PriFairFed: A Local Differentially Private Federated Learning Algorithm for Client-Level Fairness

Chuang Hu, *Member, IEEE,* Nanxi Wu, Siping Shi, Xuan Liu, Wenhan Wu, Bing Luo, *Senior Member, IEEE,* Ye Wang, Jiawei Jiang, *Member, IEEE,* Dazhao Cheng *Senior Member, IEEE*

***Abstract*—Local Differential Privacy (LDP) is a mechanism used to protect training privacy in Federated Learning (FL) systems, typically by introducing noise to data and local models. However, in real-world distributed edge systems, the non-independent and identically distributed nature of data means that clients in FL systems experience varying sensitivities to LDP-introduced noise. This disparity leads to fairness issues, potentially discouraging marginal clients from contributing further. In this paper, we explore how to enhance client-level performance fairness under LDP conditions. We model an FL system with LDP and formulate the problem PriFair using regularization, which assigns varied noise amplitudes to clients based on federated analytics. Additionally, we develop PriFairFed, a Tikhonov regularization-based algorithm that eliminates variable dependencies and optimizes variables alternately, while also offering a theoretical privacy guarantee. We further experimented with the algorithm on a real-world system with 20 Raspberry Pi clients, showing up to a 73.2% improvement in client-level fairness compared to existing state-of-the-art approaches, while maintaining a comparable level of privacy.**

***Index Terms*—Federated Learning, Local Differential Privacy, Performance Fairness, Tikhonov Regularization.**

I. Introduction

FEDERATED LEARNING (FL) [1] is a popular learning technique that allows users to train a unified model in a distributed way. Clients have no need to exchange local data, instead, each client individually train a model on private local data in parallel. They only send model updates to the server, and the server receives the updates and aggregates them to obtain a global model. However, a potentially malicious third party can recover local information from updates sent to the server [2]. To protect private local data, Local Differential Privacy (LDP), first formalized in [3], perturbs data or gradients locally with rigorously designed noise before sending them to the central server.

However, the calibrated noise inevitably introduces errors to the model outputs and affects different groups differently due to the non-independent and identically distributed (non-IID) data in FL, introducing a *fairness issue*. Given noise determined by equal privacy budgets, groups that performed worse before using LDP suffer more performance degradation [4] thus being underrepresented. For instance, in sentiment analysis of tweets, DP disproportionately degrades accuracy for users writing in African-American English. When learning language models, DP punishes more on users with bigger vocabularies [4]. Such marginal groups are penalized in societal and economic decisions, leading to reduced future participation and contribution. Another example is the LDP mechanisms which are adopted by Google [5] and Microsoft [6] for gathering potentially sensitive client data to train models. This method allows respondents to confidentially answer questions on sensitive topics, such as illegal activities or personal preferences. However, [7] has shown that affirmative responses to sensitive questions typically require stronger privacy protection than negative ones, especially when analyzed through machine learning models. This discrepancy may lead to inconsistent performance, where some user groups receive less privacy protection, while uniformly increasing the privacy budget results in reduced model accuracy. Such imbalances highlight the urgent need to ensure robust privacy protection and consistent performance fairness across different respondent groups. Hence our key question arises: *how to design a fair FL system to mitigate LDP impacts?*

The fairness issues arising from privacy mechanisms are worthy of research but have not been fully resolved. As clients are naturally grouped by attributes in FL, *client-level performance fairness* notion emerges to measure the fairness level of an FL system, which relaxes accuracy parity [8] to **ensure a more uniform performance distribution across clients** [9]. Current approaches [9]–[12] address fairness concerns but do not take LDP into consideration.

To design a fair FL system under LDP, the inherent challenge is to allocate heterogeneous privacy budgets to different clients due to restricted information and non-IID data in FL settings. In more detail, **1) the impact of random noise on local objectives is implicit.** The optimization of the privacy budget allocation strategy needs to expose the variable of random noise. However, unlike output and gradient perturbation, the noise of input perturbation is applied to local data and indirectly affects local and global objectives. In order to analyze the noise impact, we have to capture the changes in local data distribution caused by noise, which is difficult due to non-iid local data. **2) The privacy budget and model parameters, our decision variables, can not be optimized simultaneously.** The local model is learned based on perturbed data, while clients have to perturb data with allocated privacy

Chuang Hu, Nanxi Wu, Wenhan Wu, Jiawei Jiang and Dazhao Cheng are with the School of Computer Science, Wuhan University. E-mail: {handc, nancywu, wenhanwu, jiawei.jiang, dcheng}@whu.edu.cn.

Siping Shi is with the Department of Computing, Hong Kong Polytechnic University. E-mail: cssshi@comp.polyu.edu.hk.

Xuan Liu is with the Department of Electrical and Electronic Engineering, Hong Kong Polytechnic University. E-mail: xuan18.liu@connect.polyu.hk

Bing Luo is with the Data Science Research Center, Duke Kunshan University. E-mail: bing.luo@dukekunshan.edu.cn.

Ye Wang is with the Faculty of Science and Technology, University of Macau. E-mail: wangye@um.edu.mo.

budgets, which are unknown before clients send local updates to the server for global optimization. It's critical to eliminate the mutual dependency between variables.

In this work, we formulate the problem PriFair, which aims to improve client-level performance fairness under input perturbation LDP. Inspired by the Excessive Risk Gap [13] in centralized learning, we generalize it to FL settings and make it a fairness regularization term. To tackle challenge 1, we employ Tikhonov regularization to incorporate the noise amplitude into objectives. To tackle challenge 2, we propose PriFairFed, an algorithm alternately trained to derive privacy budget allocation strategy and global model, which solves the problem in polynomial time as a quadratic programming. Experiment results show PriFairFed's effectiveness in preserving fairness and privacy.

Our contributions are summarized as follows:

• We formulate PriFair, a novel problem improving client-level fairness under LDP in FL. It provides a paradigm for the edge computing community to study fairness issues caused by DP (Section III).

• We propose a Tikhonov regularization-based algorithm to solve the problem as a quadratic programming and alternately optimize for privacy strategy and the global model, which can be generalized to other private training scenarios (Section IV).

• We provide a theoretical analysis of PriFairFed, including its privacy-preserving ability and the convergence rate which is $\mathbf{O}(\frac{1}{T})$, where T is the total training round (Section V).

• We evaluate the performance of PriFairFed in a real-world system containing 20 Raspberry Pi as clients, compared with state-of-the-art fairness methods. Experiment results show that PriFairFed reduces the variance of performance up to 73.6% while maintaining the similar privacy-preserving ability of the global model (Section VI).

The rest of this paper is organized as follows. Section II states the background on FL and reviews the related work about LDP and fairness in FL. Section III describes the system structure and formulates the optimization problem PriFair. Section IV introduces Tikhonov regularization to reformulate problem PriFair as PriFair-T and provides the details of the algorithm PriFairFed. We analyze the privacy and convergence rate of PriFairFed in Section V and present the experimental evaluation in Section VI. Finally, we come to a conclusion in Section VII.

II. Background and Related Work

A. Background on FL

FL is a classic paradigm of distributed machine learning, which keeps the training data of edge devices locally, conducting data mining and analysis in a federated manner to avoid exchanging data. Specifically, the training progress of FL is divided into two parts: the client-side local training and the server-side global aggregation. In each global round, the central server selects a subset of N clients to participate in the training and sends them the newest global model θ_t. Client i executes the local training to get model $\theta_{i,t+1}$ and sends it back. Then the server aggregates them to obtain a new global model. After T global rounds, FL gets the optimal global model [1]. The global aggregation is formulated as:

$$\theta_{t+1} = \sum\nolimits_{i=1}^{N} \frac{d_i}{D} \theta_{i,t+1} \tag{1}$$

where d_i is the data size of client i and $D = \sum_i d_i$.

B. LDP in FL

The model update phase in federated learning can lead to information leakage; therefore, we utilize the LDP paradigm to quantitatively measure the privacy loss of each client. LDP is a distributed implementation of Differential Privacy (DP) [14], which applies to FL and perturbs local information to protect it from a potentially malicious server. In our research, clients perturb local data with Gaussian noise.

We chose Gaussian noise for its ability to integrate smoothly with the Tikhonov regularization framework used in our algorithm. Gaussian noise is particularly well-suited for achieving a balance between privacy and fairness, especially in non-IID data scenarios common in FL [15]–[17]. Additionally, Gaussian noise aligns well with the relaxed differential privacy guarantees we aim for, providing better utility while maintaining sufficient privacy protection.

