2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) | 978-1-6654-7177-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDCS54860.2022.00086

Distributionally Robust Federated Learning for Differentially Private Data

Siping Shi*, Chuang Hu†, Dan Wang*, Yifei Zhu‡, Zhu Han§
* Department of Computing, The Hong Kong Polytechnic University
† Department of Computer Science, Wuhan University
‡ UM-SJTU Joint Institute, Shanghai Jiao Tong University
§ Department of Electrical and Computer Engineering, University of Houston
Email: {cssshi, csdwang}@comp.polyu.edu.hk, hchuchuang@gmail.com, yifei.zhu@sjtu.edu.cn, zhan2@uh.edu

***Abstract*—Local differential privacy (LDP) is a prominent approach and widely adopted in federated learning (FL) to preserve the privacy of local training data. It also nicely provides a rigorous privacy guarantee with computational efficiency in theory. However, a strong privacy guarantee with local differential privacy can degrade the adversarial robustness of the learned global model. To date, very few studies focus on the interplay between LDP and the adversarial robustness of federated learning. In this paper, we observe that LDP adds random noise to the data to achieve privacy guarantee of local data, and thus introduces uncertainty to the training dataset of federated learning. This leads to decreased robustness. To solve this robustness problem caused by uncertainty, we propose to leverage the promising distributionally robust optimization (DRO) modeling approach. Specifically, we first formulate a distributionally robust and private federated learning problem (DRPri). While our formulation successfully captures the uncertainty generated by the LDP, we show that it is not easily tractable. We thus transform our DRPri problem to another equivalent problem, under the Wasserstein distance-based uncertainty set, which is named the DRPri-W problem. We then design a robust and private federated learning algorithm, RPFL, to solve the DRPri-W problem. We analyze RPFL and theoretically show it satisfies differential privacy with a robustness guarantee. We evaluate algorithm RPFL by training classifiers on real-world datasets under a set of well-known attacks. Our experimental results show our algorithm RPFL can significantly improve the robustness of the trained global model under differentially private data by up to 4.33 times.**

***Index Terms*—Federated learning, local differential privacy, distributional robust optimization**

I. Introduction

Federated learning is becoming an increasingly important paradigm in many data-intensive applications ranging from prevalent personal mobile apps [1] [2] and privacy-intensive verticals, e.g., health, finance [3] [4]. In this paradigm, the local model updates instead of local private data are sent to the server for global model updating. Though the local data is kept in individual clients, the shared local model updates (i.e., gradients) may nonetheless reveal sensitive information of the local data to either a third-party or the central server [5]. For instance, an adversary can recover the training data from even a small portion of original gradients as shown in [5]. To provide privacy protection for data in local clients, local differential privacy (LDP) is widely adopted as its rigorous privacy guarantee and computational efficiency [6] [7] [8].

In LDP-protected federated learning, the data or gradient is locally perturbed before disclosing the sensitive information to the server. The perturbation is generally implemented by adding artificial noise, and the magnitude of the added noise is related to the privacy budget of each client. A smaller privacy budget indicates less privacy loss and leads to a larger magnitude of the added noise, thereby a stronger privacy guarantee. However, while LDP provides a strong privacy guarantee, the robustness of the learned model is largely decreased. Here, robustness refers to the accuracy of the learned model when encountering adversarial attacks. Particularly, empirical evaluation in [9] demonstrates that the accuracy of the model learned with LDP is greatly degraded compared to without LDP when defending against backdoor attacks. Theoretical analysis in [10] also shows that a higher level privacy guarantee with LDP leads to lower robustness of the learned global model.

To date, studies on the interplay between LDP and the adversarial robustness of federated learning are very few. We observe that LDP adds random noise to the data to achieve a local differential privacy guarantee. This random noise introduces uncertainty to the training dataset of federated learning, and thus decreases the adversarial robustness of federated learning. Since the added noise in each client is determined based on the privacy budget of each client, the uncertainty of the training dataset is highly related to the privacy budget. Therefore, we need to jointly consider the uncertainty and privacy budget to improve the adversarial robustness of federated learning.

Distributionally Robust Optimization (DRO) [11] is an increasingly popular modeling approach that can make a decision under uncertainty. The uncertain parameters are governed by any probability distribution from within an uncertainty set and a decision is sought that minimizes the worst-case cost under all possible distributions in the uncertainty set.

This work is supported by the National Key Research and Development Program of China under Grant No. 2020YFE0200500 and by GRF 15210119, 15209220, 15200321, ITF-ITSP ITS/070/19FP, CRF C5026-18G, C5018-20G, PolyU 1-ZVPZ. Yifei Zhu's work is supported by the SJTU Explore-X grant. Zhu Han's work is partially supported by NSF CNS-2107216 and CNS-2128368. The corresponding author is Chuang Hu.

2575-8411/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDCS54860.2022.00086

Recently, DRO has been explored in many areas of machine learning, like reinforcement learning [12], adversarial training [13], model generalization [14], and so on. Compared to existing heuristic robust methods in federated learning, such as GeoMed [15] computes the geometric median of all local model updates to generate the global model, DRO can provide a robustness guarantee in theory.

Intuitively, there are several reasons to apply DRO to solve the robustness problem caused by LDP. Firstly, it enables to model the uncertainty into optimization problems. Thus, the uncertainty caused by differential privacy noise in federated learning can be involved in the DRO problem. Secondly, by constructing a proper uncertainty set, DRO can provide a robustness guarantee with high confidence in the theory, thereby theoretically connecting LDP with robustness in federated learning. Thirdly, DRO problems can often be solved exactly and in polynomial time. There are no extra computations incurred in DRO problems compared to the stochastic optimization used in vanilla federated learning.

In this paper, we propose to leverage the DRO paradigm to solve the robustness problem in federated learning caused by LDP. Specifically, we first formulate a distributionally robust and private federated learning problem (DRPri). In DRPri, we need to determine the model parameter and privacy budget of each participating training client to maximize the model performance with privacy constraints. While our formulation successfully captures the uncertainty generated by the LDP, it is not easy to be solved due to the infinite-dimensional uncertainty set. Then, we transform our DRPri problem to another equivalent problem, DRPri-W, by constructing the uncertainty set with Wasserstein distance-based ball and connecting the privacy budget to the radius of the ball. Finally, we design a robust and private federated learning algorithm, RPFL, to solve the DRPri-W problem. The contributions of this work are summarized as follows:

- We propose to utilize the DRO paradigm to solve the robustness problem existed in locally differentially private federated learning, and design an algorithm RPFL to solve it with high robustness and privacy guarantee.
- With theoretical analysis, we prove the algorithm RPFL is robust and privacy-preserving with high accuracy, and the convergence rate can reach to $O\left(\frac{1}{K}\right)$, where K is the total training rounds.
- We evaluate the algorithm RPFL through training classification models on Adult and Loan datasets comprehensively. Experimental results show that RPFL improves the robustness of the learned global model under LDP by up to 4.33 times compared with the benchmarks.

The rest of this paper is organized as follows. Section II discusses the related work. Section III models the system and formulate the DRPri problem. Section IV constructs the uncertainty set and reformulates the DRPri problem. Section V introduces the proposed algorithm RPFL in detail and Section VI demonstrates theoretical analysis. And the evaluation results of RPFL based on extensive experiments are presented in Section VII. Finally, a conclusion is drawn in Section VIII.

II. Related Work

A. Federated Learning with Differential Privacy

Differential Privacy (DP) is increasingly popular in enhancing the privacy of federated learning. The implementation of differential privacy in federated learning can be classified into central differential privacy (CDP) and local differential privacy (LDP). For CDP, the local model updates are directly sent to the server without noise, and the server adds noise to the aggregated global model [6]. Obviously, CDP can only guarantee the privacy of the global model, and cannot provide protection for the privacy of local data in each client.

For LDP, each client perturbs its information locally and only sends a noisy result to the server, thereby protecting both the clients and the server against private information leakage [7] [8]. The approaches to achieve LDP can be divided by the object it perturbs: one is gradient perturbation that add noise to the local gradient or local model [7], and the other is data perturbation which directly add noise to the training data [16]. Since data perturbation can also lead to the privacy preserving in gradient or local model, we adopt it in our paper to achieve private federated learning.

