Incentive Mechanism Design for Federated Learning with Multi-Dimensional Private Information

Ningning Ding*, Zhixuan Fang*, and Jianwei Huang†‡*
*Department of Information Engineering, The Chinese University of Hong Kong
†School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen
‡Shenzhen Institute of Artificial Intelligence and Robotics for Society

***Abstract*—As an emerging machine learning technique, federated learning has received significant attention recently due to its promising performance in mitigating privacy risks and costs. In federated learning, the model training is distributed over users and coordinated by a central server. Users only need to send the most updated learning model parameters to the server without revealing their private data. While most of the existing work of federated learning focused on designing the learning algorithm to improve the training performance, the incentive issue for encouraging users' participation is still under-explored. Such a fundamental issue can significantly affect the training efficiency, effectiveness, and even the practical operability of federated learning. This paper presents an analytical study on the server's optimal incentive mechanism design, in the presence of users' multi-dimensional private information including training cost and communication delay. Specifically, we consider a multi-dimensional contract-theoretic approach, with a key contribution of summarizing users' multi-dimensional private information into a one-dimensional criterion that allows a complete order of users. We further perform the analysis in three different information scenarios to reveal the impact of the level of information asymmetry on server's optimal strategy and minimum cost. We show that weakly incomplete information does not increase the server's cost. However, the optimal mechanism design under strongly incomplete information is much more challenging, and it is not always optimal for the server to incentivize the group of users with the lowest training cost and delay to participate.**

***Index Terms*—federated learning, incentive mechanism, multi-dimensional contract, information asymmetry**

I. Introduction

A. Background and Motivations

The unprecedented amount of data generated by users' mobile devices has a great potential in powering intelligent learning models in many aspects of our life. However, the privacy concerns from users often make it risky (or even illegal) to store all the users' data in a centralized location. This motivates the emergence of federated learning, which can enable effective learning while protecting users' privacy.

A typical federated learning application platform (e.g., Google Keyboard, or Gboard in short) usually consists of (i) a population of users who use their local data to collaboratively train a shared learning model and (ii) a central server who coordinates the training. Specifically, each user computes the updated parameters of the global learning model based on his local data and sends the parameters to the server repeatedly; the server iteratively updates the global model based on the users' inputs and feeds the aggregated global model back to users. Users and server repeat this process until a desirable model accuracy is achieved, e.g., the training error is smaller than a threshold [1]. Different from a traditional centralized model training where the central server acquires and stores users' raw data, federated learning allows users to keep the local data on their own devices and only share the intermediary model parameters, which well protects users' data privacy.

However, with all the promising benefits, federated learning also comes with challenges to tackle. First, most existing studies usually make an optimistic assumption that users are willing to participate in the training process (e.g., [2]). That may not be realistic without proper incentives, as users incur various costs during the training process [3]. Second, the server can selectively incentivize appropriate users' participation to enhance training efficiency as well as effectiveness (e.g., [4]). However, this may not be easy to achieve if the server does not know the information regarding users' communication delay and training costs. The communication delay depends on each user's device configuration and time availability, which are often unknown to the server, especially when there are a large number of heterogeneous users in federated learning. Moreover, the training costs are also users' private information and will not be easily accessible by the server due users' privacy concerns. The above two problems motivate our first key question in this paper: *How to incentivize users with multi-dimensional private information to train the federated learning model truthfully and efficiently?*

Although the server may not know each user's private information, it may have the knowledge about statistics of such information through market research [5]. For example, the server may know the numbers of different types of users (which is denoted as *weakly incomplete information*) or only knows the user type distribution (which is denoted as *strongly incomplete information*). Different levels of information asymmetry require the server to design different optimal strategies to achieve the highest possible model accuracy with the lowest possible costs. This motives our second key question: *How does the server's knowledge of users' private information influence its strategy and cost?*

B. Contributions

We summarize our key contributions as follows:

- *Incentive issue with multi-dimensional private information.* To the best of our knowledge, this is one of the first ana-

This work is supported by the Shenzhen Institute of Artificial Intelligence and Robotics for Society, the General Research Fund CUHK 14219016 from Hong Kong UGC, and the Presidential Fund from the Chinese University of Hong Kong, Shenzhen.

ISBN 978-3-903176-29-4

lytical studies on a multi-dimensional incentive mechanism design for federated learning, considering different levels of information asymmetry. We model the server's trade-off between model accuracy and payment to users, and show that such a trade-off eventually translates into the trade-off between different dimensions of users' private information.
- *Multi-dimensional contract and server preference characterization.* We analytically solve the server's optimal contract design problem, which is challenging in the presence of users' multi-dimensional private information. Specifically, we are able to project two-dimensional information into a one-dimensional criterion, which characterizes the server's complete ordered preference of different users.
- *Investigation on effect of multiple information asymmetry levels.* We reveal the influence of information asymmetry level on the optimal contract, and show that the complexity of contract design increases with information incompleteness. We demonstrate that: 1) comparing with the complete information benchmark, weakly incomplete information does not increase the server's cost, but strongly incomplete information does; 2) choosing the group of users with lowest training cost and delay is not always optimal for the server when information is strongly incomplete.

C. Related Work

Studies on federated learning started in 2016, and most literature has focused on improving training efficiency and effectiveness (e.g., [6]), enhancing security (e.g., [7]), and preserving privacy (e.g., [8]). Most of the results are derived under an optimistic assumption that users are willing to participate in the federated learning, which may not be realistic without proper incentives to the users.

A carefully designed incentive mechanism can elicit honest behaviors of data owners and enhance training efficiency in federated learning [4]. Although federated learning has been increasingly widely implemented in practice, there are only a few important earlier work on the incentive mechanism design, with a few limitations. First, these existing work usually modeled the server's profit along one dimension, e.g., time consumption (e.g., [2], [9]) or training data size (e.g., [10]). Second, these literature did not consider various possible information scenarios. Specifically, Kang et al. [2] only considered the weakly incomplete information where the server knows the number of different types of users, while Sarikaya et al. [9] and Feng et al. [10] assumed a complete information scenario where the server knows the private information of users (e.g., costs). Third, these studies assumed that the server can only make one-dimension decision, which is not flexible enough when users have multi-dimensional private information. Building upon these earlier work, we consider a more general and practical model of multi-dimensional private information, provide a more comprehensive mechanism design, and investigate the effect of information completeness.

