Improving Federated Learning With Quality-Aware User Incentive and Auto-Weighted Model Aggregation

Yongheng Deng, Feng Lyu, *Senior Member, IEEE*, Ju Ren, *Senior Member, IEEE*, Yi-Chao Chen, *Member, IEEE*, Peng Yang, *Member, IEEE*, Yuezhi Zhou, *Senior Member, IEEE*, and Yaoxue Zhang, *Senior Member, IEEE*

Abstract—Federated learning enables distributed model training over various computing nodes, e.g., mobile devices, where instead of sharing raw user data, computing nodes can solely commit model updates without compromising data privacy. The quality of federated learning relies on the model updates contributed by computing nodes training with their local data. However, with various factors (e.g., training data size, mislabeled data samples, skewed data distributions), the model update qualities of computing nodes can vary dramatically, while inclusively aggregating low-quality model updates can deteriorate the global model quality. To achieve efficient federated learning, in this paper, we propose a novel framework named *FAIR*, i.e., Federated leArning with qualIty awaReness. Particularly, *FAIR* integrates three major components: 1) learning quality estimation: we adopt the model aggregation weight (learned in the third component) to reversely quantify the individual learning quality of nodes in a privacy-preserving manner, and leverage the historical learning records to infer the next-round learning quality; 2) quality-aware incentive mechanism: within the recruiting budget, we model a reverse auction problem to stimulate the participation of high-quality and low-cost computing nodes, and the method is proved to be truthful, individually rational, and computationally efficient; and 3) auto-weighted model aggregation: based on the gradient descent method, we devise an auto-weighted model aggregation algorithm to automatically learn the optimal aggregation weights to further enhance the global model quality. Based on real-world datasets and learning tasks, extensive experiments are conducted to demonstrate the efficacy of *FAIR*.

Index Terms—Edge computing, incentive mechanism, learning quality, mobile computing, model aggregation, federated learning

1 Introduction

With the rapid development of the Internet of Things (IoT), a gigantic amount of data is continuously generated at the network edge, providing opportunities to enable learning-based intelligent services [1], [2]. Traditionally, the centralized learning framework requires aggregating large amounts of training data into a cloud center for model training. However, it can lead to leakage of user privacy [3]. Besides, both the data delivery overhead for power-constrained mobile devices and the cost of data maintenance in the cloud, are prohibitive for practical system implementation and operation [4]. Recently, with the emerging of the mobile edge computing (MEC) technology, mobile devices can be equipped with significant computing and storage capability to support local computing and model training [5], [6]. MEC has also promoted the research of federated learning [7], [8], [9], which allows a community of computationally-capable nodes to collaboratively build global learning models without compromising user privacy. Specifically, federated learning is a distributed learning framework, where distributed computing nodes independently train the global model with their local data and only the model updates are committed to the cloud server for aggregation. In this way, distributed model updates can be aggregated to enhance the global model quality in a privacy-preserving manner.

The success of federated learning is highly dependent on the participation of a large number of nodes that contribute sufficient training data. However, it is computation- and communication- consuming for participating nodes to collaboratively train the federated learning models. Therefore,

- *Yongheng Deng, Ju Ren, Yuezhi Zhou, and Yaoxue Zhang are with the Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing 100084, China. E-mail: dyh19@mails.tsinghua.edu.cn, {renju, zhangyx}@tsinghua.edu.cn, zhouyz@mail.tsinghua.edu.cn.*
- *Feng Lyu is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China. E-mail: fenglyu@csu.edu.cn.*
- *Yi-Chao Chen is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: yichao@cs.sjtu.edu.cn.*
- *Peng Yang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. E-mail: yangpeng@hust.edu.cn.*

Manuscript received 9 October 2021; revised 26 July 2022; accepted 27 July 2022. Date of publication 1 August 2022; date of current version 23 August 2022.
This work was supported in part by the National Key R&D Program of China under Grant 2022YFC2009800, in part by the National Natural Science Foundation of China under Grants 62002389, 62001180, 62122095, 62072472, and U19A2067, in part by a grant from the Guoqiang Institute, Tsinghua University, the Young Elite Scientists Sponsorship Program by CAST under Grant YESS20200238, in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2019B010137005, in part by the Natural Science Foundation of Hunan Province, China under Grants 2020JJ2050 and 2021JJ20079, in part by the Key R&D Program of Hunan Province of China under Grant 2022GK2013, and in part by the Young Talents Plan of Hunan Province of China under Grant 2021RC3004.
(Corresponding author: Feng Lyu.)
Recommended for acceptance by V. Cardellini.
Digital Object Identifier no. 10.1109/TPDS.2022.3195207

establishing an efficient incentive mechanism for federated learning is crucial. To this end, 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 [10]. In [11], Le *et al.* proposed an auction-based incentive mechanism to maximize the social welfare of the wireless federated learning services market. Pandey *et al.* adopted a two-stage Stackelberg game model to optimize the utility of the server and participating nodes jointly [12]. Although these incentive mechanisms are useful for motivating mobile nodes to participate in federated learning, none of them considers the learning quality of participating nodes, which results in suboptimal federated learning performance in terms of global model accuracy and convergence speed. For example, for nodes with incorrectly labeled or non-IID (not independent and identically distributed) local datasets, aggregating their generated model updates inclusively can deteriorate the performance of federated learning significantly.

To bridge this gap, in this paper, we investigate quality-aware federated learning, where the individual learning quality of nodes is estimated to facilitate precise user incentive and model aggregation. Our goal is to select high-quality nodes within the learning budget in a privacy-preserving way and weight their model updates to maximize the global model quality. To this end, we have to tackle the following technical challenges. First, as there are various factors that can affect the learning quality of participating nodes, e.g., the training data size, mislabeled data samples, non-IID data distribution, it is quite challenging to quantify their impacts on the global learning model with refined mathematical models. Besides, due to the privacy issue, it is usually inaccessible to the information of these quality-influencing factors, posing additional challenges. Second, even with the estimated learning quality, it is non-trivial to recruit suitable nodes for learning tasks by determining both-satisfied payments, especially when the learning budget is limited, since participants are usually strategically selfish with quite different data/computing/communication resources. Third, for the model updates committed by the recruited participating nodes, how to aggregate them is also crucial to the global model quality, hereby expecting an efficient model aggregation algorithm that can assign appropriate weights to them with considering their differential qualities.

To address the aforementioned challenges, we propose a learning optimization framework named *FAIR*, i.e., Federated leArning with qualIty awaReness, to determine the learning task allocation with payments, and conduct model aggregation. Functionally, *FAIR* integrates three major technical components: 1) learning quality estimation, 2) quality-aware incentive mechanism, and 3) auto-weighted model aggregation. We first design an online quality quantification method to quantify the individual learning quality of nodes in a privacy-preserving manner, and leverage the historical quality records to infer the next-round learning quality to assist in learning task allocation. With the estimated quality, a reverse auction case is then built to stimulate user participation, where mobile users submit their bids and the platform serves as the auctioneer. To maximize the learning quality, we formulate a Learning Quality Maximization (LQM) problem, which is proved to be NP-hard, and thus we devise a light-weight algorithm to determine the learning task allocation and reward distribution within the recruiting budget. Finally, we devise an auto-weighted model aggregation algorithm based on the gradient descent method that can automatically learn the optimal model aggregation weights to further enhance the global learning model.

Theoretical analysis demonstrates that the proposed *FAIR* is truthful, individually rational, and computationally efficient. To evaluate the performance of *FAIR*, we build an simulation system based on real-world datasets and widely adopted learning models. Extensive experiments under various scenarios are carried out, and the results demonstrate the efficacy of *FAIR*. Particularly, *FAIR* advances in both the user incentive and model aggregation, jointly contributing to the superior federated learning performance that can outperform the benchmark approaches significantly.

We highlight our major contributions as follows.

- We investigate the quality-aware federated learning, where the individual learning quality of nodes is estimated to facilitate precise user incentive and model aggregation. It is critical in practical federated learning scenarios, but to our best knowledge, is rarely seen in the literature.
- We propose *FAIR* to determine the learning task allocation and the learning payment to nodes, and conduct model aggregation with automatically learned weights. In *FAIR*, we design and implement three key components: 1) learning quality estimation, 2) quality-aware incentive mechanism, and 3) auto-weighted model aggregation.
- Extensive experiments are conducted to demonstrate the efficacy of *FAIR*, where the incentive mechanism can facilitate more high-quality model updates, and the devised aggregation algorithm can effectively aggregate the model updates, collectively contributing to an advanced globe learning model.

The remainder of this paper is organized as follows. We give the system description and problem definition in Section 2. In Section 3, we present the system overview of *FAIR* with highlighted design goals. We elaborate on the design of *FAIR* in Section 4, and conduct the theoretical analysis in Section 5. Extensive experiments are conducted to evaluate the performance of *FAIR* in Section 6, and the related work is reviewed in Section 7. Finally, we conclude this paper and direct our future work in Section 8.