B. Distributionally Robust Optimization

Distributionally robust optimization (DRO) [11] is a prominent tool that can model problems under uncertainty. It aims to find a robust solution not only performs well on a fixed problem instance (parameterized by a distribution) but simultaneously for a range of problems, each determined by a distribution in an uncertainty set. The construction of the uncertainty set is the key step in DRO. Particularly, there are two approaches to generate the uncertainty set. One is the moment-based method [17] which captures the uncertainty in moments of the distributions. The other is discrepancy-based methods that form the uncertainty set with all probability distributions whose discrepancy to the empirical probability distribution is sufficiently small. Various discrepancy functions have been explored to generate discrepancy-based uncertainty sets, such as Wasserstein distance metric [18], ϕ-divergence [19], etc.

Wasserstein distance-based DRO has been widely applied in machine learning [12] [13] [14] [20], as it enables to contain continuous distributions and provide tractable formulation. It constructs the uncertainty set as a Wasserstein ball in the space of probability distributions centered around the empirical distribution. Wasserstein distance-based DRO has also been explored to be applied in training models with differentially private data [20]. The work in [20] connects the privacy budget of local data with the radius of the Wasserstein ball to train a robust model. This work motivates us to utilize Wasserstein distance-based DRO to solve the robustness problem in private federated learning.

C. Federated Learning with Robustness

Robustness in federated learning mainly refers to the capability to defend against various adversarial attacks like

backdoor attack [21] and poisoning attacks [22] during the training time. Many prior efforts have been done for defending against these attacks. Specifically, for backdoor attack, norm bounding is proposed in [23] to defend by simply ignoring the local model updates in which the norm is out of the threshold. For poisoning attacks, most of existing defense schemes are statistics based methods. Representative examples include Trimmed mean [24] uses trimmed average value as the agregated model, and GeoMed [15] computes the geometric median of all local models to update the global model.

Most of these defense methods are trying to mitigate the negative impact introduced by the adversarial attacks, and assume the normal local models or gradients are not noisy. Thus, they cannot be directly applied in federated learning with LDP scenarios. Moreover, the existing defense methods are mainly heuristic, and can not provide theoretical robustness guarantee. Therefore, more considerations are required to improve the robustness of federated learning with LDP.

III. System Modeling and Problem Formulation

We present a federated learning system with robustness and privacy guarantee. The considered system consists of a server and N clients, and performs K rounds of training. In round k, the server first distributes the global model and privacy budget to each participating client. Then each client performs robust local training with the noisy data which is perturbed based on the allocated privacy budget, and sends the updated local model to the server. Finally, the server aggregates all the received local models to update the global model for the next round of training. Since the total privacy budget in the server is limited and the privacy requirement varies in each client, we need to improve the accuracy of the learned global model under these constraints. We first introduce the privacy budget allocation model in III-A. Then, we model the local differentially private data in III-B. And we show the distributionally robust local training model in III-C. Finally, we formulate the problem in III-D.

A. Privacy Budget Allocation Model

We know sharing the model or gradients learned from the local dataset has the risk of privacy leakage. Specifically, an attacker can infer or reconstruct the training data from the learned model or gradient. To quantitatively measure the level of privacy loss for each client, some notions are required. We adopt local differential privacy as it is a useful and versatile notion of privacy for the individual client.

Definition 1. *((ϵ, δ)-Local differential privacy) A randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-local differential privacy((ϵ, δ)-LDP) if, for any two input x and x' in set $\mathcal{X}$, and for any output set $O \subseteq range(\mathcal{M})$, we have*

$$\mathbb{P}\left[\mathcal{M}(x) \in O\right] \leq exp(\epsilon) \cdot \mathbb{P}\left[\mathcal{M}(x') \in O\right] + \delta. \quad (1)$$

For (ϵ, δ)-LDP, the privacy loss is captured by ϵ which is also called *privacy budget*. Larger ϵ indicates larger privacy loss and lower privacy guarantee. In other words, $\epsilon = 0$ indicates no privacy loss, while $\epsilon = \infty$ gives no privacy guarantee. In federated learning, the data owned by each client is highly related to the performance of the learned global model. And more private information of the local data can lead to more accurate global model. However, with the privacy concern increased, the individual client is unwilling to dedicate their data or the information of the data with no rewards.

To obtain a global model with high accuracy, the server of federated learning system is required to pay for the privacy loss. The payoff for private information of data is proportional to the privacy loss of the data. Assume the unit price of privacy loss is p, and the privacy budget allocated to each client in round k is $\epsilon_{i,k}$. For the server, since the monetary budget paid for privacy loss of all clients in each round is limited to V, we have the constraint in below

$$\sum_{i=1}^{N} \epsilon_{i,k} \cdot p \leq V, \quad \forall k = 1, \ldots, K. \quad (2)$$

For each client, monetary reward can incentivize them to contribute private information of data with some degree of privacy loss. Moreover, each client i has the baseline of privacy loss, which can be denoted as b_i. We have

$$\epsilon_{i,k} \leq b_i, \quad \forall k = 1, \ldots, K. \quad (3)$$

$$\epsilon_{i,k} \geq 0, \quad \forall k = 1, \ldots, K. \quad (4)$$

Let $\epsilon_k \triangleq (\epsilon_{1,k}, \epsilon_{2,k}, \ldots, \epsilon_{N,k})$ be the privacy budget allocation strategy in round k, then $\epsilon \triangleq (\epsilon_1, \ldots, \epsilon_K)$ is a decision variable to be optimized.

B. Local Differentially Private Data Model

For each client i, the local dataset for model training is denoted as D_i with size d_i, then the set of local data for all clients is $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ with size $D = \sum_{i=1}^{N} d_i$. The local dataset D_i can be represented as the set of data and labels, and we have $D_i \triangleq \{(x_j, y_j)\}_{j=1}^{d_i}$. Note the gradient or model trained with D_i has the potential to leak the privacy of local training data, protection mechanism should be directly applied on the local data. Adding Gaussian noise to each data sample can be an effective way to provide privacy protection, and the local dataset D_i is then transformed to a noisy dataset $\tilde{D}_i \triangleq \{(\tilde{x}_j, y_j)\}_{j=1}^{d_i}$ in which $\tilde{x}_j := x_j + w_{i,k}$, where $w_{i,k} \sim N(0, \sigma_{i,k}^2)$. The standard deviation $\sigma_{i,k}$ determines the magnitude of noise which added to each local data of client i in round k, and larger $\sigma_{i,k}$ indicates larger noise.

Theoretically, a larger privacy budget means a lower privacy guarantee which leads to smaller noise. Thus the value of $\sigma_{i,k}$ is inversely proportional to the privacy budget. In round k, when the privacy budget allocated to client i is $\epsilon_{i,k}$, we can achieve $(\epsilon_{i,k}, \delta_{i,k})$-LDP after T rounds local training, if $\sigma_{i,k} = \frac{C_i(T, \delta_{i,k})}{\epsilon_{i,k}}$, where $C_i(T, d_i, \delta_{i,k})$ is a coefficient determined by T, d_i, and $\delta_{i,k}$. The explicit expression of $C(T, d_i, \delta_{i,k})$ will be introduced later in Theorem 1 of section VI-A. When the value of noise is determined, each client i first perturbs their local data with the Gaussian noise, and then trains the local model based on the noisy data $\tilde{D}_i$.

C. Distributionally Robust Local Training Model

In each round of federated learning, all participating clients perform local training with their own local dataset. The goal of local training is to obtain a model which has minimum expectation of loss on the local data distribution $\mathbb{P}_i$. Since $\mathbb{P}_i$ is not precisely known, we often rely on the known empirical training dataset $\tilde{D}_i$. However, the learned global model has low robustness due to the uncertain LDP noise in $\tilde{D}_i$.

DRO is utilized to model the uncertain LDP noise in $\tilde{D}_i$ with an *uncertainty set*. The uncertainty set consists of probability distributions generated from the observed training samples that characterized by certain known properties of $\mathbb{P}_i$. It plays a key role and implicitly defines robustness. The higher probability that the uncertainty set $\mathcal{P}_i$ contains the true underlying local data distribution $\mathbb{P}_i$, the higher robustness of the learned model. Let θ_i be the local model of client i and $\ell(\cdot)$ is the loss function. Then, we can learn the robust model by solving the distributionally robust optimization problem in

$$\hat{J}_i := \min_{\theta_i} \sup_{\mathbb{G}_i \in \mathcal{P}_i} \mathbb{E}^{\mathbb{G}_i}\{\ell(x, \theta_i, y)\}, \tag{5}$$

which is to minimize the worst-case expected loss over $\mathcal{P}_i$. In each round of federated learning, all the participating clients perform local training by solving problem (5).

