A Two-Stage Incentive Mechanism Design for Quality Optimization of Hierarchical Federated Learning

ZHOU LI\textsuperscript{1,2}, (Member, IEEE), HUI DU\textsuperscript{2}, AND XIN CHEN\textsuperscript{2}

\textsuperscript{1}Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing Information Science and Technology University, Beijing 100101, China
\textsuperscript{2}School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China

Corresponding author: Zhuo Li (lizhuo@bistu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFF0604502, in part by the National Natural Science Foundation of China under Grant 61872044, and in part by the Beijing Municipal Program for Top Talent.

\section*{ABSTRACT} In this paper, we investigate the aggregated model quality maximization problem in hierarchical federated learning, the decision problem of which is proved NP-complete. We develop the mechanism MaxQ to maximize the sum of local model quality, which consists of two stages. In the first stage, an algorithm based on matching game theory is proposed to associate mobile devices with edge servers, which is proved able to achieve the stability and $\frac{1}{2}$-approximation ratio. In the second stage, we design an incentive mechanism based on contract theory to maximize the quality of models submitted by mobile devices to edge servers. Through thorough experiments, we analyse the performance of MaxQ and compare it with the existing mechanisms FAIR and EHFL, under different deep learning models ResNet18, ResNet50 and AlexNet, individually. It is found that the model quality can be improved by 8.20% and 7.81%, 10.47% and 11.87%, 10.98% and 11.97% under different models, respectively.

\section*{INDEX TERMS} Hierarchical federated learning, maximization of model quality, matching game, contract theory, incentive mechanism design.

\section*{I. INTRODUCTION} According to the forecast of Cisco [1], by 2023 the total number of Internet users will reach 5.3 billion, and the number of network devices will reach 29.3 billion. Ericsson predicts that the average monthly total mobile network traffic will exceed 300 EB by 2026 [2]. In traditional machine learning, a large amount of user data will be uploaded to the cloud server for centralized machine learning. However, this may leakage users’ privacy information [3], and bring huge delay and energy consumption [4]. To address these challenges, Google proposed federated learning (FL) in 2016 [5]. In FL, model training is performed on mobile devices, and only parameters update of the local trained models needs to be transmitted to the cloud server for global aggregation, which makes the size of the uploaded data significantly smaller and thus reduces energy consumption and latency. In addition, since users do not have to upload original data, their privacy can be protected.

In FL, the cloud server is mainly used as parameter server [6]. As the number of mobile devices involved in the model training process increases [7], FL faces the challenge of overloaded transmission in the infrastructure networks and computation in the cloud. With the development of edge computing, computing devices on the edge of networks can perform as parameter servers [8], [9], and hierarchical federated learning (HFL) is proposed [10], the architecture of which is cloud-edge-client based and different with the traditional cloud-client one. In HFL, edge servers can aggregate parameter updates from the nodes in their region, uploading the results to the cloud server. This new kind of federated learning technology is used in many applications, such as anomaly detection [11], unmanned aerial vehicular networks [12], and power transformer fault diagnosis [13], etc. The framework of HFL is shown as Figure 1.
Model quality may be the most important performance metric in different HFL applications [14], such as autonomous driving [15], [16], health monitoring [17]. Due to the size and quality of data set on mobile devices is usually different, their obtained local model quality always differs with each other. Due to the participation of mobile devices in model training will consume their own resources [18], it is necessary to design effective incentive mechanisms to encourage them to contribute high qualified model updates in the training process. However, there are two key challenges in the incentive mechanism design for HFL. First, the associated mobile devices with a particular edge server may change dynamically due to their mobility, i.e., they can enter or leave the region of the edge server at any possible time. Second, it is difficult for edge servers to evaluate the real contribution of mobile devices until the models training process is finished and the quality of their submitted model update is measured. On the other side, it is also hard for the mobile devices to know exactly the real reward they can get from the edge server before they submit their local updates.

To address these two challenges, we design mechanism MaxQ for the maximization of HFL model quality, which consists of two stages. In the first stage, we determine the association relationship between the devices and edge servers based on many-to-one matching game theory [19]. In the second stage, we address the issue of information asymmetry between the two parties based on contract theory, in which edge servers can offer different rewards to devices according to their different local model quality. After that, in the process of model aggregation, larger weights are given to model update with higher quality, reducing the influence of updates with lower quality. In this way, MaxQ can select mobile devices with high estimated quality participating in model training, encourage them to contribute high-quality model updates, and thus improve the HFL performance.

The main contributions of this paper can be summarized as follows.

- We define the global model quality maximization problem MQM for HFL, and we prove that its decision problem is NP-complete.
- We develop mechanism MaxQ to maximize the global model quality, which consists of two stages. In the first stage, we design an algorithm based on matching game theory to associate mobile devices with edge servers, which can achieve the stability and $1/2$-approximation ratio. In the second stage, we design an incentive mechanism based on contract theory, the optimality of which is also proved.
- We take thorough simulation experiments and find that MaxQ can effectively accelerate the model convergence speed, while improving the model quality by 8.20% and 7.81%, 10.47% and 11.87%, 10.98% and 11.97% on ResNet18, ResNet50 and AlexNet models respectively, compared with the existing methods FAIR and EHFL.

The remainder of this paper is organized as follows. We define the system model and quality maximization problem of HFL in Section 2. The mechanism MaxQ is proposed in Section 3. We investigate its performance through simulations in Section 4. Section 5 briefly introduces the related work, before concluding in Section 6.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we introduce the system model of hierarchical federated learning and the problem of model quality maximization. The main notations used in this paper are listed in Table 1.
TABLE 1. Key Symbols.

| Symbol | Definitions |
|--------|-------------|
| $\theta_{ij}^t$ | Data quality of the mobile device $n_j$ in iteration $t$ |
| $\theta_{ij}$ | Data quality of the mobile device $n_j$ selected by the edge server $m_i$ in iteration $t$ |
| $d_j$ | Size of data samples set of the mobile device $n_j$ |
| $d_{ij}$ | Size of data sample set of the mobile device $n_j$ selected by the edge server $m_i$ |
| $d_{ij}^t$ | Size of data samples set of the mobile device $n_j$ selected by the edge server $m_i$ in iteration $t$ |
| $q_j$ | Model quality of the mobile device $n_j$ in iteration $t$ |
| $q_{ij}$ | Model quality of the mobile device $n_j$ selected by the edge server $m_i$ in iteration $t$ |
| $B_i^t$ | Budget of the edge server $m_i$ in iteration $t$ |
| $r_{ij}$ | Reward paid by edge server $m_i$ to mobile device $n_j$ in iteration $t$ |
| $X_j^t(M)$ | Priority list of edge servers for the mobile device $n_j$ |
| $M_j^t(N)$ | Priority list of mobile devices for the edge server $m_i$ |
| $K_i(N)$ | Priority list of mobile devices |
| $R_j(M)$ | Priority list of edge servers |

