FAIR: Quality-Aware Federated Learning with Precise User Incentive and Model Aggregation

Yongheng Deng[1], Feng Lyu[2]*, Ju Ren[2], Yi-Chao Chen[3], Peng Yang[4], Yuezhi Zhou[1], Yaoxue Zhang[1]
[1]Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing, China
[2]School of Computer Science and Engineering, Central South University, Changsha, China
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[4]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China
*Corresponding Author
{dyh19@mails., zhouyz@mail., zhangyx@}tsinghua.edu.cn, {fenglyu, renju}@csu.edu.cn,
yichao@cs.sjtu.edu.cn, yangpeng@hust.edu.cn

IEEE INFOCOM 2021 - IEEE Conference on Computer Communications | 978-1-6654-0325-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOM42981.2021.9488743

***Abstract*—Federated learning enables distributed learning in a privacy-protected manner, but two challenging reasons can affect learning performance significantly. First, mobile users are not willing to participate in learning due to computation and energy consumption. Second, with various factors (e.g., training data size/quality), the model update quality of mobile devices can vary dramatically, inclusively aggregating low-quality model updates can deteriorate the global model quality. In this paper, we propose a novel system named FAIR, i.e., Federated leArning with qualIty awaReness. FAIR integrates three major components: 1) learning quality estimation: we leverage historical learning records to estimate the user learning quality, where the record freshness is considered and the exponential forgetting function is utilized for weight assignment; 2) quality-aware incentive mechanism: within the recruiting budget, we model a reverse auction problem to encourage the participation of high-quality learning users, and the method is proved to be truthful, individually rational, and computationally efficient; and 3) model aggregation: we devise an aggregation algorithm that integrates the model quality into aggregation and filters out non-ideal model updates, to further optimize the global learning model. Based on real-world datasets and practical learning tasks, extensive experiments are carried out to demonstrate the efficacy of FAIR.**

I. Introduction

With the rapid development of Internet of Things (IoT), a large amount of data is continuously generated at the network edge, which provides opportunities to enable learning-based intelligent services [1]–[3]. Traditionally, the centralized learning framework requires a gigantic amount of training data to be aggregated to a cloud center for model training. However, it can lead to a disclosure of user privacy [4]. Besides, both the data delivery overhead for power-constrained mobile devices and the data maintenance cost at the cloud, are prohibitive during the system implementation and operation [5]. Recently, with the emerging technology of mobile edge computing (MEC), mobile devices can be equipped with significant computing and storage capability to enable local computing and model training [6]–[8]. MEC has also pushed forward the research of federated learning [9], which allows a community of computationally-capable nodes to collaboratively build a global learning model without compromising user privacy. Specifically, federated learning is a distributed learning framework, where all nodes independently train the global model based on local data and only model updates are committed to the cloud server for aggregation. In this way, distributed model updates can be aggregated to improve the global model quality in a privacy-preserving manner.

Despite the promising merits of federated learning, technical challenges still exist. First, the success of federated learning is highly dependent on node participation. However, without satisfactory rewards, it is conceivable that computing nodes are not willing to participate in federated learning at the cost of computation and transmission resources. Second, due to various factors, such as training data size/quality and computational capability, the quality of model updates contributed by participating nodes varies significantly. It is non-trivial to recruit suitable nodes to participate in federated learning, especially when the recruiting budget is limited. One plausible approach is to select as many participating nodes as possible. However, inclusively aggregating excessive low-quality model updates can deteriorate the global model quality and inflict model convergence problems [10], [11], which has been verified with field experiments.

There have been some efforts to improve the performance of federated learning, which however cannot well tackle the above challenges. Particularly, they designed federated learning algorithms to accelerate learning convergence [12], proposed control algorithms to determine the frequency for global aggregation [13], or focused on the security and privacy enhancement for federated learning systems [14]. Although these researches have made contributions to federated learning, they are based on a common assumption that there are enough volunteer participants for federated learning. However, volunteer participation is not realistic in practice, because model learning consumes enormous resources including energy, computation, and bandwidth, which is usually prohibitive to resource-scarce mobile nodes. To address this problem, recently, a few works have investigated the incentive mechanism for federated learning [15]–[17]. Specifically, in the work [15], Kang *et al.* investigated the reputation of mobile nodes and designed the incentive mechanism based on contract theory. Likewise, in the studies of [16] and [17], the authors respectively considered the communication efficiency and leveraged reinforcement

978-1-6654-0325-2/21/$31.00 ©2021 IEEE

learning to achieve user incentives. However, none of t
considers the quality of model updates, which can significa
affect the learning performance.

To bridge this gap, in this paper, we investigate the qua
aware federated learning, where the individual learning qu
is estimated to facilitate the precise user incentive and m
aggregation. Particularly, in a multi-task learning scen
we propose a distributed learning system named FAIR,
Federated leArning with qualIty awaReness, to determine
learning task allocation and corresponding payment, and
duct model aggregation. Functionally, FAIR integrates t
major technical components: 1) learning quality estimatio
quality-aware incentive mechanism, and 3) model aggrega
We first adopt the loss reduction during the learning process to quantify the individual learning quality, and leverage the historical quality records to infer the current learning quality. With the estimated quality, a reverse auction case is then built to motivate user participation, where mobile users submit their bids and the platform acts as the auctioneer. To maximize the collective learning quality of all the participants, within the recruiting budget, we formulate a Learning Quality Maximization (LQM) problem, which is proved to be NP-hard. To make real-time decisions at low time complexity, we devise a greedy algorithm to determine the learning task allocation and reward distribution based on Myerson's theorem. Finally, we devise a new aggregation algorithm that integrates the model quality into aggregation and filters out non-ideal model updates, to further enhance the global learning model.

Theoretical analysis indicates that, the proposed FAIR is truthful, individually rational, and computationally efficient. To evaluate the performance of FAIR, we build an emulation system based on real-world datasets and widely adopted learning models. We conduct extensive experiments under various distributed learning scenarios, and the results demonstrate the efficacy of FAIR. Particularly, FAIR advances in both the user incentive and model aggregation, collectively contributing to the superior federated learning performance that can outperform the benchmarks significantly.

We highlight our major contributions as follows.

- We investigate the quality-aware federated learning, where the individual learning quality is estimated to facilitate precise user incentive and model aggregation. It is crucial in practical distributed learning scenarios, but to our best knowledge, is rarely seen in the literature.
- We propose FAIR to determine the learning task allocation and the corresponding payment, and conduct model aggregation in real time. In FAIR, we design and implement three major components: 1) learning quality estimation, 2) quality-aware incentive mechanism, and 3) model aggregation.
- Extensive experiments are carried out to demonstrate the efficacy of FAIR, where the incentive mechanism can stimulate more high-quality model updates, and the devised aggregation algorithm can effectively aggregate the model updates, collectively contributing to a superior globe learning model.

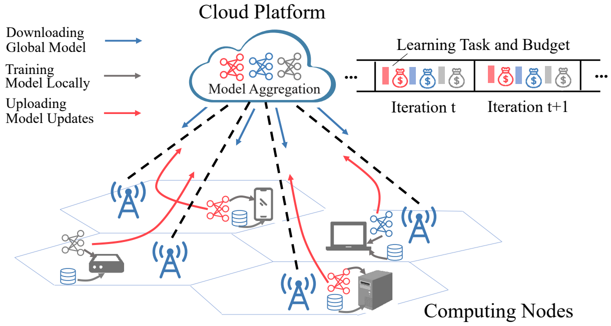


Fig. 1. An overview of the distributed federated learning system.

The remainder of this paper is organized as follows. We describe the system scenario and give the problem definition in Section II. In Section III, we present the overview of FAIR with highlighting design goals. We elaborate on the design of FAIR in Section IV, and conduct the theoretical analysis on FAIR in Section V. Extensive experiments are carried out to evaluate the performance in Section VI, and the related work is reviewed in Section VII. Finally, we conclude this paper and direct our future work in Section VIII.