D. Problem Formulation

The goal of the server is to maximize the model performance with limited privacy budget and individual privacy requirement constraints. That is to minimize the loss of the learned global model under the budget constraint, while guarantee the robustness and privacy. Given the privacy budget V of each round, the privacy loss baseline b_i of each client i, the dataset $\mathcal{D}$ of all clients, and the loss function $\ell(\cdot)$, we need to determine the privacy budget allocation strategy ϵ and the model θ with privacy constraints (2), (3) and (4). Therefore, our problem DRPri is formulated as in below,

$$\begin{aligned} \min_{\epsilon,\theta} \quad & \sum_{i=1}^{N} \frac{d_i}{D} \sup_{\mathbb{G}_{i,k} \in \mathcal{P}_{i,k}} \mathbb{E}^{\mathbb{G}_{i,k}}\{\ell(x, \theta, y)\} \\ s.t. \quad & \sum_{i=1}^{N} \epsilon_{i,k} \cdot p \le V, \quad \forall k = 1, \ldots, K, \\ & \epsilon_{i,k} \le b_i, \quad \forall k = 1, \ldots, K, \\ & \epsilon_{i,k} \ge 0, \quad \forall k = 1, \ldots, K. \end{aligned} \tag{6}$$

To solve DRPri, we first should construct the uncertainty set $\mathcal{P}_{i,k}$. A good uncertainty set should be not only rich enough to contain the true distribution $\mathbb{P}_{i,k}$ with high confidence, but also be small enough to exclude distributions which would cause overly conservative decisions. The detailed introduction of the uncertainty set construction will be shown in Section IV. Then, we need to design an effective algorithm to optimize the privacy budget allocation strategy ϵ and global model θ simultaneously. Since the model θ is derived based on the local dataset which is perturbed based on the allocated privacy budget, it is difficult to optimize them at the same time. In Section V, we demonstrate an algorithm to overcome it.

IV. Uncertainty Set Construction and Problem Reformulation

The uncertainty set is a key ingredient of DRPri problem in (6). To achieve robustness, the constructed uncertainty set $\mathcal{P}_{i,k}$ should satisfy tractability and reliability. Specifically, for tractability, the constructed uncertainty set $\mathcal{P}_{i,k}$ must be possible to solve DRPri problem efficiently. For reliability, the constructed uncertainty set $\mathcal{P}_{i,k}$ should contain the true underlying distribution $\mathbb{P}_{i,k}$ with high probability. In this section, we propose to use the Wasserstein distance metric to construct $\mathcal{P}_{i,k}$ as a ball to guarantee tractability and reliability in Section IV-A. Moreover, since the inner supreme problem in (6) requires to compute with infinite distributions, tractable reformulation is demonstrated in Section IV-B.

A. Wasserstein Distance based Uncertainty Set

In practice, the uncertainty set $\mathcal{P}_{i,k}$ is typically chosen as a ball of radius $\rho_{i,k}$ around the empirical distribution $\tilde{\mathbb{D}}_{i,k}$ which represents the local noisy dataset $\tilde{D}_{i,k}$, that is the distance between each distribution and the empirical distribution is no more than $\rho_{i,k}$. The choice of distance metric is significant and determines how large $\rho_{i,k}$ must be. In this paper, we choose the widely used Wasserstein distance as the distance metric and the definition is in below.

Definition 2. *(Wasserstein Distance) Let $\mathcal{M}(\Xi^2)$ be the set of probability distributions on $\Xi \times \Xi$. For two probability distribution $\mathbb{P}_1, \mathbb{P}_2$ in the same probability space Ξ, we define the r-Wasserstein distance between them as follows:*

$$\mathrm{W}_r\left(\mathbb{P}_1, \mathbb{P}_2\right) := \left\{ \inf_{\pi \in \Pi(\mathbb{P}_1, \mathbb{P}_2)} \int_{\Xi \times \Xi} d^r\left(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2\right) \mathrm{d}\pi\left(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2\right) \right\}^{1/r} \tag{7}$$

where $d\left(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2\right)$ is an arbitrary norm function and $\Pi\left(\mathbb{P}_1, \mathbb{P}_2\right)$ denotes the set of all probability measures on $\Xi \times \Xi$ with marginal probabilities $\mathbb{P}_1$ and $\mathbb{P}_2$ on $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$, respectively.

Then, with 1-Wasserstein Distance, the uncertainty set $\mathcal{P}_{i,k}$ is defined as $\mathcal{P}_{i,k} = \left\{\mathbb{G}_{i,k} : \mathrm{W}_1\left(\mathbb{G}_{i,k}, \tilde{\mathbb{D}}_{i,k}\right) \le \rho_{i,k}\right\}$. Obviously, Definition 2 presents high computational complexity, as it requires to calculate the double integral under the joint distribution. To reduce the computational complexity, we can use an alternative approach according to [25] to calculate 1-Wasserstein distance as in the following lemma.

Lemma 1. *(Kantorovich duality [25]). The 1-Wasserstein distance admits a reformulation due to Kantorovich's duality:*

$$W_1\left(\mathbb{P}_1, \mathbb{P}_2\right) = \sup_{f \in \mathbb{L}} \left\{ \int_{\Xi} f\left(\boldsymbol{\xi}\right) \mathbb{P}_1\left(\mathrm{d}\boldsymbol{\xi}\right) - \int_{\Xi} f\left(\boldsymbol{\xi}\right) \mathbb{P}_2\left(\mathrm{d}\boldsymbol{\xi}\right) \right\}, \tag{8}$$

where, $\mathbb{L}$ is the set of all Lipschitz functions with Lipschitz constant upper bounded by one.

We can see the Kantorovich dual representation demonstrates 1-Wasserstein distance can be calculated only by two single integrals with respect to marginal distributions $\mathbb{P}_1$ and $\mathbb{P}_2$. It has much lower computational complexity compared to the original Wasserstein distance defined in Definition 2. As

mentioned before, we need to choose a proper $\rho_{i,k}$ to enable the constructed uncertainty set $\mathcal{P}_{i,k}$ contain the underlying distribution $\mathbb{P}_i$ with $1-\gamma$ confidence level. Before computing $\rho_{i,k}$, we assume the underlying distribution $\mathbb{P}_i$ is light-tailed.

Assumption 1. *(Light-tailed distribution). Suppose the distribution P is light-tailed. That is, $\mathrm{E}_{\boldsymbol{\xi}\sim P}\left[\exp\left(\|\boldsymbol{\xi}\|^a\right)\right] < \infty$ for some $a > 1$.*

With Assumption 1, we can compute the radius $\rho_{i,k}$ based on the measure concentration theorem proposed in [26], which characterizes the rate at which the empirical distribution $\tilde{\mathbb{D}}$ converges to the true distribution $\mathbb{P}^*$ in the sense of the Wasserstein distance. The measure concentration is described in the following lemma.

Lemma 2. *(Measure concentration [26]). Suppose Assumption 1 hold for $\mathbb{P}^*$, then we have*

$$\mathbb{P}^N\left(W_1\left(\mathbb{P}^*,\tilde{\mathbb{D}}\right)\geq\rho\right)\leq\begin{cases}c_1\exp(-c_2N\rho^{\max(d,2)}), & \text{if} \quad \rho\leq 1,\\ c_1\exp(-c_2N\rho^{a}), & \text{if} \quad \rho>1,\end{cases} \tag{9}$$

for all $N \geq 1$, $d \neq 2$, and $\rho > 0$, where N is the size of the observed training dataset, d is the dimension of the data sample, a is defined in Assumption 1, and c_1, c_2 are two positive constants only depending on a and d.

According to Lemma 2 and Theorem 3 of [20], the radius $\rho_{i,k}$ of Wasserstein ball can be computed in

$$\rho_{i,k} = \eta_i + \sigma_{i,k}, \tag{10}$$

where

$$\eta_i=\begin{cases}\left(\frac{\log(c_1/\gamma)}{c_2 d_i}\right)^{1/\max\{p,2\}}, & d_i\geq\frac{\log(c_1/\gamma)}{c_2},\\ \left(\frac{\log(c_1/\gamma)}{c_2 d_i}\right)^{1/a}, & d_i<\frac{\log(c_1/\gamma)}{c_2},\end{cases} \tag{11}$$

for p equals to the total dimensions of data sample (x_j, y_j). Therefore, with the uncertainty set $\mathcal{P}_{i,k}$ determined, the objective function of DRPri problem in (6) can be rewritten as

$$\min_{\epsilon,\theta}\quad\sum_{i=1}^{N}\frac{d_i}{D}\sup_{\mathbb{G}_{i,k}:\mathrm{W}_1\left(\mathbb{G}_{i,k},\tilde{\mathbb{D}}_i\right)\leq\rho_{i,k}}\mathbb{E}^{\mathbb{G}_{i,k}}\{\ell(x,\theta,y)\}. \tag{12}$$

The inner problem in (12) requires to take a supremum over the probability density function, which is an infinite-dimensional optimization problem. It is challenging to solve this problem by computing on all possible distributions directly. We need to relax the inner supremum problem and derive a more tractable reformulation.