2 System Description and Problem Definition

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

2.1 System Description

As shown in Fig. 1, we consider a distributed federated learning system, where there are one cloud platform and

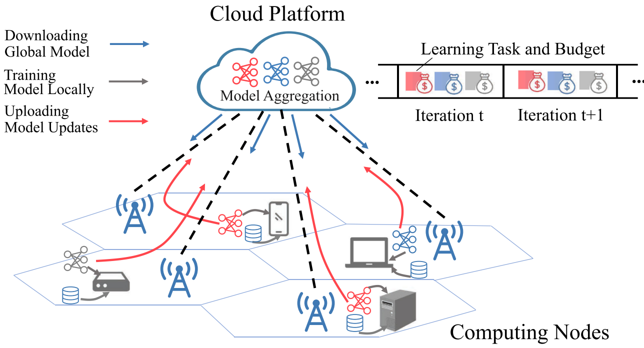


Fig. 1. An overview of the distributed federated learning system.

various distributed computing nodes denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$. The system operates as a federated learning services market in a time-slotted manner, where the time span is partitioned into T consecutive slots, and in each iteration t, a set of learning tasks $\mathcal{L}^t = \{l_1^t, l_2^t, \ldots, l_j^t, \ldots\}$ with budgets $\mathcal{B}^t = \{B_1^t, B_2^t, \ldots, B_j^t, \ldots\}$ are submitted to the cloud platform. Here, each task l_j^t represents a model training task submitted by companies or users, e.g., it can be a traffic sign recognition task that requires computing nodes with traffic sign data samples training the submitted model. In each iteration t, the cloud platform publishes the received learning tasks $\mathcal{L}^t$ to computing nodes, and recruits a set of participating nodes for each $l_j^t \in \mathcal{L}^t$ within budget B_j^t. Specifically, each node i notifies the cloud platform the set of tasks $\mathcal{L}_i^t \subseteq \mathcal{L}^t$ that it is willing to participate in, based on which the cloud platform allocates the learning tasks to nodes and determines according payments. Note that, since the local data of computing nodes are diverse, they can participate in different learning tasks, but each node is constrained to participate in at most one task in each iteration as the computing capacity of nodes is usually limited. Once a computing node is recruited to participate in a learning task, it downloads the corresponding global model from the cloud platform, trains the model locally using local data, and commits the model updates to the cloud platform for aggregation. The cloud platform separately aggregates the received model updates from participants to update the model for each task and finally the updated model of each task is sent back to its submitters. Note that if the model accuracy reached in iteration t is not satisfactory, the model can still be submitted along with a budget in iteration $t+1$ for further learning. Table 1 shows the key notations that are used in this paper.

2.2 Problem Definition

In each iteration t, given the learning budget, the platform has to determine which learning task is executed by which computing nodes (i.e., the learning task allocation) at what price (i.e., determining the payment). In order to complete the submitted learning tasks with high quality in every iteration, we cast the following *quality-aware federated learning* problem.

Definition 1 (Quality-Aware Federated Learning Problem). *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 the quality of all aggregated learning models is maximized?*

TABLE 1
Key Notations

Notation	Definition
T	The number of total time slots
t	The index of time slot
$\mathcal{N}$	The set of distributed computing nodes
i	The index of node in $\mathcal{N}$
$\mathcal{L}^t$	The set of learning tasks in iteration t
$\mathcal{L}_i^t$	The set of tasks that the node i can participate in
l_j^t	The jth learning task in $\mathcal{L}^t$
$\mathcal{B}^t$	The set of learning budgets in iteration t
B_j^t	The learning budget issued for the task l_j^t
$s_{i,j}^t$	The indicator of whether task l_j^t is allocated to node i
$r_{i,j}^t$	The reward paid to node i in iteration t
$c_{i,j}^t$	The learning cost of node i on task l_j^t
$u_{i,j}^t$	The utility of node i in iteration t
$\mathbb{B}_i^t$	The bid information of node i in iteration t
$b_{i,j}^t$	The bid price of node i on task l_j^t
$\widehat{q}_{i,j}^t$	The estimated quality of node i on task l_j^t
$q_{i,j}^t$	The learning quality of node i on task l_j^t
$D_{i,j}^t$	The data size of node i used to train the task l_j^t
$\mathbf{M}^t$	The learning task allocation results
$\mathcal{M}_j^t$	The set of winner nodes for task l_j^t in iteration t
$\mathbf{R}^t$	The payment determination results
$\mathcal{R}_j^t$	The set of payments for winner nodes $\mathcal{M}_j^t$
$\lambda_{i,j}^t$	The aggregation weight of node i on task l_j^t
$\boldsymbol{w}_{i,j}^t$	The local model parameters of node i on task l_j^t
$\boldsymbol{w}_j^t$	The aggregated model parameters of task l_j^t

A binary variable $s_{i,j}^t \in \{0,1\}$ is used to indicate whether the task l_j^t 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 $\mathcal{M}_j^t = \{i\}_{s_{i,j}^t=1, \forall i \in \mathcal{N}}$ the nodes assigned with task l_j^t, and $r_{i,j}^t$ represents the payment reward to node $i \in \mathcal{M}_j^t$. For each learning task l_j^t, the model updates $\boldsymbol{w}_{i,j}^t$ received from nodes $i \in \mathcal{M}_j^t$ will be aggregated with weight $\lambda_{i,j}^t$ to update the global model $\boldsymbol{w}_j^t$. Then, the quality-aware federated learning problem can be formulated as

$$\max_{\mathbf{M}^t, \mathbf{R}^t, \boldsymbol{\lambda}^t} \quad \sum_{l_j^t \in \mathcal{L}^t} Acc(\boldsymbol{w}_j^t), \tag{1}$$

$$\boldsymbol{w}_j^t = \sum_{i \in \mathcal{M}_j^t} \lambda_{i,j}^t \boldsymbol{w}_{i,j}^t, \tag{2}$$

$$s.t. \quad s_{i,j}^t \in \{0,1\}, \quad \forall i \in \mathcal{N}, \forall l_j^t \in \mathcal{L}^t, \tag{3}$$

$$\sum_{i \in \mathcal{N}} r_{i,j}^t s_{i,j}^t \le B_j^t, \quad \forall l_j^t \in \mathcal{L}^t, \tag{4}$$

$$s_{i,j}^t = 0, \quad \forall l_j^t \notin \mathcal{L}_i^t, \forall i \in \mathcal{N}, \tag{5}$$

$$\sum_{l_j^t \in \mathcal{L}^t} s_{i,j}^t \le 1, \quad \forall i \in \mathcal{N}, \tag{6}$$

$$\sum_{i \in \mathcal{M}_j^t} \lambda_{i,j}^t = 1, \quad \forall l_j^t \in \mathcal{L}^t, \tag{7}$$

where we utilize model accuracy as a measurement of the model quality. In addition, $\mathbf{M}^t = \{s_{i,j}^t\}_{\forall i \in \mathcal{N}, \forall l_j^t \in \mathcal{L}^t}$, $\mathbf{R}^t =$

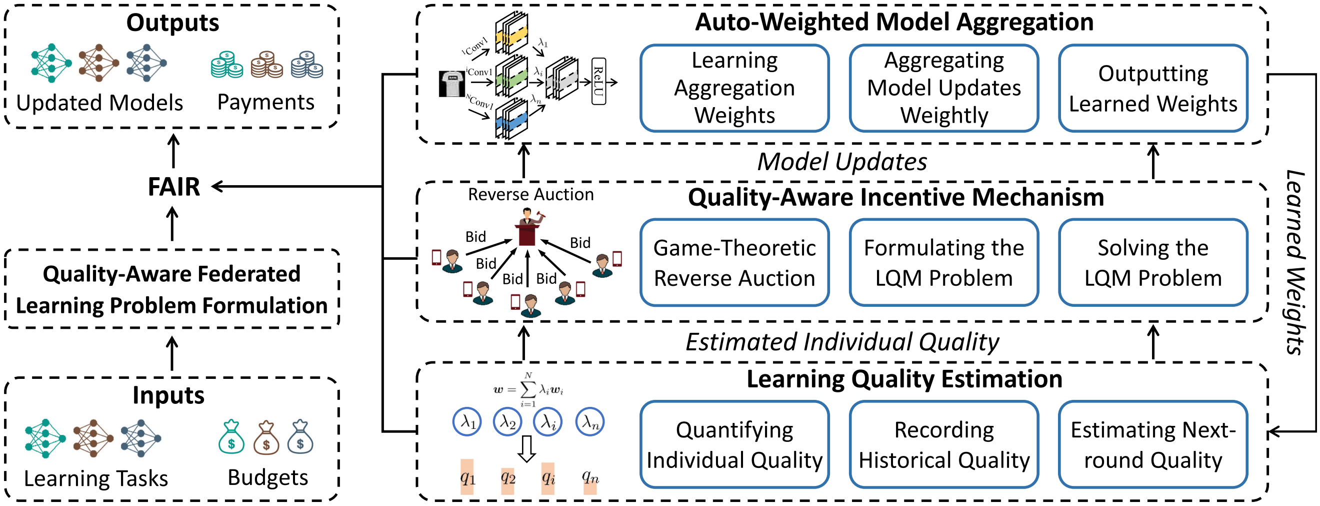


Fig. 2. Architecture of *FAIR*.

$\{r^t_{i,j}\}_{\forall i\in\mathcal{N},\forall l^t_j\in\mathcal{L}^t}$, and $\lambda^t = \{\lambda^t_{i,j}\}_{\forall i\in\mathcal{N},\forall l^t_j\in\mathcal{L}^t}$ represent the learning task allocation results, the payment determination results, and the model aggregation weights, respectively. Constraint (4) represents that for each learning task, the sum of payments should not exceed the learning budget provided by the task submitter. Constraint (5) means that each node can only be allocated with the learning tasks that it can participate in. In constraint (6), each node is limited to participate in at most one learning task in every iteration. Constraint (7) limits the aggregation weights of the received model updates.