II. System Description and Problem Definition

In this section, we first describe the targeted scenario of a distributed learning system with multiple learning tasks, then formally define the quality-aware federated learning problem, and finally conduct the problem tractability analysis.

A. System Description

As shown in Fig. 1, we consider a distributed learning system, where there are one cloud platform and various mobile computing nodes. Denote by $\mathcal{N} = \{1, 2, \ldots, N\}$ the set of mobile computing nodes, which can be recruited to conduct model training locally. The system operates in a time-slotted manner and the time span is partitioned into T consecutive slots with equal duration. We focus on a multi-task learning scenario and assume that the model iterates once in a slot. In each iteration, the cloud platform publishes a set of learning tasks, $\mathcal{L}^t = \{l_1^t, l_2^t \ldots\}$, where l_j^t represents the jth learning task published in iteration t. Typically, a learning task demands both training data variety and learning model quality. Note that, the learning task is labeled with the iteration t since the aggregated model quality is different in each iteration. In addition, for each learning task l_j^t in the iteration t, the cloud platform issues a learning budget, B_j^t, to recruit suitable computing nodes to learn the model collaboratively. Denote by $\mathcal{B}^t$ the set of learning budget in iteration t. As computing nodes have limited computing capability with different data varieties, they can participate in different sets of learning tasks. We denote the task set that the node i can participate in as $\mathcal{L}_i^t \subseteq \mathcal{L}^t$. Furthermore, each computing node is constrained to participate in at most one learning task in each iteration.

B. Problem Definition

In each iteration t, the platform has to determine which learning task is executed by which computing node (i.e., the learning task allocation) at what price (i.e., determining the payment), considering the learning budget. Once a computing node is recruited to participate in a learning task, the node will download the corresponding global model, train the model with its own data samples, and submit the local model updates to the platform. After the platform receives the model updates from different nodes, it will aggregate them to update the global models of learning tasks. With the evolving of iteration, the aggregated model quality can be enhanced with more training data over time. Once the model quality reaches a threshold, the learning task terminates and the global learning model can be adopted by the system in all subsequent iterations. As the system has to use the global learning model in each iteration, the aggregated model quality of each learning task should be maximized in every iteration. Formally, our quality-aware federated learning problem is cast as follows.

Definition 1 (Quality-Aware Federated Learning). *For each iteration t, given the sets of learning tasks $\mathcal{L}^t$ and learning budgets $\mathcal{B}^t$, how to allocate the learning tasks, distribute payments, and aggregate the model updates, such that the sum of qualities of all aggregated learning models is maximized?*

A binary variable $s^t_{i,j} \in \{0, 1\}$ is used to indicate whether the task l^t_j is allocated to node i in iteration t, which equals 1 if the task is allocated to the node, and equals 0 otherwise. Denote by $r^t_{i,j}$ the payment reward to recruit node i to participate in learning task l^t_j in iteration t, and denote by $c^t_{i,j}$ the learning cost of the participating node with respect to the computation and energy consumption. Then, the *quality-aware federated learning* problem can be formulated as

$$\max_{\mathbf{M}^t, \mathbf{R}^t} \quad \sum_{l^t_j \in \mathcal{L}^t} f(\mathbf{M}^t), \tag{1}$$

$$s.t. \quad s^t_{i,j} \in \{0, 1\}, \quad \forall i \in \mathcal{N}, \forall l^t_j \in \mathcal{L}^t, \tag{2}$$

$$\sum_{i \in \mathcal{N}} r^t_{i,j} s^t_{i,j} \le B^t_j, \quad \forall l^t_j \in \mathcal{L}^t, \tag{3}$$

$$s^t_{i,j} = 0, \quad \forall l^t_j \notin \mathcal{L}^t_i, \forall i \in \mathcal{N}, \tag{4}$$

$$\sum_{l^t_j \in \mathcal{L}^t} s^t_{i,j} \le 1, \quad \forall i \in \mathcal{N}, \tag{5}$$

where $\mathbf{M}^t = \{s^t_{i,j}\}_{\forall i \in \mathcal{N}, \forall l^t_j \in \mathcal{L}^t}$ is the learning task allocation results in the iteration t, $\mathbf{R}^t = \{r^t_{i,j}\}_{\forall i \in \mathcal{N}, \forall l^t_j \in \mathcal{L}^t}$ is the payment determination results in the iteration t, and $f()$ is the model aggregation function with inputs of model updates. The constraint (3) represents that for each learning task, the sum of payments should not exceed the learning budget provided by the task publisher. The constraints (4) means that each node can only be assigned with learning tasks that it can participate in. In constraint (5), the system limits each node to participate in at most one learning task in every iteration.

C. Problem Difficulties

The formulated *quality-aware federated learning* problem is intractable directly due to the following reasons. First, the problem is hard to be mathematically externalized in terms of the model aggregation function since there is a lack of an appropriate metric to quantify the learning qualities of both the individual model updates and global aggregated models. Additionally, the quality of model updates to be uploaded is unknown ahead before learning, which brings challenges to the task allocation. Second, it is rather challenging to allocate multiple learning tasks with both-satisfied payments, especially when the learning budget is limited. On the one hand, the nodes have different computation and communication costs, and it is hard to model the learning cost of each participating node. On the other hand, participants are usually strategically selfish, and they tend to claim a higher cost than the real one in order to increase individual learning profit. Third, with the model updates, how to aggregate them is also crucial to the aggregated model quality, hereby expecting an efficient model aggregation mechanism that can effectively aggregate the model updates with different learning qualities. In this paper, we propose FAIR to systematically address those challenges, and thus provide a solution to the *quality-aware federated learning* problem.

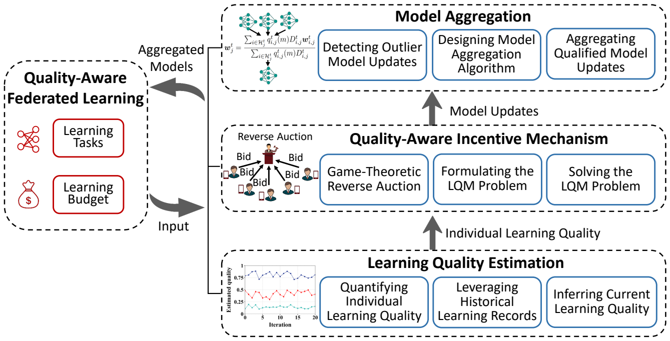

Model Aggregation
$w^t_j = \frac{\sum_{i \in \mathcal{H}^t_j} q^t_{i,j}(m) D^t_{i,j} w^t_{i,j}}{\sum_{i \in \mathcal{H}^t_j} q^t_{i,j}(m) D^t_{i,j}}$
Detecting Outlier Model Updates
Designing Model Aggregation Algorithm
Aggregating Qualified Model Updates
Aggregated Models
Quality-Aware Federated Learning
Learning Tasks
Learning Budget
Input
Model Updates
Reverse Auction
Bid
Bid
Bid
Bid
Bid
Quality-Aware Incentive Mechanism
Game-Theoretic Reverse Auction
Formulating the LQM Problem
Solving the LQM Problem
Individual Learning Quality
Learning Quality Estimation
Estimated quality
1
0.75
0.5
0.25
0
0
5
10
15
20
Iteration
Quantifying Individual Learning Quality
Leveraging Historical Learning Records
Inferring Current Learning Quality

Fig. 2. Architecture of FAIR.

