IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-0926-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798096

Quality-Aware Incentive Mechanism Design Based on Matching Game for Hierarchical Federated Learning

DU Hui[†*], LI Zhuo[†*§], CHEN Xin[*]
[†]Beijing Key Laboratory of Internet Culture and Digital Dissemination Research,
Beijing Information Science and Technology University
[*]School of Computer Science, Beijing Information Science and Technology University
duhui199801@163.com, {lizhuo, chenxin}@bistu.edu.cn
[§]Corresponding author

***Abstract*—To protect user privacy and combined with mobile edge computing, hierarchical federated learning (HFL) is proposed. In HFL, we investigated the aggregated model quality maximization problem. Since the global model quality is influenced by the local model quality, we transformed the aggregated model quality maximization into the sum of local model quality maximization. And we proposed the model quality maximization mechanism MaxQ based on matching game to select high quality mobile devices. In MaxQ, the allocation of mobile devices to each edge server is realized so that the sum of the local model quality is maximized. And we proved that MaxQ has a $\frac{1}{2}-$approximation ratio. Finally, through a large number of simulation experiments, compared with FAIR and EHFL, the model quality of MaxQ is improved by 10.8% and 12.2%, respectively.**

***Index Terms*—Hierarchical Federated Learning, Maximization of Model Quality, Matching Game, Incentive Mechanism Design**

I. Introduction

According to 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. And Ericsson [2] predicts that the average monthly total mobile network traffic will exceed 300 EB by 2026. In conventional machine learning, a large number of user data will be uploaded to the cloud server to more effective learning. However, this will leakage users privacy information [3], and with increasingly data generated, uploading it to the cloud server will generate a lot of delay and energy consumption [4]. Facing these challenges, Google proposed federated learning in 2016 [5]. In federated learning, model training is performed only on mobile devices, and only the trained local models need to be uploaded to the cloud server for aggregation into global models. Thus federated learning not only reduces energy consumption and transmission latency but also protectes privacy of users.

In conventional federated learning, cloud servers are mainly used as parameter servers [6]. Due to the scale of mobile devices reaching millions [7], which face the problem of slow communication between parameter servers, leading to inefficient model training. With the emergence of edge computing, the edge server is used as the parameter server [8], [9]. Due to the limited number of mobile devices under the edge server, there are too few mobile devices, resulting in low model quality. Therefore, L. Liu et al. [10] proposed cloud-edge-client hierarchical federated learning and overcomes the above problems. Framework of hierarchical federated learning is shown in Fig. 1.

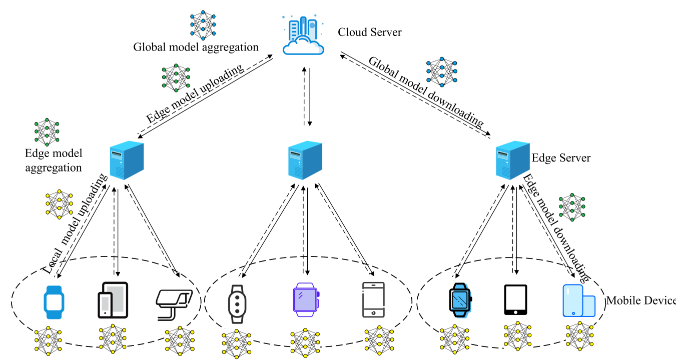

Fig. 1. Framework of hierarchical federated learning

With the development of artificial intelligence [11], many novel applications have been generated. Model quality is closely related to the performance of applications, such as autonomous driving [12], [13], health monitoring [14], etc. The quality of deep learning model will be closely related to safety problems. In practice, each mobile device has different data quatity and data quality. Therefore, the quality of the local model obtained by each mobile device participating in the model training is different. In hierarchical federated learning, because participating in model training needs to consume its own resources [15], it is necessary to design an effective incentive mechanism to incentive mobile devices to participate to training high quality models.