2.3 Problem Difficulties

The formulated quality-aware federated learning problem is intractable directly for the following reasons. The first challenge comes from the task allocation. To boost the quality of the learning models, it would be promising to select high-quality participants for each learning task, i.e., the nodes whose model updates would contribute more to the convergence of the task's model. However, how to quantify the learning quality of nodes becomes a hurdle. Because there are various factors of participating nodes, such as data size used for training, the noise level of data labels, and data distribution skewness, which can affect the learning quality intricately, but there is no established model to quantify their impacts on the learning qualities of both the individual model update and aggregated global model. Besides, due to the privacy issue, it is usually inaccessible to the raw data of each node, and thus we cannot achieve the data profiles directly, such as the data label noise, data distribution. Additionally, the learning quality of a node in each iteration is unknown before learning, which exacerbates the challenge of task allocation. The second challenge arises in determining 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 difficult to model the learning cost of each participating node. On the other hand, participants are often strategically selfish, and they tend to claim a higher cost than the real one in order to increase their individual learning profits. Model aggregation encounters another challenge, i.e., how to aggregate the received model updates from participants is important, but finding the optimal aggregation weights for them is nontrivial.

In this paper, we propose *FAIR* to systematically address those challenges, and thus provide an efficient solution to the quality-aware federated learning problem.

3 Overview of *FAIR*

In this section, we first present the design overview of *FAIR* and then highlight its design goals.

3.1 Design Overview

As shown in Fig. 2, *FAIR* integrates three major components: 1) *learning quality estimation*, 2) *quality-aware incentive mechanism*, and 3) *auto-weighted model aggregation*. Specifically, to mathematically pinpoint the optimization problem, we first utilize the model aggregation weights learned in the *auto-weighted model aggregation* component to quantify the individual quality of each participating node in a privacy-preserving manner, and then leverage the historical quality records to estimate the next-round quality. With the estimated individual quality, we then model the interaction between the platform and distributed 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 price $b^t_{i,j}$. Working with the reverse auction mechanism, we formulate the *Learning Quality Maximization* (LQM) problem (proved to be NP-hard), and devise a light-weight algorithm to allocate the learning tasks with payments based on participants' individual quality and bid information. After that, with the model updates from the selected participating nodes, we finally devise a model aggregation algorithm based on the gradient descent method, which can automatically learn the aggregation weights to efficiently aggregate the received model updates. Note that, the aggregation weights learned by the *auto-weighted model aggregation* component are inversely fed to the *learning quality estimation* component for quality quantification in a privacy-preserving manner.

By doing so, the original *quality-aware federated learning* problem can be transformed and solved along with the following two optimization directions. On the one hand, *FAIR* estimates the individual learning quality of candidate nodes and adopts the quality-aware incentive mechanism to recruit as many high-quality model updates as possible. On the other hand, *FAIR* automatically learns the optimal aggregation weights to efficiently aggregate the recruited model updates, which can further enhance the learning qualities of aggregated global models.

3.2 Design Goals

FAIR aims to maximize the sum of the quality of all aggregated learning models in each iteration, while ensuring *truthfulness*, *individual rationality*, and *computational efficiency*. Since computing nodes are strategically selfish, the truthfulness goal is set to avoid nodes announcing untruthful bid prices. To quantify the benefits of computing nodes 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.,*

$$u_{i,j}^t = \begin{cases} r_{i,j}^t - c_{i,j}^t, & \text{if } i \in \mathcal{M}_j^t; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Then, the design goals are defined as follows.

Definition 3 (Truthfulness). *A mechanism is truthful if, in each iteration t, no computing 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 is truthful in iteration t, its utility is $u_{i,j}^t$, otherwise $\widehat{u}_{i,j}^t$. We have $u_{i,j}^t \geq \widehat{u}_{i,j}^t$ 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 learning task allocation, payment determination, and model aggregation can be conducted within a polynomial time.*

4 Design of *FAIR*

In this section, we elaborate on the main components of *FAIR:* 1) learning quality estimation, 2) quality-aware incentive mechanism, and 3) auto-weighted model aggregation.

4.1 Estimating Learning Quality

4.1.1 Learning Quality Quantification

In order to encourage high-quality nodes to participate in federated learning, the first item is to quantify the individual learning qualities of computing nodes. For federated learning systems, constrained by the storage/computing capacity and data resources, the data size of each participating node that can be used for model training varies dramatically. Besides, the issues of mislabeled samples and skewed data distributions are also common in local data sets. These factors can collectively affect the individual learning quality of participating nodes, which has been verified by field experiments [13], [14], but their impacts are complicated and hard to be mathematically modeled for quantification. Even more, the information of these affecting factors is inaccessible directly due to the privacy issue, posing another dimensional challenge. One seemingly plausible approach is to evaluate the accuracy of individual model update to quantify the participant's learning quality. For instance, we can test the accuracy of participant's local training model on a small test dataset maintained by the server as their individual learning quality. The weakness is that the quantified quality cannot precisely reflect the contribution of a participant to the global aggregated model, resulting in unexpected bias in implementing the quality-aware incentive mechanism. Therefore, we define the quality of a node as the contribution of its provided model updates to the convergence of the global model. As the overall contribution of a participant is correlated with other participants and varies with time, we propose an online quality quantification approach in this paper. The main idea is that, the contribution of participants can be embodied by the aggregation weight of its model update during the model aggregation. Therefore, we adopt the aggregation weight to reversely quantify the individual learning quality in real time. Specifically, for each learning task l_j^t in iteration t, we define the learning qualities of its L participating nodes as $\boldsymbol{q}_j^t = [q_{1,j}^t, q_{2,j}^t, \ldots q_{L,j}^t]$, and the aggregation weights used for aggregating the model updates from participating nodes are defined as $\boldsymbol{\lambda}_j^t = [\lambda_{1,j}^t, \lambda_{2,j}^t, \ldots, \lambda_{L,j}^t]$. We use a *softmax* function to transform $\boldsymbol{q}_j^t$ to $\boldsymbol{\lambda}_j^t$

$$\begin{aligned} \boldsymbol{\lambda}_j^t &= softmax(\boldsymbol{q}_j^t), \\ \lambda_{i,j}^t &= \frac{\exp(q_{i,j}^t)}{\sum_{l=1}^{L} \exp(q_{l,j}^t)}. \end{aligned} \tag{9}$$

The online quality quantification mechanism of *FAIR* works as follows. After receiving the model updates from participating nodes, *FAIR* automatically learns the optimal aggregation weight $\lambda_{i,j}^t$ (detailed in Section 4.3), and then the quality value $q_{i,j}^t$ can be obtained based on the learned aggregation weight $\lambda_{i,j}^t$. In our implementation, we define $\boldsymbol{q}_j^t$ as a learnable vector, and utilize Eq. (9) to obtain the learnable vector $\boldsymbol{\lambda}_j^t$, such that $\boldsymbol{\lambda}_j^t$ and $\boldsymbol{q}_j^t$ can be learned simultaneously. In this way, the quantified $q_{i,j}^t$ can not only describe the individual learning quality, but also reflect the overall contribution of each participant.

4.1.2 Learning Quality Estimation

With the learning quality quantification mechanism, *FAIR* can obtain the individual quality of participating nodes after receiving their committed model updates. However, to allocate the learning tasks properly, it is necessary to have the quality values of all candidate nodes ahead at the beginning of each iteration round. Therefore, in each iteration, *FAIR* first estimates the individual quality of candidate participants to assist in allocating tasks with payments. As the system runs iteratively, *FAIR* leverages the historical quality records of a participant to estimate its next-round quality. Specially, in the first iteration of each learning task, we let all candidate nodes participate in the learning process to get their initial quality values. Afterwards, supposing that node i has participated in the learning task l_j in iteration $t_0, t_1, \ldots, t_r$, we can use 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 vary 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 based on their freshness [15]. Specifically, we employ an exponential forgetting function to assign the weights, which assigns larger weights to the recent quality records and smaller weights to the stale ones [16]. The most recent quality record receives the weight of 1 and the other records are weighted 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 [17]. Then, the estimated quality value $\widehat{q}_{i,j}^t$ can be obtained as

$$\widehat{q}_{i,j}^t = \frac{\sum_{k=0}^{r} \rho^{t_r-t_k} q_{i,j}^{t_k}}{\sum_{k=0}^{r} \rho^{t_r-t_k}}. \tag{10}$$

4.2 Quality-Aware Incentive Mechanism

After estimating the individual quality for each candidate node, we then solve the defined *quality-aware federated learning* problem in two steps. Within the learning budget, we first encourage high-quality and low-cost computing nodes to participate in the learning tasks via a quality-aware incentive mechanism. Then, with model updates, we devise an aggregation algorithm to further enhance the learning performance. In this subsection, we focus on the design of the quality-aware incentive mechanism. Specifically, in each iteration, we model a reverse auction case where each node i submits the 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. Here, the reverse auction mechanism is chosen for the following reasons. First, as the model learning costs tend to vary among nodes, it is hard and inappropriate for the platform to pay a fixed price for each participating node. Therefore, we employ a reverse auction to have each node bid to the platform with the price of completing a task in accordance with the learning cost. Second, reverse auction is beneficial to solve the defined optimization problem, since the computing nodes are competing with each other to win over a learning task, which can encourage nodes to improve their learning quality and reduce their bid price via designing a delicate incentive scheme. Therefore, based on the reverse auction mechanism, 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 quality of the selected nodes is maximized?*

The defined LQM problem can be formulated as

$$\begin{aligned} \max_{\mathcal{M}_j^t, \mathcal{R}_j^t} \quad & \sum_{l_j^t \in \mathcal{L}^t} \sum_{i \in \mathcal{M}_j^t} \widehat{q}_{i,j}^t, \\ s.t. \quad & (3), (4), (5), (6), \\ & truthfulness, \\ & individual\ rationality, \\ & computational\ efficiency. \end{aligned} \tag{11}$$

Inputs. The LQM problem takes the learning 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 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 of the LQM problem include all the constraints of the *quality-aware federated learning* problem, as well as the goals of truthfulness, individual rationality, and computational efficiency. For the LQM problem, we have the following Theorem.