II. System Model

We consider a typical federated learning platform (e.g., the popular Gboard system) where the model training is distributed over N users and coordinated by a central server. To simulate users' participation under incomplete information, we will propose a contract-based incentive mechanism where the server provides a set of contract items for each user to choose. In the following, we first introduce the federated learning process, then formulate the contract, and finally specify the users' payoff and the server's cost, respectively.

A. Federated Learning Process

As an illustrative example, Gboard is a Google keyboard software which relies federated learning to help users predict the next word (to be typed) based on the current word (that has been just typed). Since the typing data from each mobile user is limited, Gboard relies the data from millions of users to achieve an effective prediction. It asks users to use their local data about input behaviors to cooperatively train a global learning model. Each user only needs to share model parameters with the server without uploading his raw data.

Specifically, consider an example of data (x_i, y_i), where x_i is the input (e.g., a word entered by a user on the keyboard) and y_i is the label (e.g., the word entered by the user following x_i). The objective of learning is to find the proper model parameter w that can predict the label y_i based on the input x_i. Denote the prediction value as $\tilde{y}(x_i; w)$. The gap between the prediction $\tilde{y}(x_i; w)$ and the ground truth label y_i is characterized by the prediction loss function $f_i(w)$. If user k uses a set $\mathcal{S}_k$ of data with data size s_k to train the model, the loss function of user k is the average prediction loss on all data $i \in \mathcal{S}_k$, i.e.,

$$F_k(w) = \frac{1}{s_k} \sum_{i \in \mathcal{S}_k} f_i(w).$$

The optimal model parameter w^* minimizes global loss function, which is a weighted average of all users' loss functions:

$$w^* = \arg\min_w f(w) = \arg\min_w \sum_{k=1}^{N} \frac{s_k}{s} F_k(w), \qquad (1)$$

where s is the total data size of all users [1].

We consider the widely adopted synchronous update scheme that proceeds in rounds of communication, i.e., all users enter a new global training round simultaneously; the server sends the global parameter to all users at the same time and waits for all users' updates. The key advantage of the synchronous algorithms is that they have provable convergence (e.g., [4], [11]). A typical synchronous federated learning algorithm with one-step local update works as Algorithm 1 [1].

Users have to perform both communication and computation in the federated learning. Communication usually takes time. McMahan et al. [1] shows that mobile users usually have a limited upload bandwidth and may even wait for some time before uploading. Meanwhile, computation time becomes shorter and shorter, as each user's on-device dataset is small compared to the total dataset size and modern mobile phones have relatively fast processors. In this paper, we focus on the synchronous federated learning that each user only conducts one step of gradient update in each round, in which case we can reasonably assume that communication time dominates in each round of training. We will discuss the more general case

Algorithm 1: Synchronous federated learning

Input : Number of iterations D, learning rate η, number of users N, and each user's data.
Output: Model parameter w_D

initialize w_0
for *round* $d = 0$; $d < D$; $d++$ **do**
  Server executes:
  select a set $\mathcal{K}$ of users
  send current global parameter w_d to users
  Each user $k \in \mathcal{K}$ **executes:**
  compute local parameter: $w_{d+1}^k \leftarrow w_d - \eta \nabla F_k(w_d)$
  return w_{d+1}^k to server
  Server executes:
  aggregate all users' updates: $w_{d+1} \leftarrow \sum_{k \in \mathcal{K}} \frac{s_k}{s} w_{d+1}^k$
end

of multiple-step local updates in Appendix XI of the technical report [12]. Moreover, we assume that the powerful server has a large enough bandwidth, so that the communications from multiple users to the server do not interfere with each other.

Because users will suffer time/energy costs due to the model training, the sever needs to properly incentivize users' participation by providing rewards. An effective incentive mechanism usually offers heterogeneous rewards for different types of users.

B. User's Types

We consider a population of N users on the federated learning platform. Users are distinguished by two-dimensional private information: the marginal data-usage cost θ and the communication time t. For the convenience of presentation, we refer to a user with $\pi_i \triangleq (\theta_i, t_i)$ as a type-i user. We consider all users belonging to a set $\mathcal{I} = \{1, ..., I\}$ of I types. Each type $i \in \mathcal{I}$ has N_i users, with $\sum_{i \in \mathcal{I}} N_i = N$. Though each user could have data different from others, we assume that the data is i.i.d. among all users and each user's type does not change in the entire training process.

In the presence of users' private information, it is difficult for the server to predict the users' behaviors without complete information. To this end, we propose to design a contract mechanism to elicit the private information.

C. Contract Formulation

Contract theory is a promising and widely adopted theoretic tool for dealing with problems with private information. Therefore, we propose a contract theoretic framework to tackle the incentive mechanism design problem.

1) Server's Contract: The server will propose a contract that specifies the relationship among users' communication time, training data size, and reward for the entire training process. Specifically, the contract $\mathcal{C} = (t_{\max}, \boldsymbol{\phi})$ contains a maximum communication time $t_{\max}$ (for all user types) and I contract items $\boldsymbol{\phi} = \{\phi_i\}_{i \in \mathcal{I}}$ (one for each type). The term $t_{\max}$ is the maximum communication time in each global round set by the server for all users, i.e., users with $t_i \leq t_{\max}$ are able to finish the transmission of the parameters in time. Each contract item $\phi_i \triangleq (s_i, r_i)$ specifies the relationship between each type-i user's data size and reward. The term s_i is the required training data size for each type i user in each global round. The term r_i is the reward (e.g., money) for each type i user in each global round, if the user completes the training task with required time and data size[1]. The server offers a zero contract item for any user type i with $t_i > t_{\max}$.

2) Users' Choices: At the beginning of the training process, each user decides whether to participate in the training and (if yes) which contract item to choose. If a user chooses the contract item ϕ_i, it needs to use s_i data examples to train the model and sends local updates to the server in time $t_{\max}$. In return, it will get r_i reward in this global round. Users will not participate if their payoff (defined in Section II-D) is negative.

Under such a contract, we specify the users' payoff in Section II-D and the server's cost in Section II-E.

D. Users' Payoff

Each user's payoff in each global round is the difference between the reward offered by the server and the cost of data usage in model training.