In this paper, to maximize the global model quality, the global model quality maximization mechanism MaxQ is designed to achieve matching between multiple edge servers and multiple mobile devices. In each training round, multiple mobile devices under one edge server participate in local

978-1-6654-0926-1/22/$31.00 ©2022 IEEE

model training, and a mobile device can only train a model for one edge server. In practice, mobile devices can participate or leave at any time. First, based on the theory of many-to-one matching game [16], the allocation of mobile devices to each edge server is realized so that the profit of both parties is maximized. That is, for the edge server, the profit represents the desired edge model quality of aggregation; for the mobile device, the profit represents the utility. Then, after the matching is finished, the corresponding mobile devices are available under each edge server. Finally, when the models are aggregated, higher weights are given to high quality local models to obtain high quality aggregated models and to reduce the impact of low quality models on the quality of aggregated models.

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

- In hierarchical federated learning, since the global model quality is affected by the local model quality, in order to maximize the global model quality, this paper transforms the global model quality maximization problem into the sum of local model quality maximization problem MQM, and proves that the MQM problem is NP-hard.
- In hierarchical federated learning, this paper proposes MaxQ, a matching mechanism between edge servers and mobile devices, based on a many-to-one matching game for maximizing the quality of aggregation models, and proves the stability of the matching game.
- Through a large number of simulation experiments, the MaxQ mechanism proposed in this paper can effectively accelerate the model convergence and improve the model quality by 10.8% and 12.2%, respectively, compared with FAIR and EHFL.

The remainder of this paper is organized as follows. Section II briefly introduces the related works. We define the system model and maximization of quality of global model problem in Section III. The Maximization of Quality of Global Model mechanism MaxQ is proposed in Section IV. We analyze the simulation results in Section V, before concluding in Section VI.

II. Related Work

Google proposed federated learning in 2016 [5]. However, cloud-based federated learning has slow communication efficiency between cloud servers and mobile devices, and edge-based federated learning has limited number of participating mobile devices. Therefore, Liu et al. proposed cloud-edge-client hierarchical federated learning [10].

Compared to cloud-based federated learning, hierarchical federated learning will significantly reduce the communication between mobile devices and cloud servers. and update efficiently between mobile devices and edge servers, thus significantly reducing the number of iterations between mobile devices and cloud servers [17]. In the existing work of hierarchical federated learning, J. Wang et al. [18] studied HF-SGD model with multi-level parameter aggregation in hierarchical federated learning. B. Xu et al. [19] proposed a device allocation scheme based on energy consumption in hierarchical federated learning. However, the above work does not propose an effective incentive mechanism to encourage mobile devices to participate to training high quality models.

Among the existing work on model quality-related incentives, several have used the amount of training data to measure client contributions as the accuracy of learning models is related to the size of training samples [20]. Furthermore, by combining game theory with deep reinforcement learning, Zhan et al. [21] proposed a federated learning incentive mechanism based on deep reinforcement learning. To motivate clients to participate in model training, Zhan et al. describe the interaction between the parameter server and clients as a Stackelberg game. Pandey et al [22] proposed a new crowd-sensing platform by constructing a communication efficiency cost model that considers the communication efficiency during the exchange of model parameters. The incentive mechanism is a value-based compensation strategy, such as a bonus, which is proportional to the level of federated learning participation. A two-stage Stackleberg countermeasure approach is used to solve the initial optimization problem of maximizing the benefits of both parties. In addition, an admission control scheme is provided for the client to ensure a certain local accuracy. Y. Deng et al [23] proposed a learning quality maximization problem in federated learning based on an auction mechanism. However, in the above work, the case of multiple edge servers in hierarchical federated learning is not considered. Therefore, this paper realizes matching between edge servers and mobile devices based on matching game theory.

III. System Model and Problem Definition

In this section, the hierarchical federated learning model and the global model quality maximization problem are modeled.