III. Overview of FAIR

A. Design Overview

As shown in Fig. 2, FAIR integrates three major components: 1) learning quality estimation; 2) quality-aware incentive mechanism, and 3) model aggregation. Specifically, to mathematically pinpoint the optimization problem, we first adopt the loss reduction to quantify the individual learning quality, and leverage the historical learning records to predict the current learning quality. With the estimated learning quality, we then model the interaction between the platform and computing nodes as a game-theoretic reverse auction to cast a quality-aware incentive mechanism. In the incentive mechanism, during each iteration t, the platform announces the learning task set $\mathcal{L}^t$ to the computing nodes, and each node i submits its bid information $\mathbb{B}^t_i = \{(l^t_j, b^t_{i,j})\}_{\forall l^t_j \in \mathcal{L}^t_i}$ to the platform. The tuple $(l^t_j, b^t_{i,j})$ consists of the learning task l^t_j that the node wants to participate in, and the corresponding bid price $b^t_{i,j}$. In this way, the original quality-aware federated learning problem can be transformed and solved by following two optimization directions. On the one hand, working with the reverse auction, we formulate the *Learning Quality Maximization* (LQM) problem to allocate the learning tasks with payments, and the objective is to recruit more high-quality model updates within the learning budget. On the other hand, with the model updates of participating nodes, we devise a model aggregation algorithm to detect outlier model updates and effectively aggregate the qualified model updates to further improve the aggregated model quality.

B. Design Goals

FAIR aims to maximize the sum of qualities of all aggregated learning models in each iteration while ensuring *truthfulness*, *individual rationality*, and *computational efficiency*. As computing nodes are strategically selfish, the truthfulness goal is set to avoid nodes declaring untruthful bid price. To quantify the benefits of computing nodes when participating in a learning task, the node utility is defined.

Definition 2 (**Node Utility**). *In iteration t, the utility gain of node i by participating in learning task l_j^t is the difference between the reward and learning cost, i.e.,*

$$u_{i,j}^t = \begin{cases} r_{i,j}^t - c_{i,j}^t, & \text{if } i \in \mathcal{M}_j^t; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Then, the design goals are defined as follows.

Definition 3 (**Truthfulness**). *A mechanism is truthful if, in each iteration t, no node can increase its utility by reporting untruthful bid price with $b_{i,j}^t > c_{i,j}^t$. Formally, for each node i with true bid price, i.e., $b_{i,j}^t = c_{i,j}^t$, if the node i is truthful in iteration t, its utility is U_i, otherwise $\widehat{U}_i$. We have $U_i \geq \widehat{U}_i$ for each node.*

Definition 4 (**Individual Rationality**). *A mechanism is individually rational if the utility of each node i in each iteration t is non-negative, i.e., $u_{i,j}^t \geq 0$.*

Definition 5 (**Computational Efficiency**). *A mechanism is computationally efficient if the task allocation, payment determination, and model aggregation can be conducted within a polynomial time.*

IV. Design of FAIR

A. Estimating Learning Quality

After receiving model updates contributed by participating nodes, the system can evaluate the learning quality of model updates. However, the learning quality is unknown ahead before learning, and thus in each iteration, FAIR first estimates the learning quality of candidate participants to assist in allocating tasks with payments.

1) Learning Quality Quantification: In federated learning, both the volume and quality of the training data can affect the learning quality significantly. The quantification of the learning quality should adequately reflect how useful that the local model updates can contribute to the global model. One plausible approach is to adopt each node's local model accuracy tested on a global dataset as the learning quality. However, in this approach, the test on each local model is required in each iteration, which can inflict significant overhead. Different from the accuracy measurement, the loss value is calculated in training with no additional overhead. Therefore, we leverage the loss reduction in each iteration to quantify the training data quality. Specifically, suppose iteration t starts at time t_s and ends at time t_e. At time t_e, the received local model updates are aggregated to update the global models, and the next iteration starts. Hence, participating nodes are required to submit their local model updates at time $t_{i,j}^u$ within $[t_s, t_e]$, otherwise, their local model updates will be rejected with no system rewards. Suppose the average test loss value of task l_j^t's global model at time t_s is $loss_j(t_s)$ and the average training loss value of node i's local model at time t_e is $loss_{i,j}(t_e)$. We define the training data quality of node i in iteration t as

$$m_{i,j}^t = loss_j(t_s) - loss_{i,j}(t_e). \tag{7}$$

Combining the amount of data (denoted by $D_{i,j}^t$) used for training, the learning quality of node i in iteration t is defined as follows

$$q_{i,j}^t = \begin{cases} m_{i,j}^t D_{i,j}^t, & \text{if } t_s < t_{i,j}^u \leq t_e; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

2) Learning Quality Estimation: With the learning quality quantification, we then proceed to estimate the current learning quality of each candidate. Particularly, as the distributed learning runs iteratively, we leverage the historical quality records of a participant in learning tasks, to estimate its model update quality. Supposing that node i has participated in the learning task l_j in iteration $t_0, t_1, \ldots, t_r$, we can utilize the quality records $(q_{i,j}^{t_0}, q_{i,j}^{t_1}, \ldots, q_{i,j}^{t_r})$ to estimate the quality, $q_{i,j}^t$, that is contributed in iteration t, where $t > t_r$. The learning quality of one node may change with time, and intuitively, the recent quality records are more informative than the stale quality records. Therefore, instead of giving all quality records the same weight, we weight them according to their freshness [18]. Specifically, we adopt an exponential forgetting function to assign the weights, which gives larger weights to the recent quality records and smaller weights to the stale ones [19]. The most recent quality record receives the weight of 1 and other record weights are determined by their relative position to the most recent quality record. The according weights of $(q_{i,j}^{t_0}, q_{i,j}^{t_1}, \ldots, q_{i,j}^{t_r})$ are $(\rho^{t_r - t_0}, \rho^{t_r - t_1}, \ldots, 1)$, where $0 < \rho \leq 1$ is the forgetting factor [20]. Above all, the estimated quality value $\widehat{q}_{i,j}^t$ can be obtained as

$$\widehat{q}_{i,j}^t = \frac{\sum_{k=0}^{r} \rho^{t_r - t_k} q_{i,j}^{t_k}}{\sum_{k=0}^{r} \rho^{t_r - t_k}}. \tag{9}$$

B. Quality-Aware Incentive Mechanism

After estimating the learning quality for each candidate, we then solve the defined *quality-aware federated learning* problem in two steps. Within the learning budget, we first encourage high-quality computing nodes to participate in the learning tasks via a quality-aware incentive mechanism. Then, with model updates, we devise an algorithm to further enhance the model aggregation performance. In this subsection, we focus on the design of quality-aware incentive mechanism. In each iteration, we model a reverse auction case to stimulate high-quality candidate participants, where each node i submits its bid information $\mathbb{B}_i^t$, and for each learning task l_j^t, FAIR selects a set of winner nodes $\mathcal{M}_j^t \subset \mathcal{N}$ and determines a payment set $\mathcal{R}_j^t = \{r_{i,j}^t\}_{\forall i \in \mathcal{M}_j^t}$ within the learning budget.

Specifically, we can formulate the LQM problem as follows.