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

$$Pr[\mathcal{M}(x) \in \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{M}(x') \in \mathcal{O}] + \delta. \tag{2}$$

Informally, Def. 1 states that the probability of any output does not change much when a record is changed in a dataset, limiting the amount of information that the output reveals about any individual. ϵ represents privacy loss or privacy budget and a smaller ϵ indicates higher privacy guarantee. δ captures the probability of failure of the algorithm $\mathcal{M}$ to satisfy ϵ-LDP.

Some current researches on LDP in FL mainly focus on providing tighter privacy guarantees against various privacy and adversarial attacks [18]–[20]. [21] leverages distributionally robust optimization (DRO) to model the uncertainty brought by LDP noise in order to mitigate its impact on model performance when facing membership inference attacks [22] (MIA) and backdoor attacks. Others study how to better allocate privacy budgets and encourage local contribution to improve task performance and computing efficiency under LDP [23], [24]. [18] hides the contribution of clients during training to balance the trade-off between privacy loss and model performance. [25] introduced the first language model based on LDP and Long Short-Term Memory (LSTM) that sacrifices less precision. [26] presents practicable approaches to large-scale model training, loosening LDP constraints by limiting the attacker's ability to allocate smaller privacy budgets while achieving task performance close to non-private algorithms. Existing works successfully reduce the impact of LDP on task performance, but ignore the different behavior across clients. In this study, we pay more attention to fairness issues caused by LDP with a carefully designed privacy budget allocation strategy.

C. Related Work

Fairness in FL. The literature on fairness in FL can be divided into two groups: those focusing on collaborative fairness [27], [28] and performance fairness [9], [12], [29], [30]. Collaborative fairness forces clients to converge to different models aligning with their contributions. In contrast, performance fairness acquires small discrepancies in the performance across various groups. It includes, for example, individual fairness [31] and group fairness [32], [33]. The former encourages similar data points to be treated similarly, while it's hard to design the problem-specific distance metrics [34]. Group fairness requires some statistical property of protected groups to be similar to that of the whole population. However, it's hard to extend to FL as the notion of a "good" outcome for a device is not clear [35].

As clients are naturally grouped by attributes in FL, we focus on client-level performance fairness, which acquires uniform accuracy distribution across clients. Current research in federated learning (FL) highlights significant client-level performance fairness challenges, particularly in heterogeneous environments where differences in client device capabilities, data distribution, and participation behavior can lead to uneven model performance across clients. Empirical studies [36], [37] show that such heterogeneity can cause substantial drops in both accuracy and fairness, with some clients disproportionately benefiting from the global model while others experience degraded performance. Addressing these fairness issues is crucial for ensuring equitable outcomes in real-world FL systems. [12] proposed an Agnostic FL (AFL) framework, which is robust to unknown testing distribution. [38] further develop the notation of AFL and use kernel functions to reweigh training samples. [9] up-weighted clients with lower performance and their objective can be tuned based on the desired amount of fairness. They are both successful methods of improving fairness. However, they didn't consider the impact of privacy mechanisms, where the introduction of DP perturbations can degrade fairness.

Mitigation of Fairness Issues Caused by LDP. The mitigation of fairness issues caused by LDP in FL settings is an under-studied topic. Most existing works [39] that take both privacy and fairness into account mostly consider Central Differential Privacy (CDP) and focus on group fairness, enforcing statistical properties like equal odds [40] and demographic parity [8] on sensitive attributes between protected groups. They require access to sensitive attributes and are non-decomposable and non-convex [41], which are not applicable to FL settings. Moreover, they either not provide convergence guarantee [42]–[44] or have to train with full batch of data to converge [45]. By contrast, we achieve fairness from the perspective of clients, which ensures the model does not overfit any client at the expense of others. We avoid direct access to sensitive properties and theoretically provides convergence guarantee.

III. Model and Problem Formulation

A. Privacy Budget Allocation Model

As described in II-B, to prevent attackers from inferring private data from the local updates, it is necessary to quantitatively perturb client data based on LDP. The client data varies in terms of quantity, quality, and distribution, leading to disparate privacy budget requirements. According to the parallel composition theorem proposed in [46], the total privacy budget of the system is the maximum one among clients, denoted as b. In each global training round t, the privacy budget $\epsilon_{i,t}$ of client i satisfies:

$$\epsilon_{i,t} > 0, \quad \forall i \in \{N\}, \forall t \in \{T\}, \tag{3}$$

$$\epsilon_{i,t} \leq b, \quad \forall i \in \{N\}, \forall t \in \{T\}. \tag{4}$$

where we set the baseline of privacy loss b based on the largest client data size.

Therefore, the privacy budget allocation strategy for the t-th round is $\epsilon_t \triangleq (\epsilon_{1,t}, ..., \epsilon_{N,t})$. By utilizing local insights from clients, we optimize the privacy budget allocation strategy to mitigate the fairness issue while maintaining privacy.

B. Local Training Model

Each client holds a dataset denoted as D_i, with size d_i, $D_i \triangleq \{(x_j, y_j)\}_{j=1}^{d_i}$. To protect local information from the server, the client perturbs local data beforehand, transforming it into $\tilde{D}_i \triangleq \{(\tilde{x}_j, y_j)\}_{j=1}^{d_i}$, where $\tilde{x}_j \coloneqq x_j + w_{i,t}$ and $w \sim \mathcal{N}(0, \sigma_{i,t}^2)$ represents noise following a Gaussian distribution with standard deviation $\sigma_{i,k}$. σ controls the magnitude of noise and a greater σ provides stronger privacy protection.

In the local training phase, each client minimizes the empirical loss function $l(\theta)$ with perturbed data $\tilde{D}$ to obtain a local model:

$$\min_{\theta_{i,t}} f_i(\theta_{i,t}) \coloneqq \mathbb{E}[l_i(\theta_{i,t}; \tilde{D}_i)] \tag{5}$$

Based on the relationship between privacy budget and noise magnitude (see Section V for details), by optimizing the privacy budget allocation strategy, we will obtain the optimal choice of σ. Clients then sample noise following the distribution $\mathcal{N}(0, \sigma^2)$ for input perturbation.

C. Privacy Impact Model

We adopt client-level fairness that requires a more uniform distribution of the performance. We take "performance" to be the final testing accuracy of applying the trained global model on the test data of each client. To measure uniformity, we mainly use the variance of the performance distribution.

Definition 2 (Uniform Performance Distribution)**.** *For trained models θ and $\tilde{\theta}$, we formally say that model θ provides a more fair solution to the federated learning system than model $\tilde{\theta}$ if the test performance distribution of model θ is more uniform than that of $\tilde{\theta}$ across N clients, i.e.,*

$$Var\{A_i(\theta)\}_{i\in\{N\}} < Var\{A_i(\tilde{\theta})\}_{i\in\{N\}}, \tag{6}$$

where $A_i(\cdot)$ denotes the testing accuracy on client $i \in \{N\}$, $\{N\}$ denotes an integer set $\{N\} = \{1, ..., N\}$ and $Var\{\cdot\}$ denotes the variance.

Notice that LDP causes a discrepancy in performance decrease due to non-IID data in FL. To achieve client-level fairness, it's crucial to limit the gap among *performance changes*.

To formalize the performance changes caused by private mechanisms, *excessive risk* [47] is widely adopted in centralized settings, which defines the difference between the CDP private and non-private risk functions. We generalize it to an analogous *distributed excessive risk*:

Definition 3 (Distributed Excessive Risk)**.** *If a client with index i participates in distributed learning and transforms its own data from D_i into noisy data $\tilde{D}_i$, then the distributed excessive risk for client i is:*

$$R_i(\theta_i; D_i) = \mathbb{E}[l(\theta_i; \tilde{D}_i)] - l(\theta_i^*; D_i). \tag{7}$$

where the expectation is defined over the randomness of LDP and $\theta_i^* = \underset{\theta_i}{argmin} l(\theta_i; D_i)$.

To measure the performance degradation of a client deviating from the global, we define *distributed excessive risk gap* as follows:

Definition 4 (Distributed Excessive Risk Gap, ERG)**.** *In a global training round, the excessive risk gap ξ_i of client $\forall i \in \{N\}$ can be calculated as:*

$$\xi_i = |R_i(\theta) - R(\theta)| \tag{8}$$

where $R_i(\theta)$ and $R(\theta)$ are shorthands respectively represent the excessive risk $R_i(\theta_i; D_i)$ of client i and $R(\theta) = \frac{1}{N}\sum_{i=1}^{N} R_i(\theta_i; D_i)$ of the global, and N is the number of participating clients.

Smaller ERGs indicate a fairer FL system. We incorporate a regularization term expressed with ERG to the vanilla global objective of FL, which penalizes the discrepancies of performance changes. We formulate the regularization term as follows:

$$\lambda \frac{1}{N} \sum\nolimits_i \xi_i^2 \tag{9}$$

where N is the number of participating clients, λ is a positive scalar that tunes fairness and training accuracy and the square is used to avoid absolute value. A larger λ results in less difference in performance decrease among clients caused by LDP, i.e., higher fairness.