B. Tractable Reformulation

The inner worst-case expectation problem is the key to reformulate the problem (12). Specifically, we can convert the inner supremum problem of (12) to a minimization problem through Lagrange duality approach as in [18]. However, the reformulated minimization is still complex as it requires to compute over the whole probability space Ξ. To avoid this, We can simplify the distributionally robust optimization problem by finding an upper bound for the worst-case expectation problem. We consider the loss function is Lipschitz continuous for convenient analysis, which is a typical setting in federated learning [6]. Thus, we have the following assumption.

Assumption 2. *The loss function ℓ is $G(\theta)$-Lipschitz continuous: for all ξ_1 and ξ_2, $|\ell(\theta,\xi_1)-\ell(\theta,\xi_1)| \leq G(\theta)\cdot\|\xi_1-\xi_2\|$.*

With the Assumption 2 and Lemma 1, we can find the upper bound of the inner supremum problem of (12) according to the following lemma.

Lemma 3. *Let Assumption 2 hold, then*

$$\sup_{\mathbb{G}:\mathrm{W}_1\left(\mathbb{G},\tilde{\mathbb{D}}\right)\leq\rho}\mathbb{E}^{\mathbb{G}}\{\ell(x,\theta,y)\}\leq\mathbb{E}^{\tilde{\mathbb{D}}}\{\ell(x,\theta,y)\}+G(\theta)\cdot\rho. \tag{13}$$

Proof. First, for the inner expectation problem, we have

$$\begin{aligned}\mathbb{E}^{\mathbb{G}}\{\ell(x,\theta,y)\}=&\mathbb{E}^{\mathbb{G}}\{\ell(x,\theta,y)\}-\mathbb{E}^{\tilde{\mathbb{D}}}\{\ell(x,\theta,y)\}\\&+\mathbb{E}^{\tilde{\mathbb{D}}}\{\ell(x,\theta,y)\}\\\leq&\left|\mathbb{E}^{\mathbb{G}}\{\ell(x,\theta,y)\}-\mathbb{E}^{\tilde{\mathbb{D}}}\{\ell(x,\theta,y)\}\right|\\&+\mathbb{E}^{\tilde{\mathbb{D}}}\{\ell(x,\theta,y)\}.\end{aligned} \tag{14}$$

Then, for the first term in the right hand side of the in the inequality (14), we compute the expectation with integral on the joint distribution on (x, y) and (x', y') with marginals $\mathbb{G}$ and $\tilde{\mathbb{D}}$. Thus,

$$\begin{aligned}&\left|\mathbb{E}^{\mathbb{G}}\{\ell(x,\theta,y)\}-\mathbb{E}^{\tilde{\mathbb{D}}}\{\ell(x,\theta,y)\}\right|\\=&\left|\int_{x,y,x',y'}(\ell(x,\theta,y)-\ell(x',\theta,y'))\times\Pi(\mathrm{d}x,\mathrm{d}y,\mathrm{d}x',\mathrm{d}y')\right|\\\leq&\int_{x,y,x',y'}|\ell(x,\theta,y)-\ell(x',\theta,y')|\times\Pi(\mathrm{d}x,\mathrm{d}y,\mathrm{d}x',\mathrm{d}y')\\\leq&\int_{x,y,x',y'}\frac{|\ell(x,\theta,y)-\ell(x',\theta,y')|}{\|(x,y)-(x',y')\|}\\&\qquad\times\|(x,y)-(x',y')\|\Pi(\mathrm{d}x,\mathrm{d}y,\mathrm{d}x',\mathrm{d}y')\\\leq&G(\theta)\cdot\rho.\end{aligned} \tag{15}$$

Combine (15) and (14), we can derive the lemma. □

From Lemma 3, we can see the inner worst-case problem can be transformed to a regularized sample-averaged problem. Therefore, problem in (12) can then be reformulated to DRPri-W problem as in below:

$$\begin{aligned}\min_{\epsilon,\theta}\quad&\sum_{i=1}^{N}\frac{d_i}{D}\mathbb{E}^{\tilde{\mathbb{D}}_{i,k}}\{\ell(x,\theta,y)\}+\rho_{i,k}\cdot G(\theta)\\s.t.\quad&\sum_{i=1}^{N}\epsilon_{i,k}\cdot p\leq V,\quad\forall k=1,\ldots,K,\\&\epsilon_{i,k}\leq b_i,\quad\forall k=1,\ldots,K,\\&\epsilon_{i,k}\geq 0,\quad\forall k=1,\ldots,K.\end{aligned} \tag{16}$$

The above minimization problem in (16) is computationally much easier to solve in comparison to the problem in (12). It no longer involves taking a supremun over the probability

density function and it does not require to solve an infinite dimensional optimization as in (16). There are a lot of methods in literature to solve the minimization problem. However, we cannot directly utilize them to solve DRPri-W problem in (16) as the two decision variables ϵ and θ are not independent. To overcome this, we design an algorithm to obtain the privacy budget allocation strategy ϵ first, and then optimize the model θ. Details are introduced in Section V.

V. Robust and Private Federated Learning Algorithm

The objective of DRPri-W problem in (16) is to minimize the expected loss through optimizing the privacy budget allocation strategy and model. Since the model is learned based on the perturbed data, we should first determine the privacy budget allocation strategy ϵ and then derive the model θ. For the privacy budget allocation strategy ϵ, we need to derive it in a federated way rather than a centralized way, since the private local data cannot be sent to the server for loss calculation. For the model θ, once the privacy budget allocated to each client is determined, we can train the model with gradient decent method on the differentially private data. In the following, we first introduce the federated privacy budget allocation mechanism in Section V-A, then we design a robust and private federated learning algorithm (RPFL) to achieve the optimal model in Section V-B.

A. Federated Privacy Budget Allocation

In our federated learning system, the model θ is learned based on the differentially private data implemented by adding designed noise. Since the noise added to the local data of client i in round k is calculated based on the allocated privacy budget $\epsilon_{i,k}$, we first optimize the privacy budget allocation strategy ϵ of DRPri-W problem. When given a determined model $\hat{\theta}$, to get an optimal ϵ, the objective function of DRPri-W problem in (16) is

$$\min_{\epsilon} \quad \sum_{i=1}^{N} \frac{d_i}{D} \cdot \left(\mathbb{E}^{\mathbb{D}_{i,k}} \{ \ell(x_j + w_{i,k}, \hat{\theta}, y_j) \} + (\eta_i + \frac{C_i}{\epsilon_{i,k}}) G(\hat{\theta}) \right). \tag{17}$$

To solve (17), the server needs to obtain the expected loss $\mathbb{E}^{\mathbb{D}_{i,k}} \{ \ell(x_j + w_{i,k}, \hat{\theta}, y_j) \}$ which is based on noisy data from all the participating clients. However, each client cannot perturb their local data before privacy budget allocation. Thus, we approximate $\mathbb{E}^{\mathbb{D}_{i,k}} \{ \ell(x_j + w_{i,k}, \hat{\theta}, y_j) \}$ with first order Taylor expansion, and we have

$$\begin{aligned} \mathbb{E}^{\mathbb{D}_{i,k}} \{ \ell(x_j + w_{i,k}, \hat{\theta}, y_j) \} \approx & \mathbb{E}^{\mathbb{D}_{i,k}} \{ \ell(x_j, \hat{\theta}, y_j) \\ & + w_{i,k} \nabla \ell(x_j, \hat{\theta}, y_j) \} \\ = & \frac{1}{d_i} \sum_{j=1}^{d_i} \ell(x_j, \hat{\theta}, y_j), \end{aligned} \tag{18}$$

as $w_{i,k} \sim N(0, \sigma_{i,k}^2)$ and the expectation of $w_{i,k} \nabla \ell(\tilde{x}_j, \hat{\theta}, \tilde{y}_j)$ equals to 0. Therefore, with the approximation in (18), each client is only required to compute the expected loss based on the original local dataset. Therefore, the privacy budget allocation problem is