We assume that a user's training cost (e.g., time and energy costs) is proportional to the used data size, i.e., $\theta_i s_i$ [3]. Hence, if a type-i user chooses the contract item ϕ_i, his payoff is[2]:

$$U(\theta_i, t_i, \phi_i) = \begin{cases} r_i - \theta_i s_i, & \text{if } t_i \leq t_{\max}, \\ -\theta_i s_i, & \text{if } t_i > t_{\max}. \end{cases} \quad (2)$$

We assume that in each global iteration, every user locally performs one step of mini-batch stochastic gradient decent (SGD) to compute the model parameters. Thus from the global perspective, it is equivalent to a mini-batch SGD with batch size $B = \sum_{i \in \mathcal{I}} \mathbb{1}_{t_i \leq t_{\max}} N_i s_i$, where

$$\mathbb{1}_{t_i \leq t_{\max}} = \begin{cases} 1, & \text{if } t_i \leq t_{\max}, \\ 0, & \text{if } t_i > t_{\max}, \end{cases}$$

means that only users with $t_i \leq t_{\max}$ are eligible to train the model. Note that we are able to analyze the general case where users perform multiple steps of local updates and the results turn out to be similar to the one-step case. Due to space limit, both analysis and results of the general case are given in Appendix XI of the technical report [12].[3]

E. Server's Cost

With a fixed training time, the server's cost is determined by the accuracy loss of global model and the total payment to users.

[1]The server is able to know each user's training data size which is the weight in parameter aggregation step (i.e., (1)) of federated learning [13].

[2]Since we consider synchronous federated learning, our model can be applied to the case where users' payoffs include an additional homogeneous time cost term $\alpha t_{\max}$. Such a time cost only makes the optimal rewards increase by a constant. Thus, we normalize the time cost to zero.

[3]Considering one-step update is for the convenience of modeling the global training accuracy in Section II-E, so that we can derive explicit solutions as well as comprehensible insights. That is because there is no theoretical results of the global accuracy of federated learning in the presence of users' multiple updates in each global iteration. However, this assumption is not restrictive. First, if users perform multiple steps of local updates, we are able to derive similar results by using a general accuracy function $f(S, E, K)$, where S is the total training data size, E is the number of data passes, K is the number of global iterations, and $f(S, E, K)$ is a convex decreasing function [14], [15]. Second, we will show in simulation (Fig. 5) that even if users perform multiple steps of updates in each global iteration, our proposed mechanism still has a good performance.

First, we characterize the expected accuracy loss of the global model. We use T to denote the total training time. Thus, the number of global iterations is denoted by $D = T/t_{\max}$. The model accuracy loss after D rounds is measured by the difference between the prediction loss with parameter w_D and that with the optimal parameter w^*, i.e., $f(w_D) - f(w^*)$ (defined in Section II-A). The expected difference is bounded by $O(1/\sqrt{BD} + 1/D)$ when users use mini-batch SGD [15], [16], where B is the batch size. Thus, server's expected loss in model accuracy decreases as the number of iterations D and batch size B increase. Note that a smaller value means less expected accuracy loss.

Next, we consider the server's total payment to all users in the entire training process. If all users choose the respective contract items, the total payment is the product of the number of global iterations and the payment to all users in each iteration, i.e., $D \cdot \sum_{i\in\mathcal{I}} \mathbb{1}_{t_i \le t_{\max}} N_i r_i$.

To summarize, the server's cost is:

$$W(t_{\max}, \boldsymbol{\phi}) = \gamma_1 \min\{\left(\frac{1}{\sqrt{\frac{T}{t_{\max}} \sum_{i\in\mathcal{I}} \mathbb{1}_{t_i \le t_{\max}} N_i s_i}} + \frac{t_{\max}}{T}\right), C\} + \gamma_2 \frac{T}{t_{\max}} \sum_{i\in\mathcal{I}} \mathbb{1}_{t_i \le t_{\max}} N_i r_i. \quad (3)$$

The first term on the right hand side of (3) characterizes the server's expected loss in accuracy, where γ_1 indicates the server's valuation on accuracy loss. The second term on the right hand side of (3) represents the server's payment to users, where γ_2 indicates the server's valuation on payment. We use $C \in (\frac{1}{\sqrt{\frac{T}{t_{\max}}}} + \frac{t_{\max}}{T}, \infty)$ to characterize the server's finite (and possibly large) accuracy loss when there is no data for training (i.e., $\sum_{i\in\mathcal{I}} \mathbb{1}_{t_i \le t_{\max}} N_i s_i = 0$).

III. Optimal Multi-dimensional Contract Design

In this section, we analyze the server's optimal incentive mechanism. To understand the impact of incomplete information, we consider three information scenarios:

1) *Complete information scenario (benchmark):* The server knows each user's type. This provides a lower bound of the server's minimum cost for all information scenarios.
2) *Weakly incomplete information scenario:* The server knows the total number of users as well as the specific number of each user type, but does not know which user belongs to which type.
3) *Strongly incomplete information scenario:* The server knows the total number of users and the distribution of user types, but does not know the specific number of each user type.

In each scenario, we first derive the condition for a feasible contract, and then characterize the optimal contract. Feasibility and optimality of the contract are defined as follows:

Definition 1 (Contract Feasibility)**.** *A contract is* feasible *if each user achieves the maximum payoff under the contract item designed for his type.*

Definition 2 (Contract Optimality)**.** *A contract is* optimal *if it minimizes the server's cost among all feasible contracts.*

A. Complete Information Scenario

In this subsection, we study the server's optimal contract in the scenario where the server knows the type of each user. This makes it possible for the server to monitor and make sure that each type of users accepts will not accept any contract item not designed for that type. Even in this case, the server still needs to ensure that each user achieves a non-negative payoff, so that the user will accept the corresponding contract item. In other words, a contract is feasible if and only if it satisfies Individual Rationality (IR) constraints:

Definition 3 (Individual Rationality)**.** *A contract is individually rational if each type-i user receives a non-negative payoff by accepting the contract item ϕ_i intended for his type, i.e.,*

$$U(\theta_i, t_i, \phi_i) \ge 0, \forall i \in \mathcal{I}. \quad (4)$$

Thus, in the complete information scenario, the optimal contract $\mathcal{C}^{opt}_{complete} = (t^*_{\max}, \boldsymbol{\phi}^*)$ is the solution to the following optimization problem:

Problem 1 (Contract Design under Complete Information)**.**

$$\begin{aligned} \min_{t_{\max}, \boldsymbol{\phi}} \quad & W(t_{\max}, \boldsymbol{\phi}) \\ s.t. \quad & \text{IR Constaints in (4).} \end{aligned} \quad (5)$$

We will solve Problem 1 in two steps. First, for any given data size s_i, we derive the server's optimal reward $r^*_i(s_i)$ (Lemma 1). Second, we substitute the optimal reward $r^*_i(s_i)$ into the server's objective function and derive the optimal data size s^*_i as well as the optimal maximum communication time $t^*_{\max}$ (Theorem 1).