A. Hierarchical Federated Learning Model

In the hierarchical federated learning framework, there is one cloud server, multiple edge servers and multiple mobile devices. We assume a set of edge servers $\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$, a set of mobile devices $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$. Model training is defined in discrete time slots $\mathcal{T} = \{t_1, t_2, \ldots, t_T, \ldots\}$. And the model is iterated once in each time slots. In hierarchical federated learning, first, cloud server distribute a machine learning model to each edge servers. In iteration t, budget of edge server m_i is B_i^t for model training$(i = 1, 2, \ldots, M)$. Then, edge server distribute model to each mobile device, and a set of mobile devices training model for edge server m_i is $\mathcal{N}_i^t, \mathcal{N}_i^t \subset \mathcal{N}$. In practice, mobile device may arrive or leave at any time. We assume mobile device n_j arrive at iteration a_j and leave at d_j. Data samples set of mobile device n_j is $\mathcal{D}_j$, and the quatity of data samples is $d_j = |\mathcal{D}_j|$. In iteration t, data quality of mobile device n_j is θ_j^t, and estimation of model quality is $\hat{q}_j^t$. Moreover, due to the limited computing power of mobile devices, each iteration can only train the model for one edge server.

B. Estimation of Model Quality

In federated learning, the quantity and quality of training data will significantly affect the quality of the model [23]. The quantification of model quality should fully reflect the contribution of local model updating to the global model. A feasible method is to use the local model accuracy of each device tested on the global data set as the model quality. However, in this method, each local model needs to be tested in each iteration, which will cause great overhead. Different from the accuracy, the loss function is calculated in the training process without additional overhead. Therefore, we use the loss function reduction in each iteration to quantify the training data quality.

We assume each mobile device start local model training whose loss function is $loss_j(\tau)$ in time slot τ, and updates local model during Δ time slots, whose loss function is $loss_j(\tau+\Delta)$ in time slot $\tau+\Delta$. Therefore, In iteration t, data quality of mobile device n_j is

$$\theta_j^t = loss_j(\tau+\Delta) - loss_j(\tau). \tag{1}$$

According to the data sample quatity and data quality of the mobile device n_j, in the iteration t, the model quality [24] of the mobile device n_j is

$$q_j^t = 1 - e^{-\varphi(\theta_j^t d_j)^{\upsilon}}, \tag{2}$$

where φ and υ is weight coefficient.

Assuming mobile device n_j participate in local model training in iteration $t_0, t_1, \ldots, t_k$, we estimate 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 $t_{k+1} > t_k$. The model quality will change over time, and the recent model quality can better reflect the current model quality than the old model quality. Therefore, we use the exponential forgetting function to allocate the weight, which provides a greater weight for the recent model quality record and a smaller weight for the old model quality. The weight of the latest model quality is as 1, and the weight of other model quality is determined by its relative position to the latest model quality. The weight corresponding to historical model quality $(q_j^{t_0}, q_j^{t_1}, \ldots, q_j^{t_k})$ is $(\rho^{t_k-t_0}, \rho^{t_k-t_1}, \ldots, \rho^{t_k-t_{k-1}}, 1)$ where $0 \leq \rho \leq 1$ is forgetting factor. So in iteration t estimation of model quality of moblie device n_j [23] is

$$\hat{q_j^t} = \frac{\sum_{i=0}^{k} \rho^{t_k-t_i} q_j^{t_i}}{\sum_{i=0}^{k} \rho^{t_k-t_i}}. \tag{3}$$

C. Problem Definition

In iteration t, according to its own budget B_i^t and estimation of model quality of moblie device n_j, edge server m_i determines which mobile devices train the model, and pay reward r_{ij}^t to moblie device n_j. The estimation of model quality of mobile device n_j training model for the edge server m_i is $\hat{q_{ij}^t}$. Once the mobile device is selected by the edge server, the mobile device uses its own data to train the model and uploads the local model update to the edge server. After receiving local model updates from different mobile devices, each edge server aggregates local models for update the edge model, then publishes its edge model to its selected mobile device. Until the edge model quality reaches the threshold, each edge server uploads the edge model to the cloud server. After receiving the aggregated models of all edge servers, the cloud server aggregates them into global models, and publishes the global models to each edge server until the global model reaches the threshold. Since at the beginning of each global iteration, the edge model is the global model distributed from the cloud server and the local model is the edge model distributed from the edge server. Therefore, the quality of each aggregated model should be maximized in each iteration. The aggregation model quality is influenced by the local model, so maximizing the aggregation model quality can be transformed to maximizing the sum of local model quality. The local model quality sum maximization problem (MQM) is defined as