D. Problem Formulation

We face a **privacy and fairness improving problem (PriFair)**: given the interpolation parameter λ, fairness regularization term (9), privacy baseline b, local dataset D and local objectives of all clients, determining the privacy budget allocation strategy ϵ and global model θ, to minimize the expected loss of the aggregated global model while encouraging a more uniform performance distribution under constraints (3) and (4). We formulate PriFair as follows:

$$\begin{aligned} \textbf{PriFair}: \quad \min_{\epsilon,\theta} \quad & G(f_1(\theta; \tilde{D}_1), ..., f_N(\theta; \tilde{D}_N)) + \lambda \tfrac{1}{N} \textstyle\sum_i \xi_i^2 \\ \text{s.t.} \quad & 0 < \epsilon_{i,t} \leq b, \quad \forall i \in N, \forall t \in T. \end{aligned} \tag{10}$$

where $G(\cdot)$ is a function that aggregates the local objectives in an average way, i.e., $G(f_1(\theta), ..., f_N(\theta)) = \frac{1}{N}\sum_{i=1}^{N} f_i(\theta)$.

E. Problem Analysis

The fairness issue in our research is caused by LDP random noise added to local data. Without careful designs, random noise can disproportionately decrease the data availability across clients, thus it's crucial to mitigate the impact on performance by optimizing the noise allocation.

Existing LDP works [48], [49] consider gradient perturbation, directly adding noise to local updates, and the impact of random noise can be explicitly expressed in local and global optimization objectives. However, we implement LDP with input perturbation, which adds noise to local data. Gaussian noise appears in local objectives as a random variable w added in the noisy sample $\tilde{x}$ and indirectly influences optimization objectives.

In order to analyze the noise impact on problem PriFair, we have to capture the changes in local data distribution of each client caused by noise. It's obviously difficult in FL due to small amount of and non-iid local data.

Therefore, we need other tools that can expose the existence of random noise in optimization objectives. We introduce the found tool, a Tikhonov regularization-based approach, in Section IV-A. What's more, it's hard to simultaneously optimize the global model θ and the privacy budget vector ϵ in PriFair as they are not independent. We present an algorithm in Section IV-B to solve the interdependence problem by alternately optimizing the decision variables.

IV. Solution to PriFair

We assign varied magnitudes of Gaussian noise to clients for diverse levels of privacy protection. As analyzed in Section III-E, in order to assign appropriate amount of random noise to each client, a more intuitive way is needed to analyze the impact of noise on the optimization objectives. In view of the relation between regularization and random noise [50], we introduce a special class of regularizers, Tikhonov regularization, to reformulate our objectives.

A. A Tikhonov Regularization-based Perturbation Technique

Regularization technique was first introduced to control the bias and variance of a model. It adds a penalty term to modify the error function, sacrificing model flexibility for a more robust network. Besides directly modifying the network structure and regularization, adding random noise to data also improve the generalization of the network. Therefore, if there exists connection between regularization and input perturbation with the same purpose of generalization, regularization can be used to formalize the impact of noise on the model.

Tikhonov regularization is a class of regularizers on linear models, which takes the form:

$$\Omega(y) = \sum\nolimits_{r=0}^{R} \int_a^b h_r(x) (\frac{d^r y}{dx^r})^2 dx \tag{11}$$

where $h_r \leq 0$ for $r = 0, ..., R-1$ and $h_R > 0$.

In this section, we leverage Tikhonov regularization to reformulate the optimization objective and incorporate the noise amplitude into the model, providing a formal method to quantify the impact of LDP noise. Tikhonov regularization, commonly used to control bias and variance in machine learning models, allows us to explicitly capture the effect of random noise introduced during local client training as a form of input perturbation [50]. In problem PriFair, the server minimizes the aggregated expected sum-of-squares loss from clients, where the training-with-noise error term is replaced by a Tikhonov regularized error function. This enables us to systematically expose and quantify the influence of noise on the model. By decomposing the loss into two components—expected error and noise-induced regularization—we can directly optimize the noise allocation and adjust privacy budgets across clients, thereby improving fairness without sacrificing privacy.

According to the equations (18) and (19) in [50], when the local model minimizes the sum-of-squares error on noisy data, the expected error in the local objective can be reformulated as a combination of two parts. One denotes the expected error before adding noise, and the other denotes the regularization term in the form of a L_2-norm of the global model parameters. The noise amplitude σ in our problem acts as a bias-variance trade-off parameter in the regularized local objective function. The above is formulated in Lemma 1.

Lemma 1. *[50] For network training, for a set of discrete input vectors $\boldsymbol{x}$ and corresponding target output vectors $\boldsymbol{t}$, with probability distribution function given by $p(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n}\sum_q \delta(\boldsymbol{x} - \boldsymbol{x}^q)\delta(\boldsymbol{t} - \boldsymbol{t}^q)$, the sum-of-squares error function on perturbed data becomes*

$$\tilde{E} = E + \sigma^2 E^R, \tag{12}$$

in which σ^2 is controlled by the amplitude of the noise, E is the standard sum-of-squares error on clean data defined as

$$E = \frac{1}{2n}\sum\nolimits_j ||y(x^j) - t^j||^2, \tag{13}$$

and the extra term E^R is given by

$$E^R = \frac{1}{2n}\sum\nolimits_j \sum\nolimits_m \sum\nolimits_i (\frac{\partial y_m^j}{\partial x_i^j})^2. \tag{14}$$

where n is the amount of input samples labeled by index j, i labels the dimension of input samples and m labels the dimension of output vectors. According to Lemma 1, the local objective becomes:

$$\min_{\theta_{i,t}} f_i(\theta_{i,t}) \coloneqq \mathbb{E}[l_i(\theta_{i,t}; \tilde{D}_i)] = \frac{1}{2d_i}[\sum_{j=1}^{d_i} h(x_j, y_j) + \sigma_{i,t}^2 \|\theta_{i,t}\|^2]. \tag{15}$$

where h denotes a sum-of-squares loss function.

In PriFair, the client perturbs local data and performs the optimization of local model θ_i based on designed noise amplitude σ_i, which is determined by the optimal privacy budget allocation strategy ϵ. Therefore, we first optimize the privacy budget allocation strategy of problem PriFair. At the beginning of each global round t, the client performs local analysis based on a given initial model $\tilde{\theta}_t$. The server collects

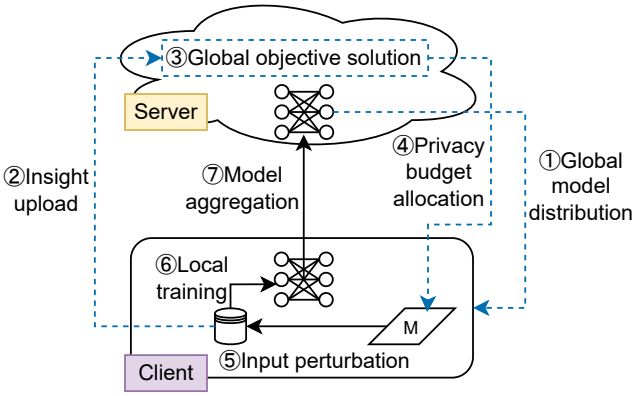


Fig. 1: The workflow of PriFairFed Algorithm.

local information for federated analysis and solves PriFair to get an optimal ϵ. The fairness regularization term (8) of PriFair, including the local expected loss on noisy data $\mathbb{E}[l(\theta_{i,t}; \tilde{D}_i)]$, can be obtained during the local analysis and transformed based on 1.