Lemma 1. *For any given data size s_i (even if it is not optimal), it is optimal for the server to choose the reward as $r^*_i(s_i) = \theta_i s_i,\ \forall i \in \mathcal{I}$.*

Proof of Lemma 1 is given in Appendix I of the technical report [12]. Lemma 1 shows that the server will design the contract such that all users get a zero payoff in the complete information scenario.

Based on Lemma 1, we can derive the optimal data size for each type that minimizes the server's cost. To illustrate the impact of choosing each user based on the server's cost, we have the following lemma:

Lemma 2. *The server's cost of only choosing type i is*

$$G(\theta_i, t_i) \triangleq \frac{\gamma_1 t_i}{T} + \left(2^{\frac{1}{3}} + 2^{-\frac{2}{3}}\right) \gamma_2{}^{\frac{1}{3}} \gamma_1{}^{\frac{2}{3}} \theta_i^{\frac{1}{3}}, \quad (6)$$

Proof of Lemma 2 is given in Appendix II of the technical report [12]. Lemma 2 characterizes the server's trade-off between users' different dimensions of private information (i.e., θ and t). Thus, we can transform users' two-dimensional private information into a one-dimensional criterion, which indicates the server's preference on different user types:

Definition 4 (Preference)**.** *The server has a higher preference on type j than type i (denoted by $j \succ i$) if and only if*

$$G(\theta_j, t_j) < G(\theta_i, t_i).$$

Fig. 1 illustrates how the server's preference on user types changes over the parameter space of (θ, t). More specifically,

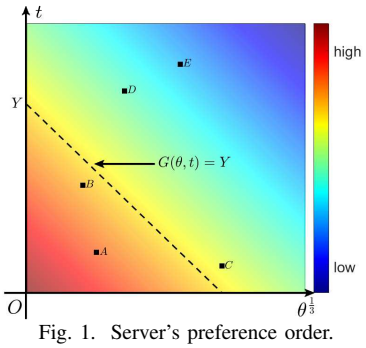

Fig. 1. Server's preference order.

the server's preference on users' types decreases from the red area (low cost and delay) to the blue area (high cost and delay). The server has the same preference on users whose $(\theta_i^{\frac{1}{3}}, t_i)$ on the same line $G(\theta, t) = Y$, where Y is an arbitrary constant. Among all I user types, we will denote the set of user types which have the same highest server preference as follows:

$$\mathcal{J}^{prefer} \triangleq \arg\min_{j \in \mathcal{I}} G(\theta_j, t_j).$$

For example, suppose that there are five user types A, B, C, D, and E with (θ, t) shown in Fig 1. The server's preference order is $A \succ B \succ C \succ D \succ E$ and $\mathcal{J}^{prefer} = \{A\}$.

Theorem 1 characterizes the optimal contract for the server in the complete information scenario under different cases of the set $\mathcal{J}^{prefer}$:

Theorem 1. *In the complete information scenario,*

1) if $\mathcal{J}^{prefer} = \{j\}$, the server's optimal contract is $t^*_{\max} = t_j$, $\phi_j^* = (\frac{1}{N_j \frac{T}{t_j}[\frac{2\gamma_2\theta_j}{\gamma_1}]^{\frac{2}{3}}}, \frac{\theta_j}{N_j \frac{T}{t_j}[\frac{2\gamma_2\theta_j}{\gamma_1}]^{\frac{2}{3}}})$, and $\phi_i^* = \mathbf{0}, \forall i \neq j$.

2) if $|\mathcal{J}^{prefer}| > 1$, the server's optimal contract is to select any one type $j \in \mathcal{J}^{prefer}$ with $t^*_{\max} = t_j$, $\phi_j^* = (\frac{1}{N_j \frac{T}{t_j}[\frac{2\gamma_2\theta_j}{\gamma_1}]^{\frac{2}{3}}}, \frac{\theta_j}{N_j \frac{T}{t_j}[\frac{2\gamma_2\theta_j}{\gamma_1}]^{\frac{2}{3}}})$, and $\phi_i^* = \mathbf{0}, \forall i \neq j$.

3) if $|\mathcal{J}^{prefer}| > 1$, offering the only positive contract to one type $j \in \mathcal{J}^{prefer}$ leads to the same minimum server cost.

Proof of Theorem 1 is given in Appendix III of the technical report [12]. Theorem 1 shows that the server only provides a positive contract item for the single most preferred user type and offers the same zero contract item for all other user types. Moreover, the optimal contact always exists but may not be unique, as the most preferred type may not be unique. On the other hand, it is never optimal to select (provide a positive contract item) to multiple user types, even if they all belong to the set $\mathcal{J}^{prefer}$, as this would increase the cost of the server.

Intuitively, having less information would lead to a different behavior of the server. However, we will show in next section that server's optimal contract under weakly incomplete information is the same as that in complete information scenario.

B. Weakly Incomplete Information Scenario

In this subsection, we study the server's optimal contract in the weakly incomplete information scenario. The server does not know which user belongs to which type, but knows the specific number of each user type (i.e., $N_i, \forall i \in \mathcal{I}$).