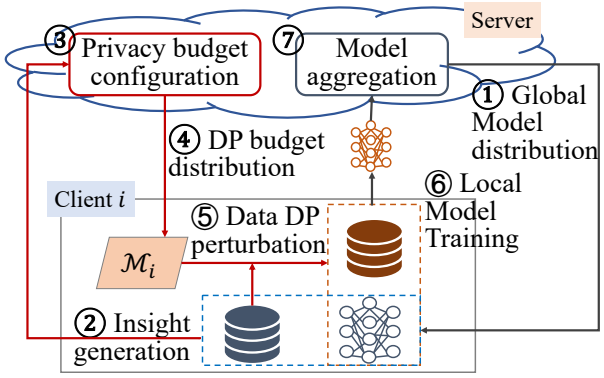


Fig. 1: Workflow of the proposed algorithm RPFL.

$$\begin{aligned} \min_{\epsilon} \quad & \sum_{i=1}^{N} \frac{d_i}{D} \left(\frac{1}{d_i} \sum_{j=1}^{d_i} \ell(x_j, \hat{\theta}, y_j) + (\eta_i + \frac{C_i}{\epsilon_{i,k}}) G(\hat{\theta}) \right) \\ s.t. \quad & \sum_{i=1}^{N} \epsilon_{i,k} \cdot p \le V, \quad \forall k = 1, \ldots, K, \\ & \epsilon_{i,k} \le b_i, \quad \forall k = 1, \ldots, K, \\ & \epsilon_{i,k} \ge 0, \quad \forall k = 1, \ldots, K. \end{aligned} \tag{19}$$

Obviously, problem in (19) is a typical minimization problem and easy to solve. In each round, each client performs analytics at first. That is each client computes the expected loss of the local dataset with the received global model $\hat{\theta}$. Then they send the analytics result to the server, and the server can derive the optimal privacy budget allocation strategy ϵ^* by solving the minimization problem in (19). Once the privacy budget allocation strategy of each round is determined, each client can perform distributionally robust local training with the allocated privacy budget.

B. Robust and Private Federated Learning Algorithm

With the privacy budget allocation strategy ϵ is determined, DRPri-W problem in (16) is reduced to a classical federated learning problem and we can apply the gradient descent method to solve it. We design a robust and private federated learning algorithm (RPFL). The overall RPFL algorithm is shown in Algorithm 1, which includes K rounds training and each training round consists of two phases: the privacy budget allocation phase and the global model updating phase. We will introduce them in detail one by one.

In the privacy budget allocation phase, the server derives and allocates a proper privacy budget for each client in a federated way. Specifically, in each global training round, the server first distributes the global model to all participating clients. Secondly, each client performs local analytics as shown in Line 7 – 9. During local analytics, each client computes the expected loss with the received global model and local dataset and sends the result to the server. Thirdly, upon receiving all the local analytics result, the server solves the convex minimization problem in (19) and obtain the optimal privacy

budget allocation strategy. Finally, the server broadcasts the determined privacy budget to all participating clients. Figure 1 also demonstrates the workflow of the privacy budget allocation phase, which is represented in the red arrow.

In the global model updating phase, the server updates the global model with the local model updates which are learned through differentially private data. Particularly, the global model updating phase includes three steps. In the first step, each client perturbs their local data with the allocated privacy budget and transforms their local dataset into a noisy dataset, which is illustrated in Line 16. In the second step, as shown in Line 18 – 20, each client computes the local model updates which are based on the gradient descent method with noisy dataset and the global model, and then sends the updates to the server. In the last step, the server aggregates all the received local model updates with FedAvg and updates the global model for the next round of training. The process of the global model updating phase is also shown in Figure 1 which is represented in the black arrow.

Algorithm 1: Robust and private federated learning algorithm (RPFL)

Input: Local dataset $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$, the objective function $\ell(\cdot)$, total privacy budget V in each round, baseline of privacy loss b_i in each client i, and the confidence level $1 - \gamma$;

Output: Learned global model θ^* and the privacy budget allocation strategy ϵ^* ;

1 $\theta \leftarrow$ `Random(`θ`)`,
2 **for** $k = 1, 2, \ldots, K$ **do**
3 // Phase I: Privacy budget allocation
4 **Central Server**:
5 Broadcast θ to each client i.
6 **Client** i:
7 $\theta_i \leftarrow \theta$,
8 $s_i \leftarrow \frac{1}{d_i} \sum_{j=1}^{d_i} \ell(x_j, \theta_i, y_j)$,
9 Send s_i to central server.
10 **Central Server**:
11 Solving (19) with the uploaded s_i to obtain ϵ_k^*,
12 Broadcast ϵ_k^* to each client i.
13 // Phase II: Global model updating
14 **Client** i:
15 **for** $(x_j, y_j) \in D_i$ **do**
16 $\tilde{x}_i \leftarrow x_j + w_{i,k}$,
17 $\tilde{y}_i \leftarrow y_j$,
18 **for** $t = 1, 2, \ldots, T$ **do**
19 $\theta_i \leftarrow \theta_i - \eta \cdot \frac{1}{d_i} \sum_{j=1}^{d_i} \nabla \ell(\tilde{x}_j, \theta_i, \tilde{y}_j)$,
20 Send θ_i to central server.
21 **Central Server**:
22 $\theta \leftarrow \sum_{i=1}^{N} \frac{d_i}{D} \cdot \theta_i$
23 **return** θ, ϵ^*

VI. Theoretical Analysis

In this section, we provide theoretical analysis to show that our proposed RPFL algorithm can guarantee privacy and utility. First, we show the privacy analysis in Section VI-A, including the privacy leakage of each training round and all training rounds. Then, we give the utility analysis in Section VI-B to show the convergence rate of algorithm RPFL.

We make the following assumptions on the loss functions. These assumptions are widely used in federated learning system. Specifically, logistics regression and softmax classifier are typical loss functions satisfied these assumptions.

Assumption 3. *ℓ is μ-strongly convex: for all θ_1 and θ_2, $\ell(\theta_1) \geq \ell(\theta_2) + (\theta_1 - \theta_2)^T \nabla \ell(\theta_2) + \frac{\mu}{2} \|\theta_1 - \theta_2\|_2^2$.*

Assumption 4. *ℓ is L-smooth: for all θ_1 and θ_2, $\ell(\theta_1) \leq \ell(\theta_2) + (\theta_1 - \theta_2)^T \nabla \ell(\theta_2) + \frac{L}{2} \|\theta_1 - \theta_2\|_2^2$.*

A. Privacy Analysis

With local data perturbed with the derived Gaussian noise, not only the privacy of data itself is protected, but also the privacy of the local model learned with the noisy data is preserved. For Algorithm 1, we first analyze the privacy loss in each round of training and then the total privacy loss of the proposed algorithm.

Privacy leakage of each training round: In each training round, the local model is updated with the local data in several epochs. Since the perturbation is performed on local data, it also affects the privacy of the local model in each round. The following theorem shows the relation between the input perturbation and the privacy of the local model.

Theorem 1. *Let Assumption 2 and 3 hold and $\epsilon_{i,k}, \delta_{i,k} > 0$ for all $i = 1, \ldots, N$ and all $k = 1, \ldots, K$. In Algorithm 1, if we have*

$$\sigma_{i,k}^2 = c \frac{G^2 T \ln(1/\delta_{i,k})}{d_i (d_i - 1) \sqrt{\mu} \epsilon_{i,k}^2}, \tag{20}$$

then the local model $\theta_{i,k}$ learned in round k satisfies $(\epsilon_{i,k}, \delta_{i,k})$-local differential privacy for some constant c.

Proof. In round k, each client i performs T epochs of gradient decent to update their local model. As in [27], we assume the loss function $\ell(x, \theta, y)$ is $\ell(y\theta^T x)$. For client i in the t-th epoch, the local model $\theta_{t+1,k}$ is

$$\theta_{t+1,k} = \theta_t - \eta \underbrace{\frac{1}{d_i} \sum_{j=1}^{d_i} y_i^j \nabla \ell \left(y_i^j \theta_t^T \left(x_i^j + w_{i,k} \right) \right) \left(x_i^j + w_{i,k} \right)}_{M_{t,i}}, \tag{21}$$

where $w_{i,k} \sim \mathcal{N}\left(0, \sigma_{i,k}^2\right)$ and η denotes the learning rate. From Equation (21), we can see only the aggregated gradients may disclose privacy of the training data, and we define it as the mechanism $M_{t,i}$.