Definition 6 (**The LQM Problem**). *In each iteration t, in accordance with the bids information, how to select a set of winner nodes $\mathcal{M}_j^t$ with payments $\mathcal{R}_j^t$ for each learning task l_j^t, such that the sum of the estimated learning qualities of the selected nodes is maximized?*

The defined LQM problem can be formulated as

$$\max_{\mathcal{M}_j^t, \mathcal{R}_j^t} \quad \sum_{l_j^t \in \mathcal{L}^t} \sum_{i \in \mathcal{M}_j^t} \widehat{q}_{i,j}^t, \tag{10}$$

$$s.t. \quad \sum_{i \in \mathcal{N}} r_{i,j}^t s_{i,j}^t \leq B_j^t, \quad \forall l_j^t \in \mathcal{L}^t, \tag{11}$$

$$r_{i,j}^t \geq b_{i,j}^t, \quad \forall l_j^t \in \mathcal{L}^t, \forall i \in \mathcal{M}_j^t, \tag{12}$$

$$s_{i,j}^t \in \{0,1\}, \quad \forall i \in \mathcal{N}, \forall l_j^t \in \mathcal{L}^t, \tag{13}$$

$$s_{i,j}^t = 0, \quad \forall l_j^t \notin \mathcal{L}_i^t, \forall i \in \mathcal{N}, \tag{14}$$

$$\sum_{l_j^t \in \mathcal{L}^t} s_{i,j}^t \leq 1, \quad \forall i \in \mathcal{N}. \tag{15}$$

Inputs. The LQM problem takes the task set $\mathcal{L}_i^t$ of each node $i \in \mathcal{N}$, the bid price $b_{i,j}^t, \forall i \in \mathcal{N}, \forall l_j^t \in \mathcal{L}_i^t$, the learning budget $B_j^t, \forall l_j^t \in \mathcal{L}^t$, and the quality estimation value $\widehat{q}_{i,j}^t, \forall i \in \mathcal{N}, \forall l_j^t \in \mathcal{L}^t$ as inputs. *Outputs.* FAIR determines the value of the binary variable $s_{i,j}^t$ for each $i \in \mathcal{N}$ and $l_j^t \in \mathcal{L}^t$. If $s_{i,j}^t = 1$, the node i will be included into the selected nodes set $\mathcal{M}_j^t$ which means that the learning task l_j^t will be allocated to the node i. Also, FAIR determines the learning reward $r_{i,j}^t$ in the set $\mathcal{R}_j^t$ for each winner node. *Constraints.* The constraints for LQM problem include all the constraints of the *quality-aware federated learning* problem, but the Constraint (12) is additionally required, which guarantees that the payment to each node is larger than its claimed bid price. For the LQM problem, we have the following Theorem.

Theorem 1. *The LQM problem is NP-hard.*

Proof. To prove its NP-hardness, we devise a polynomial reduction from a classic NP-hard problem, i.e., *Multiple Knapsack Problem with Assignment Restrictions* (MKAR) [21], which is a variant of the well-known NP-hard problem of *Multiple Knapsack Problem* (MKP) [22], to our formulated LQM problem. An instance of the MKAR problem can be given as follows. Suppose there is an item set $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ with specified value v_i and weight w_i for each item $o_i \in \mathcal{O}$, as well as a knapsack set $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$ with specified capacity c_j for each knapsack $b_j \in \mathcal{B}$. For each item $o_i \in \mathcal{O}$, a set $\mathcal{B}_i \subseteq \mathcal{B}$ of knapsacks that can hold item o_i is specified. To maximize the total value of assigned items, for each knapsack $b_j \in \mathcal{B}$, we need to choose a subset $\mathcal{O}_j \subseteq \mathcal{O}$ of items to be assigned to knapsack b_j, such that: 1) each item is assigned to at most one knapsack; 2) each $\mathcal{O}_j$ is a subset of $\mathcal{A}_j$, where $\mathcal{A}_j \subseteq \mathcal{O}$ is the set of items that can be assigned to knapsack b_j; 3) total weight of items assigned to a knapsack is no more than the capacity of the knapsack. Hereafter, based on the instance of the MKAR problem, we construct an instance of the LQM problem. First, we transform the item set $\mathcal{O}$ and knapsack set $\mathcal{B}$ into node set $\mathcal{N}$ and learning task set $\mathcal{L}^t$, respectively. Then, we assume that each node has the same bid price and quality value for each learning task, that is, $b_{i,j}^t = b_i^t$ and $\widehat{q}_{i,j}^t = \widehat{q}_i^t$ for all $l_j^t \in \mathcal{L}_i^t$. Next, we set $r_{i,j}^t = b_{i,j}^t$ for all $l_j^t \in \mathcal{L}^t, i \in \mathcal{M}_j^t$. Finally, we set $v_i = \widehat{q}_i^t$, $w_i = r_i^t$ for all $i \in \mathcal{N}$, and $B_j^t = c_j$ for all $l_j^t \in \mathcal{L}^t$. In this way, each instance of the MKAR problem is polynomial-time reducible to an instance of the LQM problem. Therefore, the LQM problem is an NP-hard problem, which concludes the proof. □

Algorithm 1: Solving the LQM problem.

```
Input  : (1) bid price $b_{i,j}^t$; (2) budget $B_j^t$; (3) task set $\mathcal{L}_i^t$;
         (4) quality estimation values $\widehat{q}_{i,j}^t$.
Output : (1) the task allocation results $s_{i,j}^t$; (2) the
         payments $r_{i,j}^t$.
Initialize $\mathcal{N}_j^t \leftarrow \emptyset$, $p_j^t \leftarrow 0$ for each $l_j^t \in \mathcal{L}^t$;
Initialize $x_i^t \leftarrow 1$ for each $i \in \mathcal{N}$;
Initialize $r_{i,j}^t \leftarrow 0, s_{i,j}^t \leftarrow 0$ for each $i \in \mathcal{N}$, $l_j^t \in \mathcal{L}^t$;
foreach $i \in \mathcal{N}$ do
    foreach $l_j^t \in \mathcal{L}_i^t$ do
        $\mathcal{N}_j^t \leftarrow \mathcal{N}_j^t + \{i\}$;
    end
end
while $\exists\ x_i^t = 0$ and $\exists\ p_j^t = 0$ do
    Initialize $\mathcal{M}_j^t \leftarrow \emptyset$ for each $l_j^t \in \mathcal{L}^t$;
    foreach $l_j^t \in \mathcal{L}^t$ do
        if $p_j^t = 0$ then
            Sort all $i \in \mathcal{N}_j^t$ in descending order of $\frac{q_{i,j}^t}{b_{i,j}^t}$;
            Find the smallest $k$ such that
                $\sum_{i=1}^{k} \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t x_i^t > B_j^t$;
            for $i \leftarrow 1$ to $k-1$ do
                $\mathcal{M}_j^t \leftarrow \mathcal{M}_j^t + \{i\}$;
                $r_{i,j}^t \leftarrow \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t$;
            end
        end
    end
    Find the task $l_k^t$ with maximum $\sum_{i \in \mathcal{M}_k^t} q_{i,k}^t x_i^t$;
    Set $p_k^t \leftarrow 1$;
    foreach $i \in \mathcal{M}_k^t$ do
        if $x_i^t = 1$ then
            $s_{i,k}^t \leftarrow 1$;
            $x_i^t \leftarrow 0$;
        end
    end
end
return $(s_{i,j}^t, r_{i,j}^t)$;
```

Given the NP-hardness of the LQM problem, in FAIR, we devise a heuristic algorithm to solve the LQM problem with truthfulness, individual rationality, and computational efficiency. Myerson's theorem [23] of truthfulness has proved that a mechanism for auction problems is truthful if and only if the winner selection problem is *monotone* and the payment of each winner is a *critical value*:

- *Monotonicity.* If node i wins in iteration t by claiming a cost bid price $b_{i,j}^t$ for performing the learning task, it will still win with any cost bid $\widehat{b}_{i,j}^t < b_{i,j}^t$.
- *Critical payment.* If node i wins with the bid price $b_{i,j}^t$, it can also win with other bid $\widehat{b}_{i,j}^t$, but bidding with $b_{i,j}^t$ makes it get the maximum payment, and then $b_{i,j}^t$ is said

to be the critical payment of node i. That is, a critical payment is the maximum bid value for a bid to win.