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

Proof. To prove its NP-hardness, we design a polynomial reduction from a classic NP-hard problem, i.e., *Multiple Knapsack Problem with Assignment Restrictions* (MKAR) [18], which is a variant of the NP-hard problem of *Multiple Knapsack Problem* (MKP) [19], 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 its capacity. Afterwards, 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 NP-hard, which concludes the proof. □

Given the NP-hardness of the LQM problem, we design a heuristic algorithm to solve the LQM problem with truthfulness, individual rationality, and computational efficiency. Myerson's theorem [20] of truthfulness has demonstrated 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 bid price $b_{i,j}^t$ for performing the learning task, it will still win with any 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 price $\widehat{b}_{i,j}^t$, 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 price value for a bid to win.

Algorithm 1. Solving the LQM Problem

Input: (1) bid price $b^t_{i,j}$; (2) budget B^t_j; (3) task set $\mathcal{L}^t_i$; (4) quality estimation values $q^t_{i,j}$.

Output: (1) the task allocation results $s^t_{i,j}$; (2) the payments $r^t_{i,j}$.

1 Initialize $\mathcal{N}^t_j \leftarrow \emptyset$, $p^t_j \leftarrow 0$ **for each** $l^t_j \in \mathcal{L}^t$;
2 Initialize $x^t_i \leftarrow 1$ **for each** $i \in \mathcal{N}$;
3 Initialize $r^t_{i,j} \leftarrow 0, s^t_{i,j} \leftarrow 0$ **for each** $i \in \mathcal{N}$, $l^t_j \in \mathcal{L}^t$;
4 **foreach** $i \in \mathcal{N}$ **do**
5   **foreach** $l^t_j \in \mathcal{L}^t_i$ **do**
6         $\mathcal{N}^t_j \leftarrow \mathcal{N}^t_j + \{i\}$;
7   **end**
8 **end**
9 **while** $\exists\ x^t_i = 1$ **and** $\exists\ p^t_j = 0$ **do**
10   Initialize $\mathcal{M}^t_j \leftarrow \emptyset$ **for each** $l^t_j \in \mathcal{L}^t$;
11   **foreach** $l^t_j \in \mathcal{L}^t$ **do**
12   **if** $p^t_j = 0$ **then**
13   Sort all $i \in \mathcal{N}^t_j$ in descending order of $\frac{q^t_{i,j}}{b^t_{i,j}}$;
14   Find the smallest k such that $\sum_{i=1}^{k} \frac{b^t_{k,j}}{q^t_{k,j}} q^t_{i,j} x^t_i > B^t_j$;
15   **for** $i \leftarrow 1$ **to** $k-1$ **do**
16                 $\mathcal{M}^t_j \leftarrow \mathcal{M}^t_j + \{i\}$;
17                 $r^t_{i,j} \leftarrow \frac{b^t_{k,j}}{q^t_{k,j}} q^t_{i,j}$;
18   **end**
19   **end**
20   **end**
21   Find the task l^t_k with maximum $\sum_{i \in \mathcal{M}^t_k} q^t_{i,k} x^t_i$;
22   Set $p^t_k \leftarrow 1$;
23   **foreach** $i \in \mathcal{M}^t_k$ **do**
24   **if** $x^t_i = 1$ **then**
25             $s^t_{i,k} \leftarrow 1$;
26             $x^t_i \leftarrow 0$;
27   **end**
28   **end**
29 **end**
30 **return** $(s^t_{i,j}, r^t_{i,j})$;

We use 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 $\mathcal{N}^t_j$ that can participate in the learning task l^t_j (lines 4-8). The main loop (lines 9-29) is then executed until there is no node can participate in the learning tasks or all tasks have been allocated to nodes for execution. In the main loop, the algorithm first selects a subset of winner nodes $\mathcal{M}^t_j \subseteq \mathcal{N}^t_j$ for each task l^t_j that can approximately maximize the sum of the estimated quality of l^t_j's winner nodes (lines 11-20). Specifically, the algorithm sorts node $i \in \mathcal{N}^t_j$ in descending order by $q^t_{i,j}/b^t_{i,j}$, i.e., the quality contribution per unit bid price (line 13). The value of $q^t_{i,j}/b^t_{i,j}$ is a ranking indicator for node i. Then, the algorithm greedily includes nodes into the winner node set $\mathcal{M}^t_j$ according to the rankings until the total payment exceeds the budget B^t_j (lines 14-18). Here, we determine the reward of each participating 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 $q^t_{i,j}/b'_{i,j} = q^t_{k,j}/b^t_{k,j}$. This means the critical payment of node i is the bidding price $b'_{i,j} = \frac{b^t_{k,j}}{q^t_{k,j}} q^t_{i,j}$. The critical payment $b'_{i,j}$ is used as the payment to node i (line 17). Finally, the algorithm finds the task l^t_k with maximum $\sum_{i \in \mathcal{M}^t_k} q^t_{i,k} x^t_i$ (line 21), and allocates task l^t_k to the obtained winner nodes (lines 22-28).

Obviously, Algorithm 1 satisfies the constraints (3), (4), (5), and (6). The constraints truthfulness, individual rationality and computational efficiency will be proved in Section 5.

4.3 Auto-Weighted Model Aggregation

In each iteration t, for each learning task l^t_j, after one or multiple gradient-descent updates, each winner node i will commit their local model parameters $\boldsymbol{w}^t_{i,j}$ to the platform, and then the platform will aggregate them to update the global model parameters $\boldsymbol{w}^t_j$. Given a set of winner nodes $\mathcal{M}^t_j$ as well as their local model parameters, the objective of *FAIR* is to find the optimal aggregation weight $\boldsymbol{\lambda} = \{\lambda^t_{i,j}\}_{i \in \mathcal{M}^t_j}$, such that the loss function $L(\boldsymbol{w}^t_j)$ of the aggregated model of l^t_j can be minimized, i.e., getting $\boldsymbol{\lambda}^* = \arg\min L(\boldsymbol{w}^t_j)$, which can be formulated as

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & L(\boldsymbol{w}^t_j), \\ s.t. \quad & \boldsymbol{w}^t_j = \sum_{i \in \mathcal{M}^t_j} \lambda^t_{i,j} \boldsymbol{w}^t_{i,j}, \\ & \sum_{i \in \mathcal{M}^t_j} \lambda^t_{i,j} = 1, \\ & \lambda^t_{i,j} \geq 0,\ \forall i \in \mathcal{M}^t_j. \end{aligned} \tag{12}$$

In (12), the local model parameters $\{\boldsymbol{w}^t_{i,j}\}_{i \in \mathcal{M}^t_j}$ are fixed, which can be treated as constants, while the aggregation weight variables $\boldsymbol{\lambda}$ are the ones that need to be optimized. To this end, we propose an auto-weighted model aggregation mechanism to gradually optimize the aggregation weights by the gradient descent method. Generally, the cloud platform has a small data set for each learning task, which is used to test the global model accuracy achieved in each iteration. The amount of data maintained in the platform is relatively smaller compared to the data aggregated from all participating nodes, and thus is not enough to train a high-accuracy model. However, the data set usually has a balanced data distribution with correct labels, which can be regarded as a *benchmark dataset* [21]. FAIR utilizes the benchmark dataset to optimize the model aggregation weights to generate high-quality aggregated models. Specifically, for each learning task, the data samples in the benchmark dataset are first fed into the global model for forward propagation. Specially, when the input passes through a linear transformation layer of the global model, the input will actually be fed into the linear transformation layer of each local model, and then the output of each linear transformation layer is aggregated with weights $\boldsymbol{\lambda}$ and then put forward to the next layer of the global model. Then the loss function can be calculated by the model output $\widehat{y}$ and the target y. Afterwards, the backpropagation process is performed to calculate the gradient of the loss function to the model aggregation weights $\boldsymbol{\lambda}$. With the calculated gradients,

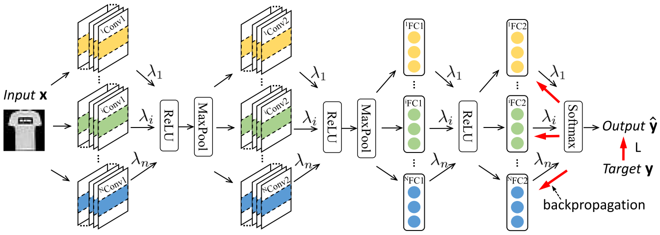


Fig. 3. Auto-weighted model aggregation.

the gradient descent process is executed to update λ. Taking a simple convolutional neural network (CNN)[1] as an example, as shown in Fig. 3, the linear transformation layers of the CNN, e.g., convolutional layers, fully connected layers, are transformed into the local models of the participating nodes weighted with parameters λ, while the nonlinear transformation layers remain unchanged. The images in the benchmark dataset are first fed forward to the first convolutional layer of each local model, then the outputs are weighted by λ and pass through the ReLU activation as well as the MaxPool layer. Likewise, the inputs pass through each layer of the CNN model to calculate the loss function L and then backpropagation and gradient descent are performed to update the weight parameters λ. Overall, compared with directly training the global model with the benchmark dataset, there are the following two differences. One is that in the forward propagation process, the output of the linear transformation layer is the weighted sum of the output of each local model linear transformation layer. The second is that the gradient descent updates the weights of the model aggregation rather than the parameters of the global model. By doing so, the optimal aggregation weights can be automatically learned.