If noise amplitude $\sigma_{i,t}$ and privacy loss $\epsilon_{i,t}$ of client i satisfy $\sigma_{i,t} = \frac{Q_i(K, \delta_{i,t}, d_i)}{\epsilon_{i,t}}$, the local model satisfies $(\epsilon_{i,t}, \delta_{i,t})$-LDP, where Q_i is a coefficient determined by the client local training epochs K, local data size d_i and LDP failure probability $\delta_{i,t}$ (see Section V for details). Thus we can substitute ϵ for σ in objectives and fairness regularization term:

$$\begin{aligned} R_i(\theta_{i,t}; \epsilon_{i,t}) &= \mathbb{E}[l(\theta_{i,t}; \tilde{D}_i)] - l(\theta_i^*; D_i) \\ &= \frac{1}{2d_i}\left[\sum_{j=1}^{d_i} h(x_j, y_j) + \sigma_{i,t}^2 ||\theta_{i,t}||^2 - l(\theta_i^*; D_i)\right] \\ &= \frac{1}{2d_i}\left[\sum_{j=1}^{d_i} h(x_j, y_j) + \left(\frac{Q_i}{\epsilon_{i,t}}\right)^2 ||\theta_{i,t}||^2 \right. \\ &\quad \left. - l(\theta_i^*; D_i)\right], \\ R(\theta_t; \epsilon_t) &= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2d_i}\left[\sum_{j=1}^{d_i} h(x_j, y_j) + \left(\frac{Q_i}{\epsilon_{i,t}}\right)^2 ||\theta_{i,t}||^2 \right. \\ &\quad \left. - l(\theta_i^*; D_i)\right], \quad \forall i \in \{N\}, \forall t \in \{T\}. \end{aligned} \tag{16}$$

Combining (15)(16)(18), PriFair is reformulated to problem **PriFair-T** as follows:

$$\begin{aligned} \textbf{PriFair} - \textbf{T}: \quad & \min_{\epsilon, \theta} \quad G(f_1(\theta; \tilde{D}_1), ..., f_N(\theta; \tilde{D}_N)) \\ & \qquad +\lambda \tfrac{1}{N}\textstyle\sum_i [R_i(\theta_{i,t}; \epsilon_{i,t}) - R(\theta_t; \epsilon_t)]^2 \\ & \text{s.t.} \quad \epsilon_{i,t} > 0, \tfrac{1}{\epsilon_{i,t}^2} \geq \tfrac{1}{b^2}, \quad \forall i \in N, \forall t \in T. \end{aligned} \tag{17}$$

Now we have incorporated noise impact into the global objective. Obviously, problem (17) is a typical quadratic programming problem for $\frac{1}{\epsilon^2}$ and easy to solve.

B. Fair and Private FL Algorithm: PriFairFed

The local model is learned based on noisy data, but clients have to perturb data with allocated privacy budgets, which together with model parameter is the solution of the global optimization objective. To overcome the mutual dependency between the model and privacy budgets, we approximate the local model with the original one distributed by the server at the start of each round, enabling clients to provide local insights for global optimization. Once the server determines

Algorithm 1 The PriFairFed Algorithm

Input: Local dataset $D = \{D_1, D_2, ..., D_N\}$, local objective function $l(\cdot)$, privacy baseline b and local learning rate η
Parameter: Positive scalar λ
Output: Privacy budget allocation strategy ϵ^* and global model θ^*

1: $\hat{\theta} \leftarrow Random(\theta)$.
2: **for** $t = 1, 2, ..., T$ **do**
3: // Phase I: Privacy budget allocation
4: **Central Server** :
5: Broadcast initial model $\hat{\theta}$ to each client i.
6: **Client** i:
7: $\theta_i \leftarrow \hat{\theta}$,
8: $s_{i1} \leftarrow l(\theta_i; D_i)$, $s_{i2} \leftarrow ||\hat{\theta}_{i,t}||^2$,
9: Send s_{i1}, s_{i2}, data size d_i and optimal local model loss without noise $l(\theta_i^*; D_i)$ to the central server.
10: **Central Server** :
11: Solving (17) with uploaded local insights to obtain ϵ_t^* and corresponding $(\sigma_t^2)^*$.
12: Broadcast ϵ_t^* to each client i.
13: // Phase II: Global model updating
14: **Client** i:
15: **for** $(x_j, y_j) \in D_i$ **do**
16: $\tilde{x}_j \leftarrow x_j + w_{i,k}$, $\tilde{y}_j \leftarrow y_j$
17: **end for**
18: **for** $k = 1, 2, ..., K$ **do**
19: $\theta_i \leftarrow \theta_i - \eta \cdot \frac{1}{d_i} \sum_{j=1}^{d_i} \nabla l[\theta_i; (\tilde{x}_j, \tilde{y}_j)]$,
20: **end for**
21: Send θ_i to the central server.
22: **Central Server**:
23: $\theta \leftarrow \frac{1}{N} \sum_{i=1}^{N} \theta_i$
24: **end for**
25: **return** θ, ϵ^*

the privacy budget allocation strategy ϵ^* by solving PriFair-T, clients perturb local data with allocated privacy budgets, conduct stochastic gradient descent to train local models, and upload local updates.

We design a **private and fair FL (PriFairFed) algorithm**, which can be easily combined with other aggregation algorithms. As shown in Alg. 1, each global training round consists of two stages: the privacy budget allocation stage and the model updating stage. The global model converges after T rounds of global aggregation.

In the privacy budget allocation stage, the server first distributes the initial global model θ to each client (Line 5). Clients then compute expected loss with the received model and L_2-norm of the model parameters with respect to clean data (Line 7-8). Each client uploads computing result s_{i1}, s_{i2}, local data size d_i and optimal local model loss without noise $l(\theta_i^*; D_i)$ to the server (Line 9), in which $l(\theta_i^*; D_i)$ is considered as prior knowledge and pre-computed only once by clients during idle time. The server solves the problem (17) based on local analytics results, obtaining the optimal privacy budget allocation strategy ϵ^* (Line 11) and allocating it to each client (Line 12). Fig. 1 shows the workflow of the privacy budget allocation stage in dotted lines.

In the model updating stage, clients first perturb local data based on privacy budget ϵ^* obtained in the first stage (Line 16) where noise amplitude $\sigma_{i,t} = \frac{Q_i(K, \delta_{i,t}, d_i)}{\epsilon_{i,t}}$. Clients then perform stochastic gradient descent on noisy data for K local epochs (Line 18-19) and send back local updates (Line 21). Finally, the server aggregates local models (Line 23) to get the next round global model. Fig. 1 demonstrates the workflow of the model updating stage in solid lines.

V. Privacy and Utility Analysis

In this section, we first analyze the privacy leakage of each training round and all training rounds. Then, we show the convergence rate of the algorithm PriFairFed.

Following general settings in FL research, we make the assumptions below on local loss function $l(\theta)$.

Assumption 1. *$l(\theta)$ is $G(\theta)$-Lipschitz continuous: for all ζ_1 and ζ_2, $|l(\theta; \zeta_1) - l(\theta; \zeta_2)| \leq G(\theta) \cdot ||\zeta_1 - \zeta_2||$.*

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

These assumptions can be satisfied as we analyze with linear regression, softmax classifier and mean squared error (MSE).

A. Privacy Analysis

Clients receive the distributed privacy budget and train their local models with perturbed data. Thus, both the privacy of local data and models are preserved. In this section, we analyze the privacy leakage and the privacy loss of PriFairFed.

Privacy loss of each training round: In each training round, the client individually trains the local model with noisy data for several local epochs, introducing noise into the model. The following theorem shows the relation between noise amplitude and the privacy guarantee of the local model.

Theorem 1. *Let Assumption 1 and 2 hold and all $\epsilon_{i,t}, \delta_{i,t} > 0$ for all $i = 1, ..., N$ and $t = 1, ..., T$. In Alg. 1, if we have*

$$\sigma_{i,t}^2 = c \frac{G^2 K ln(1/\delta_{i,t})}{d_i(d_i - 1)\sqrt{\mu}\epsilon_{i,t}^2}, \tag{18}$$

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

Proof. In round t, each client i performs K epochs of gradient descent to update their local model. As in [51], we assume the loss function $l(\theta, x, y)$ is $l(y\theta^T x)$. For client i in the k-th epoch, the local model $\theta_{i,k+1}$ is

$$\theta_{t,k+1} = \theta_{t,k} - \eta \underbrace{\frac{1}{d_i} \sum_{j=1}^{d_i} y_i^j \nabla l(y_i^j \theta_k^T (x_i^j + w_{i,t}))(x_i^j + w_{i,t})}_{\mathcal{M}_{i,k}}, \tag{19}$$

where $w_{i,t} \sim \mathcal{N}(0, \sigma_{i,t}^2)$ and η denotes the learning rate. From Equation (19), we can see only the aggregated gradients may disclose the privacy of the training data, and we define it as the mechanism $\mathcal{M}$.

