5\}$ and $d' = \{a_1, a_2, a_3, a_5\}$ be two datasets, where the two datasets are different in a_4. If a_i represents a sample, d and d' are sample-adjacent. If a_i represents the samples of client i, d and d' are client-adjacent.

A standard paradigm for providing a privacy-preserving approximation of the query function $h : \mathcal{D} \rightarrow \mathcal{R}^m$ is to add noise proportional to the sensitivity S_h of the query function h. The sensitivity S_h is defined as the maximum of absolution ℓ_2 difference $\max_{d,d' \in \mathcal{D}} ||h(d) - h(d')||_2$, where d and d' are adjacent. The Gaussian noise mechanism is the one of such privacy-preserving approximations, defined by $\mathcal{M}(d) = h(d) + \mathcal{N}(0, S_h^2 \sigma^2 I)$, where $\mathcal{N}$ is a normal distribution with the mean 0 and the standard deviation $S_h \sigma$, I is identity matrix, and $S_h^2 \sigma^2 I$ represents a covariance matrix.

Besides, there is a property in DP that makes it particularly useful in applications, that is sequential composition [39], [40].

Theorem 1: (Sequential composition). Suppose m mechanisms $\{\mathcal{M}_1, \ldots, \mathcal{M}_m\}$ satisfy (ε_i, δ)-DP, respectively. Then, a mechanism formed by $(\mathcal{M}_1(d), \ldots, \mathcal{M}_m(d))$ satisfies $(\sum_i^m \varepsilon_i, m\delta)$-DP.

Density-based Spatial Clustering of Applications with Noise (DBSCAN): DBSCAN [26], [41] is a density-based clustering non-parametric algorithm that groups closely packed points together and marks points that are located alone in low-density regions as outliers.

Given a set of points in some space, let *eps* specify the radius of a neighborhood with respect to some points, and m' represents the minimum number of points required to form a dense region. These points can be classified as core points, directly (density-) reachable points, reachable points and outliers, as follows:

Core points: if at least m' points are within distance *eps* of point p (including p), p is defined as core point;

Directly reachable points: if point q is within distance *eps* from core point p, point q is directly reachable from point p;

Reachable points: if there is a path $p_1, \cdots, p_n$ with $p_1 = p$ and $p_n = q$, and each p_{i+1} is directly reachable from p_i, point q is reachable from p;

Outliers: If all points are not reachable from any other point, these points are outliers or noise points.

In DBSCAN, if p is a core point, p and all points that are reachable from it form a cluster.

IV. Problem Formulation

In this section, we present the threat model, assumptions, goals, and system model of the proposed Ada-PPFL.

System Model: The system model of Ada-PPFL is shown in Fig. 1, including client layer and server layer. The clients are responsible for local model training, while the server performs model update aggregation and the detection operations of malicious model updates. The data processing procedures of Ada-PPFL are shown in the right part of Fig. 1, and the details are described below.

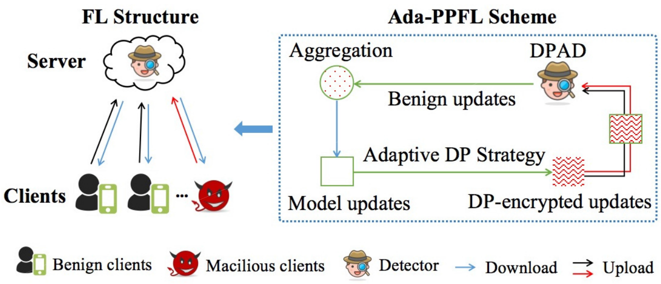


Fig. 1. The system model of Ada-PPFL.

1) Local clients download the initial global model or aggregated global model updates from the server.
2) Local clients train model on their data to generate new model updates, and perform an adaptive DP protection for the new model updates before they are uploaded (The specific design is introduced in Section V).
3) The server receives DP-protected model updates uploaded by the clients, and then employs DPAD algorithm to detect any anomalies in these updates. (The specific design is shown in Section V).
4) The 'benign updates' after detection are aggregated to get the global model updates of the next round.

Threat Model: In this paper, we consider the non-fully trusted FL, where the honest-but-curious server and the malicious clients coexist. 1) *Honest-but-curious server*. She/He follows aggregation protocols honestly, but can launch any attack to steal or compromise clients' privacy information. For example, she/he can launch model reversal attacks or inference attacks to reconstruct private data of the victim by analyzing the model updates of the clients, and she/he can further compromise the privacy of the clients by sharing the entire protocol view with others. 2) *Malicious clients*. They can maliciously upload arbitrary values to the server by poisoning sample data and model updates to undermine the model convergence or corrupt the global model.

Assumptions: 1) If the model updates are only slightly modified by malicious clients, they are considered to remain benign. 2) To defend against collusion attacks, we assume the proportion of benign clients is greater than 50%. In a real FL scenario, it is reasonable to assume that the number of benign clients is much larger than the number of malicious clients.

Goals: There are two design goals of the proposed Ada-PPFL. 1) *Privacy-preserving*. Ada-PPFL should ensure that each client's model updates satisfy (ε, δ)-DP in each training round. 2) *High-accuracy*. Ada-PPFL should ensure that the global model achieves high prediction accuracy.

V. Design of Ada-PPFL

This section describes the design of Ada-PPFL in detail, including the main training process of Ada-PPFL, the adaptive DP strategy, and the proposed detection algorithm DPAD.

A. The Main Process of Ada-PPFL

The goal of Ada-PPFL is to obtain a high-performance global model while ensuring strong privacy protection for participating clients. On the client-side of Ada-PPFL, we develop an adaptive DP strategy for noise addition to provide client-level privacy protection and reduce the impact of noise in the prediction accuracy of global model. On the server-side of Ada-PPFL, we propose a DP-tolerant anomaly detection with adaptive scaling for clustering to identify and remove the malicious model updates, ensuring the high prediction accuracy of global model. The main process of Ada-PPFL is described in Algorithm 1.

In Algorithm 1, Step 5 - Step 7 present the process on the clients. The function $\mathcal{A}$ in Step 7 represents the adaptive DP strategy, and will be described in Section V-B in detail. Step 9 is the detection process on the server, and the proposed DPAD is presented in Section V-C in detail.

Algorithm 1 The Main Process of Ada-PPFL

```
1  Initialization: Samples of $n$ clients $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, model weight $\omega_0$ and $\omega_{-1} = zeros(|\omega_0|)$, client selection probability $q \in (0, 1]$, FedAvg is the update method, and the sample number in each client is the same;
2  for each global round $T = 1, 2, \ldots$ do
3      Learning rate $\eta_T$;
4      $\mathcal{U}_T \leftarrow$ (sample clients with probability $q$);
5      for each client $i \in \mathcal{U}_T$ in parallel do
6          Local training:
7          $\bar{\omega}^i_T = \mathcal{A}(d_i, \omega_{T-1}, \eta_T)$;
8      The selected clients upload their $\bar{\omega}^i_T$ to server;
9      The server detects these updates by DPAD and removes the malicious model updates;
10     The set of the remaining clients is $\mathcal{U}^l_T$, where $\mathcal{U}^l_T \subseteq \mathcal{U}_T$;
11     The remaining set of model updates $\bar{\omega}_T = \{\bar{\omega}^i_T | i \in \mathcal{U}_T\}$;
12     Update (i.e., aggregation) in server: $\omega_T \leftarrow \frac{1}{|\mathcal{U}^l_T|} \sum_{i \in \mathcal{U}^l_T} \bar{\omega}^i_T$;
```

B. The Adaptive DP Strategy

The adaptive DP strategy is designed to obtain strong privacy protection while ensuring high prediction accuracy of the global model. To achieve the goals, the client-level privacy protection is designed, and a selection criterion is developed to support the adaptive noise addition for the local training. The design of the adaptive DP strategy for client i is presented in Algorithm 2.

As described in Step 2 - Step 8 of Algorithm 2, we train a model on client's samples by using FedAvg with a diminishing learning rate. The learning rate of the round T is represented as η_T, which is decreasing over time. Step 9 - Step 11 achieve the client-level DP of Ada-PPFL. In Step 12 - Step 17, the selection criterion is developed to select the model updates that contribute the high prediction accuracy for model aggregation.

The detail designs of client-level privacy protection and the selection criterion for each client are presented as follows.

Client-level privacy protection: In order to achieve client-level privacy protection for each client, we first define a query function on the client-adjacent datasets, and then design a bounded-sensitivity for the query function. When the noises added to the model updates are based on the bounded-sensitivity, the client's model updates satisfy (ε, δ)-DP, achieving client-level privacy protection.

Algorithm 2 Adaptive DP Strategy for Client i (i.e., Function $\mathcal{A}$)

```
   Input: the samples of client $i$ ($i.e.$, $d_i$), global model weight $\omega_{T-1}$ at global round $T-1$, learning rate $\eta_T$, gradient clipping $C$;
   Output: The uploaded weight $\bar{\omega}^i_T$ of client $i$ at global round $T$;
1  Initialization: Batch number $b$, local epoch times $E$, gradient calculations $\nabla F_i$, DP mechanism in round $T$ is $\mathcal{M}_T$, learning rate $\eta_0$, conversion ratio $\lambda$, $\mathbf{w}^i_t$ is the gradient of client $i$ at time $t$, privacy budget $\varepsilon$;
2  $\mathcal{B} \leftarrow d_i$ splits into $b$ batches;
3  $\mathbf{w}^i_t = \omega_{T-1}$, $t = E(T-1)b$ and $k = 0$;
4  for each local epoch from 1 to E do
5      for batch $B \in \mathcal{B}$ do
6          $g^i_{t+k} = \nabla F_i(B|\mathbf{w}^i_{t+k})$;
7          $k = k + 1$;
8          $\mathbf{w}^i_{t+k} = \mathbf{w}^i_t - \eta_T g^i_{t+k-1}$;
9  The gradient at global round $T$ is $\hat{g}^i_T = g^i_t$, where $t = ETb$;
10 Clip gradient $\bar{g}^i_T = \hat{g}^i_T / \max(1, \frac{\|\hat{g}^i_T\|_2}{C})$;
11 Compute the sensitivity at global round $T$, $S_T \leftarrow 2\eta_T C$;
12 Noise scale $\sigma_T \leftarrow \{S_T/\varepsilon$ for $Q\}$;
13 Update the model weights of round $T$, i.e., $\omega^i_T = \omega_{T-1} - \eta_T \bar{g}^i_T$;
14 if $\lambda \cdot ||\omega^i_T - \omega^i_{T-1}||_2 + ||\mathcal{N}(0, I\sigma^2_{T-1})||_2 < 2 \cdot ||\mathcal{N}(0, I\sigma^2_T)||_2$ then
15     $\bar{\omega}^i_T = \bar{\omega}^i_{T-1}$;
16 else
17     $\bar{\omega}^i_T = \mathcal{M}_T(d_i) = \omega^i_T + \mathcal{N}(0, I\sigma^2_T)$;
```

1) The Query Function of Ada-PPFL: To formally guarantee client-level privacy, we apply DP to model training by using the notion of client-adjacent datasets. The whole process of local training in a round is defined as a query function Q of the client-level DP.

$$Q(d_i|\omega_T) = \omega_T - \eta_T \nabla F_i(d_i|\omega_T),$$

where $\nabla F_i(d_i|\omega_T)$ is the gradient calculations based on ω_T and b batches sampled from d_i in a training round. The sensitivity of Q can be represented as follows.

$$S_Q = \max_{d,d' \in \mathcal{D}} ||Q(d) - Q(d')||_2.$$

Different from the DP with adding noise to the aggregated values, our scheme adds noise in the model updates of each client. If the training samples of each client are a dataset, the samples of any two clients are client-adjacent datasets.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

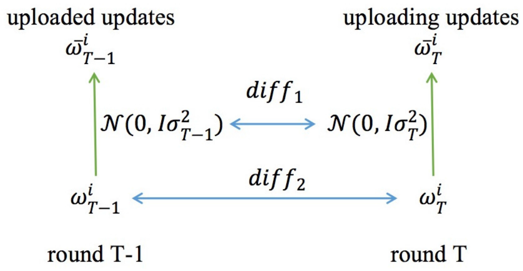


Fig. 2. Two rounds of model updates.