Since the server cannot force a user to accept certain contract item in this case, it needs to design the contract to further ensure the Incentive Compatibility (IC) constraints:

Definition 5 (Incentive Compatibility)**.** *A contract is incentive compatible if each type-i user maximizes his own payoff by choosing the contract item ϕ_i intended for his type, i.e.,*

$$U(\theta_i, t_i, \phi_i) \geq U(\theta_i, t_i, \phi_j), \forall i, j \in \mathcal{I}. \tag{7}$$

The optimal contract $\mathcal{C}^{opt}_{W-incomplete} = (t^*_{\max}, \boldsymbol{\phi}^*)$ under weakly incomplete information is the solution to Problem 2:

Problem 2 (Contract Design under Weakly Incomplete Information)**.**

$$\begin{aligned} \min_{t_{\max}, \boldsymbol{\phi}} \quad & W(t_{\max}, \boldsymbol{\phi}) \\ s.t. \quad & \text{IR Constaints in (4), IC Constaints in (7).} \end{aligned} \tag{8}$$

As the total number of IR and IC constraints is I^2, it is quite complex to solve Problem 2 directly. In the following, we first transform IR and IC constraints into a smaller number of equivalent constraints (Lemma 3). Then, for any given data size s_i, we derive the server's optimal reward $r_i^*(s_i)$ (Lemma 4). Finally, we derive the optimal data size s_i^* and the optimal maximum communication time $t^*_{\max}$ (Theorem 2).

We use $\mathcal{I}'$ to denote the set of user types with communication time no larger than $t_{\max}$, i.e., $\mathcal{I}' = \{i | t_i \leq t_{\max}\}$. We denote $I' = |\mathcal{I}'|$ and reindex the user types in $\mathcal{I}'$ by $\{i_{\mathcal{I}'}\}_{i \in \{1,...,I'\}}$ in the ascending order of marginal cost θ, because as long as the communication time t is no larger than $t_{\max}$, it does not matter anymore. Lemma 3 characterizes contract feasibility:

Lemma 3. *Under weakly incomplete information, a contract $\mathcal{C} = (t_{\max}, \boldsymbol{\phi})$ is feasible if and only if the followings are true:*

a) for user types in $\mathcal{I}'$, the contract items satisfy the following three conditions:

a.1) $r_{I'_{\mathcal{I}'}} - \theta_{I'_{\mathcal{I}'}} s_{I'_{\mathcal{I}'}} \geq 0$;

a.2) $r_{1_{\mathcal{I}'}} \geq ... \geq r_{I'_{\mathcal{I}'}} \geq 0$ and $s_{1_{\mathcal{I}'}} \geq ... \geq s_{I'_{\mathcal{I}'}} \geq 0$;

a.3) $r_{i+1_{\mathcal{I}'}} + \theta_{i_{\mathcal{I}'}}(s_{i_{\mathcal{I}'}} - s_{i+1_{\mathcal{I}'}}) \leq r_{i_{\mathcal{I}'}} \leq r_{i+1_{\mathcal{I}'}} + \theta_{i+1_{\mathcal{I}'}}(s_{i_{\mathcal{I}'}} - s_{i+1_{\mathcal{I}'}})$, $i \in \{1, ..., I'\}$.

b) for any user type $i \notin \mathcal{I}'$, $s_i = r_i = 0$.

Proof of Lemma 3 is given in Appendix IV of the technical report [12].

Constraint $(a.1)$ ensures that each type of users can get a non-negative payoff by accepting the contract item of type-$I'_{\mathcal{I}'}$ users (with maximum marginal cost $\theta_{I'_{\mathcal{I}'}}$ in $\mathcal{I}'$) as $r_{I'_{\mathcal{I}'}} - \theta_{i_{\mathcal{I}'}} s_{I'_{\mathcal{I}'}} \geq r_{I'_{\mathcal{I}'}} - \theta_{I'_{\mathcal{I}'}} s_{I'_{\mathcal{I}'}} \geq 0, \forall i \in \{1, ..., I'\}$. This corresponds to the IR constraints. Both constraints $(a.2)$ and $(a.3)$ are related to IC constraints. Constraint $(a.2)$ shows that the server should request more data from a user type with a lower marginal cost and provide more reward in return. Constraint $(a.3)$ characterizes the relationship between any two neighbor contract items. The results in (b) mean that users with $t_i > t_{\max}$ cannot finish the communication in time, so the required data size and reward in contract are zero.

Based on Lemma 3, Lemma 4 characterizes the optimal rewards for any feasible data size:

Lemma 4. *For any given data size $\boldsymbol{s} = \{s_i\}_{i \in \mathcal{I}}$ (even if it is not optimal), it is optimal to choose reward satisfy:*

- *for any user type $i \in \mathcal{I}'$,*

$$r_i^*(\boldsymbol{s})=\begin{cases}\theta_i s_i, & \text{if } i=I'_{\mathcal{I}'};\\ \theta_i s_i+\sum_{j=i+1_{\mathcal{I}'}}^{I'_{\mathcal{I}'}}(\theta_j-\theta_{j-1})s_j, & \text{if } i=1_{\mathcal{I}'},...,(I'-1)_{\mathcal{I}'}.\end{cases}$$

- *for any user type $i \notin \mathcal{I}'$, $r_i^*(\boldsymbol{s})=0$.*

Proof of Lemma 4 is given in Appendix V of the technical report [12]. Based on Lemma 3 and Lemma 4, we can significantly simplify Problem 2. The following theorem characterizes the server's optimal contract in weakly incomplete information scenario:

Theorem 2. *Under weakly incomplete information, the server's optimal contract $\mathcal{C}^{opt}_{W-incomplete}$ is the same as that in complete information scenario in Theorem 1, i.e., $\mathcal{C}^{opt}_{complete}$.*

Proof of Theorem 2 is given in Appendix VI of the technical report [12]. Here we discuss some intuitions about Theorem 2. Recall that in the complete information scenario, the server's optimal contract is to only choose the most preferred user type. Under weakly incomplete information, the server knows the exact number of each user type. Thus, the server can focus on designing a contract to only attract the most preferred type, so that it achieves the same minimum cost as complete information scenario. Next, we will show that when the server does not know the number of each type, it needs to design a more complex contract to deal with all possible situations.

C. Strongly Incomplete Information Scenario

In this section, we consider a scenario where the information about users' types is strongly incomplete. The server does not know the specific number of each user type, but only knows the total number of users N and the distribution of users' types, i.e., the probability of a user being type i (θ_i, t_i) as p_i.