We utilize the above theorem and devise the greedy algorithm to solve the LQM problem in each iteration t. As shown in Algorithm 1, in each iteration t, the algorithm first picks the candidate nodes $\mathcal{N}_j^t$ that can participate in the learning task l_j^t (lines 4-8). Then, the main loop (lines 9-29) is executed until there is no node that can participate in the learning tasks or all tasks have been allocated to nodes for execution. In the main loop, the algorithm first chooses a subset of winner nodes $\mathcal{M}_j^t \subseteq \mathcal{N}_j^t$ for each task l_j^t that can approximately maximize the sum of the estimated qualities of l_j^t's winner nodes (lines 11-20). Specifically, the algorithm sorts node $i \in \mathcal{N}_j^t$ in descending order by $q_{i,j}^t / b_{i,j}^t$, i.e., the quality contribution per unit bid (line 13). The value of $q_{i,j}^t / b_{i,j}^t$ is a ranking indicator for node i. Then, the algorithm greedily includes node i into the winner node set $\mathcal{M}_j^t$ according to its ranking until the total payment exceeds the budget B_j^t (lines 14-18). Here, we determine the reward of node i according to its critical payment. Denoting by k the node with the highest ranking among all loser nodes, the maximum bidding price $b'_{i,j}$ that can substitute node i as the winner satisfies $q_{i,j}^t / b'_{i,j} = q_{k,j}^t / b_{k,j}^t$. This means the critical payment of node i is the bidding price $b'_{i,j} = \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t$. The critical payment $b'_{i,j}$ is used as the payment to node i (line 17). Finally, the algorithm finds the task l_k^t with maximum $\sum_{i \in \mathcal{M}_k^t} q_{i,k}^t x_i^t$ (line 21), and allocates task l_k^t to the obtained winner nodes (lines 22-28).

Obviously, algorithm 1 satisfies the constraints (11), (13), (14), and (15). The constraint (12) as well as truthfulness and computational efficiency will be proved in Section V.

Algorithm 2: Model aggregation algorithm.

```
   Input  : (1) winner node set $\mathcal{M}_j^t$; (2) local model
            parameters $\boldsymbol{w}_{i,j}^t$; (3) data size used for training
            $D_{i,j}^t$; (4) data quality $m_{i,j}^t$.
   Output : aggregated model parameters $\boldsymbol{w}_j^t$.
1  Initialize high-quality node set $\mathcal{H}_j^t \leftarrow \{i : i \in \mathcal{M}_j^t\}$;
2  Set $D \leftarrow \sum_{i \in \mathcal{M}_j^t} m_{i,j}^t D_{i,j}^t$;
3  Set $\boldsymbol{w}_j^t \leftarrow \sum_{i \in \mathcal{M}_j^t} \frac{m_{i,j}^t D_{i,j}^t}{D} \boldsymbol{w}_{i,j}^t$;
4  foreach $i \in \mathcal{M}_j^t$ do
5      Compute $d_{i,j}^t \leftarrow similarity(\boldsymbol{w}_j^t, \boldsymbol{w}_{i,j}^t)$;
6  end
7  Compute $\bar{\mu}_d$, $\hat{\mu}_d$, $\sigma_d$;
8  if $\bar{\mu}_d > \hat{\mu}_d$ then
9      foreach $i \in \mathcal{H}_j^t$ do
10         if $d_{i,j}^t > \hat{\mu}_d + \eta\sigma_d$ then
11             $\mathcal{H}_j^t \leftarrow \mathcal{H}_j^t - \{i\}$;
12         end
13     end
14 else
15     foreach $i \in \mathcal{H}_j^t$ do
16         if $d_{i,j}^t < \hat{\mu}_d - \eta\sigma_d$ then
17             $\mathcal{H}_j^t \leftarrow \mathcal{H}_j^t - \{i\}$;
18         end
19     end
20 end
21 Set $D \leftarrow \sum_{i \in \mathcal{H}_j^t} m_{i,j}^t D_{i,j}^t$;
22 Set $\boldsymbol{w}_j^t \leftarrow \sum_{i \in \mathcal{H}_j^t} \frac{m_{i,j}^t D_{i,j}^t}{D} \boldsymbol{w}_{i,j}^t$;
23 return $\boldsymbol{w}_j^t$;
```

C. Model Aggregation

In each iteration t, for each learning task l_j^t, after one or multiple gradient-descent updates, each winner node i will upload their local model parameters $\boldsymbol{w}_{i,j}^t$ to the platform, and then the platform will aggregate them to update the global model parameters $\boldsymbol{w}_j^t$. The following Federated Averaging algorithm has been widely adopted in state-of-the-art researches (e.g., [9], [13]):

$$\boldsymbol{w}_j^t = \frac{\sum_{i=1}^{N} D_{i,j}^t \boldsymbol{w}_{i,j}^t}{\sum_{i=1}^{N} D_{i,j}^t}, \tag{16}$$

where $D_{i,j}^t$ is the amount of data used by node i to train the learning model of task l_j^t in iteration t. Unlike the existing model aggregation algorithms for federated learning, we aggregate the model updates considering not only the training data size of each node, but also the training data quality. That is, in iteration t, given a set of winner nodes $\mathcal{M}_j^t$ of task l_j^t and their local model parameters $\boldsymbol{w}_{i,j}^t$, the aggregated model parameters $\boldsymbol{w}_j^t$ used to update the global model of task l_j^t can be computed by:

$$\boldsymbol{w}_j^t = \frac{\sum_{i \in \mathcal{M}_j^t} m_{i,j}^t D_{i,j}^t \boldsymbol{w}_{i,j}^t}{\sum_{i \in \mathcal{M}_j^t} m_{i,j}^t D_{i,j}^t}, \tag{17}$$

where $D_{i,j}^t$ and $m_{i,j}^t$ are the amount of data used for training and the data quality of node i for task l_j^t, respectively. In addition, to avoid the negative impacts of low quality models, we design a computationally efficient method to detect and filter out low-quality local model updates [24].

Specifically, Algorithm 2 shows the design of our model aggregation algorithm. First, we aggregate the local model updates received from winner nodes using (17) and we use one of the distance models, *cosine similarity* to calculate the similarity between $\boldsymbol{w}_j^t$ and $\boldsymbol{w}_{i,j}^t$ (lines 2-6). Second, we calculate the mean ($\bar{\mu}_d$), median ($\widehat{\mu}_d$), and standard deviation (σ_d) of the similarity (line 7). Since the incentive mechanism in FAIR eliminates most unreliable nodes before learning, the majority of the received updates should be high-quality. Therefore, the median ($\widehat{\mu}_d$) can reflect the direction of high-quality local model updates. To be specific, we compare the value of $\bar{\mu}_d$ and $\widehat{\mu}_d$, and if $\bar{\mu}_d > \widehat{\mu}_d$, the low-quality updates should have a similarity value $d_{i,j}^t$ higher than $\widehat{\mu}_d$. We treat the local model updates whose similarity value $d_{i,j}^t > \widehat{\mu}_d + \eta\sigma_d$, as low-quality updates (lines 8-13). The parameter η is a preset threshold that can control the range. Similarly, when $\bar{\mu}_d \leq \widehat{\mu}_d$, the model updates will also be considered as low quality if $d_{i,j}^t < \widehat{\mu}_d - \eta\sigma_d$ (lines 14-20). In this way, we get the high-quality node set $\mathcal{H}_j^t$ whose local model updates are qualified. Finally, we aggregate the qualified local model updates as the aggregated results (lines 21-23).

V. Performance Analysis

In this section, we theoretically prove the truthfulness, individual rationality, and computational efficiency of FAIR.