Denote D_i and $D_i^{'}$ as the adjacent datasets of 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 distribution on D_i and $D_i^{'}$ over mechanism $\mathcal{M}_{i,k}$. We have:

$$P_i = A + B \cdot w_{i,t} + \underbrace{\frac{1}{d_i} y_i^{d_i} \nabla l(y_i^{d_i} \theta_k^T (x_i^{d_i} + w_{i,t}))(x_i^{d_i} + w_{i,t})}_{C}, \tag{20}$$

$$P_i^{'} = A + B \cdot w_{i,t} + \underbrace{\frac{1}{d_i^{'}} y_i^{d_i^{'}} \nabla l(y_i^{d_i^{'}} \theta_k^T (x_i^{d_i^{'}} + w_{i,t}))(x_i^{d_i^{'}} + w_{i,t})}_{C'}, \tag{21}$$

where

$$A = \frac{1}{d_i} \sum\nolimits_{j=1}^{d_i - 1} y_i^j \nabla l(y_i^j \theta_k^T (x_i^j + w_{i,t})) x_i^j, \tag{22}$$

$$B = \frac{1}{d_i} \sum\nolimits_{j=1}^{d_i - 1} y_i^j \nabla l(y_i^j \theta_k^T (x_i^j + w_{i,t})). \tag{23}$$

Note that $w_{i,t} \sim \mathcal{N}(0, \sigma_{i,t}^2)$, we have $P \in \mathcal{N}(A + C, B\sigma_{i,t}^2)$ and $P^{'} \in \mathcal{N}(A + C^{'}, B\sigma_{i,t}^2)$.

To analyze the total privacy of K epochs, we utilize the moments accountant method proposed in [52]. The λ-th moment of mechanism $\mathcal{M}_{i,k}$ on dataset D_i and $D_i^{'}$ is defined as

$$\alpha_{\mathcal{M}_{i,k}}(\lambda) = ln\left(\mathbb{E}_{O \sim P}\left[\left(\frac{P_i}{P_i^{'}}\right)^{\lambda}\right]\right) = \lambda D_{\lambda+1}(P_i || P_i^{'}), \tag{24}$$

where the second equality is derived by Definition 2.1 in [52].

According to Lemma 2.5 in [53], Equation (24) can be transformed to

$$\alpha_{\mathcal{M}_{i,k}}(\lambda) = \frac{\lambda(\lambda+1)||C - C^{'}||^2}{2B\sigma_{i,t}^2}. \tag{25}$$

Since the loss function is G-Lipschitz continuous according to Assumption 1, 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 [54]. Thus we have:

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

where θ^* is the optimal model parameter. Then the lower bound of B defined in Equation (20) and (21) can be derived based on Equation (27):

$$B \geq \frac{d_i - 1}{d_i} \sqrt{2\mu(l(\theta_K) - l(\theta^*))}, \tag{28}$$

where θ_T is the model of the last epoch.

Since $l(\theta_T) - l(\theta^*)$ can be considered as a constant, with Equation (26) and (28), for some constant c_0, Equation (25) can be transferred to

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

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

$$\sum\nolimits_{k=1}^{K} \alpha_{\mathcal{M}_{i,k}} \leq c_1 \frac{\lambda^2 G^2 K}{\sqrt{\mu}\sigma_{i,t}^2 d_i(d_i - 1)}. \tag{30}$$

Let $\alpha_{\mathcal{M}_i}(\lambda)$ be the bound of all possible $\alpha_{\mathcal{M}_{i,k}}(\lambda)$. With Theorem 2.1 stated in [52] and Equation (29), we have

$$\alpha_{\mathcal{M}_i}(\lambda) \leq \sum\nolimits_{k=1}^{K} \alpha_{\mathcal{M}_{i,k}} \leq c_1 \frac{\lambda^2 G^2 K}{\sqrt{\mu}\sigma_{i,t}^2 d_i(d_i - 1)}. \tag{31}$$

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

$$\alpha_{\mathcal{M}_i}(\lambda) \leq c_1 \frac{\lambda^2 G^2 K}{\sqrt{\mu}\sigma_{i,t}^2 d_i(d_i - 1)} \leq \frac{\lambda\epsilon_{i,t}}{2}, \tag{32}$$

and as a result, we have

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

According to Theorem 2.2 in [52], the local model $\theta_{i,t}$ of client i in k-th round learned with mechanism $\mathcal{M}_i$ satisfies $(\epsilon_{i,t}, \delta_{i,t})$-local differential privacy. □

Privacy loss of total training rounds: Based on the above analysis of privacy loss in each training round, since PriFairFed is a k-fold adaptive algorithm, we can use moments accountant defined in [52] to obtain the total privacy leakage.

Theorem 2. *Let Assumption 1 and 2 hold and $\epsilon_{i,t}, \delta_{i,t} > 0$ for all $i = 1, ..., N$ and all $t = 1, ..., T$. With $\sigma_{i,t}$ satisfies Eq. (18), the Alg. 1 guarantees $\left(c_0\sqrt{T}\epsilon_{m,\tilde{t}}, \delta_{m,\tilde{t}}\right)$-differential privacy for some constant c_0, where $\epsilon_{m,\tilde{t}} = max\{\epsilon_{i,t}|\forall i = 1, ..., N, \forall t = 1, ..., T\}$.*

Proof. In Algorithm 1, the training process ends after T global rounds. We first analyze the privacy leakage in each global round and then derive the total privacy loss of all training rounds. According to Theorem 1, we know each local model guarantees $(\epsilon_{i,t}, \delta_{i,t})$-local differential privacy if $\sigma_{i,t}^2 = c\frac{G^2 K ln(1/\delta_{i,t})}{d_i(d_i-1)\sqrt{\mu}\epsilon_{i,t}^2}$. Since the global model in t-th round is aggregated by the local model $\theta_{i,t}$ of each client i, according to the parallel composition theorem proposed in [46], the global model in t-th round satisfies $(\epsilon_{m,t}, \delta_{m,t})$-differential privacy, where $\epsilon_{m,t} = max\{\epsilon_{1,t}, ..., \epsilon_{N,t}\}$. Then for all training rounds, each of them satisfies $(\epsilon_{m,\tilde{t}}, \delta_{m,\tilde{t}})$-differential privacy, where $\epsilon_{m,\tilde{t}} = max\{\epsilon_{i,t}|\forall i = 1, ..., N, \forall t = 1, ..., T\}$. Similar to formula (31), we have

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum\nolimits_{t=1}^{T} \alpha_{\mathcal{M}_t} \leq c_2 \frac{\lambda^2 G^2 K T}{\sqrt{\mu}\sigma_{m,\tilde{t}}^2 d_m(d_m - 1)}. \tag{34}$$

With Theorem 2.1 stated in [52] and assume Algorithm 1 achieves $(\hat{\epsilon}, \delta_{m,\tilde{t}})$-differential privacy, we can obtain

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

Therefore, there exists a constant c_0 to make the total privacy loss $\hat{\epsilon}$ satisfies $\hat{\epsilon} = c_0\sqrt{T}\epsilon_{m,\tilde{t}}$, which leads to Theorem 2. □

B. Utility Analysis

We define $\mathcal{U}(T)$ as the utility of the global model learned after T rounds. Followed with [55], the utility can be represented as the multiplicative inverse of the convergence rate. We have $\mathcal{U}(T) = \frac{1}{\mathbf{E}[l(\theta_T)] - l(\theta^*)}$, where θ^* is the optimal model parameter that minimizes the global model loss. Thus, we utilize $\mathcal{U}(T)$ by deriving the upper bound of the convergence rate of Alg. 1.

Assumption 3. *$l(\theta)$ is L-smooth: for all θ_1 and θ_2, $l(\theta_1) \leq l(\theta_2) + (\theta_1 - \theta_2)^T \nabla l(\theta_2) + \frac{L}{2}||\theta_1 - \theta_2||^2$.*

L-smooth is a common assumption as in [56].

Theorem 3. *Let Assumption 1 to 3 hold and the value of the gradient is upper bounded with V (i.e., $\nabla l(\theta) \leq V$), with $\sigma_{i,t}$ satisfies Eq. (18), we have*

$$\mathbf{E}\left[l\left(\theta_T\right)\right] - l^* \leq \frac{LV^2}{\mu^2 T}\left(1 + p\sigma^2\right), \tag{36}$$

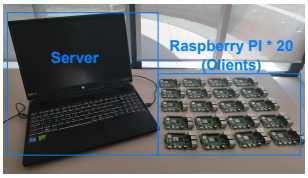


Fig. 2: System Architecture in the experiment.

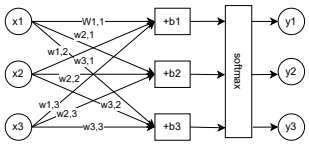


Fig. 3: The architecture of the neural network used.

where $\sigma = max\{\sigma_{i,t}|\forall i = 1, ..., N, \forall t = 1, ..., T\}$, p *is the dimension of each training data point.*

Proof. In Algorithm 1, Gaussian noise is added to local data before performing gradient descent. Then the global model parameter is updated as below:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \eta(\frac{1}{D}\sum_{j=1}^{D}\nabla l(\theta_t, x_j + w_{j,t}, y_j)) \\ &\approx \theta_t - \eta(\frac{1}{D}\sum_{j=1}^{D}\nabla l(\theta_t, x_j, y_j) + \frac{1}{D}\sum_{j=1}^{D} a_{j,t}w_{j,t}),\end{aligned} \tag{37}$$

where $a_{j,t} = ||\nabla_{x_j}\nabla l(\theta_t, x_j, y_j)|| \geq I$ and $D = \sum_{i=1}^{N} d_i$, and the approximation is derived by the first order Taylor expansion at point x_j.