A. HIERARCHICAL FEDERATED LEARNING PROCESS

For hierarchical federated learning, we assume that there is one cloud server, multiple edge servers and many mobile devices. We denote the set of edge servers as $M = \{m_1, m_2, \ldots, m_M\}$ and the set of mobile devices as $N = \{n_1, n_2, \ldots, n_N\}$. Model training is taken in discrete time slots $T = \{t_1, t_2, \ldots, t_T\}$ and the model can be updated once in each time slot.

The cloud server first distributes a machine learning model to each edge server. We assume that in iteration $t$ the budget of edge server $m_i$ is $B_i^t (i = 1, 2, \ldots, M)$ for the recruitment of mobile devices. According to the budget and the estimated capacity of different mobile devices, $m_i$ can determine the devices $X_j^t$ participating in the model training process in its region, with reward $r_{ij}^t$ paid to the selected device $n_j$ in iteration $t$. Once selected, $n_j$ uses its own data set $D_j$ with size $d_j = |D_j|$ to train the model and uploads the update to $m_i$, the quality of which is denoted as $q_{ij}^t$. Due to the computing capacity of mobile devices is limited, it is assumed that each device can participate in the model training process for only one edge server in each iteration. After receiving all local model updates from the selected devices, the edge server aggregates them and obtains a new edge model, which is then published to the selected mobile devices again. This process continues until the quality of the global model satisfies a certain requirement.

B. DEFINITION OF MODEL QUALITY

The size of training data samples and their quality can significantly affect the model quality in federated learning [20]. An intuitive way to measure the model quality is to test its accuracy on real data set, which will cause great overhead because the test has to be taken in each iteration. In this paper, we propose to use a function of the reduction of loss function after each iteration to measure the quality.

We assume that mobile device $n_j$ starts the local model training with the loss function $loss_j(\tau) - loss_j(\tau - \Delta)$, and updates the local model during $\Delta$ time slots, resulting in a loss function $loss_j(\tau)$ in the time slot $\tau$. We define

$$\theta_j^t = loss_j(\tau) - loss_j(\tau - \Delta).$$

According to the size of data samples and corresponding improvement in loss function, the model quality of mobile device $n_j$ in iteration $t$ can be defined as

$$q_{ij}^t = 1 - e^{-\varphi(\theta_{ij}^t)^v},$$

where $\varphi$ and $\nu$ are weight coefficients [21].

C. PROBLEM FORMULATION

Since the quality of aggregated global model is influenced by the local models, the optimization problem of maximizing the aggregated model quality can be transformed into maximizing the sum of local models quality, which is formulated as the following MQM (Model Quality Maximization) problem.

$$\max_{X'^t, R'^t} \sum_{m_i \in M} \sum_{n_j \in N_j^t} x_{ij}^t q_{ij}^t$$

s.t. \hspace{1cm} \sum_{n_j \in N_j^t} r_{ij}^t x_{ij}^t \leq B_i^t, \quad \forall m_i \in M, \hspace{1cm} (4)

$$\sum_{m_i \in M} x_{ij}^t \leq 1, \quad \forall n_j \in N_j^t,$$

$$x_{ij}^t \in \{0, 1\}, \quad \forall m_i \in M, \quad \forall n_j \in N_j^t, \hspace{1cm} (6)$$

where $X'^t = \{x_{ij}^t | m_i \in M, n_j \in N_j^t, t \in T\}$ is the matching matrix denoting the association relationship between edge servers and mobile devices, and $R'^t = \{r_{ij}^t | m_i \in M, n_j \in N_j^t, t \in T\}$ is the reward matrix per edge server. Constraint (5) indicates that the reward paid by $m_i$ in iteration $t$ cannot exceed its maximum budget $B_i^t$, and constraint (6) indicates that each mobile device can only participates in the model training of at most one edge server in each iteration.

For the hardness of MQM, we have the following result shown as Theorem 1, i.e., its corresponding decision problem is NP-complete.

**Theorem 1:** The decision problem of MQM is NP-complete.

**Proof:** The decision problem of MQM can be defined as follows. In iteration $t$, if the model quality of each mobile device can reach the threshold $q^t$, under the constraints that the sum of rewards of each edge server paid to mobile devices cannot exceed its maximum budget and each mobile device can participate in the training process of at most one edge server model.

The decision problem of the multi-dimensional knapsack problem (MKAR) with allocation restrictions is known as an NP-complete problem [22]. An instance of the decision problem for the MKAR is as follows. The item set is defined as $\mathcal{O} = \{o_1, o_2, \ldots, o_d\}$, and the value of each item $o_j \in \mathcal{O}$ is
Finally, we set the reward and model quality of each mobile device is same of set instance of the decision problem of MQM. For each item most one backpack. The total weight of items allocated to of allocated items, and each item can only be allocated to at most one backpack. The total weight of items allocated to each backpack cannot exceed its capacity. For each instance of the decision problem of MKAR, we can reduce it to an instance of the decision problem of MQM. For each item set and knapsack set, there is a corresponding mobile device set and edge server set, respectively. We assume the reward and model quality of each mobile device is same of different edge servers, i.e., \( r_i = r_j \), \( q_i = q_j \), \( \forall m_i \in M \). Finally, we set \( v_j = q_j \), \( w_j = r_j \), \( \forall n_j \in N \) and \( c_i = B_j \), \( \forall m_i \in M \). In this way, each instance of the decision problem of MKAR can be reduced to an instance of the decision problem of MQM in polynomial time. Therefore, the decision problem of MQM is NP-complete.