Therefore, the sensitivity of Q at the global round T in Ada-PPFL can be rewritten as

$$S_Q(\cdot|\omega_{T-1}) = \max_{d_i, d_j \in \mathcal{U}_T} ||Q(d_i|\omega_{T-1}) - Q(d_j|\omega_{T-1})||_2,$$

where $\mathcal{U}_T$ represents the samples of all clients in the global round T, ω_{T-1} is the model weight of global model in round $T-1$, $i, j \in [1, N]$ and $i \neq j$.

2) Bounded-Sensitivity for Query Function: The output of Q is varied with the different inputs and cannot be bounded. Without a priori knowledge on the bound of the size of updates, it is hard to calculate the sensitivity S_Q. Towards this end, the works in [36] and [38] proposed to clip updates (gradients/weights) with norm (i.e., ℓ_2). For example, in [38], the server sets a threshold C for the updates, and each model weight ω^i_t is replaced by $\omega^i_t / \max(1, \frac{||\omega^i_t||_2}{C})$, where t represents update round. This clipping ensures that the weight ω^i_t is preserved when $||\omega^i_t||_2 < C$, otherwise the weights are scaled down to be the norm of C. Towards this end, we also use a fixed clipping C to bound the sensitivity in this paper, and the sensitivity of query function Q in the round T can be represented as $S_T \leftarrow 2\eta_T C$.

Selection criterion: In the steps of adding noise, we adopt an adaptive strategy (motivated by [42]) to obtain stronger privacy protection by slightly sacrificing prediction accuracy. The main idea is that the clients adaptively select to use some new DP-protected local model updates and discard others, and then the discarded ones are replaced by the model updates of the prior round. According to Fig. 2, the details of the selection criterion are described as follows. When the replacement is performed, the sacrificed prediction accuracy $diff_2$ is $||\omega^i_T - \omega^i_{T-1}||_2$. Since the model updates and noise $\mathcal{N}$ follow different distributions, we use the constant λ to represent the conversion ratio of model updates to noise, and the converted noise is denoted as $\lambda \cdot diff_2$. Moreover, as the uploaded model updates of the previous round will not provide any more information to the server, we don't need to allocate privacy budget for the model updates of the current round. Then, the reduced noise is $\mathcal{N}(0, I\sigma^2_T)$, while the privacy protection is increased by ε. When $\lambda \cdot diff_2 + diff_1 < ||\mathcal{N}(0, I\sigma^2_T)||_2$, the replacement can help to reduce the added noise and thus improve the prediction accuracy, where $diff_1 = ||\mathcal{N}(0, I\sigma^2_{T-1})||_2 - ||\mathcal{N}(0, I\sigma^2_T)||_2$.

In summary, the selection criterion can be rewritten as: the model updates of the current round are replaced by the model updates of the prior round if $\lambda \cdot diff_2 + diff_1 < ||\mathcal{N}(0, I\sigma^2_T)||_2$, otherwise, the model updates of the current round are DP-protected by adding the noise $||\mathcal{N}(0, I\sigma^2_T)||_2$.

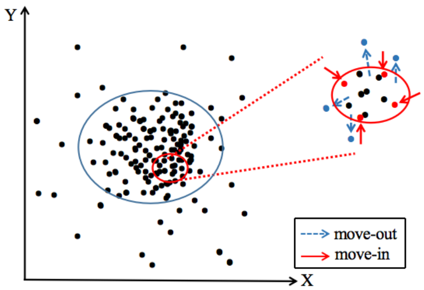


Fig. 3. An example to illustrate why DPAD perserves the density of updates over 2D space. After DP-protection, given a local subspace, the number of move-in updates and move-out updates are roughly the same, so the density over the subspace is changed slightly.

C. DPAD Algorithm

The DP-tolerant anomaly detection (DPAD) algorithm is designed to identify and remove the malicious model updates from the DP-protected model updates. After DP-protection, given a local subspace, the number of move-in updates and move-out updates are roughly the same, so the density over the subspace is changed slightly. Fig. 3 shows an example to explain the reason, where the density (over the 2D space) determines the cluster boundary shape in DPAD. Therefore, DPAD is *quasi-cluster-boundary-preserving*, indicating that the anomalies after protection are still high likely to be anomalies before protection. The details of DPAD algorithm are presented in Algorithm 3.

In Step 2 of Algorithm 3, we first use multi-dimensional scaling (MDS) to build a two-dimensional space for model updates. Because the generation of clusters in DPAD is based on spatial density, which is not suitable for high-dimensional model updates (i.e., curse of dimensionality).

In Step 3 of Algorithm 3, we adaptively adjust eps through a linear correlation between eps and noise scale, i.e., $eps = k\sigma_T + r$, where k and r are constant. The parameter eps is a reflection of the similarity between model updates. How to set eps is important. If eps is too large, the malicious model updates will easily be divided into clusters. On the contrary, if eps is too small, the number of benign clients' model updates in the cluster is small, and the DP-added noise will not be able to offset each other well during aggregation, which will decrease in the aggregation accuracy. Thus, we adaptively adjust eps to adapt to the changes caused by the model updates and the DP-added noise during training, instead of scaling similarity. Besides, according to Algorithm 2, the noise scale is calculated from model updates. Thus, the relation of eps with model updates and noise scale can be converted to a relation between eps and noise scale.

Step 4 - Step 20 of Algorithm 3 show the processes of clustering based on spatial-density. In Step 6 of Algorithm 3, the function $Q'(P, eps)$ is to obtain all model updates of the eps-neighborhood of P, and saved in the set Np. If the distance between the model update $\bar{\omega}^i_T$ and P is lower or equal to eps, $\bar{\omega}^i_T$ is the neighborhood of P, where the distance function is Euclidean metric. Step 10 - Step 20 of Algorithm 3 expand the set CLU based on the core point P.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3 DPAD

```
    Input: $\mathcal{U}_T$, clipping threshold $C$, and the DP-protected
           model updates $\bar{\omega}_T = \{\bar{\omega}^i_T | i \in \mathcal{U}_T\}$;
    Output: new $\bar{\omega}_T$;
 1  Initialization: $m'$, $CLU = \varnothing$ and $j = 0$;
 2  Map $\bar{\omega}_T$ to a 2-dimensional space, i.e.,
      $\bar{\omega}_T = MDS(\bar{\omega}_T, 2)$;
 3  $eps = k\sigma_T + r$, which is an adaptive setting;
 4  for $P$ in $\bar{\omega}_T$ do
 5      Mark $P$ as visited model updates;
 6      Compute $\mathsf{Np} = Q'(P, eps)$ to obtain all model
          updates within $P$'s $eps$-neighborhood;
 7      if $size(\mathsf{Np}) < m'$ then
 8          Mark $P$ as NOISE;
 9      else
10          $Clu_j = \varnothing$, and $j = j + 1$;
11          Add $P$ to cluster $Clu_j$;
12          for $P'$ in $\mathsf{Np}$ do
13              if $P'$ is not visited then
14                  Mark $P'$ as visited;
15                  $\mathsf{Np}' = Q'(P', eps)$;
16                  if $size(\mathsf{Np}') \geq m'$ then
17                      $\mathsf{Np}$ is equal to $\mathsf{Np}$ joined with $\mathsf{Np}'$;
18                  if $P'$ is not yet member of any cluster
                      then
19                      Add $P'$ to cluster $Clu_j$;
20          Put $Clu_j$ into the set $CLU$;
21  Cluster that satisfies $\max\{size(Clu_j)|Clu_j \in CLU\}$ is
      marked as $Clu$;
22  return $\bar{\omega}_T = Clu$;
```

After clustering, the target cluster with the maximum number of points is obtained, in which the values are the benign model updates, described in Step 21 of Algorithm 3.

VI. Theoretical Analysis

In this section, we present the convergence analysis and security analysis for Ada-PPFL.

A. Convergence Analysis

The convergence of the proposed Ada-PPFL scheme is analyzed with *Smooth and Strongly Convex cost function* and *Non-Smooth and Convex cost function*. The update mode of Ada-PPFL is inspired by the FedAvg method. Let ω^i_t be the model updates of the i-th client at the t-th (or time t) local model update. Let $\mathcal{I}_T$ be the set of global aggregation steps, i.e., $\mathcal{I}_T = \{kEb | k = 1, 2, \ldots\}$, where E is local epoch time and b is the number of batches in an epoch. If $t \in \mathcal{I}_T$, the local model updates of these rounds are clipped and noised, and then they will be uploaded to the server for aggregation.

Update Description: The updates of Ada-PPFL without the replacement can be described as

$$\mathbf{v}^i_{t+1} = \mathbf{w}^i_t - \eta_T \nabla F_i\left(\omega^i_t, \xi^i_t\right),$$

$$\mathbf{w}^i_{t+1} = \begin{cases} \mathbf{v}^i_{t+1}, & \text{if } t+1 \notin \mathcal{I}_T, \\ \sum_{i\in\mathcal{U}^l_T} p_i(\omega_{T-1} - \frac{\acute{\eta}_t \hat{g}^i_T}{\alpha^i_T} + \tilde{n}^i_T), & \text{if } t+1 \in \mathcal{I}_T, \end{cases}$$

where $\omega_{T-1} = \mathbf{w}_{(T-1)Eb}$, $\mathbf{v}^i_{t+1}$ represents the immediate result of the local update from $\mathbf{w}^i_t$, and $\hat{g}^i_T = (\omega_{T-1} - \mathbf{v}^i_{t+1})/\acute{\eta}_t$. ξ^i_t is the batch of samples uniformly chosen from the client's samples, and $\alpha^i_T = \max(1, ||\hat{g}^i_T||_2/C)$. $\acute{\eta}_t$ is the learning rate at time t and $\acute{\eta}_t = \eta_T$ when $T - 1 < \lfloor \frac{t}{Eb} \rfloor \leq T$. $\sum_{i\in\mathcal{U}^l_T} p_i = 1$, and $p_i = 1/|U^l_T|$.

The aggregation with all benign client participation at round t ($t \in \mathcal{I}_T$) can be written as

$$\mathbf{w}_t \leftarrow \sum_{i\in\mathcal{U}^l_T} p_i \mathbf{w}^i_t.$$

Based on the above analysis, the convergence results are analyzed under *smooth and strongly convex costs* and *non-smooth and convex costs*, respectively.

Assumptions of Smooth and Strongly Convex Costs: we review the following assumptions on the functions $F_1, \ldots, F_M$ in [43], where $[1, M] \in \mathcal{U}^l_T$.

Assumption 1: $F_1, \ldots, F_M$ are all L-smooth for all $\mathbf{v}$ and $\mathbf{w}$ satisfying $F_i(\mathbf{v}) \leq F_i(\mathbf{w}) + (\mathbf{v} - \mathbf{w})^T \nabla F_i(\mathbf{w}) + \frac{L}{2}||\mathbf{v} - \mathbf{w}||^2_2$.

Assumption 2: $F_1, \ldots, F_M$ are all μ-strongly convex for all $\mathbf{v}$ and $\mathbf{w}$ satisfying $F_i(\mathbf{v}) \geq F_i(\mathbf{w}) + (\mathbf{v} - \mathbf{w})^T \nabla F_i(\mathbf{w}) + \frac{\mu}{2}||\mathbf{v} - \mathbf{w}||^2_2$.

Assumption 3: Let ξ^t_i be sampled from the i-th client's local data uniformly at random. The variance of stochastic batch gradients in each client is bounded, i.e., $\mathbb{E}||\nabla F_i(\mathbf{w}^t_i, \xi^t_i) - \nabla F_i(\mathbf{w}^t_i)||^2 \leq \acute{\sigma}^2_i$ for $i \in \mathcal{U}^l_T$.