$$\max \quad \sum_{m_i\in\mathcal{M}} \sum_{n_j\in\mathcal{N}} x_{ij} \hat{q_{ij}^t} \tag{4}$$

$$s.t. \sum_{n_j\in\mathcal{N}_i^t} r_{ij}^t x_{ij}^t \leq B_i^t, \forall m_i \in \mathcal{M}, \tag{5}$$

$$\sum_{m_i\in\mathcal{M}} x_{ij}^t \leq 1, \forall n_i \in \mathcal{N}, \tag{6}$$

$$x_{ij}^t \in \{0,1\}, \forall m_i \in \mathcal{M}, \forall n_j \in \mathcal{N}_i^t, \tag{7}$$

where constraint (5) indicates that the reward of the edge server m_i to the mobile device in iteration t cannot exceed the maximum budgetB_i^t; constraint (6) indicates that each mobile device participates in the model training of at most one edge server in the iteration t .

Theorem 1. *The decision problem for the MQM problem proposed in this paper is an NP-complete problem.*

We omit the proof due to the length of limitation.

IV. Maximization of Quality of Global Model Mechanism MaxQ

In this section, we propose Maximization of Quality of Global Model mechanism **MaxQ** to maximize quality of global model. In MaxQ, the edge server should choose the mobile device that trains the high-quality model to maximize the global model quality. and the mobile device should select edge server with high reward to maximize its profit. Each mobile device can only participate in the local model training of one edge server. And there are multiple mobile devices training model under each edge server. Therefore, there exist a many to one matching between the edge server and the mobile device.

In iteration t, matching matrix between edge servers and mobile devices is

$$\mathbf{X}^t = \begin{pmatrix} x_{11}^t & x_{12}^t & \cdots & x_{1n}^t \\ x_{21}^t & x_{22}^t & \cdots & x_{2n}^t \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1}^t & x_{n2}^t & \cdots & x_{nn}^t \end{pmatrix}, \tag{8}$$

where $x_{ij}^t = 1$ indicates that the mobile device is allocated to the edge server, and the two sides form a match. When the edge server and the mobile device can not improve own revenue by establishing a new match, the match is stable.

In iteration t, we assume edge server m_i requires N_i^t mobile devices where $N_i^t = |\mathcal{N}_i^t|$, and the expected rewards of edge server m_i to each mobile device is a $(r_{i1}^t, r_{i2}^t, \ldots, r_{iN_i^t}^t)$ where $r_{ik}^t \geq r_{ij}^t$ if $k \leq j$. Since the mobile device does not participate in the model training at this time, it is unable to determine its own reward. We assume that reward of mobile device m_j from edge server n_i is expressed in R_{ij}^t, and the probability of reward r_{ik}^t for mobile device m_j is p_{ik}^t, i.e. $P\{R_{ij}^t = r_{ik}^t\} = p_{ik}^t$, where $\sum_{k=1}^{N_i^t} p_{ij}^t = 1$.

The profit function of mobile device n_j selecting edge server m_i is defined as

$$u_j^t(m_i) = \mathbb{E}(R_{ij}^t) - cd_j, \tag{9}$$

where $\mathbb{E}(R_{ij}^t)$ is mathematical expectation of R_{ij}^t, and c is the cost of processing each data sample when the mobile device participates in model training. According to the profit function $u_j^t(m_i)$, we establish a priority list of edge servers for mobile device n_j that

$$\mathcal{N}_j^t(\mathcal{M}) = \{u_j^t(m_{j_1}), u_j^t(m_{j_2}), \ldots, u_j^t(m_{j_M})\}. \tag{10}$$

The profit function of edge server m_i selecting mobile device n_j is defined as

$$v_i^t(n_j) = \hat{q}_j^t. \tag{11}$$

According to the profit function $v_i^t(n_j)$, we establish a mobile devices priority list of for edge server m_i that

$$\mathcal{M}_i^t(\mathcal{N}) = \{v_i^t(n_{i_1}), v_i^t(n_{i_2}), \ldots, v_i^t(n_{i_N})\}. \tag{12}$$