III. DESIGN OF MECHANISM MaxQ FOR THE MODEL QUALITY MAXIMIZATION

In this section, mechanism MaxQ for the maximization of model quality is proposed, which consists of two stages. We realized the association between mobile devices and edge servers based on matching games to make the HFL process feasible in the first stage. Then, based on contract theory, we maximize the quality of the local model of mobile device training associated with each edge server in the second stage, that is, the sum of the quality of the local models of all devices. Finally, in order to improve the quality of the global model, we use the model quality as the weight of the model aggregation.

A. MATCHING GAME FOR ASSOCIATING MOBILE DEVICES WITH EDGE SERVERS

We use matching game theory to associate proper mobile devices with the edge server. Because each mobile device can participate in the model training of only one edge server, but for the server there may be more than one devices associated with, we construct an one to many matching game between the server and devices.

First, we estimate the model quality of different mobile devices. Assuming mobile device \( n_j \) participates in local model training in iteration \( t_0, t_1, \ldots, t_k \), we estimate its model quality \( q_j^{t_k+1} \) in iteration \( t_{k+1} \) using historical model quality \((q_j^{t_0}, q_j^{t_1}, \ldots, q_j^{t_k})\) where \( t_{k+1} > t_k \). Due to the model quality may change with time and the quality measured in a more recent time slot may reflect the change trend more accurately, an exponential forgetting function is used to allocate different weights to quality values obtained in different slots, providing greater weights for the more recent model quality records and smaller weights for the earlier ones. The weight of the latest model quality is set as 1, and the weights of other model quality are calculated according to its relative time distance to the current slot. Weights allocated to historical model quality \((q_j^{t_0}, q_j^{t_1}, \ldots, q_j^{t_k})\) are denoted as \((\rho^{t_0}, \rho^{t_1}, \ldots, \rho^{t_{k-1}}, 1)\), where \( 0 \leq \rho \leq 1 \) is the forgetting factor. For iteration \( t_{k+1} \), the estimated model quality of \( n_j \) is calculated as

\[
q_j^{t_{k+1}} = \sum_{t=0}^{k} \rho^{t_{k+1}-t} q_j^t.
\]

We define the problem of maximizing the estimated model quality as

\[
\max \sum \sum x_{ij} \hat{q}_{ij} \quad \text{s.t. (4) - (6)}.
\]

We use matching game theory to solve this problem. In iteration \( t \), we define the matching matrix between edge servers and mobile devices as

\[
X = \begin{pmatrix}
x_{11} & x_{12} & \ldots & x_{1N} \\
x_{21} & x_{22} & \ldots & x_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
x_{M1} & x_{M2} & \ldots & x_{MN}
\end{pmatrix},
\]

where \( x_{ij} = 1 \) indicates that mobile device \( n_j \) is allocated to edge server \( m_i \), and the two parties form a matching. When the server and the device can not improve their revenue through establishing a new matching, the matching is stable.

\( m_i \) requires \( N_i \) mobile devices and the possible reward values it can pay are \((r_{i1}, r_{i2}, \ldots, r_{iN_i})\), where \( r_{ik} \geq r_{ij} \) if \( k \leq j \). \( R_i^t \) is the random variable denoting the reward of \( m_i \) for the device \( n_j \) and \( P(R_i^t = r_{ik}) = p_{ij}^t \), where \( \sum_{k=1}^{N_i} p_{ij}^t = 1 \). The real reward value is unknown before the model training process is accomplished. The profit function of the device \( n_j \) associated with \( m_i \) is defined as

\[
u_i(m_i) = \mathbb{E}(R_i^t) - cd_i,
\]

where \( \mathbb{E}(R_i^t) \) is the mathematical expectation of \( R_i^t \), and \( c \) is the cost for processing every piece of data sample in the model training process. According to \( u_i(m_i) \), we can establish a priority list of edge servers for the device \( n_j \) as,

\[
N_j^t(M) = \{u_i^t(m_{j1}), u_i^t(m_{j2}), \ldots, u_i^t(m_{jM})\},
\]

where \( u_i^t(m_{j1}) \geq u_i^t(m_{j2}) \geq \cdots \geq u_i^t(m_{jM}) \).

The profit function of the edge server \( m_i \) selecting \( n_j \) is defined as

\[
v_i^t(n_j) = \hat{q}_{ij}.
\]

According to \( v_i^t(n_j) \), we can also establish a priority list of mobile devices for \( m_i \) as,

\[
N_i^t(M) = \{v_i^t(n_{i1}), v_i^t(n_{i2}), \ldots, v_i^t(n_{iN_i})\},
\]

where \( v_i^t(n_{i1}) \geq v_i^t(n_{i2}) \geq \cdots \geq v_i^t(n_{iN_i}) \).

We design the matching algorithm between edge servers and mobile devices as shown in Algorithm 1. Firstly, we...
calculate the profit function of mobile devices and edge servers, and then get the sorted list of available mobile devices \( M_i'(N) \) and available edge servers \( N_j'(M) \). We can get the priority list of mobile devices \( K_i(N) = [n_{i_1}, n_{i_2}, \ldots, n_{i_l}] \) for the edge server \( m_i \), and the priority list of edge servers \( R_j(M) = N_j'(M) \) for the mobile device \( n_j \). The matching is successful when \( n_j \) is in the priority list of mobile devices \( K_j(N) \) for \( m_j \). Then the priority lists of mobile devices for all edge servers are updated, i.e., device \( n_j \) is removed. When \( K_i(N) \) becomes empty, \( m_i \) is removed from the priority lists of edge servers for all mobile devices. If there is no change in \( K_i(N) \) before and after two matches, the best mobile device \( n_{i_1} \) in \( K_i(N) \) is removed. When \( n_j \) is not in the priority list of mobile devices \( K_j(N) \) for the edge server \( m_j \), the edge server \( m_j \) is removed from \( R_j(M) \) and then \( n_j \) is associated with \( m_j \). The procedure above continues until the matching game is successful or \( R_j(M) \) is empty.

For the performance of Algorithm 1, we have the following results.

Theorem 2: The time complexity of Algorithm 1 is \( O(MN^2) \).