Assumption 4: The expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E}||\nabla F_i(\mathbf{w}^t_i, \xi^t_i)||^2 \leq G^2$ for $i \in \mathcal{U}^l_T$ and all t.

Assumptions 1 and 2 are standard for some functions such as the ℓ_2-norm regularized linear regression, the logistic regression, and the softmax classifier.

Assumptions of Non-Smooth and Convex Costs: we make the following assumptions on the functions $F_1, \ldots, F_M$ in [43], where $[1, M] \in \mathcal{U}^l_T$.

Assumption 5: $F_1, \ldots, F_M$ are all convex (i.e., non-strongly convex) for all $\mathbf{v}$ and $\mathbf{w}$ satisfying $F_i(\mathbf{v}) \geq F_i(\mathbf{w}) + (\mathbf{v} - \mathbf{w})^T \partial F_i(\mathbf{w})$, where $\partial F_i(\mathbf{w})$ is subgradient of $F_i(\mathbf{w})$ with respect to $\mathbf{w}$.

Assumption 6: The subgradient $\partial F_i(\mathbf{w}^t_i, \xi^t_i)$ is L'-Lipschitz bounded, i.e., $||\partial F_i(\mathbf{w}^t_i, \xi^t_i)|| \leq L'$ for $i \in \mathcal{U}^l_T$ and all t.

Theorem 2: Let Assumptions 1 to 4 hold and $L, \mu, \acute{\sigma}_i, G$ be defined therein. $J = \sum_{i\in\mathcal{U}^l_T} p^2_i \acute{\sigma}^2_i + 6L\Gamma + 2(Eb-1)^2 G^2$, where Γ describes the degree of non-iid and equals $F^* - \sum_{i\in\mathcal{U}^l_T} p_i F^*_i$. F^* and F^*_i are the minimum values of F and F_i, respectively. Then

$$\mathbb{E}\left[F\left(\tilde{\omega}_T\right)\right] - F^* \leq \frac{Lv}{2(\gamma + T)},$$

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

where $v = \max\big((EbJ\beta^2 + 4z^2C^2)/(\beta\mu/\alpha_T^2 - 1), (\gamma+1)\Delta_1\big)$, $\Delta_1 = \mathbb{E}\,\|\tilde{\omega}_0 - \omega^*\|^2$, and ω^* is the model optimal weights. In addition, $\tilde{\omega}_T = \tilde{\mathbf{w}}_{t+1}$ while $T = \frac{t+1}{Eb}$.

Theorem 3: If Assumption 5 and Assumption 6 hold, i.e.,

$$\mathbb{E}[F(\tilde{\omega}_T)] - F^* \leq \frac{\sqrt{Eb}(Eb+4)L'^2(1+\ln(T-1))}{\sqrt{T-1}} + \frac{\Delta_1}{\sqrt{T-1}} + \frac{\psi(1+\ln(T-1))}{\sqrt{T-1}},$$

where T is the global round, $\mathbb{E}\,\|\tilde{\omega}_0 - \omega^*\|^2 = \Delta_1$, and $\psi = C^2[(\frac{L'}{C} - 1)^2(1+4z^2) + 4z^2]$.

The proofs of Theorem 2 and Theorem 3 are given in the Appendix.

When the replacement occurs on some clients at time $t+1 \in \mathcal{I}_T$, these clients are represented as $\mathcal{U}_T^r$. The rest of clients (without replacements) who upload model updates successfully are denoted as $\mathcal{U}_T^n$, and $\mathcal{U}_T^l = \mathcal{U}_T^r + \mathcal{U}_T^n$. Thus,

$$\tilde{\mathbf{v}}_{t+1} = (\tilde{\alpha}_{T-1}\tilde{\mathbf{w}}_{t+1-Eb} - (\tilde{\alpha}_{T-1} - 1)\tilde{\omega}_{T-2} + \tilde{\alpha}_{T-1}\tilde{n}_{T-1}) + (\tilde{\alpha}_T\tilde{\mathbf{w}}_{t+1} - (\tilde{\alpha}_T - 1)\tilde{\omega}_{T-1} + \tilde{\alpha}_T\tilde{n}_T),$$

where $\tilde{\mathbf{w}}_{t+1} = \sum_{i\in\mathcal{U}_T^n} p_i\mathbf{w}_t^i$, $\tilde{n}_T = \sum_{i\in\mathcal{U}_T^n} p_i n_T^i$, $\tilde{\alpha}_T = \sum_{i\in\mathcal{U}_T^n} p_i\alpha_T^i$, $\tilde{\mathbf{w}}_{t+1-Eb} = \sum_{i\in\mathcal{U}_T^r} p_i\mathbf{w}_{t-Eb}^i$, $\tilde{n}_{T-1} = \sum_{i\in\mathcal{U}_T^r} p_i n_{T-1}^i$, and $\tilde{\alpha}_{T-1} = \sum_{i\in\mathcal{U}_T^r} p_i\alpha_{T-1}^i$.

1) Convergence Results on Smooth and Strongly Convex Costs: According to Theorem 2, we have

$$\mathbb{E}\left[F(\tilde{\omega}_T)\right] - F^* \leq \frac{Lv_1}{2(\gamma+T)} + \frac{Lv_2}{2(\gamma+T-1)} < \frac{Lv}{(\gamma+T-1)},$$

where $v_1 = \max\left((EbJ\beta^2+4z^2C^2)/(\frac{\beta\mu}{\alpha_{T-1}^2} - 1), (\gamma+1)\Delta_1\right)$, $v_2 = \max\left((EbJ\beta^2+4z^2C^2)/(\frac{\beta\mu}{\alpha_{T-1}^2} - 1), (\gamma+1)\Delta_1\right)$, and $v = \max(v_1, v_2)$.

Specifically, we choose $\beta = \eta_0/\rho, \gamma = 1/\rho$ and $\alpha_T = \max(\alpha_T, \alpha_{T-1})$ in this paper, then $\eta_t = \frac{\eta_0}{\rho T+1}$ and

$$\mathbb{E}\left[F(\tilde{\omega}_T)\right] - F^* \leq \frac{\rho L}{\rho T + 1 - \rho}\left(\frac{\alpha_T^2(EbJ\eta_0^2 + 4C^2\rho^2)}{\rho\mu\eta_0 - \alpha_T^2\rho^2} + \frac{1+\rho}{\rho}\Delta_1\right).$$

$\alpha_T^i = \max(1, ||\hat{g}_T^i||_2/C)$ and $\max(\tilde{\alpha}_T) \leq \frac{G}{C}$, where $k \in [1, T']$ and $G \geq C$. Thus, the upper bound of α_T is a constant. Besides, $||\hat{g}_t^i||_2$ of benign model updates is always bounded. The unbounded values of the malicious model updates will be removed and are not aggregated in the server. Finally, $\mathbb{E}\left[F(\tilde{\omega}_T)\right] - F^*$ will go to zero when T is large enough. It indicates that $\tilde{\omega}_T$ converges to the optimal model weight with the convergence rate $\mathcal{O}(1/T)$.

2) Convergence Results on Non-Smooth and Convex Costs: According to Theorem 3, let $\eta_T = \frac{1}{\sqrt{T}}$ and then we have

$$\mathbb{E}\left[F(\tilde{\omega}_T)\right] - F^* \leq \left(\frac{\Delta_1}{\sqrt{T-2}} + \frac{\psi(1+\ln(T-2))}{\sqrt{T-2}} + \frac{\sqrt{Eb}(Eb+4)L'^2(1+\ln(T-2))}{\sqrt{T-2}}\right) + \left(\frac{\psi(1+\ln(T-1))}{\sqrt{T-1}} + \frac{\Delta_1}{\sqrt{T-1}} + \frac{\sqrt{Eb}(Eb+4)L'^2(1+\ln(T-1))}{\sqrt{T-1}}\right)$$
$$\leq 2\left(\frac{\psi(1+\ln(T-1))}{\sqrt{T-1}} + \frac{\sqrt{Eb}(Eb+4)L'^2(1+\ln(T-1))}{\sqrt{T-1}} + \frac{\Delta_1}{\sqrt{T-1}}\right),$$

where ψ, Δ_1, E, b and L' are constants. Thus, $\mathbb{E}\left[F(\tilde{\omega}_T)\right] - F^*$ will go to zero when T is large enough. It means that $\tilde{\omega}_T$ converges to the optimal model weight as training proceeds. Besides, the convergence rate is $\mathcal{O}(\ln T/\sqrt{T})$.

B. Security Analysis

Let $Q : \mathcal{D} \rightarrow \mathcal{R}^m$ be an arbitrary m-dimensional query function, and let its ℓ_2-sensitivity be $S_Q = \Delta_2 h = \max_{adjacent\{d,d'\}\in\mathcal{D}} ||Q(d) - Q(d')||_2$. The Gaussian mechanism adds noise scaled to $\mathcal{N}(0,\sigma^2)$ to each of the m components of the outputs (i.e., model weights).

Theorem 4: Let ε be arbitrary. For $c^2 > 2ln(1.25/\delta)$, the mechanism $\mathcal{M}$ with $\sigma \geq c\Delta_2 h/\varepsilon$ of the proposed Ada-PPFL satisfies $(qT'\varepsilon, qT'\delta)$-DP in the client-level, where the condition $c^2 > 2ln(1.25/\delta)$ denotes the bound of constant c that is necessary for (ε, δ)-DP, $\Delta_2 h = 2\eta_T C$, and $T' \leq T$. T' is the number of the rounds that are allocated privacy budget and T is the number of rounds. q is the probability of participation in training round, and then

$$\Pr[\mathcal{M}(d) \in O] \leq e^{\varepsilon}\Pr[\mathcal{M}(d') \in O] + \delta.$$

Proof: Given the client's dataset d_i and query Q, the Gaussian mechanism will return $Q(d_i) + \mathcal{N}(0, I\sigma^2)$, where the noise satisfies normal distribution. Assume that the learning rate of each client is the same in each round. The sensitivity of $Q(\cdot|\omega_{T-1})$ in the global round T of the proposed Ada-PPFL is S_T (i.e., $S_Q(\cdot|\omega_{T-1})$), where $(\cdot)$ represents all samples of round T. Then S_T can be expressed as

$$\begin{aligned} S_T &= \max_{\{d_i,d_j\}\in\mathcal{D}} ||Q(d_i) - Q(d_j)||_2 = \max_{\{\omega_T^i,\omega_T^j\}\in\omega_T^*} ||\omega_T^i - \omega_T^j||_2 \\ &= \max_{\{\bar{g}_T^i,\bar{g}_T^j\}\in\bar{g}_T^*} ||(\omega_T - \eta_T\bar{g}_T^i) - (\omega_T - \eta_T\bar{g}_T^j)||_2 \\ &= \max_{\{\bar{g}_T^i,\bar{g}_T^j\}\in\bar{g}_T^*} ||\eta_T(\bar{g}_T^i - \bar{g}_T^j)||_2. \end{aligned}$$

Since the bounded $\bar{g}_T^i = \hat{g}_T^i/max(1, \frac{\|\hat{g}_T^i\|_2}{C})$, we have $||\bar{g}_T^i||_2 \leq C$. Then, the sensitivity S_T satisfies that

$$S_T = \max_{\{\bar{g}_T^i,\bar{g}_T^j\}\in\bar{g}_T^*} ||\eta_T(\bar{g}_T^i - \bar{g}_T^j)||_2 \leq 2\eta_T C.$$

Thus, the sensitivity of Q of the global round T can be denoted as $S_T = 2\eta_T C$. The noise scale is equal to $\sigma_T = S_T/\varepsilon$, and $\mathcal{M}_T(d_i) = Q(d_i) + \mathcal{N}(0, I\sigma_T^2)$. Then we have the privacy loss

$$\frac{\Pr(\mathcal{M}_T(d_i) = o)}{\Pr(\mathcal{M}_T(d_j) = o)} = \frac{\Pr((Q(d_i) - \mathcal{N}(0, I\sigma_T^2)) = o)}{\Pr((Q(d_j) - \mathcal{N}(0, I\sigma_T^2)) = o)}$$

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

$$
\begin{aligned}
&= \frac{\Pr(\mathcal{N}(0, I\sigma_T^2) = Q(d_i) - o)}{Pr(\mathcal{N}(0, I\sigma_T^2) = Q(d_j) - o)} \\
&= \frac{\Pr(\mathcal{N}(0, I\sigma_T^2) = a_i)}{\Pr(\mathcal{N}(0, I\sigma_T^2) = a_i + S_T(\cdot|\omega_{T-1}))} \\
&= \left|\frac{exp((-1/2\sigma_T^2)||a_i||_2^2)}{exp((-1/2\sigma_T^2)||a_i + S_T||_2^2)}\right| \\
&= |exp((-1/2\sigma_T^2)(||a_i||_2^2 - ||a_i + S_T||_2^2))|,
\end{aligned}
$$

where $o \in O$ is an output and a_i represents the difference between $Q(d_i)$ and o.