5 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 $b^t_{i,j} = c^t_{i,j}$ or any other untruthful bid price $\widehat{b}^t_{i,j}$. The four bidding results of node i are as follows:

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

As $u^t_{i,j}(b^t_{i,j}) \geq u^t_{i,j}(\widehat{b}^t_{i,j})$ 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 $u^t_{i,j} = 0$. Otherwise, node i wins with truthful bid $b^t_{i,j} = c^t_{i,j}$ since we have proved that nodes bid truthfully. The node utility $u^t_{i,j} = r^t_{i,j} - c^t_{i,j} = \frac{b^t_{k,j}}{q^t_{k,j}} q^t_{i,j} - b^t_{i,j} \geq 0$ due to $\frac{q^t_{i,j}}{b^t_{i,j}} \geq \frac{q^t_{k,j}}{b^t_{k,j}}$. Therefore, $u^t_{i,j} \geq 0$ 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 $\mathcal{O}(E \cdot S \cdot M \cdot L)$, where E is the number of epochs to train the aggregation weight, S is the number of samples in the benchmark dataset of learning task l^t_j, and M is the number of nodes in winner set $\mathcal{M}^t_j$. Both time complexities are polynomial, which are computationally efficient.*

Proof. We analyze the worst case of Algorithm 1 where $|\mathcal{L}^t_i| = L$ and $|\mathcal{N}^t_j| = N$. In the worst case, the main loop in line 9 terminates after L times of iterations. Besides, the computational complexity of sorting $q^t_{i,j}/b^t_{i,j}$ (line 13) is $\mathcal{O}(N \log N)$, where finding the smallest k (line 14) is $\mathcal{O}(N)$, and finding the task l^t_k with maximum $\sum_{i \in \mathcal{M}^t_k} q^t_{i,k} x^t_i$ (line 21) is $\mathcal{O}(NL)$. Therefore, the computational complexity of Algorithm 1 is $\mathcal{O}(L^2 N \log N)$. In the model aggregation algorithm, the data samples in the benchmark dataset of each learning task are passed through the local model of each selected winner node. And we set the maximum number of passes of the entire benchmark dataset as E. Therefore, the computational complexity of the model aggregation algorithm is $\mathcal{O}(E \cdot S \cdot M \cdot L)$. □

6 Performance Evaluation

In this section, we conduct extensive experiments to evaluate the performance of *FAIR*. Specifically, we first detail the evaluation methodology with experiment setup, learning models/datasets, and benchmark design. Then, we evaluate the performance of *FAIR* in terms of incentive and model aggregation. Finally, we investigate the impact of mislabeled and

1. The CNN has the following structure: Convolutional → ReLU → MaxPool → Convolutional → ReLU → MaxPool → Fully connected → ReLU → Fully connected → Softmax.

TABLE 2
Experiment Parameters

Settings	$\|\mathcal{N}\|$	$\|\mathcal{L}^t\|$	$b_{i,j}^t$	$D_{i,j}^t$	N_e	R_e	N_d	B_j^t
I	30	1	[1,3]	[100,1000]	20	(0,1)	20	20
II	30	1	[1,3]	[100,1000]	[0,30)	0.8	0	20
III	30	1	[1,3]	[100,1000]	0	-	[0,30)	20
IV	50	3	[1,3]	[100,1000]	20	(0,1)	40	[5,20]

non-IID data as well as the learning budget on the performance of *FAIR*.

6.1 Evaluation Methodology

Experiment Setup. We build the *FAIR* simulation system by adopting the widely-used PyTorch 1.4.0 software environment. The detailed experiment settings are shown in Table 2, where the bid price, the number of data samples used for training, the number of nodes with mislabeled or non-IID training data, and the learning budget of each task are variable parameters. In Table 2, $|\mathcal{N}|$ and $|\mathcal{L}^t|$ are the number of candidate nodes and learning tasks in each iteration, respectively, and B_j^t is the learning budget for the task l_j^t. For each node i, the bid price $b_{i,j}^t$ and the amount of data $D_{i,j}^t$ used for training in task l_j^t are generated uniformly within ranges [1,3] and [100,1000], respectively. Note that, we set up the scale of computing nodes as well as the amount of training data for each node in accordance with the total number of samples in the adopted datasets. Besides, among $|\mathcal{N}|$ candidate nodes, we assume there are N_e nodes with mislabeled training data samples and N_d nodes with non-IID data distribution, where the nodes with mislabeled or non-IID data are randomly selected from $\mathcal{N}$. For the nodes with mislabeled data, the ratio of mislabeled samples is controlled by the parameter R_e, which is randomly generated among the ranges in Table 2. For the nodes with non-IID data distribution, we set 10 different non-IID levels denoted by: $H \in \{1, 2, \ldots, 7\}$ indicating that the node has only H type labels for samples and they evenly belong to H type labels; or $k \in \{0.8, 0.6, 0.4\}$, where taking an example, $\sigma = 0.8$ indicates that 80% of the data belong to one label and the remaining 20% data evenly belong to other labels. Besides, we set the local epochs of each node to be 2, i.e., the nodes communicate with the platform every 2 epochs to exchange model updates. The number of epochs E to train the model aggregation weights is set to 10.

Models and Datasets. We evaluate the performance of *FAIR* with four commonly adopted learning models, including Multi-layer Perceptron (MLP),[2] LeNet-5 [22], ResNet-18 [23], and MobileNet [24]. The above four models are trained with the following datasets: MNIST [25], Fashion-MNIST (FMNIST) [26], CIFAR-10 [27], and the Street View House Numbers (SVHN) [28] dataset, respectively. 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.

2. It has two hidden layers with 50 neurons each using ReLu activations, and its output layer has a softmax function.

Incentive Benchmarks. To examine the performance of our proposed quality-aware incentive mechanism in *FAIR*, the following reasonable benchmarks are designed.

- *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 in order to recruit as many participating nodes as possible within the budget, but it cannot guarantee the truthfulness of nodes.
- *Random mechanism:* It randomly selects a fraction of participating nodes for each learning task within the budget, which is widely used in state-of-the-art researches, e.g., [7], [29], [30], [31].

Model Aggregation Benchmarks. To compare the performance of our proposed auto-weighted model aggregation component in *FAIR*, the following two reasonable benchmarks are adopted.

- *FedAvg:* It is an efficient model aggregation algorithm for federated learning, which has been widely adopted in current federated learning frameworks. In the algorithm, the aggregation weights of committed model updates are determined by the number of data samples used for training [7], [32], [33];
- *FAIR INFOCOM:* It is a quality-aware model aggregation algorithm proposed in our previous work, where the aggregation weights are calculated based on the amount of training data and the local training loss reduction [34].

6.2 Performance of User Incentive

We first investigate the user incentive performance in *FAIR*. We adopt the experiment setting I in Table 2, where there is only one learning task with 30 candidate nodes in each iteration. Among 30 candidate nodes, 20 nodes have incorrectly labeled data samples with the mislabel rate generated randomly from range (0,1), and 20 nodes have non-IID data samples with data distributions generated from the 10 different non-IID levels as described in Section 6.1. In addition, the benchmarks adopt the Federated Averaging (FedAvg) algorithm [7] for model aggregation, and for fair comparison, we also present the performance of *FAIR* with the FedAvg model aggregation mechanism (named *FAIR FedAvg*). The model accuracy of each learning task with different incentive mechanisms is shown in Fig. 4. We can observe that for all learning tasks, the mechanism of *FAIR FedAvg* can outperform the other benchmarks significantly by converging more rapidly with a higher accuracy, which demonstrates the efficacy of the quality-aware incentive mechanism in *FAIR*. In addition, it can be seen that with integrating the auto-weighted model aggregation component in *FAIR*, the learning performance can be further enhanced in terms of both convergence speed and model accuracy. For instance, when evaluating on the *LeNet-5 FMNIST* task, after 30 communication rounds (iterations),

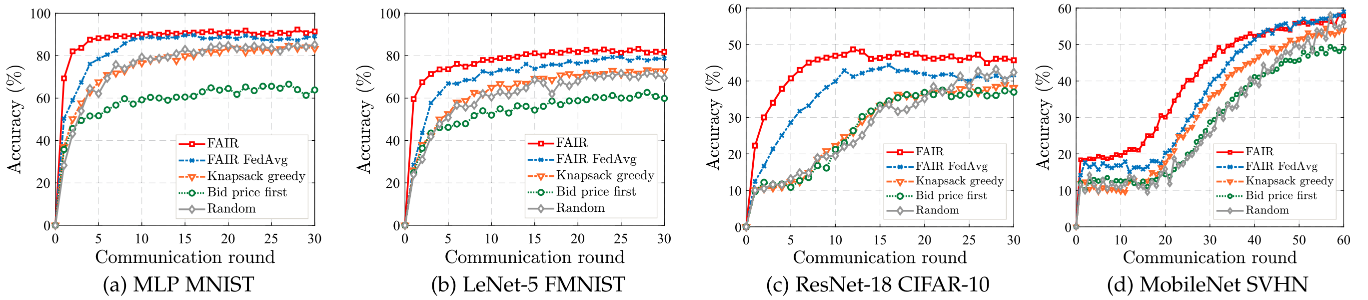


Fig. 4. Performance comparison with different incentive mechanisms.