Proof: In lines 1 to 2 and in the loop in lines 3 to 4, there is only one level of loops, which loop \( M \) and \( N \) times, respectively. In the worst case, each match is unsuccessful, then the while loop at lines 5 to 19 executes \( 2N \) loops. In the while loop, lines 5 and 10 are double loops that execute \( MN \) times in the worst case. In lines 11 to 19 is also a double loop, which executes \( MN \) times in the worst case. Since the number of edge servers is smaller than the number of mobile devices, i.e., \( M < N \). Therefore, in the worst case, Algorithm 1 is executed \( 2MN^2 \) times. All above, The time complexity of Algorithm 1 is \( O(MN^2) \).

Theorem 3: The matching game in Algorithm 1 is able to achieve stability.

Proof: For the mobile device \( n_j \) that has not form a matching relationship after \( l \) matches, we assume mobile devices that have already finished matching with the edge server \( m_{i_l}, m_{i_{l-1}}, \ldots, m_{i_1} \), where \( \nu_{i_l}(n_{i_l}) > \nu_{i_{l-1}}(n_{i_{l-1}}) > \cdots > \nu_{i_1}(n_{i_1}) \). According to Algorithm 1, there must be \( \nu_{i_l}(n_{i_l}) > \nu_{i_{l-1}}(n_{i_{l-1}}) \) if the edge server \( m_{i_1} \) is not willing to change the existing matching relationship to establish a new matching relationship with the mobile device \( n_j \).

In the \( l \)-th round of matching, if \( n_j \) that establishes matching relationship with \( m_{i_l} \), \( m_{i_l} \) is the optimal choice of \( n_j \) in that round. As a result, \( n_j \) is not willing to change the existing matching relationship.

In summary, for edge servers and mobile devices that complete the matching process, both parties will not change the existing matching relationship, so the matching can achieve stability.

Theorem 4: The approximation ratio of Algorithm 1 is \( \frac{1}{2} \).

Proof: In the iteration \( i \), we assume that \( U_i' \) is the set of mobile devices that have not been allocated, and \( M_i'(U_i') \) is the set of edge servers that can afford the reward requirement of at least one of the mobile devices in \( U_i' \). \( Q'(U_i') \) is the sum of the model quality of all mobile devices in \( U_i' \), and \( Q' \) is the sum of the model quality of all allocated mobile devices. In each edge server, since the allocated mobile devices satisfy the half-full property, we have \( Q'(U_i') \leq Q' \).

For any feasible solution, we assume that \( U_i' \subset U_i' \) is set of assigned mobile devices and \( Q'(U_i') \) is sum of the model quality of these mobile devices. Among all solutions including the optimal solution, since mobile devices in \( U_i' \) can only be allocated to the edge server in \( M_i'(U_i') \), there is \( Q'(U_i') \leq Q'(U_i') \). Therefore, the maximum model quality that can be obtained by mobile devices in the allocation is \( Q'(U_i') + Q' \) among all solutions. So the optimal solution \( Q_{opt} \leq Q'(U_i') + Q' \).

Since \( Q_{opt} \leq Q'(U_i') + Q' \leq 2Q' \), i.e. \( Q' \geq \frac{1}{2}Q_{opt} \), so Algorithm 1 has a \( \frac{1}{2} \)-approximation ratio.

Algorithm 1 Matching Algorithm Between Edge Servers and Mobile Devices

Input: \( M_i'(N), N_j'(M), N_i', M \)

Output: \( N_i' \)

1: for each \( m_i \in M \) do
2: Add the first \( N_i' \) elements in \( M_i'(N) \) to \( K_i(0)(N) \)
3: end for
4: for each \( n_j \in N \) do
5: \( R_j(M) \leftarrow N_j'(M) \)
6: end for
7: \( l \leftarrow 0 \)
8: while \( M \neq \emptyset \) or \( N \neq \emptyset \) do
9: for each \( n_j \in N \) do
10: for each \( m_i \in R_j(0)(M) \) do
11: if \( n_j \in K_i(0)(N) \) then
12: Select \( n_j \) to \( N_i' \), and delete \( n_j \) from \( N \)
13: break;
14: end if
15: end for
16: end for
17: for each \( m_i \in M \) do
18: Update all \( K_i(l)(N) \) as \( K_i(l+1)(N) \)
19: if \( K_i(l)(N) = \emptyset \) then
20: Delete \( m_i \) from \( M \)
21: for each \( n_j \in N \) do
22: Delete \( m_i \) in \( R_j(0)(M) \) as \( R_j(l+1)(M) \)
23: end for
24: end if
25: if \( K_i(l+1)(N) = K_i(l)(N) \) then
26: Delete the best mobile device in \( K_i(l+1)(N) \), and update \( K_i(l+1)(N) \)
27: end if
28: end for
29: \( l \leftarrow l + 1 \)
30: end while
31: return \( N_i' \)
B. OPTIMAL CONTRACT DESIGN BETWEEN MOBILE DEVICES AND EDGE SERVERS

After the association relationship between mobile devices and edge servers is determined, edge servers can provide devices appropriate reward contacts to motivate them improving the quality of their local trained models.

1) UTILITY OF EDGE SERVERS AND MOBILE DEVICES

In iteration $t$, the size of data samples of each mobile device in the set $N^t_i$ matched with the edge server $m_i$ can be listed in ascending order, $d^t_{i1} < d^t_{i2} < \cdots < d^t_{iN^t_i}$. Due to information asymmetry, $m_i$ does not know the accurate size of data samples of mobile device $n_j$. It only knows the probability $p^t_{ij}$ that $n_j$ having the samples $D^t_{ij}$, and $\sum_{n_j \in N^t_i} p^t_{ij} = 1$. Edge servers should provide different contracts for devices with different data samples. We denote the contract provided by $m_i$ to $n_j$ as $(r^t_{ij}(\theta^t_{ij}, d^t_{ij}), d^t_{ij})$, where $d^t_{ij}$ is the size of data samples for $n_j$ training the model, $d^t_{ij}$ is the model quality, $r^t_{ij}(\theta^t_{ij}, d^t_{ij})$ is the reward paid by $m_i$, and $r^t_{ij}(\theta, d)$ is a monotonically increasing concave function of $\theta$ and $d$. The higher the model quality is, the higher the reward will be provided.