The matching between the edge server and the mobile device is shown in algorithm 1. Firstly, we calculate the revenue function of mobile devices and edge servers, and get the sorting list of available mobile devices $\mathcal{M}_i^t(\mathcal{N})$ and available edge servers $\mathcal{N}_j^t(\mathcal{M})$. And according to $\mathcal{N}_i^t$, it is created that the priority list of mobile devices $\mathcal{K}_i(\mathcal{N}) = \{n_{i_1}, n_{i_2}, \ldots, n_{i_{N_i^t}}\}$ for edge server m_i, and the priority list of edge servers $\mathcal{R}_j(\mathcal{M}) = \mathcal{N}_j^t(\mathcal{M})$ for mobile device n_j. Then, the matching is successful when mobile device n_j is in the priority mobile device list $\mathcal{K}_{j_1}(\mathcal{N})$ of its priority edge server m_{j_1} and the priority mobile device list of all edge servers are updated. If $\mathcal{K}_i(\mathcal{N})$ is empty, edge server m_i is removed from the priority list of edge servers all mobile devices. When there is no change in $\mathcal{K}_i(\mathcal{N})$ before and after two matches, the best mobile device in $\mathcal{K}_i(\mathcal{N})$ is removed and updated. When mobile device n_j is not in the priority mobile device list $\mathcal{K}_i(\mathcal{N})$ of its priority edge server m_{j_1}, m_{j_1} is removed from $\mathcal{R}_j(\mathcal{M})$ and matched with edge server m_{j_2}, and so on until the match is successful or $\mathcal{R}_j(\mathcal{M})$ is empty.

Theorem 2. *The matching game in Algorithm 1 is able to achieve stability, and Algorithm 1 has a $\frac{1}{2}-$approximation ratio.*

Algorithm 1 Matching Algorithm between Edge Servers and Mobile Devices

Input: $\mathcal{M}_i^t(\mathcal{N})$, $\mathcal{N}_j^t(\mathcal{M})$, N_i^t, $\mathcal{M}$
Output: $\mathcal{N}_i^t$
1: **for** each $m_i \in \mathcal{M}$ **do**
2: Add the first N_i^t elements in $\mathcal{M}_i^t(\mathcal{N})$ to $\mathcal{K}_i^{(0)}(\mathcal{N})$
3: **end for**
4: **for** each $n_j \in \mathcal{N}$ **do**
5: $\mathcal{R}_j^{(0)}(\mathcal{M}) \leftarrow \mathcal{N}_j^t(\mathcal{M})$
6: **end for**
7: $l \leftarrow 0$
8: **while** $\mathcal{M} \neq \varnothing$ or $\mathcal{N} \neq \varnothing$ **do**
9: **for** each $n_j \in \mathcal{N}$ **do**
10: **for** each $m_i \in \mathcal{R}_j^{(l)}(\mathcal{M})$ **do**
11: **if** $n_j \in \mathcal{K}_i^{(l)}(\mathcal{N})$ **then**
12: Select n_j to $\mathcal{N}_i^t$, and delete n_j from $\mathcal{N}$
13: break;
14: **end if**
15: **end for**
16: **end for**
17: **for** each $m_i \in \mathcal{M}$ **do**
18: Update all $\mathcal{K}_i^{(l)}(\mathcal{N})$ as $\mathcal{K}_i^{(l+1)}(\mathcal{N})$
19: **if** $\mathcal{K}_i^{(l)}(\mathcal{N}) = \varnothing$ **then**
20: Delete m_i from $\mathcal{M}$
21: **for** each $n_j \in \mathcal{N}$ **do**
22: Delete m_i in $\mathcal{R}_j^{(l)}(\mathcal{M})$ as $\mathcal{R}_j^{(l+1)}(\mathcal{M})$
23: **end for**
24: **end if**
25: **if** $\mathcal{K}_i^{(l+1)}(\mathcal{N}) = \mathcal{K}_i^{(l)}(\mathcal{N})$ **then**
26: Delete the best mobile device in $\mathcal{K}_i^{(l+1)}(\mathcal{N})$, and update $\mathcal{K}_i^{(l+1)}(\mathcal{N})$
27: **end if**
28: **end for**
29: $l \leftarrow l + 1$
30: **end while**
31: **return** $\mathcal{N}_i^t$

We omit the proof due to the length of limitation.