The privacy loss can be written as $ln\frac{\Pr(\mathcal{M}_T(d_i)=o)}{\Pr(\mathcal{M}_T(d_j)=o)} = |(1/2\sigma_T^2)(||a_i||_2^2 - ||a_i + S_t||_2^2)|$. According to Theorem A.1. in [34], the mechanism $\mathcal{M}_T$ is (ε, δ)-DP when $\sigma \geq c\Delta_2 h/\varepsilon$, where $\Delta_2 h = 2\eta_T C$ and $c^2 > 2ln(1.25/\delta)$.

Let the mechanism be $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_k, \ldots\}$. The query function Q of each round (i.e., $Q(\cdot|\omega_{T-1})$ of global round T) is dependent on its previous global model weight (ω_{T-1}). As the global model weights are different and public in each round, each query function is independent. Thus, the mechanisms $\{\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_k, \ldots\}$ are independent. According to Theorem 2, the mechanism $\mathcal{M}$ is $(q\sum_{i=1}^{T}\varepsilon_i, q\sum_{i=1}^{T}\sigma_i)$-DP, where the number of participations per client throughout the training process is qT. In this paper, all parameters ε_i and σ_i are equal to ε and σ, respectively.

In addition, since each client's data is independent and invisible, the query function Q can only work on at most one client's data at a time. Toward this end, any two clients' datasets are taken as the adjacent datasets, which is an extra case of client-adjacent datasets.

In summary, the mechanism $\mathcal{M}$ satisfies $(qT\varepsilon, qT\sigma)$-DP in the client-level, when $\sigma \geq c\Delta_2 h/\varepsilon$, $c^2 > 2ln(1.25/\delta)$ and $\Delta_2 h = 2\eta_T C$. Moreover, some uploading model weights are replaced by the previous uploaded weights in the proposed Ada-PPFL. The actual number of client's DP-protected weights visible to the server is T'. Thus, the mechanism $\mathcal{M}$ satisfies $(qT'\varepsilon, qT'\sigma)$-DP and Theorem 4 holds. □

VII. Experiments

In this section, we first introduce the experiment setup. Then we evaluate the performance of Ada-PPFL.

A. Experiment Setup

Datasets: We conduct experiments on MNIST handwritten digits dataset[1] and CIFAR-10 dataset [2] [44]. MNIST dataset consists of 60,000 training images (28 × 28 pixels) and 10,000 test images with 10 categories (i.e, output classes or labels). CIFAR-10 dataset consists of 50,000 colour training images (3×32×32 pixels) and 10,000 colour test images in 10 classes. The training datasets are independent identically distributed (iid) and non-independent identically distributed (non-iid), and evenly partitioned into 100 clients in the following experiments (balanced setting). In the iid setting, each client is randomly assigned a uniform distribution over 10 classes. In the non-iid setting, the training data is sorted by class and then divided into 200 partitions, with each client being randomly assigned to two partitions [45].

Models: The experiments of Ada-PPFL are conducted on the Logistic Regression (LR, one connected layer with Softmax activation) and Convolutional Neural Network (CNN, two 5 × 5 convolution layers and three fully connected layers with ReLu activation). The update mode of Ada-PPFL is FedAvg, where the local epoch $E = 1$ and batch size equals 10. Besides, all clients participate in each global aggregation and m' is equal to 2 in DPAD algorithm. In the convergence validation experiments, we set up four different cost functions for the LR task. Specifically, the cost functions are given by

$$F_i(\omega) = \frac{1}{n}\sum_{k=1}^{n} \text{CrossEntropy}\ \left(f\left(\omega; x_k\right), y_k\right), \tag{1}$$

$$F_i(\omega) = \frac{1}{n}\sum_{k=1}^{n} \text{CrossEntropy}\ \left(f\left(\omega; x_k\right), y_k\right) + \lambda_1\|\omega\|_2^2, \tag{2}$$

$$F_i(\omega) = \frac{1}{n}\sum_{k=1}^{n} \text{CrossEntropy}\ \left(f\left(\omega; x_k\right), y_k\right) + \lambda_1\|\omega\|_1, \tag{3}$$

$$
\begin{aligned}
F_i(\omega) = \frac{1}{n}\sum_{k=1}^{n} &\text{CrossEntropy}\ \left(f\left(\omega; x_k\right), y_k\right) \\
&+ \lambda_1\|\omega\|_1 + \lambda_2 r(\omega),
\end{aligned} \tag{4}
$$

where $r(\omega) = \sum_{j=1}^{d}\frac{(\omega(j))^2}{1+(\omega(j))^2}$ is a smooth but non-convex regularization term and λ_2 controls the influence of $r(\omega)$. d is the size of ω and $\omega(j)$ is the j-th value of the vector ω. Cost functions in Eqs. (1), (2), (3), and (4) are convex and smooth, strongly convex and smooth, convex and non-smooth, non-convex and non-smooth, respectively. Besides, the LR tasks with Eqs. (1), (2), (3), and (4) are denoted as LR-0, LR-1, LR-2, and LR-3, respectively.

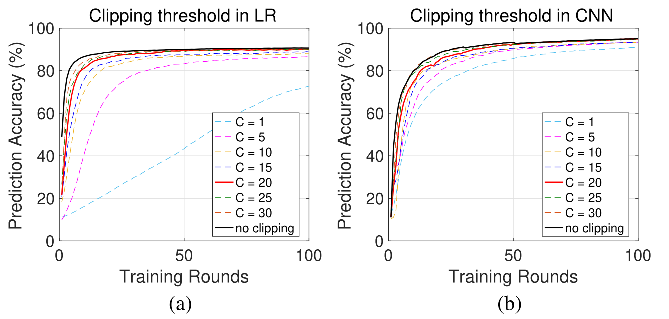
Attacks: In our experiments, three baseline attacks (including two untargeted attacks and a targeted attack) are simulated. 1) *Sign-flipping Attack:* It is an untargeted attack where the malicious clients flip the signs of their local model updates [9], [10]. 2) *Additive Noise Attack:* It is an untargeted attack where the malicious clients add Gaussian noise to their local model updates. In the experiments, the noise scale is the same for each client [9], [10]. 3) *Backdoor Attack:* It is a targeted attack, referred as model poisoning attack [11], [21], [46]. In our experiments, we examine the semantic backdoor attack, where the malicious clients attempt to manipulate the model to classify images with the label "7" as label "5" on MNIST dataset and classify images with the label "frog" as label "deer" on CIFAR-10 dataset.

Metrics: Four metrics are used, including 1) prediction accuracy, 2) privacy protection degree, 3) detection precision, and 4) accuracy difference. The prediction accuracy is evaluated by using testing data. The degree of privacy protection provided by Ada-PPFL is determined by the total privacy budget ε of the client during training. A smaller privacy

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.cs.toronto.edu/ kriz/cifar.html

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 4. (a) The prediction accuracy of Ada-PPFL with different clipping thresholds in LR task. (b) The prediction accuracy of Ada-PPFL with different clipping thresholds in CNN task.

budget indicates stronger privacy protection. The detection precision is equal to $|TP|/(|TP|+|FP|)$, where $|TP|$ is the number of benign model updates in the target cluster Clu_1, and $|FP|$ is the number of malicious model updates in Clu_1. The accuracy difference denotes the difference between two prediction accuracy, and is used to metric the similarity of two training model.

Implementations: The implementations of Ada-PPFL are achieved by PYTHON 3.7 on PC with 3.1 GHz Intel Core i5, 8 GB RAM, and macOS High Sierra operating system.

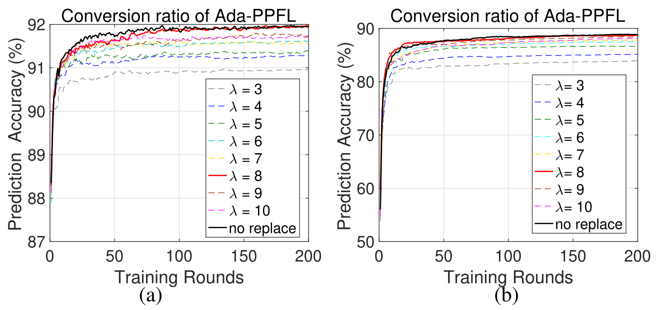
B. Algorithm Parameters

Clipping threshold C: The clipping threshold of gradients is an important factor that determines the sensitivity (or noise scale). To verify the performance of Ada-PPFL, the appropriate clipping C should be confirmed first. According to Fig. 4(a) and (b), $C = 20$ is selected as the clipping threshold of Ada-PPFL. Because the model maintains a similar prediction accuracy under this threshold as in the non-threshold case, and the smaller C means that less noise is added in DP scenario.

Conversion ratio λ: The replacement can trade a smaller loss of accuracy for a higher security, and the replacement is affected by λ. Thus, the experiments on different λ are conducted on iid and non-iid datasets. The experimental results are depicted in Fig. 5 (a) and Fig. 5 (b), which show that the variation of prediction accuracy in different λ. According to the results, the prediction accuracy at $\lambda = 8$ is high and similar to the situation without the replacement. Besides, the privacy protection is improved at conversion ratio $\lambda = 8$, which will be verified in Section VII-D.

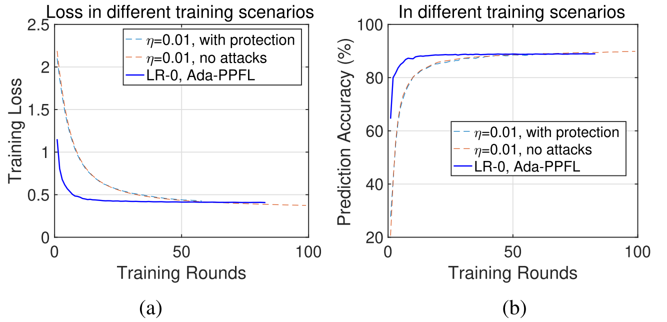
C. Convergence of Ada-PPFL

Convergence in different training setting: To verify the convergence of Ada-PPFL, the experiments of LR-0 with cross entropy cost function (i.e., Eq. (1) in Section VII-A) are conducted, where $C = 20$, $\lambda = 8$, $\eta_0 = 0.1$, $\eta_T = \eta_0/(\rho * T + 1)$, and the server is honest-but-curious. Fig. 6 (a) shows the variation of training loss during training. The training loss decreases with the number of training rounds and gradually tends to be constant. In this experiment, the threshold of the difference between two adjacent training losses is 10^{-5}. When the difference is less than 10^{-5}, the model training converges.



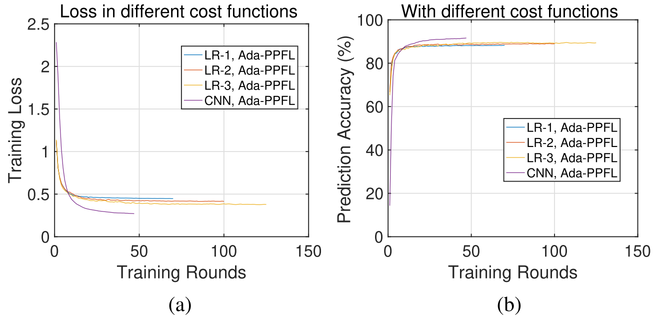
(a) (b)

Fig. 5. (a) The prediction accuracy of Ada-PPFL with different conversion ratio λ for the iid MNIST dataset. (b) The prediction accuracy of Ada-PPFL with conversion ratio λ for the non-iid MNIST dataset.