With contract $(r^t_{ij}(\theta^t_{ij}, d^t_{ij}), d^t_{ij})$, the utility of the edge server $m_i$ with which $n_j$ is associated is

$$V^t_i(r^t_{ij}) = p^t_{ij} q^t_{ij} - r^t_{ij}(\theta^t_{ij}, d^t_{ij}).$$

where $\omega > 0$ denotes the normalized parameter between model quality and rewards. Under the constraint that the payment of the edge server cannot exceed its maximum budget, its utility maximization problem with the incentive of mobile devices training a high-quality model can be defined as

$$\max_{n_j \in N^t_i} V^t_i = \sum_{n_j \in N^t_i} (N^t_i p^t_{ij} \omega q^t_{ij} - r^t_{ij})$$

where $\omega > 0$ denotes the normalized parameter between model quality and rewards. Under the constraint that the payment of the edge server cannot exceed its maximum budget, its utility maximization problem with the incentive of mobile devices training a high-quality model can be defined as

$$\max_{n_j \in N^t_i} V^t_i = \sum_{n_j \in N^t_i} (N^t_i p^t_{ij} \omega q^t_{ij} - r^t_{ij}).$$

The utility of the mobile device $n_j$ is defined as

$$U^t_j(m_i) = r^t_{ij} - cd^t_{ij},$$

where $c$ denotes the cost of the mobile device for per piece of data sample, and $d^t_{ij}$ is the size of data samples. When the mobile device $n_j$ does not participate in the model training, its reward is 0.

2) OPTIMAL CONTRACT DESIGN

To ensure the proposed contracts are feasible, each contract must satisfy individual rationality and incentive compatibility.

**Definition 1 (Individual Rationality (IR)):** The utility of the mobile devices $n_j$ participating in the model training associated with $m_i$ model is not less than 0, that is

$$r^t_{ij} - cd^t_{ij} \geq 0.$$  

**Definition 2 (Incentive Compatibility (IC)):** The utility of each mobile device cannot improve if it changes its current contract, that is

$$r^t_{ij} - cd^t_{ij} \geq r^t_{ik} - cd^t_{ik}, j \neq k, j, k \in N^t_i.$$  

We can define the optimal contract design problem for the edge server $m_i$ as follows.

$$\max V^t_i = \sum_{n_j \in N^t_i} (N^t_i p^t_{ij} \omega q^t_{ij} - r^t_{ij})$$

s.t. $r^t_{ij} - cd^t_{ij} \geq 0,$

$$r^t_{ij} - cd^t_{ij} \geq r^t_{ik} - cd^t_{ik},$$

$$d^t_{ik} < d^t_{ij} < \cdots < d^t_{in},$$

$$\sum_{n_j \in N^t_i} r^t_{ij} q^t_{ij} \leq B^t_i,$$

where (20) is the IR constraint, (21) is the IC constraint, (22) is the monotonic constraint, and (23) is the maximum budget constraint of the edge server.

In problem (19), there are a total of $N^t_i$ IR constraints and $N^t_i(N^t_i - 1)$ IC constraints. To address the issue of contract feasibility in practical applications, we reduce and relax the constraints.

**Lemma 1:** For feasible contracts $(r^t_{ij}, d^t_{ij})$ and $(r^t_{ik}, d^t_{ik})$, $r^t_{ij} \geq r^t_{ik}$ if and only if $d^t_{ij} \geq d^t_{ik}, j \neq k, j, k \in N^t_i$.

**Proof:** For feasible contracts $(r^t_{ij}, d^t_{ij})$ and $(r^t_{ik}, d^t_{ik})$, based on IC constraint, we have

$$r^t_{ij}(\theta^t_{ij}, d^t_{ij}) - cd^t_{ij} \geq r^t_{ik}(\theta^t_{ik}, d^t_{ik}) - cd^t_{ik},$$

$$r^t_{ik}(\theta^t_{ik}, d^t_{ik}) - cd^t_{ik} \geq r^t_{ij}(\theta^t_{ij}, d^t_{ij}) - cd^t_{ij}.$$  

Based on equations (24) and (25), we have

$$r^t_{ij}(\theta^t_{ij}, d^t_{ij}) - r^t_{ik}(\theta^t_{ik}, d^t_{ik}) \geq c(d^t_{ij} - d^t_{ik}),$$

$$r^t_{ik}(\theta^t_{ik}, d^t_{ik}) - r^t_{ij}(\theta^t_{ij}, d^t_{ij}) \leq c(d^t_{ij} - d^t_{ik}).$$  

**Adequacy:** If $r^t_{ij} \geq r^t_{ik}$, from equation (27) we know that $c(d^t_{ij} - d^t_{ik}) \geq 0$. Since $c > 0$, so $d^t_{ij} \geq d^t_{ik}$.

**Necessity:** If $d^t_{ij} \geq d^t_{ik}$, from equation (26) and $c > 0$ we know that $r^t_{ij}(\theta^t_{ij}, d^t_{ij}) - r^t_{ik}(\theta^t_{ik}, d^t_{ik}) \geq c(d^t_{ij} - d^t_{ik}) \geq 0$, i.e.

$$r^t_{ij} \geq r^t_{ik}.$$  

**Theorem 5 (Monotonicity):** For feasible contracts $(r^t_{ij}, d^t_{ij})$ and $(r^t_{ik}, d^t_{ik})$, $r^t_{ij} \geq r^t_{ik}$ and $d^t_{ij} \geq d^t_{ik}$, then $\theta^t_{ij} \geq \theta^t_{ik}, j \neq k, j, k \in N^t_i$.

**Proof:** For feasible contracts $(r^t_{ij}, d^t_{ij})$ and $(r^t_{ik}, d^t_{ik})$, since $r^t_{ij}(\theta, d)$ is a monotonically increasing function about $\theta$ and $d$. If $d^t_{ij} \geq d^t_{ik}$, then

$$r^t_{ij}(\theta^t_{ij}, d^t_{ij}) \geq r^t_{ik}(\theta^t_{ik}, d^t_{ik}).$$

And $r^t_{ij} \geq r^t_{ik}$, that is

$$r^t_{ij}(\theta^t_{ij}, d^t_{ij}) \geq r^t_{ik}(\theta^t_{ik}, d^t_{ik}).$$  

Therefore, based on equations (28) and (29), we have $\theta^t_{ij} \geq \theta^t_{ik}$.