Theorem 2. FAIR *is truthful.*

Proof. In each iteration t, node i might report a truthful bid price $b_{i,j}^t = c_{i,j}^t$ or any other untruthful bid price $\widehat{b}_{i,j}^t$. The four bidding results of node i are as follows:

1) $\{$*win, win*$\}$: Node i wins in iteration t with both truthful bid $b_{i,j}^t$ and untruthful bid $\widehat{b}_{i,j}^t$. In this case, the utility of node i is $u_{i,j}^t(b_{i,j}^t) = u_{i,j}^t(\widehat{b}_{i,j}^t) = \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t - c_{i,j}^t$.
2) $\{$*loss, loss*$\}$: Node i loses in iteration t with both truthful bid $b_{i,j}^t$ and untruthful bid $\widehat{b}_{i,j}^t$. In this case, the utility of node i is $u_{i,j}^t(b_{i,j}^t) = u_{i,j}^t(\widehat{b}_{i,j}^t) = 0$.
3) $\{$*win, loss*$\}$: Node i wins in iteration t with truthful bid $b_{i,j}^t$ and loses with untruthful bid $\widehat{b}_{i,j}^t$. In this case, the utility $u_{i,j}^t(b_{i,j}^t) = \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t - c_{i,j}^t = \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t - b_{i,j}^t \geq 0$. Because node i wins with bid $b_{i,j}^t$ and we have $\frac{q_{i,j}^t}{b_{i,j}^t} \geq \frac{q_{k,j}^t}{b_{k,j}^t}$ according to nodes' ranking in Algorithm 1. The utility $u_{i,j}^t(\widehat{b}_{i,j}^t) = 0$, and hence $u_{i,j}^t(b_{i,j}^t) \geq u_{i,j}^t(\widehat{b}_{i,j}^t)$.
4) $\{$*loss, win*$\}$: Node i loses in iteration t with truthful bid $b_{i,j}^t$ but wins with untruthful bid $\widehat{b}_{i,j}^t$. In this case, the utility $u_{i,j}^t(b_{i,j}^t) = 0$ and the utility $u_{i,j}^t(\widehat{b}_{i,j}^t) = \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t - c_{i,j}^t = \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t - b_{i,j}^t \leq 0$. Because node i loses with bid $b_{i,j}^t$ and we have $\frac{q_{i,j}^t}{b_{i,j}^t} \leq \frac{q_{k,j}^t}{b_{k,j}^t}$ according to node ranking. Thus, $u_{i,j}^t(b_{i,j}^t) \geq u_{i,j}^t(\widehat{b}_{i,j}^t)$ still holds.

As $u_{i,j}^t(b_{i,j}^t) \geq u_{i,j}^t(\widehat{b}_{i,j}^t)$ holds in all cases, which means that node i cannot improve its utility by reporting any untruthful bid. Therefore, we can conclude that FAIR is truthful. □

Theorem 3. FAIR *is individually rational.*

Proof. If node i loses in iteration t, its utility $u_{i,j}^t = 0$. Otherwise, node i wins with truthful bid $b_{i,j}^t = c_{i,j}^t$ since we have proved that nodes bid truthfully. The node utility $u_{i,j}^t = r_{i,j}^t - c_{i,j}^t = \frac{b_{k,j}^t}{q_{k,j}^t} q_{i,j}^t - b_{i,j}^t \geq 0$ due to $\frac{q_{i,j}^t}{b_{i,j}^t} \geq \frac{q_{k,j}^t}{b_{k,j}^t}$. Therefore, $u_{i,j}^t \geq 0$ for each node i, and FAIR is proved to be individually rational. □

Theorem 4. *The time complexity of task allocation and payment scheme in* FAIR *is* $\mathcal{O}(L^2 N \log N)$*, where* $L = |\mathcal{L}^t|$ *is the number of learning tasks in iteration t, and* $N = |\mathcal{N}|$ *is the number of nodes in set* $\mathcal{N}$*. The time complexity of the model aggregation algorithm in* FAIR *is* $\mathcal{O}(MS)$*, where* M *is the number of nodes in winner set* $\mathcal{M}_j^t$ *and* S *is the model parameter size of learning task* l_j^t*. Both time complexities are polynomial, which are computationally efficient.*

Proof. We analyze the worst case of Algorithm 1 where $|\mathcal{L}_i^t| = L$ and $|\mathcal{N}_j^t| = N$. In the worst case, the main loop in line 9 terminates after L times of iterations. Besides, the computational complexity of sorting $q_{i,j}^t / b_{i,j}^t$ (line 13) is $\mathcal{O}(N \log N)$, where finding the smallest k (line 14) is $\mathcal{O}(N)$, and finding the task l_k^t with maximum $\sum_{i \in \mathcal{M}_k^t} q_{i,k}^t x_i^t$ (line 21) is $\mathcal{O}(NL)$. Therefore, the computational complexity of Algorithm 1 is $\mathcal{O}(L^2 N \log N)$. In Algorithm 2, as the time complexity of computing cosine similarity is linear with the model size, the computational complexity of the loop in lines 4-6 is $\mathcal{O}(MS)$. Besides, the complexity to compute the median

Table I. Experiment parameters.

Settings	$\|\mathcal{N}\|$	$\|\mathcal{L}^t\|$	$b_{i,j}^t$	$D_{i,j}^t$	N_e	R_e	B_j^t
I	20	1	[1,3]	[1000,3000]	5	[0.05,0.5]	5
II	20	1	[1,3]	5000	[0,20]	0.5	10
III	100	4	[1,10]	[1000,10000]	30	[0,1]	[10,30]

of similarity ($\widehat{\mu}_q$) can be as fast as $\mathcal{O}(M)$ (line 7). Above all, the computational complexity of Algorithm 2 is $\mathcal{O}(MS)$. □

VI. Performance Evaluation

A. Evaluation Methodology

Experiment Setup. We build the FAIR emulation system by adopting the widely-used PyTorch 1.4.0 software environment. The detailed experiment settings are shown in Table I, where the bid price, computing capacity, and data quality of distributed nodes are varying parameters. Besides, in the set $\mathcal{N}$, we assume there are N_e nodes whose mislabeled data ratio in the training dataset is R_e.

Models and Datasets. We evaluate the performance of FAIR with the 6 most commonly adopted learning models, including Multi-layer Perceptron (MLP), LeNet, MobileNet, VGG-11, EfficientNet-B0, and ResNet-18. The above models are trained with four datasets: MNIST, Fashion-MNIST (FMNIST), CIFAR-10, and the Street View House Numbers (SVHN) dataset. MNIST is a dataset of handwritten digits and FMNIST is a dataset of Zalando's fashion article images, both of which have a training set of 60 thousand examples and a test set of 10 thousand examples. The CIFAR-10 dataset consists of 50 thousand training images and 10 thousand test images in 10 classes. SVHN is a real-world house number image dataset with 73 thousand training data and 26 thousand test data.

Benchmarks. To compare the performance, the following reasonable benchmarks are designed.

- **Theoretically optimal mechanism:** It adopts the depth-first search approach to find the theoretically optimal solution for the LQM problem, which however cannot guarantee the truthfulness of nodes.
- **Knapsack greedy mechanism:** It greedily selects winner nodes based on the amount of data used for training divided by the bid price, i.e., $D_{i,j}^t / b_{i,j}^t$, where the data quality and truthfulness of nodes are not considered.
- **Bid price first mechanism:** It preferentially selects nodes with the lowest bid price without guaranteeing the truthfulness of nodes.