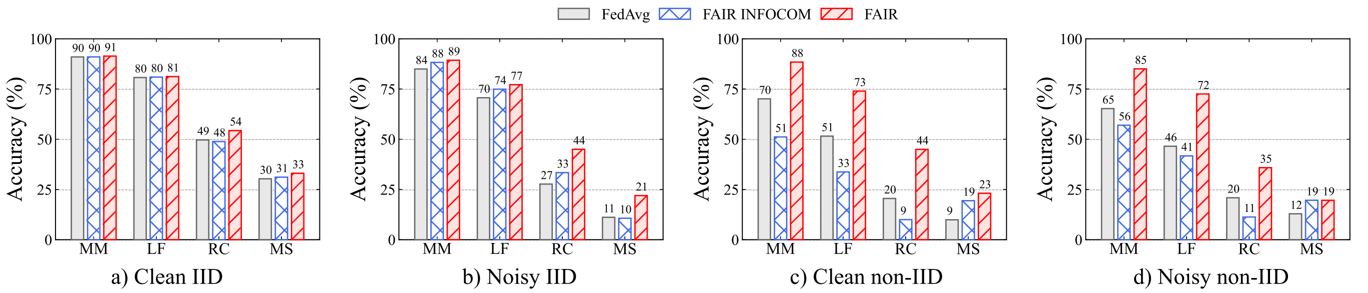


Fig. 5. The model aggregation performance of MLP MNIST (MM), LeNet-5 FMNIST (LF), ResNet-18 CIFAR-10 (RC), MobileNet SVHN (MS) under four different scenarios.

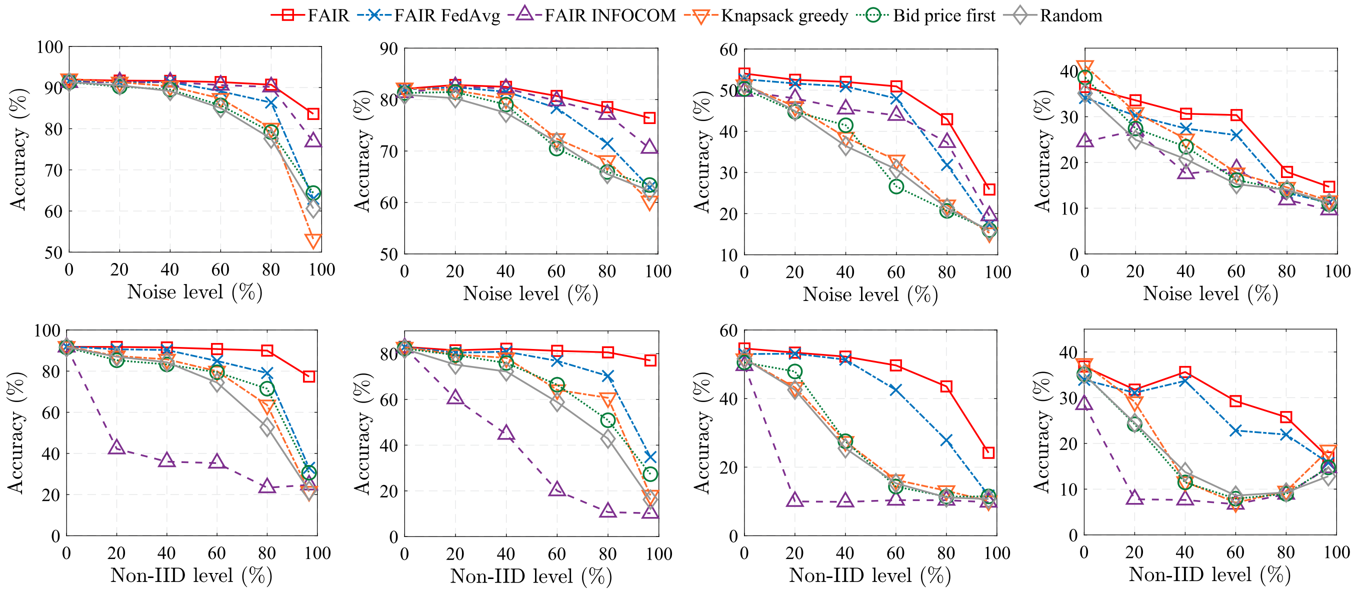
the *Knapsack greedy*, *Random*, and *Bid price first* mechanisms can only achieve accuracy scores of 72.7%, 69.5% and 59.9% respectively, while *FAIR FedAvg* can achieve a score of 78.7% and *FAIR* can further push the accuracy to 81.8%, improving the performance by 8.3%, 13.2%, 23.9%, and 12.5%, 15%, 36.6%, respectively. Besides, the *FAIR* and *FAIR FedAvg* mechanisms require only 3 and 9 iterations to achieve the model accuracy of 70%, but the *Knapsack greedy*, *Random*, and *Bid price first* mechanisms require 18, 21, and more than 30 iterations, respectively. In summary, compared to the benchmarks, *FAIR* advances dramatically in model accuracy and learning convergence speed, which is valuable to distributed learning systems in providing high-quality intelligent services at low negotiation cost.

6.3 Performance of Model Aggregation

We then examine the model aggregation performance of *FAIR*, which is evaluated with the MLP, LeNet-5, ResNet-18, and MobileNet models, trained by the MNIST, FMNIST, CIFAR-10, and SVHN dataset respectively. Each model is trained with 20 participating nodes under four different scenarios:[3] a) *Clean IID:* the local training dataset of each participating node is IID without mislabeled data samples; b) *Noisy IID:* the data distribution of the 20 participating nodes are IID, but there are 15 nodes with mislabeled data samples, and the ratio of mislabeled data samples is randomly generated from the range (0,1); c) *Clean non-IID:* for the 20 participating nodes, they do not have mislabeled data samples, but there are 15 nodes having non-IID data distribution; d) *Noisy non-IID:* for the 20 participating nodes, 15 nodes have incorrectly labeled data samples and 15 nodes have non-IID data distribution. Besides, the amount of training data samples of the 20 participating nodes is generated uniformly from the range [100,1000].

3. Note that, the incentive process is not considered in this experiment.

Fig. 5 shows the average model accuracy over 30 iterations, and we can make the following three major statements. First, *FAIR* can outperform the benchmarks in all scenarios for all learning models. For example, under the *Noisy non-IID* scenario, when evaluated on the LeNet-5 model, the *FedAvg* and *FAIR INFOCOM* mechanisms can only achieve the average accuracy score of 46% and 41% respectively, while *FAIR* can reach the score of 72%, which can improve the performance by 56.5% and 75.6%, respectively. Second, the traditional *FedAvg* algorithm is quite sensitive to the mislabeled and non-IID data, where the learning accuracy degrades dramatically when the data qualities of participating nodes are poor, but *FAIR* can work robustly under all scenarios. Taking the LeNet-5 model as an example, the model accuracy achieved by the *FedAvg* algorithm will decrease from 80% to 46% when the data condition changes from *Clean IID* to *Noisy non-IID*, while the accuracy achieved by *FAIR* just decreases from 81% to 72%. Similar observations can be also achieved for other learning models under those scenarios. Third, the *FAIR INFOCOM* mechanism can overcome the negative effect of mislabeled data samples, but fails to deal with the non-IID data condition. We can observe that, compared to the *FedAvg* mechanism, the *FAIR INFOCOM* mechanism performs robustly under the *Noisy IID* scenario, but its performance degrades significantly when there exist non-IID data distributions (e.g., under the *Clean non-IID* and *Noisy non-IID* scenarios). The phenomenon can be explained as follows. We find that when a node has non-IID data distribution, its local training loss becomes small and thus the *FAIR INFOCOM* mechanism will give it a large aggregation weight, leading to the degradation in model accuracy. However, for a node with mislabeled data samples, its local training loss is large and thus receives a small aggregation



(a) MLP MNIST (b) LeNet-5 FMNIST (c) ResNet-18 CIFAR-10 (d) MobileNet SVHN

Fig. 6. Performance comparison under different noise and non-IID levels.

weight, where the *FAIR INFOCOM* mechanism can perform well under this situation. In contrast, *FAIR* can work robustly under both scenarios with mislabeled data samples or non-IID data distribution.

6.4 Impact of Training Data Quality

After guaranteeing the overall performance, we then investigate the impact of local training data quality by comparing the performance of *FAIR* with the benchmarks under different data noise and non-IID levels. Specifically, when examining the impact of mislabeled data, the experiment setting II in Table 2 is adopted, where we set the local data of each node with IID distribution but vary the mislabeled noise level of the 30 candidate nodes. The noise level refers to the percentage of nodes within the candidate nodes that have 80% mislabeled data. Likewise, when examining the impact of non-IID data distribution, the experiment setting III in Table 2 is adopted, where we set the local data of each node with being correctly labeled but vary the non-IID level of the candidate nodes. The non-IID level refers to the percentage of candidate nodes that have only one class of labeled data.