In Theorem 2, we conduct a theoretical analysis of the performance of Algorithm 1.

The time complexity of Algorithm 1 is analyzed as follows. 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 $O(MN^2)$.

V. Performance Evaluation

In this section, the performance of Max Q mechanism proposed in this paper is analyzed through simulation experiments.

A. Simulation Environment

This paper considers a hierarchical federated learning framework with 60 mobile devices, 5 edge servers and a cloud server. It is assumed that each mobile device can train model for one edge server, each edge server has up to 10 mobile devices participating in model training. Each mobile device arrives or leaves randomly. Each model training task is an image classification task on the standard dataset CIFAR10, and the model is ResNet50. Assuming that the expected reward of the edge server to each mobile device follows the uniform distribution of $[5, 50]$, the initial model quality of each mobile device is 1.

To verify the advantages of MaxQ algorithm, FAIR [23] and EHFL [19] are used as comparison algorithms in this paper. FAIR is a node selection algorithm on quality-based reverse auction in conventional federated learning, while EHFL is an association algorithm between mobile devices and edge servers based on minimize energy consumption in hierarchical federated learning.

B. Convergence Analysis

To verify the convergence of the MaxQ mechanism, Fig. 2 show the relationship between accuracy and the number of iterations on the ResNet50 models. Fig. 3 show the relationship between the loss function and the number of iterations on the ResNet50 model. From Fig. 2, it can be seen that the accuracy of the MaxQ mechanism can be higher for the same number of iterations compared to FAIR and EHFL. The accuracy of MaxQ was able to reach 80.10%, and the accuracy of FAIR and EHFL reached 73.48% and 77.05%, respectively. From Fig. 3, it can be seen that the loss of MaxQ mechanism can be lower for the same number of iterations compared with FAIR and EHFL. Therefore, MaxQ is able to converge faster. This is due to the fact that the MaxQ mechanism selects mobile devices with high quality models for each edge server through a matching game and pays different rewards, thus speeding up the model convergence. However, EHFL does not use an incentive mechanism, and each edge server only selects mobile devices with low energy consumption to participate in model training, which will cause the selected mobile devices not to contribute their full capacity to reduce energy consumption, and thus the model convergence of EHFL is lower than that of MaxQ. In FAIR, the mobile devices are only incentive to participate in training using reverse auction in conventional federated learning. Therefore, the model convergence speed of FAIR is lower than that of MaxQ. Moreover, the shortcomings of the traditional two-layer federation learning lead to a lower convergence speed of FAIR than EHFL at low number of iterations.