**Theorem 6 (Reduct IR):** When the mobile device with the smallest size of data samples satisfies the IR constraint, i.e., $r^t_{ij} - cd^t_{ij} \geq 0$, then the other mobile devices also satisfy the IR constraint.
Proof: For feasible contracts \((r'_{ij}(\theta'_{ij}, d'_{ij}), d''_{ij})\), based on IC constraint, we have
\[
r'_{ij}(\theta'_{ij}, d''_{ij}) - cd''_{ij} \geq r'_{ij}(\theta'_{ij}, d''_{ij}) - cd''_{ij}.
\] (30)

Based on Lemma 1 and Theorem 5, if \(d''_{ij} \geq d''_{ij}\), then \(r'_{ij} \geq r'_{ij} + \theta'_{ij} \geq \theta'_{ij} + \theta'_{ij}\).

And since \(r'_{ij}(\theta, d)\) is a monotonically increasing function about \(\theta\) and \(d\), so
\[
r'_{ij}(\theta'_{ij}, d''_{ij}) - cd''_{ij} \geq r'_{ij}(\theta'_{ij}, d''_{ij}) - cd''_{ij}.\] (31)

Therefore, if \(r'_{ij}(\theta'_{ij}, d''_{ij}) - cd''_{ij} \geq 0\), then the other mobile devices also satisfy the IR constraint.

To reduce the IC constraint, we use the following definitions as [23].

Definition 3: For IC constraints between mobile devices \(n_j\) and mobile devices \(n_k\) participating in the training of the edge server \(m_i\) model:
1) if \(j > k\), IC is called downward incentive constraint (DIC);
2) if \(k = j - 1\), IC is called local downward incentive constraint (LDIC);
3) if \(j < k\), IC is called upward incentive constraint (UIC);
4) if \(k = j + 1\), IC is called local upward incentive constraint (LUIC).

Theorem 7 (Reduce IC): According to the monotonicity, the IC constraints can be reduced into local downward incentive constraints (LDIC) and local upward incentive constraints (LUIC).

Proof: We assume there are three mobile devices \(n_{ij(j+1)}\), \(n_{ij}\), \(n_{ij(j-1)}\) participating in model training for the edge server \(m_i\), satisfying \(d''_{ij(j+1)} \geq d''_{ij} \geq d''_{ij(j-1)}\). Based on IC constraint, we have
\[
r''_{ij(j+1)} - cd''_{ij(j+1)} \geq r''_{ij} - cd''_{ij}.
\] (32)
\[
r''_{ij} - cd''_{ij} \geq r''_{ij(j-1)} - cd''_{ij(j-1)}.
\] (33)

For all mobile devices participating in model training associated with the edge server \(m_i\), we have
\[
r'_{ij} - cd''_{ij} \geq r'_{ij} - cd''_{ij} \geq \ldots \geq r'_{ij} - cd''_{ij} \geq \ldots \geq r'_{ij} - cd''_{ij}.
\] (34)

Thus, they all satisfy the DIC and LDIC constraints. Similarly, it can also be proved that the mobile devices participating in model training for edge server \(m_i\) satisfy the UIC and LUIC constraints.

The IC constraint can be reduced to
\[
r''_{ij} - cd''_{ij} = r''_{ij(j-1)} - cd''_{ij}, \quad \forall n_j \in N_i.\] (35)

According to Theorems 5, 6 and 7, the optimization problem (19) is reduced and relaxed to
\[
\max V_i = \sum_{n_j \in N_i} (N_i p'_{ij} q'_{ij} - r''_{ij})
\] (36)

This is a convex optimization problem, and the optimal contract \((r'_{ij}, d''_{ij})\) can be obtained using convex optimization tools. With the reward \(r''_{ij}\) from edge server \(m_i\) to mobile device \(n_j\), \(d''_{ij}\) data samples are contributed by \(n_j\) and the edge model quality of \(m_i\) is maximized, resulting in the optimization of the global model quality.

C. MODEL AGGREGATION BASED ON MODEL QUALITY

In iteration \(t\), each associated mobile device uploads its local model parameters \(w'_{ij}\) to the edge server \(m_i\), and \(m_i\) aggregates them to update the edge model parameters \(w_i\). Different from the existing methods, we consider not only the size of training data set of each node, but also the quality of local models.

\[
w_i = \frac{\sum_{n_j \in N_i} q'_{ij} d''_{ij} w'_{ij}}{\sum_{n_j \in N_i} q'_{ij} d''_{ij}}.
\] (41)

Similarly, the global aggregated model is updated as
\[
w = \frac{\sum_{m \in M} \sum_{n_j \in N_i} q'_{ij} d''_{ij} w'_{ij}}{\sum_{m \in M} \sum_{n_j \in N_i} q'_{ij} d''_{ij}}.
\] (42)

We improve the model training process of HFL involving our proposed incentive mechanism based on matching game and contract theory, as well as the new model aggregation approach, which is shown as Algorithm 2.

IV. PERFORMANCE EVALUATION

In this section, we investigate the performance of our proposed MaxQ through simulations.

A. SIMULATION ENVIRONMENT SETTINGS

We consider a hierarchical federated learning scenario consisting of 60 mobile devices, 5 edge servers and a cloud server. It is assumed that each mobile device can train model for an edge server and each edge server can associate with up to 10 mobile devices participating in its model training. We assume that each mobile device arrives or leaves randomly. For the model training an image classification task is considered, on standard datasets CIFAR10 [24] with ResNet18 and ResNet50 models, as well as MNIST [25] with AlexNet model. The possible reward from the edge server to each mobile device lies in [5, 50]. We compare the performance of MaxQ with the existing algorithms FAIR [20] and EHFL [26], the former using quality-based reverse auction for node selection in FL and the latter considering optimization for device association with edge servers in HFL.
Algorithm 2 Improved Model Training Process of Hierarchical Federated Learning

1: for each \( t \in T \) do
2:   According to algorithm 1, the set of mobile devices training model for each edge server \( m_i \) as \( \mathcal{N}_i^t \) is obtained.
3:   for each \( m_i \in \mathcal{M} \) do
4:     Based on contract theory, the edge server \( m_i \) encourages mobile devices in \( \mathcal{N}_i^t \) to select training data set for model training.
5:     for each \( n_j \in \mathcal{N}_i^t \) do
6:       The mobile device \( n_j \) performs local model training to obtain local model parameter \( w_{ij}^t \).
7:     end for
8:     if \( t \mod l = 0 \) then
9:       The edge server \( m_i \) performs the edge aggregation model \( w_i^t \) according to equation (41).
10:   end if
11: end for
12: if \( t \mod k_l = 0 \) then
13:   The cloud server performs the edge aggregation model \( w_i^t \) according to equation (42).
14: end if
15: end for