B. Performance of User Incentive

We first investigate the user incentive performance in FAIR. We adopt the experiment setting I in Table I, where only one learning task in each iteration is considered, i.e., $|\mathcal{L}^t| = 1$. We run 6 different learning tasks, respectively. Specifically, the MLP and LeNet models are trained with MNIST and FMNIST, respectively. The MobileNet and VGG models are trained with CIFAR-10, while the EfficientNet and ResNet model are trained with SVHN. In addition, for a fair comparison, all benchmarks (including FAIR) adopt the Federated Averaging algorithm for model aggregation. After running the incentive

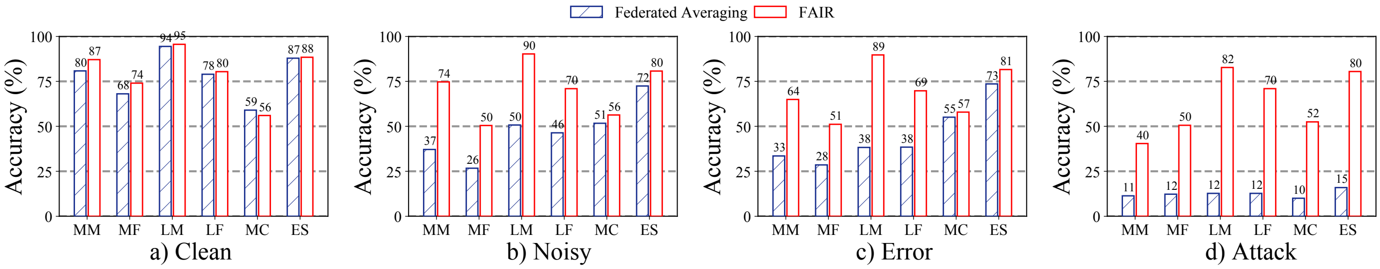


Fig. 3. The model aggregation performance of MLP-MNIST (MM), MLP-FMNIST (MF), LeNet-MNIST (LM), LeNet-FMNIST (LF), MobileNet-CIFAR10 (MC), EfficientNet-SVHN (ES) under different scenarios: a) Clean datasets; b) Noisy label datasets; c) Error label datasets; d) Attack datasets.

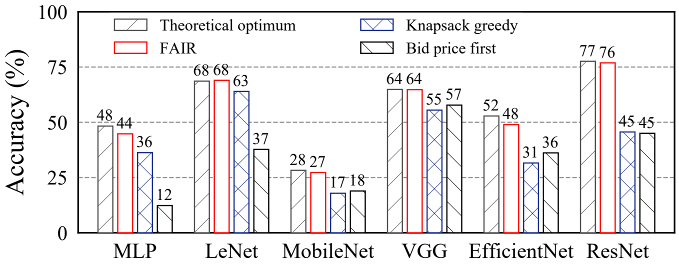


Fig. 4. The average model accuracy with different incentive mechanisms.

game 30 iterations, we plot the average accuracy results of learning models in Fig. 4. We can observe that for all learning models, FAIR can achieve the near-optimal performance since there is a negligible performance degradation when compared with the accuracy results of the theoretically optimal mechanism, but FAIR can outperform the other two benchmarks significantly. For instance, when evaluating on the ResNet model, the knapsack greedy and bid price first mechanisms can only achieve a 45% accuracy score, while FAIR can achieve a score of 76%, which can improve the performance by 68.9%.

C. *Performance of Model Aggregation*

We then compare the model aggregation performance of FAIR with the Federated Averaging (FA) algorithm. The six models are trained with respective datasets, and each model is trained with 10 nodes under 4 different scenarios[1]: a) *Clean datasets*: all distributed nodes have the original unchanged training datasets to train the model normally. At each node, the amount of data $D_{i,j}^t$ used for training is generated uniformly within the range $[1000, 10000]$; b) *Noisy label datasets*: among 10 distributed nodes, the training datasets of 5 nodes are clean, but in the training datasets of other 5 nodes, 50% of data samples are incorrectly labeled, i.e., labels are randomly generated; c) *Error label datasets*: among 10 distributed nodes, the training datasets of 7 nodes are clean, but the training datasets of other 3 nodes are incorrectly labeled; d) *Attack datasets*: one of the 10 distributed nodes will submit a random model update in each iteration while the others train normally. In this experiment, except for the *Clean datasets* scenario, we fix $D_{i,j}^t = 3000$ for other scenarios in each iteration.

Figure 3 shows the average model accuracy after running 30 iterations, and we can make two major statements. First, FAIR can outperform the FA algorithm in all scenarios for almost all models and datasets. Second, the model aggregation performance of the FA algorithm can deteriorate dramatically when degrading the learning qualities of model updates, but FAIR can work robustly under all scenarios. Taking the LeNet learning model as an example, the model accuracy achieved by the FA algorithm can decrease from 94.42% in the *Clean datasets* scenario to 12.72% in the *Attack datasets* scenario, but the accuracy achieved by FAIR only decreases from 95.60% to 82.71%. Similar observations can also be achieved for other learning models under other scenarios.

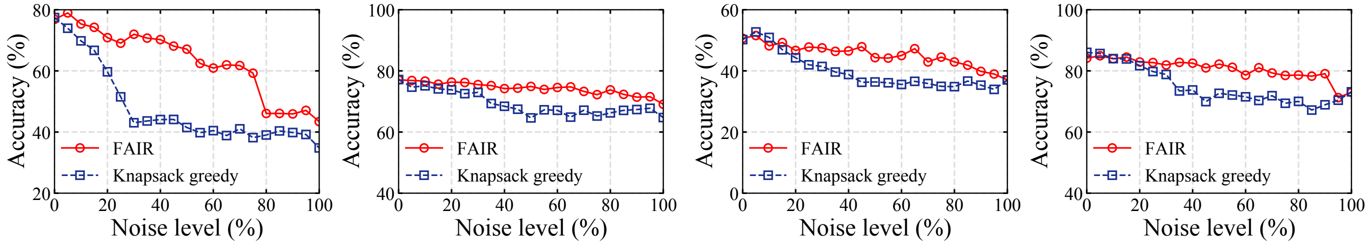
D. *Performance of Federated Learning*

We finally evaluate the federated learning performance of FAIR by integrating the user incentive and model aggregation, where the impact of data quality and learning budget is investigated. Specifically, we compare the performance of FAIR with the knapsack greedy mechanism that adopts the FA algorithm for model aggregation. The experiment setting II in Table I is adopted, where we fix $D_{i,j}^t = 5000$ for each node[2] in each iteration and vary the noise level of the federated community from 0% to 100%. The noise level refers to the percentage of nodes within the federated community that has 50% mislabeled data. In addition, the learning budget is set to be 10 in each iteration. After running 30 iterations, we plot the average accuracy of the MLP, LeNet, MobileNet, and EfficientNet models on their corresponding datasets in Fig. 5, and we can achieve two major observations. First, under almost all settings, FAIR can outperform the knapsack greedy mechanism, and the performance gap becomes significant when the noise level is within the range of 20% and 80%. Second, although the learning quality decreases with the noise level for both mechanisms, the performance of the knapsack greedy mechanism decreases dramatically within low noise levels (e.g., $\leq 40\%$), while the performance of FAIR remains stable within the low noise levels.

We further investigate the impact of the learning budgets, where the experiment setting III in Table I is adopted. There are 100 distributed nodes and 4 learning tasks in each iteration. Among the 100 nodes, the training datasets of 70 nodes are clean but the other 30 nodes have noisy training datasets with different levels of data qualities. With different learning budgets, we plot the average test loss reduction of the MLP, LeNet, MobileNet, and EfficientNet models on their corresponding datasets in Fig. 6. We can observe that for all mechanisms,

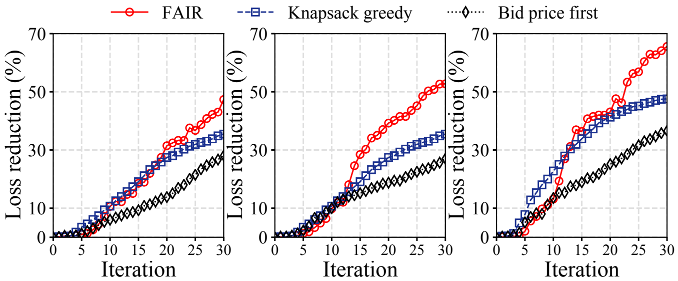
[1] Note that, the incentive process is not considered in this experiment.