(a) (b)

Fig. 6. (a) The training loss of LR-0 in three different training scenarios. (b) The prediction accuracy of LR-0 in three different training scenarios.



(a) (b)

Fig. 7. (a) The training loss for tasks with four different cost functions. (b) The prediction accuracy for tasks with four different cost functions.

In Fig. 6 (b), the schemes, i.e., the training with fixed learning rate and no attacks, the training with fixed learning rate and privacy protection, and Ada-PPFL with LR-0, converge in the 58-th, 99-th, and 83-rd rounds of training, respectively.

Convergence in different learning tasks: We examine our theoretical convergence results on the LR-1 and LR-2 task. The learning rate of LR-1 is $\eta_T = \eta_0/(\rho * T + 1)$ and the learning rate of LR-2 is $\eta_T = \eta_0/(\rho * \sqrt{T} + 1)$. As shown in Fig. 7 (a), the training losses of LR-1 and LR-2 gradually tend to be constants in 70-th and 100-th training rounds, respectively. The prediction accuracy shown in Fig. 7 (b) also tends to converge as the number of training rounds increases. Thus, the convergence analysis of Ada-PPFL in Section VI are verified.

To show the convergence results of Ada-PPFL are generally applicable, the experiments of LR-3 and CNN task (with cross entropy cost function) are conducted. The cross entropy cost function in CNN task is non-convex. The convergence results are depicted in Fig. 7 (a) and (b). Ada-PPFL converges

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

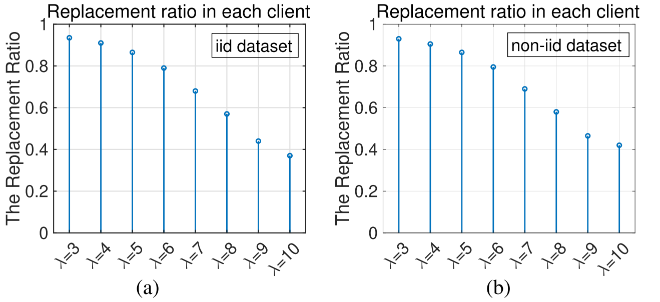


Fig. 8. (a) The client's replacement ratio under different λ for iid dataset. (b) The client's replacement ratio under different λ for non-iid dataset.

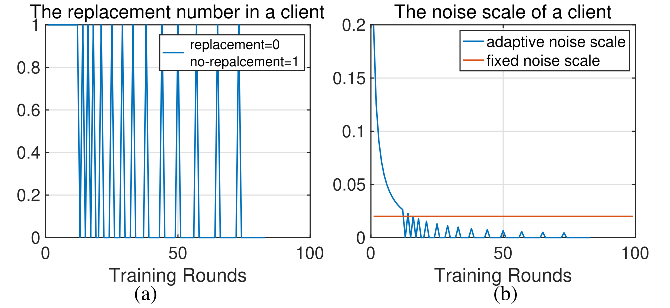


Fig. 9. (a) The number of the replacements on a client in Ada-PPFL. (b) The noise scale on clients in Ada-PPFL vs. the fixed noise scale.

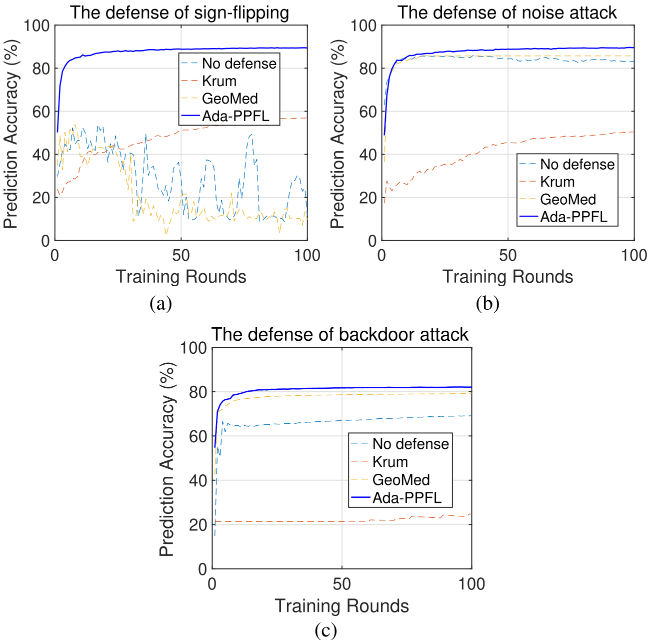


Fig. 10. (a) The prediction accuracy under sign-flipping attack. (b) The prediction accuracy under noise attack. (c) The prediction accuracy under backdoor attack.

at 125 and 47 training rounds in LR-3 and CNN tasks, respectively.

In summary, our Ada-PPFL achieves good convergence results in the common LR and CNN tasks and can adapt to different types of loss functions.

D. Privacy Protection of Ada-PPFL

In this section, we simulate the privacy budget allocated in Ada-PPFL to achieve client-level privacy protection and use privacy protection degree to measure the privacy of client data.

Privacy protection degree: The degree of privacy protection is measured by the total privacy budget during training. Fig. 8 (a) and Fig. 8 (b) show the relations between replacement ratio and parameter λ. When $\lambda = 8$, there is 35% of model updates kept locally. If the privacy budget is fixed in each training round, a 35% reduction in the total privacy budget allocated to each client, meaning a 35% increase in privacy protection guarantee.

The results of Fig. 6 show that the number of training rounds required for convergence of different schemes is also different. The replacements of each training round before convergence are shown in Fig. 9 (a), where 1 means the replacements occurred and 0 means no replacements. In the experiment, the number of 1 is equal to 25, which is far less than the number 58 in the scheme with fixed learning rate. Thus, the total privacy budget allocated to each client in Ada-PPFL is less than that of other schemes, meaning stronger privacy protection.

Besides, Fig. 9 (b) shows the noise scale added by Ada-PPFL and the scheme with fixed noise scale to each client that is usually performed by previous work. Total noise scale in Ada-PPFL with LR-0 and in the model training with fixed learning rate and protection are 0.9469 and 1.9800, respectively. That is, Ada-PPFL requires less noise to achieve privacy protection, which also ensures high prediction accuracy.

E. Attack Defense of Ada-PPFL

The defense performances of Ada-PPFL are evaluated by the prediction accuracy, the detection precision, and the prediction accuracy difference under three baseline attacks.

Ada-Pplf vs. Byzantine-Resilient Aggregation Schemes: In order to test the defense performances of the proposed Ada-PPFL, the prediction accuracy experiments are conducted on three baseline attacks in the non-fully trusted FL. Besides, the proposed Ada-PPFL is compared to two common Byzantine-resilient aggregation schemes, i.e., Krum [18] and GeoMed [19], which are often used to defend against Byzantine clients. The experimental results are shown in Fig. 10, where our Ada-PPFL can achieve high prediction accuracy under different malicious attacks and is better than Krum and GeoMed schemes. That is, our Ada-PPFL can not only defend three baseline attacks but also ensure high prediction accuracy.

Three Baseline Attacks Detection: In Ada-PPFL, the DP-protected model updates of clients are mapped to 2-dimensional space by multi-dimensional scaling (MDS) algorithm. Then, we use DPAD to detect the malicious model updates. In the experiment of detection precision, the percentage of malicious model updates is set as 30% in the sign-flipping attack and the additive noise attack. The clients with label "7" (MNIST) label "frog" (CIFAR-10) are malicious in the backdoor attack.

The experimental results on MNIST dataset are shown in Fig. 11 and Fig. 12. Fig. 11 (a) and Fig. 12 (a) show the detection precision of Ada-PPFL for the three baseline malicious attacks in LR and CNN tasks. The detection precision

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

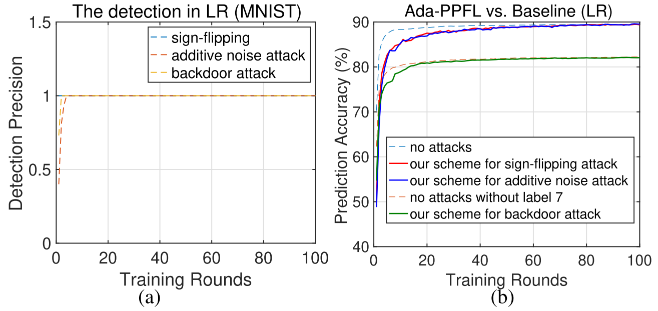


Fig. 11. (a) The detection precision of each round for three baseline attacks. (b) The prediction accuracy after Ada-PPFL.

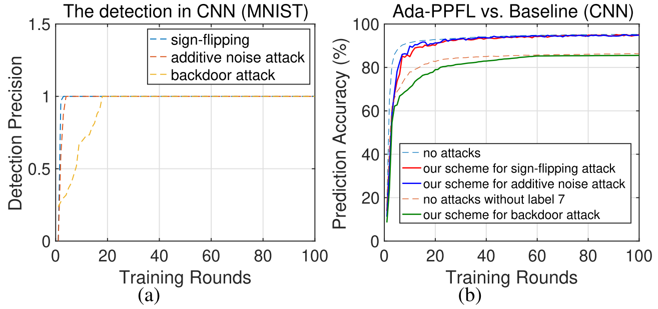


Fig. 12. (a) The detection precision of each round for three baseline attacks. (b) The prediction accuracy after Ada-PPFL.

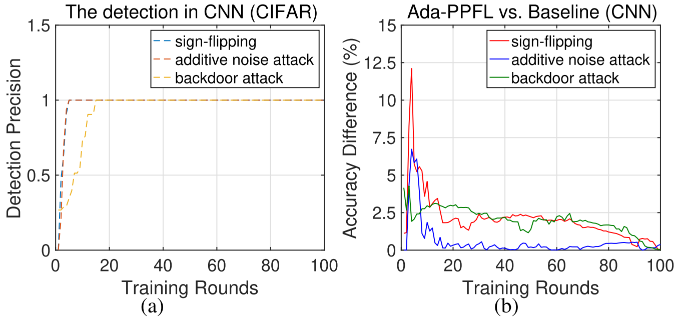


Fig. 13. (a) The detection precision of each round for three baseline attacks. (b) The prediction accuracy difference in Ada-PPFL.

of Ada-PPFL goes to 1 as training progresses, which indicates that the proposed Ada-PPFL has good detection capability for baseline attacks. Besides, Fig. 11 (b) and Fig. 12 (b) show that the prediction accuracy of Ada-PPFL is very close to that of the training without attacks, meaning that the impact of the malicious model updates is eliminated in Ada-PPFL.

In order to verify whether Ada-PPFL is applicable to different types of data, the experiments on CIFAR-10 dataset are performed. The detection precision of Ada-PPFL for the three baseline malicious attacks in CNN task is shown in Fig. 13 (a). The detection precision of Ada-PPFL also tends to 1 as the training rounds increasing. The prediction accuracy difference between Ada-PPFL and the baseline prediction accuracy is always less than 2.5%, as shown in Fig. 13 (b).

In summary, the proposed Ada-PPFL can achieve good convergence results in different types of loss functions, and perform well in handling LR and CNN tasks. Besides, Ada-PPFL enables client-level privacy protection with 35%DP-noise saving. Then, under client-level privacy protection, Ada-PPFL can effectively defend against three baseline malicious attacks with high prediction accuracy. Thus, our Ada-PPFL is an effective and generally applicable privacy-preserving federated learning scheme in the non-fully trusted setting.