Denote D_i and D_i' as the neighbouring dataset of the client i which only differs in the d_i-th sample (i.e., $\left(x_i^{d_i}, y_i^{d_i}\right)$ and

$\left(x_i^{d_i'}, y_i^{d_i'}\right)$). Let P_i and P_i' be the probability distributions on D_i and D_i' over mechanism $M_{t,i}$. We have:

$$P_i = A + B \cdot w_{i,k} + \underbrace{\frac{1}{d_i} y_i^{d_i} \nabla \ell \left(y_i^{d_i} \theta_t^T \left(x_i^{d_i} + w_{i,k}\right)\right) \left(x_i^{d_i} + w_{i,k}\right)}_{C}, \tag{22}$$

$$P_i' = A + B \cdot w_{i,k} + \underbrace{\frac{1}{d_i} y_i^{d_i'} \nabla \ell \left(y_i^{d_i'} \theta_t^T \left(x_i^{d_i'} + w_{i,k}\right)\right) \left(x_i^{d_i'} + w_{i,k}\right)}_{C'}. \tag{23}$$

where $A = \frac{1}{d_i} \sum_{j=1}^{d_i - 1} y_i^j \nabla \ell \left(y_i^j \theta_t^T \left(x_i^j + w_{i,k}\right)\right) x_i^j$, and $B = \frac{1}{d_i} \sum_{j=1}^{d_i - 1} y_i^j \nabla \ell \left(y_i^j \theta_t^T \left(x_i^j + w_{i,k}\right)\right)$.

Note that $w_{i,k} \sim N\left(0, \sigma_{i,k}^2\right)$, we have $P \sim N\left(A + C, B\sigma_{i,k}^2\right)$ and $P' \sim N\left(A + C', B\sigma_{i,k}^2\right)$.

To analyze the total privacy of T epochs, we utilize the moments accountant method which proposed in [28]. The λ-th moment of mechanism $M_{t,i}$ on dataset D_i and, D_i' is defined as

$$\alpha_{M_{t,i}}(\lambda) = \ln\left(\mathbb{E}_{o \sim P}\left[\left(\frac{P_i}{P_i'}\right)^{\lambda}\right]\right) = \lambda D_{\lambda+1}(P_i || P_i'), \tag{24}$$

where the second equality is derived by the Definition 2.1 in [28]. According to Lemma 2.5 in [29], Equation (24) can be transformed to

$$\alpha_{M_t}(\lambda) = \frac{\lambda(\lambda+1) \|C - C'\|^2}{2B\sigma_{i,k}^2}. \tag{25}$$

Since the loss function is G-Lipschitz according to Assumption 2, we have:

$$\|C - C'\| \leq \frac{2G}{d_i}. \tag{26}$$

Besides, the loss function is μ-strongly convex which also satisfies the Polyak-Łojasiewicz condition [30]. Thus we have:

$$\|\nabla \ell(x, \theta, y)\|^2 \geq 2\mu \left(\ell(x, \theta, y) - \ell(x, \theta^*, y)\right), \tag{27}$$

where θ^* is the optimal model parameter. Then the lower bound of B defined in Equation (22) and Equation (23) can be derived based on (27):

$$B \geq \frac{d_i - 1}{d_i} \sqrt{2\mu \left(\ell\left(\theta_T\right) - \ell(\theta^*)\right)}, \tag{28}$$

where θ_T is the model of last epoch. Since $\ell\left(\theta_T\right) - \ell(\theta^*)$ can be considered as a constant, with (26) and (28), for some constant c_0, Equation (25) can be transferred to

$$\alpha_{M_{t,i}}(\lambda) \leq c_0 \frac{\lambda(\lambda+1)G^2}{\sqrt{\mu}\sigma_{i,k}^2 d_i(d_i - 1)}. \tag{29}$$

By summing (29) over T epochs, for constant c_1 we have:

$$\sum_{t=1}^{T} \alpha_{M_{t,i}}(\lambda) \leq c_1 \frac{\lambda^2 G^2 T}{\sqrt{\mu}\sigma_{i,k}^2 d_i(d_i - 1)}. \tag{30}$$

Let $\alpha_M(\lambda)$ be the bound of all possible $\alpha_{M_i}(\lambda)$. With Theorem 2.1 stated in [28] and (29), we have

$$\alpha_{M_i}(\lambda) \leq \sum_{t=1}^{T} \alpha_{M_{t,i}}(\lambda) \leq c_1 \frac{\lambda^2 G^2 T}{\sqrt{\mu}\sigma_{i,k}^2 d_i(d_i - 1)}. \tag{31}$$

Taking $\sigma_{i,k}^2 = c \frac{G^2 T \ln(1/\delta_{i,k})}{d_i(d_i - 1)\sqrt{\mu}\epsilon_{i,k}^2}$ for some constant c, we can guarantee

$$\alpha_{M_i}(\lambda) \leq \frac{c_1 \lambda^2 G^2 T}{\sigma_{i,k}^2 d_i(d_i - 1)\sqrt{\mu}} \leq \frac{\lambda \epsilon_{i,k}}{2}, \tag{32}$$

and as a result, we have

$$\delta_{i,k} \leq \exp\left(\frac{-\lambda \epsilon_{i,k}}{2}\right). \tag{33}$$

According Theorem 2.2 in [28], the local model $\theta_{i,k}$ of client i in k-th round learned with mechanism M_i satisfies $(\epsilon_{i,k}, \delta_{i,k})$-local differential privacy. □

Privacy leakage of all training rounds: We have proved that the local model learned in round k and client i satisfies $(\epsilon_{i,k}, \delta_{i,k})$-local differential privacy when applied Algorithm 1. Here we focus on the total privacy leakage of all the training rounds. Since Algorithm 1 is a k-fold adaptive algorithm, we can use the moments accountant method [28] to analyze the total privacy leakage.

Theorem 2. *Let Assumption 2 and 3 hold and $\epsilon_{i,k}, \delta_{i,k} > 0$ for all $i = 1, \ldots, N$ and all $k = 1, \ldots, K$. With $\sigma_{i,k}$ satisfies Equation (20), the Algorithm 1 guarantees $\left(c_0 \sqrt{K} \epsilon_{m,\tilde{k}}, \delta_{m,\tilde{k}}\right)$-differential privacy for some constant c_0, where $\epsilon_{m,\tilde{k}} = \max\{\epsilon_{i,k} | \forall i = 1, \ldots, N, \forall k = 1, \ldots, K\}$.*

Proof. In Algorithm 1, the training process ends after K rounds. We first analyze the privacy leakage in each round, and then derive the total privacy leakage of all the training round. According to Theorem 1, we know each local model guarantees $(\epsilon_{i,k}, \delta_{i,k})$-local differential privacy if $\sigma_{i,k}^2 = c \frac{G^2 T \ln(1/\delta_{i,k})}{d_i(d_i - 1)\sqrt{\mu}\epsilon_{i,k}^2}$. Since the global model in k-th round is aggregated by the local model $\theta_{i,k}$ of each client i. According to the parallel composition theorem proposed in [31], the global model in k-th round satisfies $(\epsilon_{m,k}, \delta_{m,k})$-differential privacy, where $\epsilon_{m,k} = \max\{\epsilon_{1,k}, \ldots, \epsilon_{N,k}\}$. Then, for all the training rounds, each of them satisfies $(\epsilon_{m,\tilde{k}}, \delta_{m,\tilde{k}})$-differential privacy, where $\epsilon_{m,\tilde{k}} = \max\{\epsilon_{i,k} | \forall i = 1, \ldots, N, \forall k = 1, \ldots, K\}$. Similar to formula (31), we have

$$\alpha_M(\lambda) \leq \sum_{k=1}^{K} \alpha_{M_k}(\lambda) \leq c_2 \frac{\lambda^2 G^2 T K}{\sqrt{\mu}\sigma_{m,\tilde{k}}^2 d_m(d_m - 1)}. \tag{34}$$

With Theorem 2.1 stated in [28] and assume Algorithm 1 achieves $(\hat{\epsilon}, \delta_{m,\tilde{k}})$-differential privacy, we can obtain

$$\ln(1/\delta_{m,\tilde{k}}) \leq c_2 \frac{\hat{\epsilon}^2 \ln(1/\delta_{m,\tilde{k}})}{K\epsilon_{m,\tilde{k}}^2}. \tag{35}$$

Therefore, we there exists a constant c_0 to make the total privacy loss $\hat{\epsilon}$ satisfies $\hat{\epsilon} = c_0 \sqrt{K} \epsilon_{m,\tilde{k}}$, and we can obtain Theorem 2. □

B. Utility Analysis

We define $\mathcal{U}(K)$ as the utility of the learned global model learned after K rounds. Followed with [32], the utility can be represented as the multiplicative inverse of the convergence rate, and we have $\mathcal{U}(K) = \frac{1}{\mathbb{E}[\ell(\theta^K)]-\ell(\theta^*)}$, where θ^* is the optimal model parameter that minimizes the global loss. Therefor, we can utilize the utility $\mathcal{U}(K)$ by deriving the upper bound of the convergence rate of Algorithm 1.