Due to the uncertainty, the server needs to minimize its expected cost in this information scenario. Consider the case where $N_i = n_i$ for each user type i with $\sum_{i\in\mathcal{I}} n_i = N$. The probability for this case is

$$P(n_1,...,n_I)=\frac{N!p_1^{n_1}...p_{I-1}^{n_{I-1}}p_I^{N-\sum_{i=1}^{I-1}n_i}}{n_1!...n_{I-1}!(N-\sum_{i=1}^{I-1}n_i)!},$$

and the server's corresponding cost (if all users choose the respective contract items) is

$$W(t_{\max},\boldsymbol{\phi};n_1,...,n_I)=\gamma_2\frac{T}{t_{\max}}\sum_{i\in\mathcal{I}}\mathbb{1}_{t_i\le t_{\max}}n_i r_i+\gamma_1\min\{(\frac{1}{\sqrt{\frac{T}{t_{\max}}\sum_{i\in\mathcal{I}}\mathbb{1}_{t_i\le t_{\max}}n_i s_i}}+\frac{t_{\max}}{T}),C\}.$$

Then, the optimal contract $\mathcal{C}^{opt}_{S-incomplete}=(t^*_{\max},\boldsymbol{\phi}^*)$ is the solution to the following optimization problem:

Problem 3 (Contract Design under Strongly Incomplete Information)**.**

$$\min_{t_{\max},\boldsymbol{\phi}}\mathbb{E}[W(t_{\max},\boldsymbol{\phi})]=\sum_{(n_1,...,n_I)}P(n_1,...,n_I)W(t_{\max},\boldsymbol{\phi};n_1,...,n_I)$$

$s.t.$ IR Constaints in (4), IC Constaints in (7).

It is very challenging to directly solve Problem 3 analytically. First, we can show that even after simplifying Problem 3 based on Lemma 3 and Lemma 4, the new optimization problem is not necessarily convex. Second, even the problem is convex in some special cases, there is no closed-form optimal solution due to the high order polynomial equations in KKT conditions. This motivates us to consider a more tractable approach to compute a suboptimal contract.

If the server adopts the previously derived optimal contracts (under complete and weakly incomplete information) in strongly incomplete information scenario, it will have no data for training with a probability of $(1-p_j)^N$ when the user type j with positive contract item turns out to have $n_j = 0$. The server would be very likely to get the no data training cost when p_j and N are not large enough. Inspired by the structure of the previously derived optimal contracts that only choose the most preferred user type, we consider a simplified contract where the server only offers two kinds of contract items, one is positive for a group $\chi \subseteq \mathcal{I}$ of user types and the other is zero for the rest user types in $\mathcal{I}\backslash\chi$. We name such a contract structure as Two-Part Uniform (TPU) contract.

The optimal TPU contract $\mathcal{C}^{TPU,opt}_{S-incomplete}$ is the solution to the following problem:

Problem 4 (TPU Contract Design under Strongly Incomplete Information)**.**

$$\min_{t_{\max},\boldsymbol{\phi},\chi}\mathbb{E}[W(t_{\max},\boldsymbol{\phi})]=\sum_{(n_1,...,n_I)}P(n_1,...,n_I)W(t_{\max},\boldsymbol{\phi};n_1,...,n_I)$$

$s.t.$ IR Constaints in (4), IC Constaints in (7),

$$\phi_i=\phi_j>\mathbf{0},\forall i,j\in\chi;\phi_k=\mathbf{0},\forall k\in\mathcal{I}\backslash\chi.$$

The performance of the optimal TPU contract turns out to be close to the optimal contract $\mathcal{C}^{opt}_{S-incomplete}$ (i.e., the optimal solution of Problem 3), which can be shown through both analytical performance bounds and simulation results.

We denote by χ_m an arbitrary subset of user types in $\mathcal{I}$, and we denote χ^* as the type set that leads to the minimum server cost under the optimal TPU contract. In the following, we will first show the optimal TPU contract given an arbitrary type set χ_m, i.e., $\mathcal{C}^{TPU,opt}_{S-incomplete}(\chi_m)$ in Lemma 5, then evaluate the performance of $\mathcal{C}^{TPU,opt}_{S-incomplete}(\chi_m)$ in Theorem 3, and finally provide the guideline for finding the optimal type set χ^*.

First, we characterize the optimal TPU contract under type set χ_m, i.e., $\mathcal{C}^{TPU,opt}_{S-incomplete}(\chi_m)$. The probability of having n_{χ_m} users belonging to the types in χ_m is:

$$P(n_{\chi_m})=\binom{N}{n_{\chi_m}}P_{\chi_m}^{n_{\chi_m}}(1-P_{\chi_m})^{N-n_{\chi_m}},\qquad(9)$$

where $P_{\chi_m}=\sum_{i\in\chi_m}p_i$ is the probability that a user belongs to a type in χ_m. We denote by $T_{\chi_m}=\max\{t_i\}_{i\in\chi_m}$ the maximum communication time of user types in χ_m, and we denote by $\Theta_{\chi_m}=\max\{\theta_i\}_{i\in\chi_m}$ the maximum marginal cost of user types in χ_m. Lemma 5 presents the $\mathcal{C}^{TPU,opt}_{S-incomplete}(\chi_m)$:

Lemma 5. *The optimal TPU contract given an arbitrary type set χ_m under strongly incomplete information (i.e., $\mathcal{C}^{TPU,opt}_{S-incomplete}(\chi_m)$) is: $t^*_{\max}=T_{\chi_m}$,*

- *for all user types in χ_m:*

$$\phi^*=(\frac{1}{\frac{T}{T_{\chi_m}}[\frac{2\gamma_2\Theta_{\chi_m}}{\gamma_1}]^{\frac{2}{3}}}(\frac{\sum_{n_{\chi_m}=1}^{N}P(n_{\chi_m})\frac{1}{\sqrt{n_{\chi_m}}}}{\sum_{n_{\chi_m}=1}^{N}P(n_{\chi_m})n_{\chi_m}})^{\frac{2}{3}},\ \frac{\Theta_{\chi_m}}{\frac{T}{T_{\chi_m}}[\frac{2\gamma_2\Theta_{\chi_m}}{\gamma_1}]^{\frac{2}{3}}}(\frac{\sum_{n_{\chi_m}=1}^{N}P(n_{\chi_m})\frac{1}{\sqrt{n_{\chi_m}}}}{\sum_{n_{\chi_m}=1}^{N}P(n_{\chi_m})n_{\chi_m}})^{\frac{2}{3}});$$

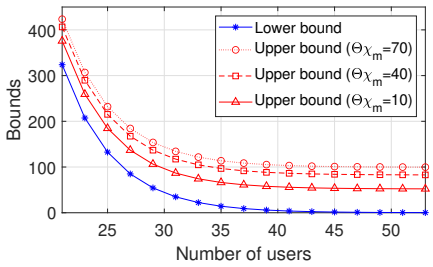

Fig. 2. Server's bounds of cost difference $\Delta W(\chi_m)$.