VIII. Conclusion

In this paper, we considered a non-fully trusted federated learning where the server is honest-but-curious, and the clients are malicious. In the non-fully trusted FL, a conflict exists between privacy-preserving and outliers detecting. To solve this conflict, we proposed an adaptive privacy-preserving scheme (Ada-PPFL) in the non-fully trusted setting. On the client side of Ada-PPFL, we designed a privacy protection model based on DP and a selection criterion to support the adaptive noise addition for clients' local training. They guaranteed strong protection of the client's privacy and improved the prediction accuracy of training. Meanwhile, we adopted a DPAD algorithm on the server side to detect the DP-protected model updates, which can precisely detect and remove the malicious model updates. Finally, the theoretical analyses and experimental results further illustrate that the proposed Ada-PPFL can provide client-level privacy protection and keep high prediction accuracy in the non-fully trusted FL.

Appendix

A. Additional Notations

We first show the update method of Ada-PPFL without the replacement, and then we define some additional notations.

$$\mathbf{v}_{t+1}^{i} = \mathbf{w}_{t}^{i} - \acute{\eta}_{t} \nabla F_{i}\left(\mathbf{w}_{t}^{i}, \xi_{t}^{i}\right),$$

$$\mathbf{w}_{t+1}^{i} = \begin{cases} \mathbf{v}_{t+1}^{i}, & \text{if } t+1 \notin \mathcal{I}_{T}, \\ \displaystyle\sum_{i \in \mathcal{U}_{T}^{l}} p_{i}\left(\omega_{T-1} - \frac{\eta_{T} \hat{g}_{T}^{i}}{\alpha_{T}^{i}} + \tilde{n}_{T}^{i}\right), & \text{if } t+1 \in \mathcal{I}_{T}, \end{cases}$$

where $T \leftarrow \lceil \frac{t+1}{Eb} \rceil$ and $\acute{\eta}_{t} = \eta_{T}$ when $t \in \{(T-1)Eb, TEb\}$.

Motivated by [43], [47], we define that $\tilde{\mathbf{v}}_{t} = \sum_{i=1}^{M} p_{i} \mathbf{v}_{t}^{i}$, $\tilde{\mathbf{w}}_{t} = \sum_{i=1}^{M} p_{i} \mathbf{w}_{t}^{i}$, $\tilde{g}_{t} = \sum_{i=1}^{M} p_{i} \nabla F_{i}\left(\mathbf{w}_{t}^{i}\right)$ and $g_{t} = \sum_{i=1}^{M} p_{i} \nabla F_{i}\left(\mathbf{w}_{t}^{i}, \xi_{t}^{i}\right)$. The learning rate of each client is the same in the same round. Then, we can get that $\tilde{\mathbf{v}}_{t+1} = \tilde{\mathbf{w}}_{t} - \acute{\eta}_{t} g_{t}$ and $\mathbb{E} g_{t} = \tilde{g}_{t}$.

B. Key Lemmas

Four key lemmas can be summarized based on Assumptions 1 to 6, described as follows.

Lemma 1: Based on Assumptions 1 and 2, if $\acute{\eta}_{t} \leq \frac{1}{4L}$, it follows that

$$\mathbb{E}\left\|\tilde{\mathbf{v}}_{t+1} - \omega^{*}\right\|^{2} \leq \left(1 - \acute{\eta}_{t} \mu\right) \mathbb{E}\left\|\tilde{\mathbf{w}}_{t} - \omega^{*}\right\|^{2} + \acute{\eta}_{t}^{2} \mathbb{E}\left\|g_{t} - \tilde{g}_{t}\right\|^{2} + 6L\acute{\eta}_{t}^{2}\Gamma + 2\mathbb{E}\sum_{i=1}^{M} p_{i}\left\|\tilde{\mathbf{w}}_{t} - \mathbf{w}_{i}^{t}\right\|^{2}.$$

Lemma 2: According to Assumption 3, it follows that

$$\mathbb{E}\left\|g_{t} - \tilde{g}_{t}\right\|^{2} \leq \sum_{i=1}^{M} p_{i}^{2} \sigma_{i}^{2}.$$

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

Lemma 3: If Assumption 4 holds, we have

$$\mathbb{E}\left[\sum_{i=1}^{M} p_i \left\|\tilde{\mathbf{w}}_t - \mathbf{w}_i^t\right\|^2\right] \leq \acute{\eta}_t^2 (Eb-1)^2 G^2,$$

where $\acute{\eta}_t$ is non-increasing and $\acute{\eta}_t \leq \eta_{t+Eb}$ for all $t \geq 0$

Lemma 4: If Assumptions 5 and 6 hold, we have

$$\mathbb{E}\left\|\tilde{\mathbf{v}}_{t+1} - \omega^*\right\|^2 \leq \mathbb{E}||\tilde{\mathbf{w}}_t - \omega^*||^2 - 2\acute{\eta}_t \mathbb{E}(F(\tilde{\mathbf{w}}_t) - F^*) + (Eb+4)\acute{\eta}_t^2 L'^2.$$

For the proof of these lemmas 1 to 3, please refer to Section .4 of [43]. We will not repeat them here.

C. Proof of Lemma 4

Proof: Notice that $\tilde{\mathbf{v}}_{t+1} = \tilde{\mathbf{w}}_t - \acute{\eta}_t g_t$, then

$$\begin{aligned} ||\tilde{\mathbf{v}}_{t+1} - \omega^*||^2 =& ||\tilde{\mathbf{w}}_t - \acute{\eta}_t g_t - \omega^* - \acute{\eta}_t \tilde{g}_t + \acute{\eta}_t \tilde{g}_t||^2 \\ =& ||\tilde{\mathbf{w}}_t - \acute{\eta}_t \tilde{g}_t - \omega^*||^2 + \acute{\eta}_t^2 ||g_t - \tilde{g}_t||^2 + \\ & 2\acute{\eta}_t \langle \tilde{\mathbf{w}}_t - \acute{\eta}_t \tilde{g}_t - \omega^*, \tilde{g}_t - g_t \rangle. \end{aligned}$$

Assume $A_1 = ||\tilde{\mathbf{w}}_t - \acute{\eta}_t \tilde{g}_t - \omega^*||^2$, then we have

$$\begin{aligned} A_1 &= ||\tilde{\mathbf{w}}_t - \omega^*||^2 - 2\acute{\eta}_t \langle \tilde{\mathbf{w}}_t - \omega^*, \tilde{g}_t \rangle + \acute{\eta}_t^2 ||\tilde{g}_t||^2 \\ &\leq ||\tilde{\mathbf{w}}_t - \omega^*||^2 - 2\acute{\eta}_t \sum_{i=1}^{M} p_i \langle \tilde{\mathbf{w}}_t - \mathbf{w}_t^i, \nabla F_i\left(\mathbf{w}_t^i\right) \rangle \\ &\quad - 2\acute{\eta}_t \sum_{i=1}^{M} p_i \langle \tilde{\mathbf{w}}_t - \omega^*, \nabla F_i\left(\mathbf{w}_t^i\right) \rangle. \end{aligned}$$

By Cauchy-Schwarz and AM-GM inequalities, we have

$$-2\acute{\eta}_t \langle \tilde{\mathbf{w}}_t - \mathbf{w}_t^i, \nabla F_i\left(\mathbf{w}_t^i\right) \rangle \leq ||\tilde{\mathbf{w}}_t - \mathbf{w}_t^i||^2 + \acute{\eta}_t^2 ||\nabla F_i\left(\mathbf{w}_t^i\right)||^2. \tag{5}$$

By the convexity of $F_i(\cdot)$ (i.e., Assumption 5) and L'-Lipschitz bounded (i.e., Assumption 6), we have

$$-2\acute{\eta}_t \langle \tilde{\mathbf{w}}_t - \omega^*, \nabla F_i\left(\mathbf{w}_t^i\right) \rangle \leq -2\acute{\eta}_t (F_i(\mathbf{w}_t^i) - F_i(\omega^*)), \tag{6}$$

$$\acute{\eta}_t^2 ||\tilde{g}_t||^2 \leq \acute{\eta}_t^2 \sum_{i=1}^{M} p_i ||\nabla F_i\left(\mathbf{w}_t^i\right)||^2 \leq \acute{\eta}_t^2 L'^2, \tag{7}$$

$$\mathbb{E}||\tilde{g}_t - g_t||^2 = \sum_{i=1}^{M} p_i^2 \mathbb{E}||\nabla F_i\left(\mathbf{w}_t^i, \xi_t^i\right) - \nabla F_i\left(\mathbf{w}_t^i\right)||^2 \leq 2L'^2, \tag{8}$$

$$\mathbb{E}\sum_{i=1}^{M} p_i ||\tilde{\mathbf{w}}_t - \mathbf{w}_t^i||^2 \leq Eb\acute{\eta}_t^2 L'^2. \tag{9}$$

More details can refer to the proof of Lemma 3 in [43]. By combining Eqs. (1), (2) and (3), it follows that

$$\begin{aligned} A_1 &= ||\tilde{\mathbf{w}}_t - \acute{\eta}_t \tilde{g}_t - \omega^*||^2 \\ &\leq ||\tilde{\mathbf{w}}_t - \omega^*||^2 + \sum_{i=1}^{M} p_i ||\tilde{\mathbf{w}}_t - \mathbf{w}_t^i||^2 + \acute{\eta}_t^2 ||\nabla F_i\left(\mathbf{w}_t^i\right)||^2 \\ &\quad - 2\acute{\eta}_t \sum_{i=1}^{M} p_i (F_i(\mathbf{w}_t^i) - F_i(\omega^*)) + \acute{\eta}_t^2 L'^2 \\ &\leq ||\tilde{\mathbf{w}}_t - \omega^*||^2 + \sum_{i=1}^{M} p_i ||\tilde{\mathbf{w}}_t - \mathbf{w}_t^i||^2 \\ &\quad - 2\acute{\eta}_t \sum_{i=1}^{M} p_i (F_i(\mathbf{w}_t^i) - F_i(\omega^*)) + 2\acute{\eta}_t^2 L'^2. \end{aligned}$$

As $\mathbb{E}\langle \tilde{\mathbf{w}}_t - \acute{\eta}_t \tilde{g}_t - \omega^*, \tilde{g}_t - g_t \rangle = 0$, and then according to Equation (4), we have

$$\begin{aligned} &\mathbb{E}||\tilde{\mathbf{v}}_{t+1} - \omega^*||^2 = \mathbb{E}||\tilde{\mathbf{w}}_t - \acute{\eta}_t \tilde{g}_t - \omega^*||^2 + \acute{\eta}_t^2 \mathbb{E}||\tilde{g}_t - g_t||^2 \\ &\leq \mathbb{E}||\tilde{\mathbf{w}}_t - \omega^*||^2 + \mathbb{E}\sum_{i=1}^{M} p_i ||\tilde{\mathbf{w}}_t - \mathbf{w}_t^i||^2 + \acute{\eta}_t^2 \mathbb{E}||\tilde{g}_t - g_t||^2 \\ &\quad - 2\acute{\eta}_t \mathbb{E}\sum_{i=1}^{M} p_i (F_i(\mathbf{w}_t^i) - F_i(\omega^*)) + 2\acute{\eta}_t^2 L'^2 \\ &\leq \mathbb{E}||\tilde{\mathbf{w}}_t - \omega^*||^2 - 2\acute{\eta}_t \mathbb{E}(F(\tilde{\mathbf{w}}_t) - F^*) + (Eb+4)\acute{\eta}_t^2 L'^2. \end{aligned}$$

Thus, Lemma 4 holds. □

D. Proof of Theorem 2

Proof: According to the above definitions and the update of Ada-PPFL, we have

$$\tilde{\mathbf{v}}_{t+1} = \begin{cases} \tilde{\mathbf{w}}_{t+1}, & \text{if } t+1 \notin \mathcal{I}_T, \\ \tilde{\alpha}_T \tilde{\mathbf{w}}_{t+1} - (\tilde{\alpha}_T - 1)\tilde{\omega}_{T-1} + \tilde{\alpha}_T \tilde{n}_T, & \text{if } t+1 \in \mathcal{I}_T, \end{cases}$$

where $\tilde{\alpha}_T = \sum_{i=1}^{M} p_i \alpha_T^i$ and $\tilde{n}_T = \sum_{i=1}^{M} p_i n_T^i$.