Theorem 3. *Let Assumption 2 to 4 hold and the value of the gradient is upper bounded with B (i.e., $\nabla\ell(\theta) \leq B$), with $\sigma_{i,k}$ satisfies Equation (20), we have*

$$\mathbb{E}\left[\ell\left(\theta^K\right)\right] - \ell^* \leq \frac{LB^2}{\mu^2 K}\left(1 + p\sigma^2\right), \tag{36}$$

where $\sigma = \max\{\sigma_{i,k}|\forall i = 1,\ldots,N, \forall k = 1,\ldots,K\}$, p is the dimension of each training data sample.

Proof. In Algorithm 1, Gaussian noise is added to each data before performing gradient decent. Then, the model parameter is updated as in below:

$$\begin{aligned}\theta^{k+1} &= \theta^k - \eta(\frac{1}{D}\sum_{j=1}^{D}\nabla\ell(x_j + w_j^k, \theta^k, y_j)) \\ &\approx \theta^k - \eta(\frac{1}{D}\sum_{j=1}^{D}\nabla\ell(x_j, \theta^k, y_j) + \frac{1}{D}\sum_{j=1}^{D}a_j^k w_j^k),\end{aligned} \tag{37}$$

where $a_j^k = \|\nabla_{x_j}\nabla\ell(x_j, \theta^k, j_j)\| \geq I$, and the approximation is derived by the first order Taylor expansion at point x_j. Assume $w_j^k \sim N(0, \sigma^2)$ and $\sigma = \max\{\sigma_{i,k}|\forall i = 1,\ldots,N, \forall k = 1,\ldots,K\}$, we have $\frac{1}{D}\sum_{j=1}^{D}a_j^k w_j^k \geq Iw_j^k = N(0, I\sigma^2)$. Then we can obtain a recurrence formula for $\|\theta^k - \theta^*\|^2$ as in below:

$$\begin{aligned}\|\theta^{k+1} - \theta^*\|^2 &= \|\theta^k - \theta^* - \eta\cdot(\frac{1}{D}\sum_{j=1}^{D}\nabla\ell(x_j + w_j^k, \theta^k, y_j))\|^2 \\ &\leq (1 - 2\eta\frac{\mu}{B})\|\theta^k - \theta^*\|^2 + \eta^2(1 + p\sigma^2).\end{aligned} \tag{38}$$

Assume $\eta = \frac{B}{\mu K}$, and according to the L-smoothness of the loss function, we can have

$$\mathbb{E}\left[\ell(\theta^K) - \ell(\theta^*)\right] \leq \frac{L}{2}\mathbb{E}\left[\|\theta^K - \theta^*\|^2\right] \leq \frac{LB^2}{\mu^2 K}\left(1 + p\sigma^2\right). \tag{39}$$

Therefore, we have Theorem 3. □

From Theorem 3, we can derive the utility $\mathcal{U}(K)$ is bound by $\frac{\mu^2 K}{LB^2(1+p\sigma^2)}$ and the Algorithm 1 will asymptotically converge to the global optimum with the rate $O(\frac{1}{K})$, where K is the training rounds.

VII. EVALUATION

In this section, we evaluate RPFL with classification tasks on two datasets. We first introduce the setup of the evaluation in Section VII-A. Then we conduct robustness analysis, privacy analysis, and examine the impact of privacy budget in Section VII-B.

A. Evaluation Setup

Dataset: We involve two different real-world datasets in our implementation, which are Adult from the UCI Machine Learning repository[1] and Loan from kaggle[2]. Adult Dataset contains nearly 49,000 records with features like age and education, and also a binary feature indicating whether an individual earns more than $50,000. The loan dataset contains roughly 887,000 loans, and each consists of loan-related information like loan size, interest rates, loan status, and so on. Preprocessing has been done on both datasets, including encoding the categorical features with integer numbers and removing unique identifiers and irrelevant attributes. The data distribution of each client is a non-identity independent distribution which is implemented with the Dirichlet process [33].

Models: We implement two different kinds of classifiers to evaluate our proposed algorithm. One is the binary classifier trained with logistics regression (LR) and adult dataset, which is used to predict whether an adult earns more than $50,000. The other is a multi-class classifier learned through multi layer perception (MLP) model with a loan dataset, which is utilized to classify the status of the loan.

Attack Setting: We include three typical attacks to evaluate the robustness and privacy of algorithm RPFL. One is the backdoor attack [21] where the attacker injects a backdoor into the final model, and the other is a label flipping attack [34], in which the attacker can change the training labels. Both attacks can lead to the accuracy degrading of the learned model. We also implement a membership inference attack (MIA) as in [35] to evaluate the privacy leakage of algorithm RPFL. MIA aims to infer whether a record is used to train a target model or not.

Benchmarks: To evaluate the performance of the proposed algorithm, we make a comparison to several existing works. GeoMed (GM) [15] and Trimmed Mean(TM) [24] is used to defend against label flipping attack. We also utilize norm bounding (Norm) and adding Gaussian noise (Weak-DP) proposed in [23] to defend against backdoor attacks. Moreover, federated learning with LDP and CDP are used to show the privacy and utility performance of our proposed algorithm. GeoMed without LDP (GM-N) and norm bounding without LDP are used to demonstrate the negative impact introduced by LDP, and we also employ FedAvg (AVG) as the baseline.

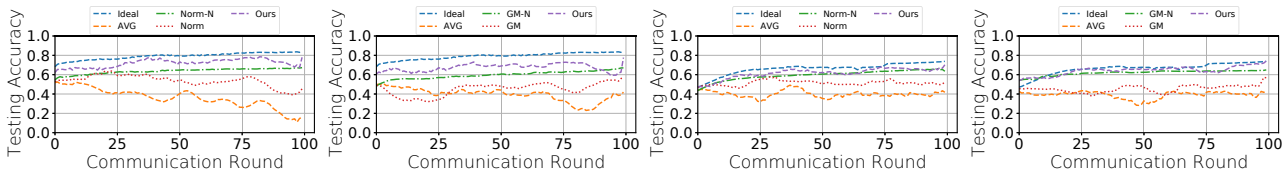
Experiment environment: Our algorithm is implemented with PyTorch and tested on a laptop equipped with an Intel Core i7 processor, 16GB memory, running with macOS. For the federated learning process, 10 clients are randomly chosen from 100 clients in each communication round, and each client executes one epoch with batch size 50.

B. Results

We evaluate RPFL from two aspects, the first aspect is macro analysis including robustness analysis and privacy

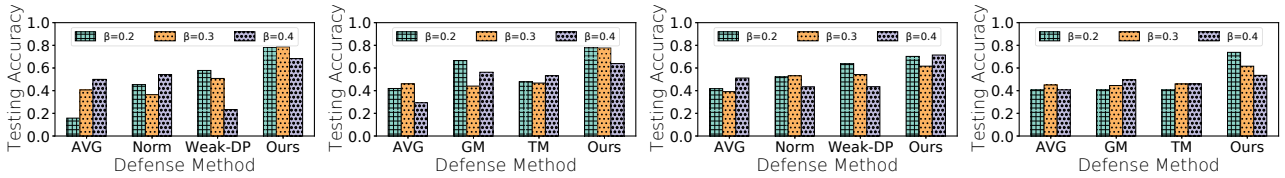
[1]https://archive.ics.uci.edu/ml/datasets/adult

[2]https://www.kaggle.com/adarshsng/lending-club-loan-data-csv



(a) Backdoor attack (LR) (b) Label flipping attack (LR) (c) Backdoor attack (MLP) (d) Label flipping attack (MLP)

Fig. 2: The testing accuracy of the learned global model when defending against attacks with different defense methods.



(a) Backdoor attack (LR) (b) Label flipping attack (LR) (c) Backdoor attack (MLP) (d) Label flipping attack (MLP)

Fig. 3: Top-1 accuracy of different defense methods when defending against attacks with various fraction of malicious clients (from 0.2 to 0.4).

analysis, and the second aspect is micro analysis including analyzing the impact of the total privacy budget. Experimental results show our method outperforms the benchmark.