Fig. 6 shows the average model accuracy over 30 iterations under different noise and non-IID levels. We can achieve the following three major observations. First, under all experimental settings, *FAIR* can outperform the other benchmarks significantly, and the performance gap becomes larger as the noise or non-IID level increases. For example, in the *ResNet-18 CIFAR-10* task, when the non-IID level reaches 60%, the other benchmarks achieve an accuracy of no more than 20%, but *FAIR* and *FAIR FedAvg* can guarantee the accuracy to 50% and 43%, respectively. Second, although the model accuracy decreases with the noise or non-IID level for all mechanisms, the performance of benchmarks start to decrease dramatically even at low noise or non-IID levels (e.g., the level of $\leq 60\%$), while the performance of *FAIR* remains stable within the low noise or non-IID levels. Taking the *LeNet-5 FMNIST* task as an example, when the non-IID level increases from 0% to 60%, the accuracy of *Knapsack greedy*, *Random*, and *Bid price first* mechanisms decrease from 82% to 64%, 59%, and 66% respectively, but *FAIR* and *FAIR FedAvg* only decrease from 83% to 81% and 77%, respectively. Third, for the mislabeled data condition, our proposed mechanisms of *FAIR*, *FAIR INFOCOM*, *FAIR FedAvg* can outperform the other benchmarks in most cases. Besides, for the non-IID condition, the mechanisms of *FAIR* and *FAIR FedAvg* can still maintain the performance superiority regardless of the non-IID level, but the *FAIR INFOCOM* mechanism fails to achieve it. As described in Section 6.3, mislabeled data results in a smaller local training loss reduction while non-IID data leads to a larger local training loss reduction. As a result, *FAIR INFOCOM* prefers to select correctly labeled but non-IID nodes, which results in the performance improvements in mislabeled data scenarios but performance drops in non-IID data scenarios.

6.5 Impact of Budget

Finally, we evaluate the performance of *FAIR* under multi-task scenarios and further investigate the impact of learning budget, where the experiment setting IV in Table 2 is adopted. There are 50 distributed nodes and 3 learning tasks in each iteration. Among the 50 nodes, the training datasets of 20 nodes have incorrectly labeled samples with the mislabeled ratio randomly generated from range (0,1), and 40 nodes have non-IID data with data distributions generated from the 10 different non-IID levels as described in Section 6.1. With different learning budgets, we plot the average accuracy of the MLP, LeNet-5, and ResNet-18 models in Fig. 7. We can observe that for all mechanisms, the average learning quality increases with the constrained budget. In addition, compared to the benchmarks, *FAIR*

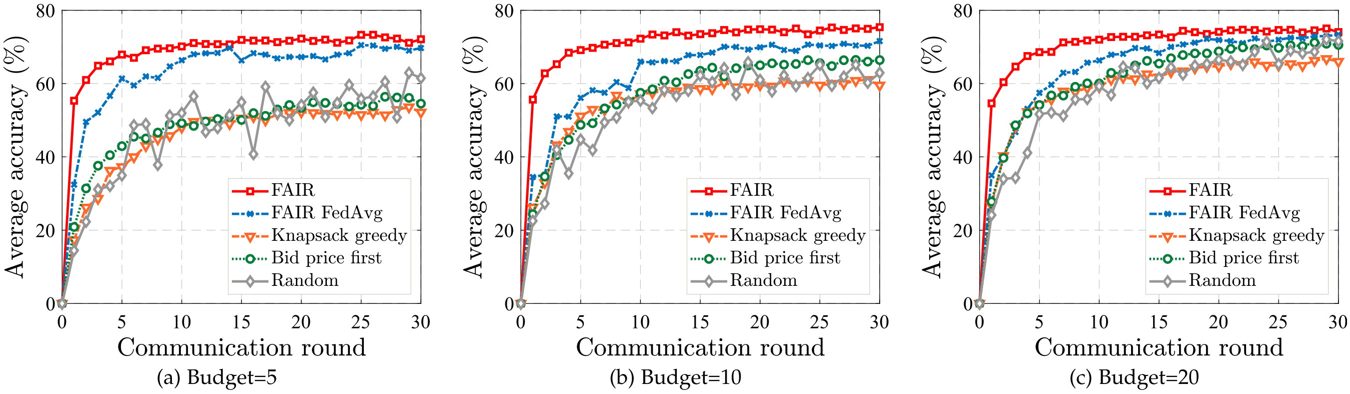


Fig. 7. The average accuracy of MLP MNIST, LeNet-5 FMNIST and ResNet-18 CIFAR-10 under different learning budgets.

advances significantly in model accuracy and convergence speed under all settings, and the performance gap becomes larger when the learning budget is set to be smaller. For example, when the learning budget is set to be 5, after 30 iterations, the *Knapsack greedy*, *Random*, *Bid price first* mechanisms can only achieve the accuracy score of 52%, 61%, and 54%, while *FAIR* and *FAIR FedAvg* can enhance the accuracy score to 72% and 70%, respectively. Besides, the *FAIR* mechanism requires fewer iterations to reach a target model accuracy, which means that it can output satisfied learning models with less budget and low negotiation cost among distributed nodes, benefiting its practical usages.

7 Related Work

Recently, there has been extensive research attention dedicated to the performance optimization of federated learning from the aspects of communication [35], [36], [37], privacy [29], robustness [38], personalization [39], and incentive [40], [41], etc. For example, Konečný *et al.* proposed to compress model updates [35], and Wang *et al.* proposed Communication-Mitigated Federated Learning to avoid uploading irrelevant updates [36], to reduce the communication overhead of federated learning. Wei *et al.* applied differential privacy to add noise to the model updates to prevent information leakage of participating nodes [29]. So *et al.* proposed a Byzantine-resilient secure aggregation framework to guarantee the robustness of federated learning against Byzantine faults [38]. In [39], Wu *et al.* proposed a personalized federated learning framework for in-home health monitoring. In this section, we mainly focus on surveying the incentive mechanism and quality quantification approaches of federated learning, since they are more relevant to our research.

7.1 Incentive Mechanism

There have been some incentive mechanisms proposed in federated learning. Pandey *et al.* devised an incentive mechanism based on the Stackelberg game to improve the global model with communication efficiency [12]. 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 [10]. In [11], Le *et al.* considered the resources and energy cost of mobile nodes and proposed an auction-based incentive mechanism to maximize the social welfare of the wireless federated learning services market. However, none of them considers the learning quality of participants, which is crucial for efficient federated learning while the quality difference among participating nodes is common in practical distributed learning systems. On the other hand, Kang *et al.* proposed an incentive mechanism combining reputation with contact theory to encourage high-reputation nodes to participate in learning [40]. Zeng *et al.* proposed an incentive mechanism FMore with multi-dimensional procurement auction to motivate high-quality nodes with low cost to participate in learning, where the data size and data category are considered [42]. Jiao *et al.* designed auction-based incentive mechanisms to maximize the social welfare of the federated learning services market, where the data size and non-IID data issue of participating nodes are considered [41]. Likewise, none of them considers the mislabeled data issue of participant's local training data, as well as the budget of federated learning tasks. Besides, they require participating nodes to report their private information, e.g., data distribution, which violates the original intent of federated learning for privacy preserving. Different from them, we consider the data size, mislabeled samples, and non-IID distribution issues simultaneously, devise a privacy-preserving approach to quantify the learning quality of nodes, and propose a quality-aware incentive mechanism to motivate high-quality nodes with low cost to participate in federated learning within the limited budget. Furthermore, rather than an incentive mechanism design alone, we also integrate an auto-weighted model aggregation mechanism to jointly build high-quality federated learning models.

7.2 Learning Quality Quantification

For fair profit distribution in the incentive mechanism, there have been some metrics proposed to quantify the contribution of each federated learning participant. Wang *et al.* proposed a deletion method, which estimates the contribution of a participant by omitting it and measuring the accuracy degradation of the retrained model [43]. However, this method requires repeating the entire federated learning process for each contribution measurement, which is computation-and communication-resource consuming. To address this challenge, Song *et al.* proposed a metric named contribution index based on Shapley value, which reconstructs the models on different combinations of datasets through model updates so as to avoid extra training [44]. Nonetheless, it still takes exponential time for contribution measurement, which is usually inapplicable in large-scale federated learning systems. Considering the contribution difference among participants, existing studies usually adopt the distance between the global model and the node local models to determine the model aggregation weights [45]. They assign a higher weight to the node whose model updates carry more different information, which, however, is not applicable when there exist low-quality nodes, since the distance between the low-quality model updates and the global model can be larger, and thus the low-quality nodes will be assigned with larger aggregation weights, resulting in model accuracy degradation. Overall, considering the efficiency and robustness, the above methods are not suitable for large-scale and reliable quality quantification. Instead, in this paper, we propose a learning quality quantification method in a privacy-preserving manner, which is applicable and effective to be integrated into *FAIR* to assist the quality-aware incentive mechanism for node selection and profit distribution.

In our previous work [34], we have demonstrated the efficiency of quality-aware federated learning. In this work, we further extend it by upgrading the components of quality estimation and model aggregation, where an online quality quantification method and auto-weighted model aggregation algorithm are newly devised, respectively. We have conducted extensive new experiments and demonstrated its superior performance in various scenarios compared to our previous work.

8 Conclusion and Future Works

In this paper, we have proposed *FAIR*, a novel quality-aware federated learning framework, which can significantly enhance the federated learning quality with quality-aware user incentive and auto-weighted model aggregation. Particularly, we have designed and implemented three major technical components in *FAIR:* 1) learning quality estimation, 2) quality-aware incentive mechanism, and 3) auto-weighted model aggregation. In addition, we have theoretically proved *FAIR* to be truthful, individually rational, and computationally efficient. Extensive experiments under various federated learning scenarios have been carried out, and the results have demonstrated the efficacy of *FAIR* in terms of user incentive and model aggregation, which can perform reliably under different levels of data quality and constrained learning budgets.