According to Lemmas 1 to 3, we have

$$\mathbb{E}||\tilde{\mathbf{v}}_{t+1} - \omega^*||^2 \leq \left(1 - \acute{\eta}_t \mu\right) \mathbb{E}\left\|\tilde{\mathbf{w}}_t - \omega^*\right\|^2 + \acute{\eta}_t^2 J,$$

where $J = \sum_{i=1}^{M} p_i^2 \acute{\sigma}_i^2 + 6L\Gamma + 2(Eb-1)^2 G^2$.

In this paper, we focus on the convergence of the global model, and the situation of $t+1 \in \mathcal{I}_T$ is analyzed, where $t+1$ is represented by T. For convenience, let $\acute{\Delta}_{t+1} = \mathbb{E}\left\|\tilde{\mathbf{w}}_{t+1} - \omega^*\right\|^2$, and it follows that

$$\begin{aligned} &\mathbb{E}||\tilde{\mathbf{v}}_{t+1} - \omega^*||^2 \\ &\quad = \mathbb{E}||\tilde{\alpha}_T \tilde{\mathbf{w}}_{t+1} - (\tilde{\alpha}_T - 1)\tilde{\omega}_{T-1} + \tilde{\alpha}_T \tilde{n}_T - \omega^*||^2 \\ &\quad = \mathbb{E}||\tilde{\alpha}_T (\tilde{\mathbf{w}}_{t+1} - \omega^*) - (\tilde{\alpha}_T - 1)(\tilde{\omega}_{T-1} - \omega^*) + \tilde{\alpha}_T \tilde{n}_T||^2 \\ &\quad \leq \left(1 - \acute{\eta}_t \mu\right) \acute{\Delta}_t + \acute{\eta}_t^2 J, \end{aligned}$$

and

$$\begin{aligned} &\mathbb{E}||\tilde{\alpha}_T (\tilde{\mathbf{w}}_{t+1} - \omega^*)||^2 = \tilde{\alpha}_T^2 \acute{\Delta}_{t+1} \\ &\leq \left(1 - \acute{\eta}_t \mu\right) \acute{\Delta}_t + \acute{\eta}_t^2 J + \mathbb{E}||(\tilde{\alpha}_T - 1)(\tilde{\omega}_{T-1} - \omega^*) + \tilde{\alpha}_T \tilde{n}_T||^2. \end{aligned}$$

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

According to $\acute{\Delta}_t \leq \left(1-\acute{\eta}_t\mu\right)\acute{\Delta}_{t-1}+\acute{\eta}_t^2 J$, $\acute{\eta}_t < 1/\mu$ and $\tilde{\alpha}_T^2 > 1$. $\Delta_T = \acute{\Delta}_{t+1}$, and we have

$$\begin{aligned}\acute{\Delta}_{t+1} \leq & \frac{\left(1-\acute{\eta}_t\mu\right)\acute{\Delta}_{t+1-Eb}}{\tilde{\alpha}_T^2}+\frac{Eb\acute{\eta}_t^2 J}{\tilde{\alpha}_T^2}+\frac{(\tilde{\alpha}_T-1)^2\Delta_{T-1}}{\tilde{\alpha}_T^2}\\ &+\mathbb{E}||\tilde{n}_T||^2\\ \leq & \left(1-\frac{\acute{\eta}_t\mu}{\tilde{\alpha}_T^2}\right)\Delta_{T-1}+Eb\acute{\eta}_t^2 J+\mathbb{E}||\tilde{n}_T||^2.\end{aligned}$$

For a diminishing stepsize, $\eta_T = \frac{\beta}{T+\gamma}$ for some $\beta > \frac{1}{\mu}$ and $\gamma > 0$. Since every $\tilde{n}_T^i \sim \mathcal{N}(0, I\sigma_T^2)$ when $t+1 \in \mathcal{I}_T$, we have $\mathbb{E}||\tilde{n}_T||^2 \approx \sigma_T^2 = 4z^2C^2\eta_T^2$. We next prove that $\Delta_T \leq \frac{v}{\gamma+T}$, where $v = \max\left((EbJ\beta^2+4z^2C^2)/(\beta\mu/\alpha_T^2-1), (\gamma+1)\Delta_1\right)$.

$$\begin{aligned}\Delta_T \leq & \left(1-\frac{\eta_T\mu}{\tilde{\alpha}_T^2}\right)\Delta_{T-1}+Eb\eta_T^2 J+\mathbb{E}||\tilde{n}_T||^2\\ = & \frac{T+\gamma-2}{(T+\gamma-1)^2}v+\left[\frac{EbJ\beta^2+4z^2C^2}{(T+\gamma-1)^2}-\frac{\beta\mu/\alpha_T^2-1}{(T+\gamma-1)^2}v\right]\\ \leq & \frac{v}{T+\gamma}.\end{aligned}$$

By the smooth of $F(\cdot)$, we have

$$\mathbb{E}\left[F\left(\tilde{\omega}_T\right)\right]-F^* \leq \frac{L}{2}\Delta_T \leq \frac{L}{2}\frac{v}{\gamma+T}.$$

Thus, Theorem 2 holds. □

E. Proof of Theorem 3

Proof: According to the definitions and the update of Ada-PPFL, we also have

$$\begin{aligned}&\tilde{\mathbf{w}}_{t+1}\\ &= \begin{cases}\tilde{\mathbf{v}}_{t+1}, & \text{if } t+1 \notin \mathcal{I}_T,\\ \frac{1}{\tilde{\alpha}_T}\tilde{\mathbf{v}}_{t+1}+(1-\frac{1}{\tilde{\alpha}_T})\tilde{\omega}_{T-1}-\tilde{n}_T, & \text{if } t+1 \in \mathcal{I}_T,\end{cases}\end{aligned}$$

where $T \leftarrow \lceil\frac{t+1}{Eb}\rceil$, $\tilde{\alpha}_T = \sum_{i=1}^M p_i\alpha_T^i$ and $\tilde{n}_T = \sum_{i=1}^M p_i n_T^i$.

According to Lemma 4, it follows that

$$\begin{aligned}\sum_{t=0}^{T'}2\acute{\eta}_t\mathbb{E}(F(\tilde{\mathbf{w}}_t)-F^*) \leq & \sum_{t=0}^{T'}(Eb+4)\acute{\eta}_t^2 L'^2\\ &+\sum_{t=0}^{T'}(\mathbb{E}||\tilde{\mathbf{w}}_t-\omega^*||^2-\mathbb{E}\left\|\tilde{\mathbf{v}}_{t+1}-\omega^*\right\|^2).\end{aligned}$$

If $t_k = kEb \in \mathcal{I}_T$, where $k = \lceil\frac{t}{Eb}\rceil$. Let $k' = \lfloor\frac{T'}{Eb}\rfloor$, and then we have

$$\begin{aligned}&\sum_{t=0}^{T'}(\mathbb{E}||\tilde{\mathbf{w}}_t-\omega^*||^2-\mathbb{E}\left\|\tilde{\mathbf{v}}_{t+1}-\omega^*\right\|^2)\\ &= \mathbb{E}||\tilde{\omega_0}-\omega^*||^2+(\mathbb{E}||\tilde{\mathbf{w}}_{t_1}-\omega^*||^2-\mathbb{E}||\tilde{\mathbf{v}}_{t_1}-\omega^*||^2)+\cdots\\ &\quad+(\mathbb{E}||\tilde{\mathbf{w}}_{t_{k'}}-\omega^*||^2-\mathbb{E}||\tilde{\mathbf{v}}_{t_{k'}}-\omega^*||^2)-\mathbb{E}||\tilde{\mathbf{v}}_{T'}-\omega^*||^2\\ &\leq \mathbb{E}||\tilde{\omega_0}-\omega^*||^2+(\mathbb{E}||\tilde{\mathbf{w}}_{t_1}-\omega^*||^2-\mathbb{E}||\tilde{\mathbf{v}}_{t_1}-\omega^*||^2)+\cdots\\ &\quad+(\mathbb{E}||\tilde{\mathbf{w}}_{t_{k'}}-\omega^*||^2-\mathbb{E}||\tilde{\mathbf{v}}_{t_{k'}}-\omega^*||^2).\end{aligned}$$

At the global round t_k, it has

$$\begin{aligned}&\mathbb{E}||\tilde{\mathbf{w}}_{t_k}-\omega^*||^2-\mathbb{E}||\tilde{\mathbf{v}}_{t_k}-\omega^*||^2\\ &= \mathbb{E}||\tilde{\mathbf{v}}_{t_k}-\omega^*+(1-\frac{1}{\tilde{\alpha}_k})(\tilde{\omega}_{k-1}-\tilde{\mathbf{v}}_{t_k})-(\tilde{n}_k)||^2\\ &\quad-\mathbb{E}||\tilde{\mathbf{v}}_{t_k}-\omega^*||^2\\ &\leq (1-\frac{1}{\tilde{\alpha}_k})^2\mathbb{E}||\tilde{\omega}_{k-1}-\tilde{\mathbf{v}}_{t_k}||^2+\mathbb{E}||\tilde{n}_k||^2\\ &= (1-\frac{1}{\tilde{\alpha}_k})^2\mathbb{E}||\tilde{\alpha}_k\tilde{\omega}_{k-1}-\tilde{\alpha}_k\tilde{\omega}_i-\tilde{\alpha}_k\tilde{n}_k||^2+||\tilde{n}_k||^2.\end{aligned}$$

In Ada-PPFL, as the clipping gradient satisfies $\bar{g}_{t_k}^i < C$, we have

$$\begin{aligned}||\tilde{\omega}_k-\tilde{\omega}_{k-1}||^2 &= ||\sum_{i=1}^M p_i(\tilde{\omega}_{k-1}-\acute{\eta}_{t_k}\bar{g}_{t_k}^i)-\sum_{i=1}^M p_i\tilde{\omega}_{k-1}||^2\\ &= ||\sum_{i=1}^M p_i\acute{\eta}_{t_k}\bar{g}_{t_k}^i||^2 \leq \acute{\eta}_{t_k}^2 C^2.\end{aligned}$$

Since $\mathbb{E}||\tilde{n}_k||^2 \approx \sigma_k^2 = 4z^2C^2\eta_k^2$, we have

$$\begin{aligned}&\sum_{t=0}^{T'}(\mathbb{E}||\tilde{\mathbf{w}}_t-\omega^*||^2-\mathbb{E}\left\|\tilde{\mathbf{v}}_{t+1}-\omega^*\right\|^2)\\ &\leq \mathbb{E}||\tilde{\omega_0}-\omega^*||^2+\sum_{k=1}^{k'}C^2\eta_k^2[(\tilde{\alpha}_k-1)^2(1+4z^2)+4z^2].\end{aligned}$$

Assume that the $\mathbb{E}[F(\tilde{\mathbf{w}}_t)]$ is smaller than $\mathbb{E}[F(\tilde{\mathbf{w}}_{t'})]$ and $T' = t+x$, where $t' \in [0, T']$, $t \neq t'$ and $x \geq 0$. As $\alpha_T^i = \max(1, ||\hat{g}_T^i||_2/C)$, we have $\max(\tilde{\alpha}_k) \leq \frac{L'}{C}$, where $k \in [1, T']$ and $L' \geq C$. Besides, using the fact that $F(\tilde{\mathbf{w}}_t)-F^* > 0$ and $\alpha_T^i = \max(1, ||\hat{g}_T^i||_2/C)$, we have

$$\begin{aligned}&(\mathbb{E}[F(\tilde{\mathbf{w}}_T)]-F^*)\sum_{t=0}^{T'}2\acute{\eta}_t \leq \sum_{t=0}^{T'}2\acute{\eta}_t\mathbb{E}(F(\tilde{\mathbf{w}}_t)-F^*)\\ &\leq \sum_{t=0}^{T'}(\mathbb{E}||\tilde{\mathbf{w}}_t-\omega^*||^2-\mathbb{E}\left\|\tilde{\mathbf{v}}_{t+1}-\omega^*\right\|^2)+\sum_{t=0}^{T'}(Eb+4)\acute{\eta}_t^2 L'^2\\ &\leq \mathbb{E}||\tilde{\omega_0}-\omega^*||^2+\sum_{k=1}^{k'}C^2\eta_k^2[(\frac{L'}{C}-1)^2(1+4z^2)+4z^2]\\ &\quad+\sum_{t=0}^{T'}(Eb+4)\acute{\eta}_t^2 L'^2,\end{aligned}$$

where $\mathbb{E}\left\|\tilde{\omega_0}-\omega^*\right\|^2 = \Delta_1$, $\psi = C^2[(\frac{L'}{C}-1)^2(1+4z^2)+4z^2]$, $\eta_k = \frac{1}{\sqrt{k}}$, and $\acute{\eta}_t = \eta_k$ when $t \in \{(k-1)Eb, kEb\}$. Then, we use the following inequalities (refer to [48])

$$\sum_{x=1}^{y}\frac{1}{\sqrt{x}} \geq \sqrt{y}, \quad \mathit{and} \quad \sum_{x=1}^{y}(\frac{1}{\sqrt{x}})^2 \leq 1+\ln(y), \tag{10}$$

where $y \geq 2$.