Robustness Analysis: We evaluate the robustness by comparing the accuracy of the learned global model when defending against backdoor and label flipping attacks. The experimental results are shown in Figure 2. It shows that our method can defend both attacks with higher accuracy compared to the benchmarks in training LR and MLP models. Result of backdoor attack is shown in Figure 2a and 2c. We can see RPFL (ours) improves the accuracy of the learned model by 3.39 times compared to the baseline FedAvg, and 0.76 times compared to the benchmark norm bounding. Moreover, comparing the result of norm bounding with and without LDP in Figure 2a, the accuracy of the learned model is decreased by about 0.31 times, and it indicates that the introduced LDP caused degraded robustness. Similar results happen on label flipping attack as shown in Figure 2b and 2d. Therefore, RPFL can effectively defend against backdoor and label flipping attacks with higher testing accuracy.

We also explore the robustness of our method further by increasing the number of attackers. The experimental result is presented in Figure 3, and the proportion of the attackers β is from 0.2 to 0.4. We can see that our method can still achieve relatively high testing accuracy even when the number of attackers is increased to 0.4. For label flipping attack, the accuracy of GM decreased from nearly 43% to less than 20%, while the accuracy of the model learned by RPFL remains more than 75%. For backdoor attack, RPFL also performs the best, as shown in Figure 3a and Figure 3c, the testing accuracy of the learned model is improved nearly 0.4 times to 4.33 times by RPFL compared to baseline method FedAvg, and 0.21 times to 2.75 times compared to benchmarks. It validates that our algorithm can preserve accuracy under strong attacks.

Privacy Analysis: We evaluate the privacy-preserving capability of the proposed algorithm based on the membership inference attack. Table I shows the attack performance under various privacy federated learning methods. To achieve a similar privacy guarantee, the privacy budget of each training round with CDP, LDP, and RPFL is all set to 7.6. There are two key takeaways from the results. First, Our method can effectively defend against membership inference attacks. The attack accuracy of RPFL can be reduced to 48%, which is 0.71 times smaller than the baseline FedAvg which has no privacy guarantee. Second, RPFL achieves almost the best privacy guarantee compared to the benchmarks. Specifically, our proposed method outperforms CDP with an attack accuracy decreasing up to 0.33 times. Thus, the privacy of training data can be well protected with the proposed method RPFL.

TABLE I: Performance of different private federated learning methods (dataset: Loan)

Accuracy \ Methods	CDP	LDP	Ours	AVG
Attack	64%	46%	48%	82%
Classification	73%	58%	80%	86%

TABLE II: Impact of total privacy budget (dataset: Loan)

Accuracy \ Budget V	25	40	60	70	75
Attack	72%	62%	54%	42%	41%
Classification	62%	67%	75%	81%	80%

We also compare the model performance of each defense method in Table I, which is presented in the classification row. We can see that the accuracy of the learned global model is decreased in all methods compared to FedAvg, and LDP decreases the most. The underlying reason is that the DP mechanism can unavoidably have bad effects on the learned global model performance to some degree. However, RPFL can reach a higher testing accuracy to 80%, which has an improvement of 0.38 times compared to LDP. Therefore, our algorithm shows better privacy and utility trade-off than other DP-related privacy-preserving methods.

Impact of Privacy Budget: The total monetary privacy budget V in constraint (2) determines the noise scale added to each local training data. We show how the privacy budget impacts the model testing accuracy and the attack accuracy in Table II. Results show that the accuracy of the learned global model increases greatly when the privacy budget increased from 25 to 70. However, when the privacy budget keeps increasing, the increasing speed of the model accuracy becomes slow. Similar results occur in the attack accuracy, and it turns to saturate with the increasing of privacy budget. The main reason underlying is that the privacy budget $\epsilon_{i,k}$ for each client i is upper bounded, and the model performance and the attack accuracy stop changing even when the privacy budget keeps increasing. It illustrates that a proper privacy budget should be carefully chosen to obtain a better global model.

VIII. Conclusion

In this paper, we study the robustness problem in locally differentially private federated learning. We first utilize the DRO paradigm to formulate the robustness and privacy federated learning problem DRPri. Then we make a tractable reformulation on the problem DRPri with the Wasserstein distance-based uncertainty set, and the reformulated problem DRPri-W is much easier to solve compared to the general problem DRPri. Finally, we design a robust and private federated learning algorithm RPFL to solve the reformulated problem DRPri-W. The performance of the proposed algorithm RPFL is evaluated with theoretical analysis and extensive experiments. Experimental results show that RPFL can improve the accuracy of the learned model for 0.4 times to 4.33 times compared to the baseline FedAvg methods, and 0.21 times to 2.75 times compared to benchmarks. Therefore, the proposed RPFL can effectively guarantee the robustness of federated learning trained with differentially private data.

References

[1] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv*, 2018.

[2] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv*, 2019.

[3] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International Journal of Medical Informatics*, vol. 112, pp. 59–67, April 2018.

[4] H. Elayan, M. Aloqaily, and M. Guizani, "Sustainability of healthcare data analysis iot-based systems using deep federated learning," *IEEE Internet of Things Journal*, pp. 1–1, Aug. 2021.

[5] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" in *Proc. of NeurIPS'20*, virtual, Dec. 2020.

[6] K. Wei, J. Li, M. Ding *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, April 2020.

[7] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, June 2021.

[8] M. Seif, R. Tandon, and M. Li, "Wireless federated learning with local differential privacy," in *Proc. of ISIT'20*, Los Angeles, CA, June 2020.

[9] M. Naseri, J. Hayes, and E. D. Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," in *Proc. of NDSS'22*, San Diego, CA, April 2022.

[10] Y. Han, Y. Cao, and M. Yoshikawa, "Understanding the interplay between privacy and robustness in federated learning," *arXiv*, 2021.

[11] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv*, vol. abs/1908.05659, 2019.

[12] E. Smirnova, E. Dohmatob, and J. Mary, "Distributionally robust reinforcement learning," *arXiv*, vol. abs/1902.08708, 2019.

[13] A. Sinha, H. Namkoong, and J. C. Duchi, "Certifying some distributional robustness with principled adversarial training," in *Proc. of ICLR'18*, Vancouver, Canada, May 2018.

[14] M. Staib and S. Jegelka, "Distributionally robust optimization and generalization in kernel methods," in *Proc. of NeurIPS'19*, Vancouver, Canada, Dec. 2019.

[15] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. of ACM POMACS'17*, vol. 1, no. 2, pp. 1–25, December 2017.

[16] K. Fukuchi, Q. K. Tran, and J. Sakuma, "Differentially private empirical risk minimization with input perturbation," in *Proc. of DS'17*, Kyoto, Japan, Oct. 2017, pp. 82–90.

[17] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, June 2010.

[18] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1–2, p. 115–166, Sep. 2018.

[19] A. Ben-Tal, D. den Hertog, A. D. Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, Fed. 2013.

[20] F. Farokhi, "Distributionally-robust machine learning using locally differentially-private data," *Optimization Letters*, pp. 1–13, June 2021.

[21] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J. Sohn *et al.*, "Attack of the tails: Yes, you really can backdoor federated learning," in *Proc. of NeurIPS'20*, virtual, Dec. 2020.

[22] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-Robust federated learning," in *Proc. of USENIX Security'20*, virtual, Aug. 2020.

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

[24] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. ICML'18*, Stockholm, Sweden, July 2018.

[25] L. Kantorovich and G. S. Rubinstein, "On a space of totally additive functions," *Vestnik, Leningrad University*, vol. 13, no. 7, pp. 52–59, 1958.

[26] N. Fournier and A. Guillin, "On the rate of convergence in wasserstein distance of the empirical measure," *Probability Theory and Related Fields*, vol. 162, no. 3-4, pp. 707–738, 2013.

[27] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1069–1109, July 2011.

[28] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. of ACM SIGSAC'16*, Vienna, Austria, Oct. 2016.

[29] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. of TCC'16*, Beijing, China, Oct. 2016.

[30] D. Csiba and P. Richtárik, "Global convergence of arbitrary-block gradient methods for generalized polyak-{1} ojasiewicz functions," *arXiv*, 2017.

[31] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proc. of ACM SIGMOD'09*, Providence, Rhode Island, June 2009.

[32] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *Proc. of IEEE ICASSP'21*, Toronto, Canada, June 2021.

[33] H. Wang, M. Yurochkin, Y. Sun, D. S. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. of ICLR'20*, Addis Ababa, Ethiopia, April 2020.

[34] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv*, vol. abs/1808.04866, 2018.

[35] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source inference attacks in federated learning," in *Proc. of IEEE ICDM'21*, Auckland, New Zealand, Dec. 2021.