B. ANALYSIS OF MODEL CONVERGENCE SPEED

First, we analyse the convergence speed of MaxQ. Figure 2 and 3 show how the model accuracy and loss functions change with the number of iterations, under ResNet18, ResNet50 and AlexNet, respectively. It can be found in Figure 2 that the accuracy of MaxQ is higher than FAIR and EHFL after the same number of iterations. After convergence, the model accuracy achieved in MaxQ is 80.18%, 80.60% and 95.03%, while that in FAIR and EHFL is 73.56% and 77.13%, 78.31% and 78.10%, 92.35% and 92.96%, respectively. Also, it can be found in Figure 3 that the loss of MaxQ is lower after the same number of iterations. This is due to the effectiveness of proposed incentive mechanism, which encourages mobile devices to provide high quality data and the model quality can be improved, resulting in faster convergence speed.

C. ANALYSIS OF THE MODEL QUALITY

We analyse the obtained model quality under different mechanisms. It can be observed from Figure 4 that the quality of MaxQ is higher than FAIR and EHFL, with the improvement 8.20% and 7.81%, 10.47% and 11.87%, 10.98% and 11.97%, respectively. This is due to the fact that MaxQ selects mobile devices with high quality models for each edge server based on matching game and contract theory. In FAIR, only the selection of mobile devices is optimized, but the selection of edge servers is not considered. In EHFL, edge servers select mobile devices with lower energy consumption during training and the ones with smaller datasets may be selected, which can not improve the model quality.

D. ANALYSIS OF TRAINING DELAY

We also investigate the training delay under our proposed method. The computing delay of local models is calculated as \( t_{\text{comp}} = \frac{eD}{f} \), where \( e \) is the number of CPU cycles per bit to process a piece of data sample (set as 20 cycles/bit in the simulation), \( D \) is the size of data samples and \( f \) is CPU cycle frequency (set as 1 GHz). Figure 5 demonstrates the average computing delay of local models on mobile devices under ResNet18, ResNet50 and AlexNet, respectively. It can be found that the average delay of MaxQ and EHFL is better than that of FAIR. Compared with the traditional two-layer FL, HFL can improve the training delay because the quality of edge models is usually better than that of local models on devices.

E. EFFECTIVENESS OF MAXQ WITH THE EXISTENCE OF SELFISH NODES

To analyse the effectiveness of MaxQ tolerating selfish nodes, we introduce certain number of selfish nodes which uploads randomly generated model parameters for edge model aggregation instead of actual training. As the number of selfish nodes increases, the quality of the aggregated model will decrease, as shown in Figure 6. It can also be found that the global model accuracy of MaxQ is higher than that of FAIR and EHFL, with an average improvement of 18.10% and 64.95%, 17.62% and 63.74%, and 21.22% and 82.35%, respectively. The performance of EHFL is worse than MaxQ and FAIR because the latter two methods both consider the model quality as weight coefficients in the process of aggregation, while it only simply takes the average of uploaded parameters. MaxQ outperforms FAIR because in FAIR model aggregation is taken only once on the parameter server in each global iteration, while in MaxQ the aggregation is also taken on the edge servers, which reduces the influence of node selfishness.

V. RELATED WORK

Google proposed federated learning (FL) in 2016 [5]. However, in cloud-based FL the communication efficiency between cloud servers and mobile devices may be low, while in edge-based FL the number of participating mobile devices may be limited. Therefore, Liu et al. proposed cloud-edge-client based hierarchical federated learning [10].

Compared with FL, HFL can significantly reduce the communication overhead between mobile devices and the parameter server [27]. In literature, Wang et al. [28] proposed HF-SGD model with multi-level parameters aggregation. Xu et al. [26] proposed a device allocation strategy according to energy consumption of different devices. However, they do not consider how to design effective incentive mechanisms.

To encourage mobile devices to train high quality models in HFL, some researchers use the size of training data samples to measure the contribution of mobile devices [29], [30]. By combining game theory with deep reinforcement learning, Zhan et al. [30] proposed an incentive mechanism.
FIGURE 2. Accuracy v.s. Number of Iterations.

FIGURE 3. Loss v.s. Number of Iterations.

FIGURE 4. Model Quality v.s. Number of Iterations.

FIGURE 5. Average Delay in each Edge Server.
for FL. They formulated the interaction between the parameter server and clients as a Stackelberg game. Pandey et al. [31] developed an incentive mechanism which is a value-based compensation strategy and the reward to mobile devices is proportional to the level of participation in FL. A two-stage Stackelberg game approach is used to solve the initial optimization problem to maximize the benefits of both parties, and an admission control scheme is introduced for clients to ensure a certain accuracy level. Deng et al. [20] investigated the learning quality maximization problem in FL based on auction theory. However, the above work does not address the issue of information asymmetry between edge servers and mobile devices.

In [32], an incentive mechanism based on contract theory was proposed. Among the existing work where contract theory is used, Kang et al. [33], [34] proposed to recruit devices with high-quality data to participate in FL. Reference [34] proposed an effective incentive mechanism combining reputation with contract theory, which encourages high reputation mobile devices with high-quality data to participate in model learning. Ding et al. [35] introduced a multi-dimensional contract design method for the context where users’ dimensional private information was considered. They also investigated the impact of the information asymmetry level on the server’s optimal policy design. Li et al. [36] transformed the incentive data holder problem into the utility maximization problem and established an incentive mechanism based on the contract theory. Contract theory based incentive mechanism design is also investigated in the related research area, e.g., mobile crowd sensing [18]. However, little existing work has considered the dynamics of the association relationship between mobile devices and edge servers, which is different from our work. We introduce contract theory on the basis of [37] to encourage mobile devices to train high-quality models, after the association relationship is determined.