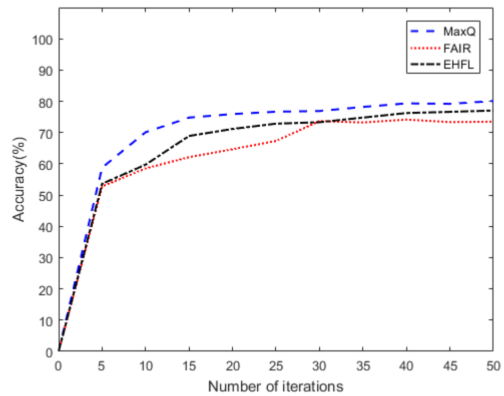

Fig. 2. Accuracy v.s. Number of Iterations

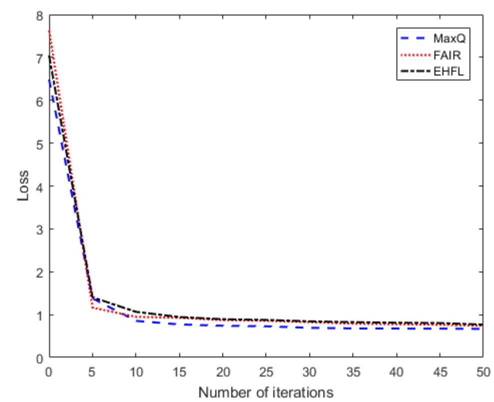

Fig. 3. Loss v.s. Number of Iterations

C. Model Quality Analysis

To verify that the MaxQ mechanism proposed in this paper can obtain higher model quality, Fig. 4 show the relationship between model quality and the number of iterations on the ResNet50 model, respectively. From Fig. 4, it can be seen that the model quality of MaxQ mechanism is higher than FAIR and EHFL, with improve 10.8% and 12.2%, respectively. And the model quality gradually decreases as the number of iterations increases. This is due to the fact that the MaxQ mechanism selects mobile devices with high quality models for each edge server through a matching game. However, FAIR is based on reverse auction in conventional federated learning to select mobile devices with high quality models to participate in model training, and there is no competition among edge servers. As a result, the model quality of FAIR is lower than MaxQ. In EHFL, each edge server selects mobile devices that generate lower energy consumption during training to participate in model training, at which point this causes the selected mobile devices to be trained with fewer datasets in order to reduce energy consumption, resulting in low model quality. However, the model quality of EFHL is slightly lower than that of FAIR due to the limitations of conventional federated learning.The definition of model quality is related to the loss function of the model training and the quatity of data involved in the training. As the number of iterations increases, the amount of change in the loss function due to model training gradually decreases, while the quatity of data is constant. Therefore, the model quality gradually decreases.

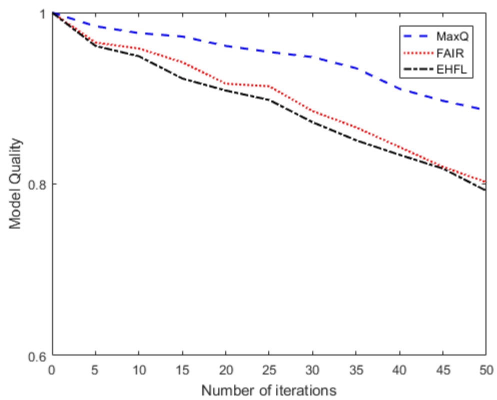

Fig. 4. Model Quality v.s. Number of Iterations

VI. Conclusion

In hierarchical federated learning(HFL), since the global model quality is influenced by the local model quality. To maximize the global model quality, this paper transformed the aggregated model quality maximization into the sum of local model quality maximization. This paper proposed the model quality maximization mechanism MaxQ based on matching game to select high quality mobile devices. In MaxQ, the allocation of mobile devices to each edge server is realized so that the sum of the local model quality is maximized. And we proved that MaxQ has a $\frac{1}{2}-$approximation ratio. Finally, through extensive simulation experiments, MaxQ is able to converge faster and improve the model quality by 10.8% and 12.2% compared with FAIR and EHFL, respectively. In the future work, we will consider the cloud-edge-client hierarchical incentive mechanism design in HFL.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (61872044), the Beijing Municipal Program for Top Talent, the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research.

References

[1] "Cisco annual internet report, 2018–2023, white paper." https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] "Ericsson mobility report, june 2021." https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[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: Springer International Publishing, 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. Surv.*, vol. 52, no. 6, oct 2019.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2017.

[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 learning," 2020.

[7] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," 2019.

[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 *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 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, 2022.

[10] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020.

[11] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[12] Y. Zou, F. Shen, F. Yan, J. Lin, and Y. Qiu, "Reputation-based regional federated learning for knowledge trading in blockchain-enhanced iov," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[13] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3975–3986, 2021.

[14] 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 *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[15] R. Saha, S. Misra, and P. K. Deb, "Fogfl: Fog-assisted federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8456–8463, 2021.

[16] K. Bando, "Many-to-one matching markets with externalities among firms," *Journal of Mathematical Economics*, vol. 48, no. 1, pp. 14–20, 2012.

[17] M. S. H. Abad, E. Ozfatura, D. GUndUz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8866–8870.

[18] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Local averaging helps: Hierarchical federated learning and convergence analysis," 2021.

[19] B. Xu, W. Xia, J. Zhang, X. Sun, and H. Zhu, "Dynamic client association for energy-aware hierarchical federated learning," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[20] M. Chen, Y. Liu, W. Shen, Y. Shen, P. Tang, and Q. Yang, "Mechanism design for multi-party machine learning," 2020.

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

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

[23] 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 *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[24] 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 of Things Journal*, vol. 7, no. 10, pp. 9575–9588, 2020.