By Eqs. (6) and (7), we have $\sum_{t=0}^{T'}\acute{\eta}_t \geq Eb\sqrt{k'}$, $\sum_{t=0}^{T'}\acute{\eta}_t^2 \leq Eb(1+\ln(k'))$, and $\sum_{k=1}^{k'}\eta_k^2 \leq 1+\ln(k')$.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

Thus, it follows that

$$\mathbb{E}[F(\tilde{\mathbf{w}}_T)] - F^* \leq \frac{\acute{\Delta}_1}{2Eb\sqrt{k'}} + \frac{\psi(1+\ln(k'))}{2Eb\sqrt{k'}} + \frac{(Eb+4)L'^2(1+\ln(k'))}{\sqrt{k'}}.$$

As $k' = \lfloor \frac{T'}{Eb} \rfloor$ and $T' = t + x$, it satisfies that $k' \geq \frac{t+x-Eb}{Eb}$, where $t \geq Eb$ and $Eb \geq 1$, and then we have

$$\begin{aligned}
&\mathbb{E}[F(\tilde{\mathbf{w}}_T)] - F^* \\
&\leq \frac{\sqrt{Eb}(Eb+4)L'^2(1+\ln(t+x-Eb))}{\sqrt{t+x-Eb}} \\
&\quad + \frac{\acute{\Delta}_1}{2\sqrt{Eb}\sqrt{t+x-Eb}} + \frac{\psi(1+\ln(t+x-Eb))}{2\sqrt{Eb}\sqrt{t+x-Eb}} \\
&\leq \frac{\acute{\Delta}_1}{\sqrt{t-Eb}} \\
&\quad + \frac{\psi(1+\ln(t-Eb))}{\sqrt{t-Eb}} \\
&\quad + \frac{\sqrt{Eb}(Eb+4)L'^2(1+\ln(t-Eb))}{\sqrt{t-Eb}}.
\end{aligned}$$

If $t \in \mathcal{I}_T$, we also have

$$\mathbb{E}[F(\tilde{\mathbf{w}}_T)] - F^* \leq \frac{\acute{\Delta}_1}{\sqrt{T-1}} + \frac{\psi(1+\ln(T-1))}{\sqrt{T-1}} + \frac{\sqrt{Eb}(Eb+4)L'^2(1+\ln(T-1))}{\sqrt{T-1}}.$$

Thus, Theorem 3 holds. □

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[2] S. Loussaief and A. Abdelkrim, “Machine learning framework for image classification,” in *Proc. 7th Int. Conf. Sci. Electron., Technol. Inf. Telecommun. (SETIT)*, Dec. 2016, pp. 58–61.

[3] S. Gao, Y. Zhang, K. Jia, J. Lu, and Y. Zhang, “Single sample face recognition via learning deep supervised autoencoders,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2108–2118, Oct. 2015.

[4] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (ISCA)*, 2010, pp. 1045–1048.

[5] J. Konečný, B. McMahan, and D. Ramage, “Federated optimization: Distributed optimization beyond the datacenter,” 2015, *arXiv:1511.03575*.

[6] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” 2016, *arXiv:1610.02527*.

[7] C. Song, T. Ristenpart, and V. Shmatikov, “Machine learning models that remember too much,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 587–601.

[8] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning,” in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 1–15.

[9] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, “RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets,” in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, 2019, pp. 1544–1551.

[10] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, “Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks,” *IEEE Trans. Signal Process.*, vol. 68, pp. 4583–4596, 2020.

[11] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, “Analyzing federated learning through an adversarial lens,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 634–643.

[12] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2020, pp. 2938–2948.

[13] K. Bonawitz et al., “Practical secure aggregation for privacy-preserving machine learning,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.

[14] C. Fang, Y. Guo, N. Wang, and A. Ju, “Highly efficient federated learning with strong privacy preservation in cloud computing,” *Comput. Secur.*, vol. 96, Sep. 2020, Art. no. 101889.

[15] R. Shokri and V. Shmatikov, “Privacy-preserving deep learning,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2015, pp. 1310–1321.

[16] K. Wei et al., “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[17] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, “Learning differentially private recurrent language models,” pp. 1–14, 2018.

[18] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 118–128.

[19] Y. Chen, L. Su, and J. Xu, “Distributed statistical machine learning in adversarial settings: Byzantine gradient descent,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 1–25, 2017.

[20] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2018, pp. 5650–5659.

[21] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, “Learning to detect malicious clients for robust federated learning,” 2020, *arXiv:2002.00211*.

[22] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, “Privacy-enhanced federated learning against poisoning adversaries,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4574–4588, 2021.

[23] J. So, B. Güler, and A. S. Avestimehr, “Byzantine-resilient secure federated learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2168–2181, Jul. 2021.

[24] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, “Beyond inferring class representatives: User-level privacy leakage from federated learning,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 2512–2520.

[25] M. Song et al., “Analyzing user-level privacy attack against federated learning,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2430–2444, Oct. 2020.

[26] M. Ester et al., “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proc. Int. Conf. Knowl. Discovery Data Mining (KDD)*, vol. 96, 1996, pp. 226–231.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.

[28] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1322–1333.

[29] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, “Exploiting unintended feature leakage in collaborative learning,” 2018, *arXiv:1805.04049*.

[30] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[31] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, and J. Li, “A training-integrity privacy-preserving federated learning scheme with trusted execution environment,” *Inf. Sci.*, vol. 522, pp. 69–79, Jun. 2020.

[32] C. Fung, C. J. Yoon, and I. Beschastnikh, “The limitations of federated learning in Sybil settings,” in *Proc. Int. Symp. Res. Attacks, Intrusions Defenses (RAID)*, 2020, pp. 301–316.

[33] P. Kairouz et al., “Advances and open problems in federated learning,” *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

[34] C. Dwork et al., “The algorithmic foundations of differential privacy,” *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[35] C. Dwork, “Differential privacy: A survey of results,” in *Proc. Int. Conf. Theory Appl. Models Comput. (TAMC)*, 2008, pp. 1–19.

[36] M. Abadi et al., “Deep learning with differential privacy,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 308–318.

[37] K. Amin, A. Kulesza, A. Munoz, and S. Vassilvtiskii, “Bounding user contributions: A bias-variance trade-off in differential privacy,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 263–271.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.

[38] G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy, "Differentially private learning with adaptive clipping," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17455–17466.

[39] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, May 2009, pp. 371–380.

[40] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, 2006, pp. 486–503.

[41] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 1–21, Sep. 2017.

[42] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2094–2106, Sep. 2014.

[43] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–26.

[44] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[45] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018, *arXiv:1806.00582*.

[46] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" 2019, *arXiv:1911.07963*.

[47] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–19.

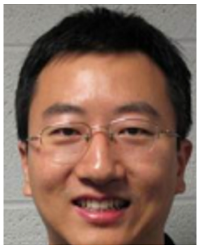
[48] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

Junqing Le (Member, IEEE) received the B.S. degree in software engineering from Southwest Jiaotong University, Chengdu, China, in 2014, and the M.S. degree in signal and information processing and the Ph.D. degree in intelligent computing and information processing from Southwest University, Chongqing, China, in 2017 and 2021, respectively. From May 2019 to May 2020, he was a Visiting Scholar with George Mason University. He is currently a Research Assistant with the College of Computer Science, Chongqing University. His current research interests include privacy protection, machine learning, and blockchain.

Di Zhang received the Ph.D. degree in intelligent computing and information processing from Southwest University, Chongqing, China, in 2021. From December 2018 to December 2020, she was a Visiting Scholar with Virginia Polytechnic Institute and State University. She is currently a Research Assistant with the College of Computer Science, Chongqing University. Her research interests include applied crypto, cloud computing security, and blockchain.

Xinyu Lei (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA, in 2021. In 2013, he was a Research Assistant with Texas A&M University at Qatar, Doha, Qatar. In 2017, he was a Research Intern with Ford Motor Company, Dearborn, MI, USA. He is currently an Assistant Professor with the Department of Computer Science, Michigan Technological University, Houghton, MI, USA. His current research interests include machine learning and cybersecurity.

Long Jiao received the Ph.D. degree from the Department of Electrical and Computer Engineering, George Mason University (GMU). In his Ph.D. study, he has published more than 20 papers in prestigious journals and conferences, including IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE WIRELESS COMMUNICATIONS, IEEE NETWORK, IEEE INFOCOM, and IEEE CNS. His research interests include cybersecurity and wireless networking with emphasis on physical layer security, cyber-physical systems/IoT security, spectrum sharing security, and machine learning application in wireless security. His research has been supported by NSF, DARPA, ARO, and Virginia Commonwealth Cyber Initiative (CCI).



Kai Zeng (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the Worcester Polytechnic Institute (WPI) in 2008. He was a Post-Doctoral Scholar with the Department of Computer Science, University of California at Davis (UCD), from 2008 to 2011. He was with the Department of Computer and Information Science, University of Michigan-Dearborn, as an Assistant Professor, from 2011 to 2014. He is currently an Associate Professor with the Department of Electrical and Computer Engineering and the Department of Computer Science, George Mason University. His current research interests include cyber-physical systems/IoT security and privacy, 5G and beyond wireless network security, network forensics, machine learning, and spectrum sharing. He was a recipient of the U.S. National Science Foundation Faculty Early Career Development (CAREER) Award in 2012, the Excellence in Post-Doctoral Research Award from UCD in 2011, and the Sigma Xi Outstanding Ph.D. Dissertation Award from WPI in 2008.

Xiaofeng Liao (Fellow, IEEE) received the B.S. and M.S. degrees in mathematics from Sichuan University, Chengdu, China, in 1986 and 1992, respectively, and the Ph.D. degree in circuits and systems from the University of Electronic Science and Technology of China, Chengdu, in 1997. From November 1997 to April 1998, he was a Research Associate with The Chinese University of Hong Kong. From October 1999 to October 2000, he was a Research Associate with the City University of Hong Kong. From 1999 to 2012, he was a Professor at Chongqing University. From March 2001 to June 2001 and from March 2002 to June 2002, he was a Senior Research Associate with the City University of Hong Kong. From March 2006 to April 2007, he was a Research Fellow with the City University of Hong Kong. He is currently a Professor and the Dean with the College of Computer Science, Chongqing University. He is also a Yangtze River Scholar of the Ministry of Education of China, Beijing, China. He holds four patents, and published four books and over 300 international journal and conference papers. His current research interests include neural networks, nonlinear dynamical systems, cryptography, and privacy protection.

Authorized licensed use limited to: Michigan Technological University. Downloaded on August 25,2023 at 01:12:25 UTC from IEEE Xplore. Restrictions apply.