[2] Note that, when the data size $D_{i,j}^t$ is fixed, the performance of knapsack greedy mechanism is equivalent to that of the bid price first mechanism.



a) MLP MNIST b) LeNet FMNIST c) MobileNet CIFAR10 d) EfficientNet SVHN

Fig. 5. The model accuracy vs. noise levels.



a) Budget=10 b) Budget=20 c) Budget=30

Fig. 6. The average test loss reduction of learning models vs. learning budgets.

the learning quality can improve with the learning budget. In addition, after 30 iterations, FAIR can achieve supreme performance under all settings when compared to the knapsack greedy and bid price first mechanisms.

VII. Related Work

A. Federated Learning

Recently, considerable research attention has been dedicated to performance optimization for federated learning. Tran *et al.* formulated an optimization problem of federated learning, aiming to balance the trade-offs between computation and communication latency, as well as the learning time and energy consumption [25]. Wang *et al.* theoretically analyzed the convergence rate of federated learning and achieved a desirable trade-off between local update and global parameter aggregation [13]. To reduce the communication overhead of federated learning, Wang *et al.* proposed Communication-Mitigated Federated Learning which avoids uploading irrelevant updates to the server [26]. Liu *et al.* proposed momentum federated learning which uses momentum gradient descent in the local update step to accelerate convergence [27]. Despite the above efforts in federated learning, they are based on volunteer participation. However, without effective incentives, it is difficult to require participants who are neither necessarily very capable nor motivated to complete the allocated learning tasks at a satisfactory level of quality.

B. Incentive Mechanism

A few researchers have proposed some incentive mechanisms for federated learning. Kang *et al.* proposed an incentive mechanism combining reputation with contact theory to encourage high-reputation nodes to participate in learning [15]. Pandey *et al.* proposed an incentive mechanism based on the Stackelberg game to improve the global model with communication efficiency [16]. Zhan *et al.* proposed a deep reinforcement learning-based incentive mechanism to determine the optimal pricing strategy for the server and the optimal training strategies for edge nodes [17]. However, none of them considers the learning quality of participants, which can bias the incentive direction. Different from them, we propose an auction-based quality-aware incentive mechanism for federated learning, which can facilitate precise user incentive and model aggregation. Although incentive mechanisms have been extensively studied in mobile crowdsourcing/crowdsensing systems [28]–[30] and offloading techniques [31]–[33], they cannot be directly applied to federated learning due to the unique characters of federated learning (e.g., learning quality). Furthermore, rather than an incentive mechanism design alone, FAIR integrates a quality-aware model aggregation algorithm to jointly build high-quality federated learning models.

VIII. Conclusion and Future Works

In this paper, we have proposed FAIR, a novel quality-aware federated learning system, which can significantly enhance the distributed learning quality with precise user incentive and model aggregation. Particularly, we have designed and implemented three technical components in FAIR: 1) learning quality estimation, 2) quality-aware incentive mechanism, and 3) model aggregation. In addition, we have theoretically proved FAIR to be truthful, individually rational, and computationally efficient. Extensive experiments under various distributed learning scenarios have been carried out, and the results have demonstrated the efficacy of FAIR in terms of the user incentive and model aggregation, as well as the distributed learning. For future work, we will further integrate the communication and computation performance into FAIR to enhance its robustness when employed in practical systems.

Acknowledgment

We would like to thank Chao Wu for his help in improving this paper. This research was supported in part by National Natural Science Foundation of China under Grants No. 62002389, 62001180, 62072472, 61702562 and U19A2067, the Key-Area Research and Development Program of Guangdong Province under Grant No. 2019B010137005, National Key R&D Program of China under Grant No. 2019YFA0706403, Natural Science Foundation of Hunan Province, China under Grant No. 2020JJ2050, 111 Project under Grant No. B18059, the Young Elite Scientists Sponsorship Program by CAST under Grant No. 2018QNRC001 and the Young Talents Plan of Hunan Province of China under Grant No. 2019RS2001.

References

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[2] N. Cheng, F. Lyu, J. Chen, W. Xu, H. Zhou, S. Zhang, and X. Shen, "Big data driven vehicular networks," *IEEE Network*, vol. 32, no. 6, pp. 160–167, 2018.

[3] Z. Xiong, J. Kang, D. Niyato, P. Wang, H. V. Poor, and S. Xie, "A multi-dimensional contract approach for data rewarding in mobile networks," *IEEE Transactions on Wireless Communications, DOI: 10.1109/TWC.2020.2997023*, Early Access, 2020.

[4] K. R. Sollins, "IoT big data security and privacy versus innovation," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1628–1635, 2019.

[5] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Computing Surveys*, vol. 52, no. 6, 2019.

[6] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246–261, 2020.

[7] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[8] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, Y. Zhang, and X. Shen, "LEAD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2020.2984261*, pp. 1–16, Early Access, Apr. 2020.

[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[10] P. Blanchard, R. Guerraoui, J. Stainer *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 119–129.

[11] R. Guerraoui, S. Rouault *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International Conference on Machine Learning*, 2018, pp. 3518–3527.

[12] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[13] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proceedings of the IEEE INFOCOM'18*, 2018, pp. 63–71.

[14] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proceedings of the IEEE INFOCOM'19*, 2019, pp. 2512–2520.

[15] 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, 2019.

[16] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3241–3256, 2020.

[17] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[18] Y. Liu and M. Liu, "An online learning approach to improving the quality of crowd-sourcing," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2166–2179, 2017.

[19] M. S. Sadik and L. Gruenwald, "DBOD-DS: Distance based outlier detection for data streams," in *International Conference on Database and Expert Systems Applications*. Springer, 2010, pp. 122–136.

[20] T. J. Brailsford, J. H. Penm, and R. D. Terrell, "Selecting the forgetting factor in subset autoregressive modelling," *Journal of Time Series Analysis*, vol. 23, no. 6, pp. 629–649, 2002.

[21] M. Dawande, J. Kalagnanam, P. Keskinocak, F. S. Salman, and R. Ravi, "Approximation algorithms for the multiple knapsack problem with assignment restrictions," *Journal of combinatorial optimization*, vol. 4, no. 2, pp. 171–186, 2000.

[22] M. S. Hung and J. C. Fisk, "An algorithm for 0-1 multiple-knapsack problems," *Naval Research Logistics Quarterly*, vol. 25, no. 3, pp. 571–579, 1978.

[23] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[24] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," *arXiv preprint arXiv:1909.05125*, 2019.

[25] N. H. Tran, W. Bao, A. Zomaya, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proceedings of the IEEE INFOCOM'19*, 2019, pp. 1387–1395.

[26] L. WANG, W. WANG, and B. LI, "CMFL: Mitigating communication overhead for federated learning," in *Proceedings of the IEEE ICDCS'19*, 2019, pp. 954–964.

[27] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 8, pp. 1754–1766, 2020.

[28] X. Gong and N. Shroff, "Incentivizing truthful data quality for quality-aware mobile data crowdsourcing," in *Proceedings of the ACM Mobihoc'18*, 2018, pp. 161–170.

[29] H. Wang, S. Guo, J. Cao, and M. Guo, "MeLoDy: A long-term dynamic quality-aware incentive mechanism for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 4, pp. 901–914, 2017.

[30] H. Gao, C. H. Liu, J. Tang, D. Yang, P. Hui, and W. Wang, "Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2589–2603, 2018.

[31] X. Zhuo, W. Gao, G. Cao, and S. Hua, "An incentive framework for cellular traffic offloading," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 541–555, 2013.

[32] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1634–1647, 2014.

[33] H. Shah-Mansouri, V. W. Wong, and J. Huang, "An incentive framework for mobile data offloading market under price competition," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 2983–2999, 2017.