- *for all user types not in* χ_m, $\phi^* = \mathbf{0}$.

We use $\Delta W(\chi_m)$ to denote the cost gap between the one achieved under optimal TPU contract given χ_m $\mathcal{C}^{TPU,opt}_{S-incomplete}(\chi_m)$ and the minimum cost achieved under the optimal contract with complete information $\mathcal{C}^{opt}_{complete}(\chi_m)$ (as in Theorem 1). Such a gap is due to two reasons: (i) the strongly incomplete information, and (ii) the simplification of TPU contract. Theorem 3 shows the bounds of $\Delta W(\chi_m)$:

Theorem 3. *In the strongly incomplete information scenario, the cost difference* $\Delta W(\chi_m)$ *has the following bounds:*

- *lower bound:*
$$LB \triangleq (1-P_{\chi_m})^N \gamma_1 \left(C - \frac{T_{\chi_m}}{T}\right)$$
- *upper bound:*
$$UB \triangleq \left(2^{\frac{1}{3}}+2^{\frac{2}{3}}\right)\gamma_1^{\frac{2}{3}}\gamma_2^{\frac{1}{3}}\Theta_{\chi_m}^{\frac{1}{3}}\left[\left(1.05+\frac{\sqrt{NP_{\chi_m}}}{e^{0.02P_{\chi_m}^{-2}N}}\right)^{\frac{2}{3}}-1\right]+LB$$

Proof of Theorem 3 is given in Appendix VII of the technical report [12].

Note that the gap between the minimum cost achieved under $\mathcal{C}^{TPU,opt}_{S-incomplete}(\chi_m)$ and the one under $\mathcal{C}^{opt}_{S-incomplete}$ is no larger than $\Delta W(\chi_m)$. Theorem 3 shows that if the server adopts the optimal TPU contract, it will have a bounded cost difference compared with the complete information scenario. First, both the lower and upper bounds decrease in the number of users N. When N becomes very large (i.e., goes to infinity), the lower bound approaches 0 and the upper bound approaches the constant $(2^{\frac{1}{3}}+2^{\frac{2}{3}})\gamma_1^{\frac{2}{3}}\gamma_2^{\frac{1}{3}}\Theta_{\chi_m}^{\frac{1}{3}}[(1.05)^{\frac{2}{3}}-1]$. Moreover, the server can decrease the upper bound by choosing a type set χ_m with a lower marginal cost Θ_{χ_m} as illustrated in Fig. 2. Especially, when N is large, the upper bound approaches to the constant, which is dominated by the Θ_{χ_m} and not related to other parameters of χ_m. This provides the guideline for us to find the optimal type set χ^* under the optimal TPU contract.

Next, we derive the optimal type set χ^*. Since there is a large number of users on federated learning platform like Gboard, we first study the asymptotic behavior under a large user population:

Proposition 1. *As the number of users* N *approaches infinity, the server will only set a positive contract item for a most preferred type in* $\mathcal{C}^{TPU,opt}_{S-incomplete}$ *(i.e.,* $\chi^* = \{j\}$ *where type* j *can be any type in* $\mathcal{J}^{prefer}$*), which achieves zero cost gap (i.e.,* $\lim_{N\to\infty} \Delta W(\chi^*) = 0$*).*

Proof of Proposition 1 is given in Appendix VIII of the technical report [12]. By the law of large numbers, the empirical value of the number of a particular type users approaches the expected value computed based on the distribution when N becomes large. Thus, the server will not encounter the situation of having no training data when it only chooses the most preferred type.

Next, we present the insight about which types are in the optimal type set χ^* under any value of N[4]. We may naturally presume that the server would prefer types with higher preference (based on Definition 4). However, the following results show that choosing some user types with lower preference (while excluding other user types with higher preferences) may minimize the server's cost, which is counter-intuitive.

Proposition 2. *Under the optimal TPU contract in the strongly incomplete information scenario* $\mathcal{C}^{TPU,opt}_{S-incomplete}$*, it is possible to exist user types* i *and* j *such that* $i \in \chi^*$, $j \notin \chi^*$, *and* $G(\theta_i, t_i) > G(\theta_j, t_j)$.

Proof of Proposition 2 is given in Appendix IX of the technical report [12]. The insights behind Proposition 2 are 1) selecting a user type with higher preference may not be optimal when the existence probability of this user type is small; 2) the server's cost is determined by the maximum communication time and maximum marginal cost of user types in the type set. Thus, the combination of several high-preference types may not have a good overall performance.

IV. Numerical Experiments

In this section, we perform numerical experiments to evaluate the performance of the proposed contracts and validate our analytical results. We first present the good performance of the contracts in three information scenarios, compared with a uniform contract benchmark defined as follows (Fig. 3). Then, we show that the server does not always choose user types with higher preference under strongly incomplete information (Fig. 4). Finally, we train a federated learning model based on a realistic dataset with users' multiple local updates, to verify the robustness of our contracts' performance (Fig. 5).

Regarding the system parameters, we choose $T = 10$, $\gamma_1 = 6751.269$, $\gamma_2 = 1$, and $C = 6.2$. There are five user types with parameters $(\theta_A^{\frac{1}{3}}, t_A) = (2.6, 1.5)$, $(\theta_B^{\frac{1}{3}}, t_B) = (2.1, 4)$, $(\theta_C^{\frac{1}{3}}, t_C) = (7.1, 1.3)$, $(\theta_D^{\frac{1}{3}}, t_D) = (3.6, 7.5)$, and $(\theta_E^{\frac{1}{3}}, t_E) = (5.6, 8.5)$. The preference order is $A \succ B \succ C \succ D \succ E$. The fractions (distribution, respectively) of each type in complete and weakly incomplete (strongly incomplete, respectively) information scenario are $p_A = p_B = p_C = p_D = p_E = 0.2$.