Assume $w_{j,t} \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = max\{\sigma_{i,t}|\forall i = 1, ..., N, \forall t = 1, ..., T\}$, we have $\frac{1}{D}\sum_{j=1}^{D} a_{j,t}w_{j,t} \geq Iw_{j,t} \sim \mathcal{N}(0, I\sigma^2)$. Then we can obtain a recurrence formula for $||\theta_t - \theta^*||^2$ as below:

$$\begin{aligned}||\theta_{t+1} - \theta^*||^2 &= ||\theta_t - \theta^* - \eta \cdot (\frac{1}{D}\sum_{j=1}^{D}\nabla l(\theta_t, x_j + w_{j,t}, y_j))||^2 \\ &\leq (1 - 2\eta\frac{\mu}{V})||\theta_t - \theta^*||^2 + \eta^2(1 + p\sigma^2).\end{aligned} \tag{38}$$

Assume $\eta = \frac{V}{\mu T}$, and according to the L-smoothness of the loss function, we can derive

$$\mathbf{E}[l(\theta_T)] - l(\theta^*) \leq \frac{L}{2}\mathbf{E}[||\theta_T - \theta^*||^2] \leq \frac{LV^2}{\mu^2 T}(1 + p\sigma^2). \tag{39}$$

Therefore we have Theorem 3. □

From Theorem 3, the utility $\mathcal{U}(T)$ is bounded by $\frac{\mu^2 T}{LV^2(1+p\sigma^2)}$ and PriFairFed will asymptotically converge to the global optimum at the rate $\mathcal{O}(\frac{1}{T})$, where T is the total global training rounds. For complete proof see Appendix B.3.

C. Complexity Analysis

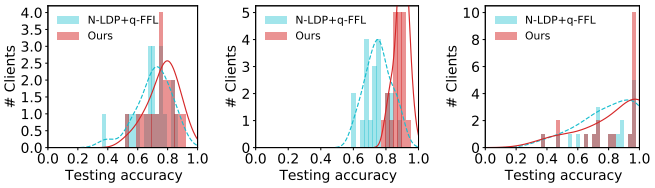
PriFairFed involves two main phases per global round: privacy budget allocation and global model updating. In the privacy budget allocation phase, the central server solves a convex quadratic programming problem to optimize privacy budgets for all clients, with a computational complexity of $O(N^2)$ per round, where N is the number of clients. After solving the problem, the server broadcasts the privacy budgets to clients, incurring a communication overhead of $O(N)$. In the global model updating phase, each client perturbs its local data based on the received privacy budget and performs stochastic gradient descent (SGD) over K epochs, where the complexity for each client is $O(d_i K)$, with d_i representing the local dataset size of client i. Once the local updates are complete, the server aggregates the models from all N clients, which has a cost of $O(N)$. Thus, the overall complexity per global round is $O(N^2 + \sum_{i=1}^{N} d_i K)$. While the dynamic privacy budget allocation introduces some overhead, it remains manageable due to the efficient nature of solving the quadratic problem. Additionally, the communication cost is linear in the number of clients. Reducing the frequency of privacy budget updates, such as allocating them every few global rounds, can further mitigate this overhead. Overall, PriFairFed maintains scalability and feasibility for large-scale federated learning deployments, even with a large number of clients.

VI. Evaluation

A. Experimental Setup

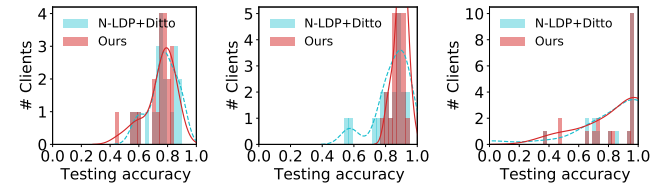
Testbed: We implement the system with PyTorch, experimenting in a real-world system consisting of 20 Raspberry Pi 4 Model B clients with ARM11 microprocessors and one laptop server, shown in Fig. 2. In the beginning, we split the data between 20 clients and all clients are selected in each round.
Dataset and Model: We evaluate our models with 4 datasets:
• *Synthetic Dataset*: Generated using a linear regression classifier, with sample sizes for each client following a log-normal distribution [57]. This setup simulates environments with varying data quantities across clients.
• *Fashion MNIST* [58]: For this image dataset, we apply a Dirichlet process [59] (denoted as $Dir(\zeta)$ with $\zeta = 0.5$) to create diverse imbalanced data distributions, mimicking real-world scenarios where clients have uneven data availability.
• *Subsampled Fashion MNIST*: This version includes only three categories—T-shirt, pullover, and shirt, with each client holding data from only one category. This setup highlights class imbalance and simulates limited class diversity.
• *Loan Dataset* [60]: A numerical dataset using a Dirichlet process (denoted as $Dir(\zeta)$ with $\zeta = 0.5$) to introduce domain shift, reflecting different.

To implement the above classification tasks, we train a linear regression classifier, as shown in Fig. 3. It comprises one input layer, a fully connected layer, and a softmax layer. We employ the mean squared error loss function due to the use of Tikhonov regularization, which provides a theoretical guarantee under MSE.
Baselines: We compare with three state-of-the-art fairness approaches which focus on client-level performance fairness and one adaptive privacy budget allocation algorithm which directly encrypts local data. We also employ FedAvg [1] as the baseline.
• *Agnostic Federated Learning (AFL) [12]* is an agnostic approach, which optimizes for the worst-performing device. Following [12], we experiment on 3 clients with subsampled Fashion MNIST.
• *q-FFL [9]* up-weights lower-performing clients and the fairness can be tuned by parameter q.
• *Ditto [61]* incorporates a regularization term into local optimization objectives to achieve an interpolation between local and global models, where the fairness can be tuned by parameter λ.



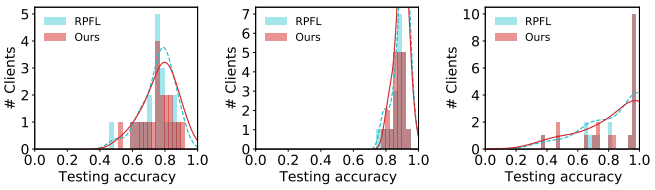
(a) Fashion MNIST (b) Loan (c) Synthetic

Fig. 4: The final testing accuracy distribution of PriFairFed and q-FFL on three chosen datasets.



(a) Fashion MNIST (b) Loan (c) Synthetic

Fig. 5: The final testing accuracy distribution of PriFairFed and Ditto on three chosen datasets.



(a) Fashion MNIST (b) Loan (c) Synthetic

Fig. 6: The final testing accuracy distribution of PriFairFed and RPFL on three chosen datasets.

• *RPFL [21]* adopts DRO to model the uncertainty induced by LDP and adaptively assigns privacy budgets to clients for data perturbation.

We experimented with 20 clients across Synthetic, Fashion MNIST, and Loan with q-FFL, Ditto, and RPFL. To maintain the same privacy level, AFL, q-FFL, and Ditto follow N-LDP, assigning equal privacy budgets to clients based on the largest amount of data across clients, while RPFL and ours assign privacy budgets adaptively no bigger than AFL and q-FFL.

Evaluation Metrics: We use two metrics to evaluate the fairness and privacy leakage level of algorithms: 1) *Variance of the Final Testing Accuracy* represents whether the model treats clients equally. A larger variance infers less fairness. And 2) *MIA Accuracy* infers whether the attacker finds the target client of a data point. A higher attack accuracy indicates less privacy.

B. Fairness Analysis

We first conduct experiments to demonstrate the effectiveness of PriFairFed (ours) in mitigating fairness issues caused by LDP in comparison with fairness benchmarks. Fig. 4, Fig. 5 and Fig. 6 illustrate the final testing performance distributions respectively for q-FFL, Ditto, and RPFL compared with PriFairFed. The privacy baselines for Fashion MNIST, Loan, and Synthetic are set to 0.7, 12, and 0.3 based on the largest data size of clients. The fairness scalar λ of PriFairFed is set to 3, 40, and 2 respectively on three datasets. For benchmark q-FFL, we employ its distributed solver q-FedAvg and set the trade-off parameter $q = 0.5$. Compared to benchmark q-FFL, PriFairFed reduces the variance by up to 73.6%. Meanwhile, the average testing accuracy improves.