VI. CONCLUSION
To optimize the model training quality in hierarchical federated learning, we investigate the problem of global model maximization, the decision problem of which is proved NP-complete. We propose an mechanism MaxQ which consists of two stages. In the first stage, we associate mobile devices with proper edge servers based on matching game theory, which is proved able to achieve the stability and that the approximation ratio is $\frac{1}{2}$. In the second stage, we propose an incentive mechanism based on contract theory to improve the quality of models submitted by mobile devices to edge servers. Through thorough experiments, we analysed the performance of MaxQ under different deep learning models ResNet18, ResNet50 and AlexNet individually, compared with the existing approaches FAIR and EHFL. It is found that MaxQ can improve the model quality by 8.20% and 7.81%, 10.47% and 11.87%, 10.98% and 11.97% with different learning models, respectively.

ACKNOWLEDGMENT
An earlier version of this paper was presented at the INFOCOM WKSHP5 A4E 2022 “Quality-Aware Incentive Mechanism Design Based on Matching Game for Hierarchical Federated Learning [DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798096].”

REFERENCES
[1] (Aug. 15, 2021). Cisco Annual Internet Report, 2018–2023, White Paper. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
[2] (Aug. 15, 2021). Ericsson Mobility Report, June 2021. [Online]. Available: https://www.ericsson.com/a/a3c2/assets/local/mobility-report/documents/2021/june-2021-ericsson-mobility-report.pdf
[3] X. Zhu, H. Li, and Y. Yu, “Blockchain-based privacy preserving deep learning,” in Information Security and Cryptology, F. Guo, X. Huang, and M. Yung, Eds. Cham, Switzerland: Springer, 2019, pp. 370–383.
[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. Surveys, vol. 52, no. 6, pp. 1–36, Oct. 2019.
[5] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” 2016, arXiv:1602.05629.
[6] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Leung, C. Miao, and Q. Yang, “Incentive mechanism design for resource sharing in collaborative edge computing,” 2020, arXiv:2006.00511.
[7] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. Brendan McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, “Towards federated learning at scale: System design,” 2019, arXiv:1902.01046.
[8] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, “Federated learning over wireless networks: Optimization model design and analysis,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2019, pp. 1387–1395.
[9] H. G. Abreha, M. Hayajneh, and M. A. Serhani, “Federated learning in edge computing: A systematic survey,” Sensors, vol. 22, no. 2, p. 450, Jan. 2022.

[10] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2020, pp. 1–6.

[11] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, M. J. Piran, and M. S. Hossain, “Toward accurate anomaly detection in industrial Internet of Things using hierarchical federated learning,” IEEE Internet Things J., vol. 9, no. 10, pp. 7110–7119, May 2022.

[12] A. Masood, T.-V. Nguyen, T. P. Truong, and S. Cho, “Content caching in HAP-assisted multi-UAV networks using hierarchical federated learning,” in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2021, pp. 1160–1162.

[13] J. Lin, J. Ma, and J. Zhu, “Hierarchical federated learning for power transformer fault diagnosis,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–11, 2022.

[14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, p. 436, Nov. 2015.

[15] Y. Zou, F. Shen, F. Yan, J. Lin, and Y. Qiu, “Reputation-based regional federated learning for knowledge trading in blockchain-enhanced IoV,” in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2021, pp. 1–6.

[16] H. Chai, S. Leng, Y. Chen, and K. Zhang, “A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of Vehicles,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 7, pp. 3975–3986, Jul. 2021.

[17] D. Y. Zhang, Z. Kou, and D. Wang, “FedSens: A federated learning approach for smart health sensing with class imbalance in resource constrained edge computing,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), May 2021, pp. 1–10.

[18] R. Saha, S. Misra, and P. K. Deb, “FogFL: Fog-assisted federated learning for resource-constrained IoT devices,” IEEE Internet Things J., vol. 8, no. 1, pp. 8456–8463, May 2021.

[19] K. Bando, “Many-to-one matching markets with externalities among firms,” J. Math. Econ., vol. 48, no. 1, pp. 14–20, Jan. 2012.

[20] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, “Fair: Quality-aware federated learning with precise user incentive and model aggregation,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), May 2021, pp. 1–10.

[21] W. Y. B. Lim, Z. Xiong, C. Miao, D. Niyato, Q. Yang, C. Leung, and H. V. Poor, “Hierarchical incentive mechanism design for federated machine learning in mobile networks,” IEEE Internet Things J., vol. 7, no. 10, pp. 9575–9588, Oct. 2020.

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

[23] Y. Zhang and Z. Han, Incentive Mechanisms for Device-to-Device Communications in Cellular Networks With Adverse Selection. Cham, Switzerland: Springer, 2017, pp. 17–41.

[24] A. Krizhevsky, “Learning multiple layers of features from tiny images,” J. Math. Econ., vol. 48, no. 1, pp. 14–20, Jan. 2012.

[25] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, “Incentive design for efficient federated learning in mobile networks: A contract theory approach,” in Proc. IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS), Aug. 2019, pp. 1–5.

[26] 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. 1070–10714, Dec. 2019.

[27] N. Ding, Z. Fang, and J. Huang, “Optimal contract design for efficient federated learning with multi-dimensional private information,” IEEE J. Sel. Areas Commun., vol. 39, no. 1, pp. 186–200, Jan. 2021.

[28] L. Li, X. Yu, X. Cai, X. He, and Y. Liu, “Contract theory based incentive mechanism for federated learning in health CrowdSensing,” IEEE Internet Things J., early access, Oct. 31, 2022, doi: 10.1109/JIOT.2022.3218008.

[29] D. Hui, L. Zhuo, and C. Xin, “Quality-aware incentive mechanism design based on matching game for hierarchical federated learning,” in Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), May 2022, pp. 1–6.

ZHOU LI (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Nanjing University, China. He is currently an Associate Professor with Beijing Institute of Science and Technology University. He has published more than 100 papers. His research interests include distributed machine learning, wireless networks, and mobile computing.

HUI DU received the B.S. degree from the Jiangxi University of Science and Technology, in 2019. He is currently pursuing the master’s degree with the School of Computer Science, Beijing Information Science and Technology University. His main research interest includes mobile edge computing.

XIN CHEN received the Ph.D. degree in computer science from the Beijing Institute of Technology, China. He is currently a Professor with the School of Computer Science, Beijing Information Science and Technology University, where he is also the Vice-President. His current research interests include performance evaluation and optimization, and wireless networking. He is a Senior Member of CCF.