We consider a *uniform contract* benchmark, which contains a single uniform contract item for all users. Specifically, $t^*_{\max} = t_E$, $\phi^* = (\frac{1}{N\frac{T}{t_E}[\frac{2\gamma_2\theta_C}{\gamma_1}]^{\frac{2}{3}}}, \frac{\theta_C}{N\frac{T}{t_E}[\frac{2\gamma_2\theta_C}{\gamma_1}]^{\frac{2}{3}}})$.

In Fig. 3, we compare the server's cost under three different information scenarios and the uniform contract: 1) comparing with complete information, weakly incomplete information does not increase server cost, but strongly incomplete information does; 2) the performance of the optimal TPU contract is very close to that of the optimal contract under strongly incomplete information, especially when the number of users

[4]Due to space limit, we provide detailed analysis regarding how to choose type set χ^* when N is finite in Appendix XII of the technical report [12].

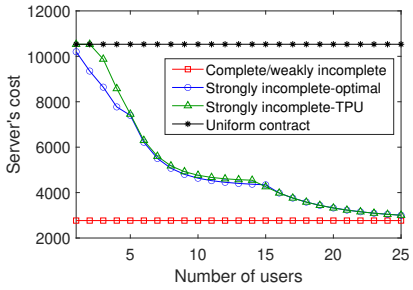

Fig. 3. Cost comparison.

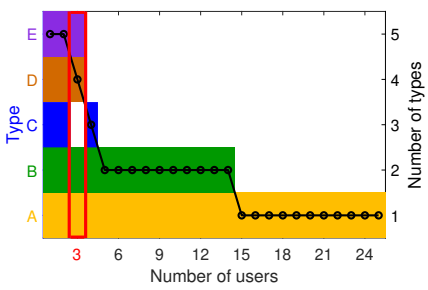

Fig. 4. Server's type set χ^* in strongly incomplete information scenario.

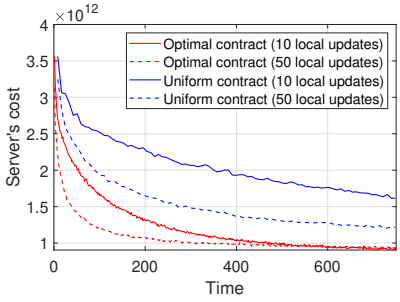

Fig. 5. Server's cost when users perform multiple updates in each global iteration under complete/weakly incomplete information.

is large; 3) all designed contracts in the three information scenarios achieve up to 73.72% cost reduction of uniform contract when N is large.

In Fig. 4, we verify our insights in Proposition 2 about the types in the optimal type set χ^* under strongly incomplete information. Interestingly, when the number of users is 3, the server chooses type A, B, D, E instead of A, B, C, D, though A, B, C, D rank top four in the order of preference. Moreover, when the total number of users decreases, the number of chosen user types increases. The server needs to ensure a high enough existence probability of chosen user types to avoid the cost of no training data, especially when N is small.

In Fig. 5, we show that the performance of our contract is robust when each user executes multiple local updates per round. Specifically, we train a federated learning model on CIFAR-10 dataset in complete/weakly incomplete information scenario[5] with $T = 750$, $\gamma_1 = 4.394 \times 10^{12}$, and $N = 500$. Other parameters remain unchanged as before. Our convolutional neural network (CNN) model consists of six 3×3 convolution layers (with 64, 64, 128, 128, 256, 256 channels, respectively, and every two followed with 2×2 max pooling), a Drop-out layer (0.5), a fully-connected layer with 10 units and ReLU activation, and a final softmax output layer. The server's cost in Fig. 5 consists of the accuracy loss in experiment and the total payment to users. Even if users perform multiple updates in each global iteration, the proposed contracts have a better performance than that of the uniform contract, up to 45.69% (37.37%, respectively) cost reduction for 10 (50, respectively) local updates per global iteration.

V. Conclusion

This paper has focused on the important issue of incentive mechanism design in federated learning. To the best of our knowledge, this is one of the first papers that deal with multi-dimensional private information for federated learning, considering different levels of information asymmetry. One of our key contributions is to identify a way to project users' two-dimensional private information into a one-dimensional criterion. It characterizes the server's complete ordered preference of different users and helps reduce the complexity of the incentive mechanism design. We have also revealed some interesting insights: First, incomplete information does not always increase the server's cost. Second, choosing user types with lower preferences (while excluding users types with higher preferences) may be optimal for the server when information is strongly incomplete.

[5]The performance of strongly incomplete information is almost the same, because the number of users N is very large in this case (Proposition 1).

References

[1] H. 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*, 2017.

[2] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, December 2019.

[3] N. Tran, W. Bao, A. Zomaya, and C. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM*, 2019.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet *et al.*, "Advances and open problems in federated learning," *arXiv:1912.04977*, 2019.

[5] Z. Wang, L. Gao, and J. Huang, "Multi-dimensional contract design for mobile data plan with time flexibility," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing - Mobihoc '18*, 2018.

[6] J. Ren, G. Yu, and G. Ding, "Accelerating dnn training in wireless federated edge learning system," *arXiv:1905.09712*, 2019.

[7] C. Fung, C. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv:1808.04866*, 2018.

[8] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *IEEE International Conference on Communications (ICC)*, 2019.

[9] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A stackelberg game perspective," *IEEE Networking Letters*, 2019.

[10] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y. Liang, "Joint service pricing and cooperative relay communication for federated learning," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2019.

[11] W. Lim, N. Luong, D. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *arXiv:1909.11875*, 2019.

[12] Online technical report. https://www.dropbox.com/s/9drkuphv506uuww/appendix.pdf?dl=0.

[13] D. Conway-Jones, T. Tuor, S. Wang, and K. Leung, "Demonstration of federated learning in a resource-constrained networked environment," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2019.

[14] S. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. Suresh, "Scaffold: Stochastic controlled averaging for on-device federated learning," *arXiv:1910.06378*, 2019.

[15] M. Li, T. Zhang, Y. Chen, and A. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[16] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *Journal of Machine Learning Research*, vol. 13, no. January, pp. 165–202, 2012.