TABLE I: Statistics of the Final Testing Accuracy Distribution for N-LDP (No Fairness Bound) and PriFairFed

Dataset	**Fairness** λ	**Avg.**	**Worst 10%**	**Best 10%**	**Variance (x10000)**
Fashion MNIST	-	75.0%	48.7%	91.3%	131.5
	50.0	75.3%	**54.7%**	91.2%	**115.5**
Loan	-	89.9%	81.6%	96.1%	17.6
	40.0	89.5%	81.4%	95.5%	**16.0**
Synthetic	-	84.9%	43.8%	100.0%	365.81
	2.0	84.7%	43.8%	100.0%	389.84

For Ditto, we follow the settings in [61] where the interpolation hyper-parameter $\lambda = 1$ and λ is not device-specific, and the iteration of each device for local optimization is 2. As shown in Fig. 5, PriFairFed maintains a performance similar to that of Ditto. However, as shown in 5b, ours reduces the variance by 86.0%, which represents a more concentrated performance distribution and reflects higher fairness.

For RPFL, each client has an individual privacy baseline based on its data size that is no bigger than the global baseline. Compared to RPFL, PriFairFed keeps almost the same variation and mean of testing performance distribution. It is important to note that while PriFairFed significantly reduces variance and improves fairness, it does not always outperform RPFL in all aspects. RPFL's use of Distributionally Robust Optimization (DRO) effectively handles the uncertainty introduced by LDP-induced noise, improving the robustness of the global model. In contrast, PriFairFed focuses more on reducing performance disparity across clients by allocating varied noise amplitudes based on federated analytics, which targets client-level fairness rather than adversarial robustness.

Additionally, we record the worst and best 10% testing accuracies along with the variance of the accuracy distributions of N-LDP and PriFairFed in Tab. I. Despite a slight decline in accuracy for the best 10%, there is an improvement for the worst 10% accompanied by a significant reduction in variance. Additionally, the average accuracy remains nearly unchanged. Here, the average accuracy is computed across all data points not the mean of all devices' accuracies. The slight decline in accuracy for the top 10% of clients reflects the trade-off between fairness and absolute performance. This trade-off is expected, given PriFairFed's objective of ensuring fairness across all clients, especially those with more challenging data distributions.

In the case of AFL, following the settings in [9], we set the privacy baseline as $\epsilon = 11$, three participating clients each hold one kind of label where the "shirt" class is the hardest one to distinguish from others. Experimental results are presented in Tab. II. Compared with AFL, PriFairFed achieves higher testing accuracy on the worst-performing device, aligning with the original optimization goal of AFL. As the fairness

TABLE II: The Final Testing Accuracy of Different Clients with PriFairFed and AFL on Subsampled Fashion MNIST

Method	Avg.	Shirt	Pullover	T-shirt
N-LDP+AFL	52.1%	57.7%	45.5%	53.0%
Ours (λ=2)	79.2%	**85.6%**	84.2%	67.7%
Ours (λ=3)	79.6%	**85.5%**	85.1%	68.2%
Ours (λ=5)	79.0%	**85.6%**	84.8%	66.5%

TABLE III: Statistics of Testing Accuracy Using a Series of Regularization Parameter λ on Dataset Fashion MNIST

Dataset	Fairness λ	Avg.	Worst 10%	Best 10%	Var.	MIA Acc.
Fashion MNIST	0.5	76.8%	48.7%	93.2%	167.9	74.0%
	1.0	77.1%	57.3%	94.9%	112.7	**78%**
	3.0	76.9%	57.2%	91.8%	**96.4**	**77.4%**
	5.0	74.7%	52.0%	85.5%	**104.8**	77%
N-LDP	-	75.0%	48.7%	91.3%	**131.5**	**71%**
FedAvg	-	76.2%	59.9%	93.0%	**97.7**	**82%**

TABLE IV: Statistics of Testing Accuracy across 50 Devices on Dataset Fashion MNIST

Method	Avg acc.	Worst 10%	Best 10%	Var.
q-FFL	66.6%	49.0%	81.3%	96.3
Ditto	**77.1%**	58.6%	89.2%	87.2
RPFL	75.4%	54.5%	**89.9%**	111.5
Ours	76.7%	**60.7%**	89.2%	**70.9**

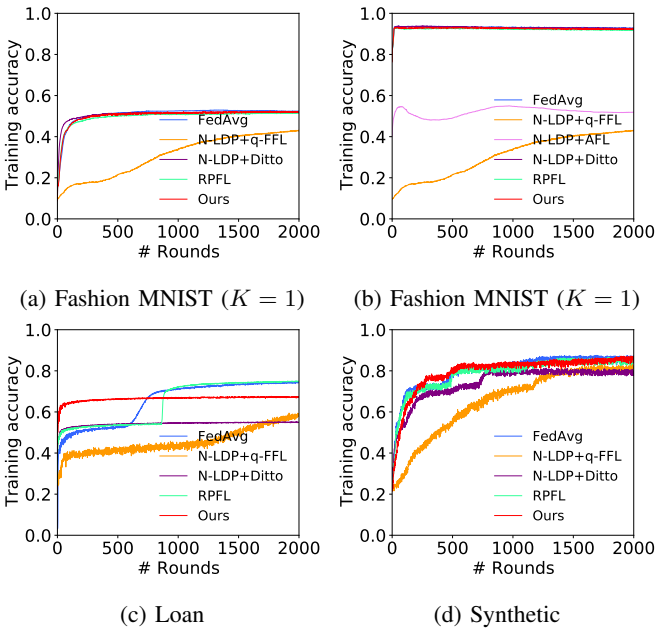
regularization parameter λ increases, the accuracy of clients holding "shirt" samples increases.

Choosing fairness parameter λ: We conduct experiments to explore how to tune the fairness level under LDP using the fairness regularization term parameter λ. We test various λ values within the range $\{0.5, 1, 3, 5\}$ on the dataset Fashion MNIST with 20 clients, as outlined in Tab. III. A larger λ leads to reduced variance, i.e., higher fairness, while potentially compromising privacy to some extent. Intuitively, with a larger λ, PriFairFed strengthens the constraints on the distribution of excessive risk gap, compelling worse-performing clients to apply less noise to reduce performance degradation (and vice versa), which consequently leads to smaller performance discrepancies among clients.

Scaling to a larger network: We also conduct experiments in a scenario with 50 clients to verify the scalability of the proposed algorithm PriFairFed. We experiment on Fashion MNIST, and the hyper-parameters and privacy settings are consistent with those described above. Results are presented in Tab. IV. Even if the network size expands to 2.5 times the original, PriFairFed still meets the fairness requirements, improving the performance of the worst 10% and significantly reducing the variance.

C. Efficiency Analysis

We also investigate the efficiency of different algorithms. The hyper-parameters and privacy budget allocation for benchmarks are the same as in Section VI-B. The fairness scalar λ is 1 ($K = 1$). When comparing with AFL, each client performed one epoch of local training. In other cases, each client performed 5 local epochs ($K = 5$). As shown in Fig. 7, in terms of communication rounds, PriFairFed converges faster than q-FFL, AFL, and Ditto, maintaining nearly the same convergence rate as RPFL. Intuitively, considering the highly heterogeneous data distribution, PriFairFed weights clients equally to avoid overfitting to certain devices, which can accelerate the training.



(a) Fashion MNIST ($K = 1$) (b) Fashion MNIST ($K = 1$)

(c) Loan (d) Synthetic

Fig. 7: The convergence of PriFairFed and SOTAs.

VII. Conclusion

In this work, we study a novel fairness problem under LDP in FL with a generalized excessive risk gap. We reformulate the problem with Tikhonov regularization, quantifying the LDP noise impact on objectives. To address the reformulated problem, we propose PriFairFed, an efficient algorithm that alternately optimizes for privacy budget allocation strategy and the global model, which can be achieved in polynomial time as a quadratic programming problem. PriFairFed flexibly controls the fairness level with a tunable parameter λ. Through theoretical analysis and real-world experiments, PriFairFed demonstrates significant improvement in fairness under LDP. Our future work includes extending the analysis to various loss functions like cross-entropy.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[3] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[4] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," *Advances in neural information processing systems*, vol. 32, 2019.

[5] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *arXiv preprint arXiv:1503.01214*, 2015.

[6] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] J. Salas, V. Torra, and D. Megías, "Towards measuring fairness for local differential privacy," in *International Workshop on Data Privacy Management*. Springer, 2022, pp. 19–34.

[8] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 1171–1180.

[9] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[10] N. Martinez, M. Bertran, and G. Sapiro, "Minimax pareto fairness: A multi objective perspective," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6755–6764.

[11] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 393–399.

[12] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4615–4625.