In our future work, we will further enhance the incentive performance of *FAIR* and conduct more practical evaluations to demonstrate its using benefits. First, *FAIR* selects nodes for learning tasks based on the historical learning quality records of them. In the future, we will try to establish a reputation assessment for each node based on its behaving performance in various learning tasks, such that more valuable information can be used for efficient node selection. Second, we will extend *FAIR* to a wider range of applications, where we will use large-scale and real-world datasets, e.g., OpenImage [46], Reddit [47], Taobao [48], to demonstrate the practical benefits of *FAIR*.

References

[1] H. Cai, F. Ye, Y. Yang, Y. Zhu, J. Li, and F. Xiao, “Online pricing and trading of private data in correlated queries,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 569–585, Mar. 2021.

[2] N. Cheng *et al.*, “Big data driven vehicular networks,” *IEEE Netw.*, vol. 32, no. 6, pp. 160–167, Nov./Dec. 2018.

[3] K. R. Sollins, “IoT Big Data security and privacy versus innovation,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1628–1635, Apr. 2019.

[4] 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 Comput. Surv.*, vol. 52, no. 6, pp. 1–6, 2019.

[5] T. Liu, Y. Zhang, Y. Zhu, W. Tong, and Y. Yang, “Online computation offloading and resource scheduling in mobile-edge computing,” *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6649–6664, Apr. 2021.

[6] F. Lyu *et al.*, “LeaD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 8, pp. 2607–2623, Aug. 2021.

[7] H. B. McMahan *et al.*, “Communication-efficient learning of deep networks from decentralized data,” 2016, *arXiv:1602.05629*.

[8] C. Wang, Y. Yang, and P. Zhou, “Towards efficient scheduling of federated mobile devices under computational and statistical heterogeneity,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 2, pp. 394–410, Feb. 2021.

[9] Y. Deng *et al.*, “SHARE: Shaping data distribution at edge for communication-efficient hierarchical federated learning,” in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 24–34.

[10] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, “A learning-based incentive mechanism for federated learning,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[11] T. H. Thi Le *et al.*, “An incentive mechanism for federated learning in wireless cellular networks: An auction approach,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4874–4887, Aug. 2021.

[12] 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 Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241–3256, May 2020.

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

[14] H. Wang, Z. Kaplan, D. Niu, and B. Li, “Optimizing federated learning on non-IID data with reinforcement learning,” in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1698–1707.

[15] Y. Liu and M. Liu, “An online learning approach to improving the quality of crowd-sourcing,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2166–2179, Aug. 2017.

[16] M. S. Sadik and L. Gruenwald, “DBOD-DS: Distance based outlier detection for data streams,” in *Proc. Int. Conf. Database Expert Syst. Appl.*, 2010, pp. 122–136.

[17] T. J. Brailsford, J. H. Penm, and R. D. Terrell, “Selecting the forgetting factor in subset autoregressive modelling,” *J. Time Ser. Anal.*, vol. 23, no. 6, pp. 629–649, 2002.

[18] M. Dawande, J. Kalagnanam, P. Keskinocak, F. S. Salman, and R. Ravi, “Approximation algorithms for the multiple knapsack problem with assignment restrictions,” *J. Combinatorial Optim.*, vol. 4, no. 2, pp. 171–186, 2000.

[19] M. S. Hung and J. C. Fisk, “An algorithm for 0-1 multiple-knapsack problems,” *Nav. Res. Logistics Quart.*, vol. 25, no. 3, pp. 571–579, 1978.

[20] R. B. Myerson, “Optimal auction design,” *Math. Operations Res.*, vol. 6, no. 1, pp. 58–73, 1981.

[21] T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung, “Overcoming noisy and irrelevant data in federated learning,” in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 5020–5027.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE Proc. IRE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[25] Y. LeCun, “The MNIST database of handwritten digits,” 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[26] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[27] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Tech. Rep. TR-2009, 2009.

[28] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2011, pp. 1–9.

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

[30] C. T. Dinh *et al.*, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2020.

[31] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[32] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-IID data," 2019, *arXiv:1907.02189*.

[33] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[34] Y. Deng *et al.*, "FAIR: Quality-aware federated learning with precise user incentive and model aggregation," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

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

[36] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 954–964.

[37] Y. Wu, Y. Song, T. Wang, L. Qian, and T. Q. S. Quek, "Non-orthogonal multiple access assisted federated learning via wireless power transfer: A cost-efficient approach," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2853–2869, Apr. 2022.

[38] J. So, B. Güler, and A. S. Avestimehr, "Byzantine-resilient secure federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2168–2181, Jul. 2020.

[39] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2818–2832, Aug. 2020.

[40] 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 Things J.*, vol. 6, no. 6, pp. 10 700–10 714, Dec. 2019.

[41] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.

[42] R. Zeng, S. Zhang, J. Wang, and X. Chu, "FMore: An incentive scheme of multi-dimensional auction for federated learning in MEC," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 278–288.

[43] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2597–2604.

[44] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2577–2586.

[45] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang, "Learning private neural language modeling with attentive aggregation," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

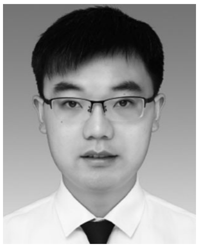
[46] I. Krasin *et al.*, OpenImages: A public dataset for large-scale multi-label and multi-class image classification, 2017. [Online]. Available: https://storage.googleapis.com/openimages/web/index.html

[47] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1025–1035.

[48] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for E-commerce recommendation in alibaba," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 839–848.

Yongheng Deng received the BS degree from Nankai University, Tianjin, China, in 2019. He is currently working toward the PhD degree with the department of computer science and technology, Tsinghua University, Beijing, China. Her research interests include federated learning, edge intelligence, distributed system and mobile/edge computing.

Feng Lyu (Senior Member, IEEE) received the BS degree in software engineering from Central South University, Changsha, China, in 2013, and the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2018. During respective Sept. 2018-Dec. 2019 and Oct. 2016-Oct. 2017, he worked as a postdoctoral fellow and was a visiting PhD student in BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently a professor with the School of Computer Science and Engineering, Central South University, Changsha, China. His research interests include vehicular networks, beyond 5G networks, Big Data measurement and application design, and edge computing. He is the recipient of the best paper award of IEEE ICC 2019. He currently serves as associate editor for *IEEE Systems Journal* and leading guest editor for *Peer-to-Peer Networking and Applications*, and served as TPC members for many international conferences. He is a member of the IEEE Computer Society, Communication Society, and Vehicular Technology Society.



Ju Ren (Senior Member, IEEE) received the BSc, MSc, and PhD degrees all in computer science, from Central South University, China, in 2009, 2012, and 2016, respectively. Currently, he is an associate professor with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include Internet-of-Things, edge computing, operating system, as well as security and privacy. He currently serves as an associate editor for *IEEE Transactions on Vehicular Technology* and *Peer-to-Peer Networking and Applications*. He also served as the general co-chair for IEEE BigDataSE'20, the TPC co-chair for IEEE BigDataSE'19, a poster co-chair for IEEE MASS'18, a track co-chair for IEEE/CIC ICCC'19, IEEE I-SPAN'18 and VTC'17 Fall, and an active reviewer for more than 20 international journals. He received many best paper awards from IEEE flagship conferences, including IEEE ICC'19 and IEEE HPCC'19, etc., the IEEE TCSC Early Career Researcher Award (2019), and the IEEE ComSoc Asia-Pacific Best Young Researcher Award (2021). He is recognized as a highly cited researcher by Clarivate (2020 & 2021).

Yi-Chao Chen (Member, IEEE) received the PhD degree in computer science with the University of Texas at Austin, in 2015. He joins Shanghai Jiao Tong University as a tenure-track associate professor in the Department of Computer Science and Engineering, in 2018. Prior to joining SJTU, he spent a year as a researcher in Huawei Future Network Theory Lab in Hong Kong and then the chief scientist in Hauoli LLC. His research interests focus on networked systems and span the areas of mobile computing, wireless networking, and cyber-security.

Peng Yang (Member, IEEE) received the BE degree in communication engineering, and the PhD degree in information and communication engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013 and 2018, respectively. He was with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, as a visiting PhD student from Sept. 2015 to Sept. 2017, and a postdoctoral fellow from Sept. 2018 to Dec. 2019. Since Jan. 2020, he has been a faculty member with the School of Electronic Information and Communications, HUST. His current research focuses on mobile edge computing, video streaming and analytics.

Yuezhi Zhou (Senior Member, IEEE) received the PhD degree in computer science and technology from Tsinghua University, Beijing, China, in 2004. He worked as a visiting scientist with the Computer Science Department at Carnegie Mellon University, in 2005. He is now working as a full professor with the Department of Computer Science and Technology, Tsinghua University. He has authored or coauthored more than 100 technical papers in international journals or conferences. His research interests include pervasive computing, distributed systems, and mobile networks. He received the Best Paper Award at IEEE International Conference on Advanced Information Networking and Applications (AINA) 2007 and IEEE International Conference on High Performance Computing and Communications (HPCC) 2014.

Yaoxue Zhang (Senior Member, IEEE) received the BS degree from the Northwest Institute of Telecommunication Engineering, China, in 1982, and the PhD degree in computer networking from Tohoku University, Japan, in 1989. Currently, he is a professor with the School of Computer Science and Engineering, Central South University, China, and a professor with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include computer networking, operating systems, ubiquitous/pervasive computing, transparent computing, and Big Data. He has published more than 200 technical papers in international journals and conferences, as well as 9 monographs and textbooks. He is a fellow of the Chinese Academy of Engineering, China, and the editor-in-chief of *Chinese Journal of Electronics*.