[13] C. Tran, M. Dinh, and F. Fioretto, "Differentially private empirical risk minimization under the fairness lens," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 555–27 565, 2021.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer, 2006, pp. 265–284.

[15] C. Wu, Y. Zhu, R. Zhang, Y. Chen, F. Wang, and S. Cui, "Fedab: Truthful federated learning with auction-based combinatorial multi-armed bandit," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15 159–15 170, 2023.

[16] R. Zhang, Y. Chen, C. Wu, and F. Wang, "Multi-level personalized federated learning on heterogeneous and long-tailed data," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2024.

[17] G. Zhou, Q. Li, Y. Liu, Y. Zhao, Q. Tan, S. Yao, and K. Xu, "Fedpage: Pruning adaptively toward global efficiency of heterogeneous federated learning," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, p. 1873–1887, dec 2023. [Online]. Available: https://doi.org/10.1109/TNET.2023.3328632

[18] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[19] T. Qi, H. Wang, and Y. Huang, "Towards the robustness of differentially private federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 18, 2024, pp. 19 911–19 919.

[20] H. Batool, A. Anjum, A. Khan, S. Izzo, C. Mazzocca, and G. Jeon, "A secure and privacy preserved infrastructure for vanets based on federated learning with local differential privacy," *Information Sciences*, vol. 652, p. 119717, 2024.

[21] S. Shi, C. Hu, D. Wang, Y. Zhu, and Z. Han, "Distributionally robust federated learning for differentially private data," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 842–852.

[22] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source inference attacks in federated learning," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1102–1107.

[23] K. Wei, J. Li, C. Ma, M. Ding, W. Chen, J. Wu, M. Tao, and H. V. Poor, "Personalized federated learning with differential privacy and convergence guarantee," *IEEE Transactions on Information Forensics and Security*, 2023.

[24] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, and H. V. Poor, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3388–3401, 2021.

[25] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[26] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," *arXiv preprint arXiv:1812.00984*, 2018.

[27] F. E. Dorner, N. Konstantinov, G. Pashaliev, and M. Vechev, "Incentivizing honesty among competitors in collaborative learning and optimization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[28] Z. Wang, X. Fan, J. Qi, C. Wen, C. Wang, and R. Yu, "Federated learning with fair averaging," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 1615–1623, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2021/223

[29] D. Y. Zhang, Z. Kou, and D. Wang, "Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 1051–1060.

[30] B. R. Gálvez, F. Granqvist, R. van Dalen, and M. Seigel, "Enforcing fairness in private federated learning via the modified method of differential multipliers," in *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[31] J. Li, T. Zhu, W. Ren, and K.-K. Raymond, "Improve individual fairness in federated learning via adversarial training," *Computers & Security*, vol. 132, p. 103336, 2023.

[32] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, and A. S. Avestimehr, "Fairfed: Enabling group fairness in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7494–7502.

[33] H. Zhao and G. J. Gordon, "Inherent tradeoffs in learning fair representations," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 2527–2552, 2022.

[34] F. Fioretto, C. Tran, P. Van Hentenryck, and K. Zhu, "Differential privacy and fairness in decisions and learning tasks: A survey," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 5470–5477, survey Track. [Online]. Available: https://doi.org/10.24963/ijcai.2022/766

[35] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[36] A. M. Abdelmoniem, C.-Y. Ho, P. Papageorgiou, and M. Canini, "A comprehensive empirical study of heterogeneity in federated learning," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 071–14 083, 2023.

[37] ——, "Empirical analysis of federated learning in heterogeneous environments," in *Proceedings of the 2nd European Workshop on Machine Learning and Systems*, 2022, pp. 1–9.

[38] W. Du, D. Xu, X. Wu, and H. Tong, "Fairness-aware agnostic federated learning," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 2021, pp. 181–189.

[39] R. Cummings, V. Gupta, D. Kimpara, and J. Morgenstern, "On the compatibility of privacy and fairness," in *Adjunct publication of the 27th conference on user modeling, adaptation and personalization*, 2019, pp. 309–315.

[40] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[41] M. Padala and S. Gujar, "Fnnc: Achieving fairness through neural networks," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI-20}, International Joint Conferences on Artificial Intelligence Organization*, 2020.

[42] M. Jagielski, M. Kearns, J. Mao, A. Oprea, A. Roth, S. Sharifi-Malvajerdi, and J. Ullman, "Differentially private fair learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3000–3008.

[43] D. Xu, S. Yuan, and X. Wu, "Achieving differential privacy and fairness in logistic regression," in *Companion proceedings of The 2019 world wide web conference*, 2019, pp. 594–599.

[44] J. Ding, X. Zhang, X. Li, J. Wang, R. Yu, and M. Pan, "Differentially private and fair classification via calibrated functional mechanism," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 622–629.

[45] H. Mozannar, M. Ohannessian, and N. Srebro, "Fair learning with private demographic data," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7066–7075.

[46] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 19–30.

[47] J. Zhang, K. Zheng, W. Mou, and L. Wang, "Efficient private erm for smooth objectives," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3922–3928. [Online]. Available: https://doi.org/10.24963/ijcai.2017/548

[48] 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, 2020.
[49] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: Federated learning with local differential privacy," in *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, 2020, pp. 61–66.
[50] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.
[51] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization." *Journal of Machine Learning Research*, vol. 12, no. 3, 2011.
[52] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
[53] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography Conference*. Springer, 2016, pp. 635–658.
[54] D. Csiba and P. Richtárik, "Global convergence of arbitrary-block gradient methods for generalized polyak-{\L} ojasiewicz functions," *arXiv preprint arXiv:1709.03014*, 2017.
[55] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2650–2654.
[56] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.
[57] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *International conference on machine learning*. PMLR, 2014, pp. 1000–1008.
[58] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[59] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
[60] Kaggle. (2021) Lending club loan data. [Online]. Available: https://www.kaggle.com/datasets/adarshsng/lending-club-loan-data-csv
[61] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International conference on machine learning*. PMLR, 2021, pp. 6357–6368.

Chuang Hu received his BS and MS degrees from Wuhan University in 2013 and 2016. He received his Ph.D. degree from the Hong Kong Polytechnic University in 2019. He is currently an Associate Professor in the School of Computer Science at Wuhan University. His research interests include edge learning, federated learning/analytics, and distributed computing.

Nanxi Wu received the B.S. degree in software engineering from Central South University. She is currently a master student at the School of Computer Science, Wuhan University. Her research interests include edge computing, federated Learning and analytics.

Siping Shi received the BS degree in computer science from Sichuan University in 2014, and the MS degree in computer applied technology from the University of Chinese Academy of Sciences in 2017. She is currently working toward the PhD degree with The Hong Kong Polytechnic University. Her research interests include edge computing, federated learning and analytics.

Xuan Liu is an undergraduate student majoring in Electronic and Information Engineering at the Hong Kong Polytechnic University. Her research interests include Trustworthy AI, Federated Learning, and Privacy. Xuan is currently working as a research assistant at the University of British Columbia, Canada.

Wenhan Wu received his B.S. degree in Computer Science and Technology from Wuhan University in 2023. He is currently pursuing a M.S. in Computer Science and Technology at the same university. His research interests include edge computing, machine learning and federated learning/analytics.

Bing Luo is an Assistant Professor of Data and Computational Science at Duke Kunshan University. Before that, he was a joint Postdoctoral Research Fellow at AIRS, The Chinese University of Hong Kong, Shenzhen, and Yale University during 2020-2022. He received the Ph.D. degree from the Department of Electrical and Electronic Engineering, University of Melbourne (UNIMELB), Australia, in 2020, and the M.S. degrees in 2012 from Beijing University of Posts and Telecommunications (BUPT), China. His research interests are federated and edge learning, network optimization, game theory, and 5G/6G wireless communications.

Ye Wang received the B.S. degree in microelectronics from Peking University, and the M.S. degree in robotics and the Doctorate of Science degree from ETH Zürich. He is currently an Assistant Professor at the University of Macau. His research interests include blockchain, financial technology, human-computer interaction, and security.

Jiawei Jiang received his PhD degree in computer science from Peking University, China, in 2018. He is currently a professor with the School of Computer Science, Wuhan University, China. His research interests include database, big data management and analytics, and machine learning systems.

Dazhao Cheng (Senior Member, IEEE) received his BS and MS degrees in Electrical Engineering from the Hefei University of Technology in 2006 and the University of Science and Technology of China in 2009. He received his Ph.D. from the University of Colorado at Colorado Springs in 2016. He was an AP at the University of North Carolina at Charlotte in 2016-2020. He is currently a professor in the School of Computer Science at Wuhan University. His research interests include big data and cloud computing.