An Incentive Mechanism for Big Data Trading in End-Edge-Cloud Hierarchical Federated Learning

Yunfeng Zhao*, Zhicheng Liu*, Chao Qiu*, Xiaofei Wang*, F. Richard Yu†, Victor C. M. Leung‡
*College of Intelligence and Computing, Tianjin University, Tianjin, China
†School of Information Technology, Carleton University, Ottawa, ON, Canada
‡College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
Email: {yfzhao97, liuzhicheng, chao.qiu, xiaofeiwang}@tju.edu.cn
richard.yu@carleton.ca, vleung@ieee.org

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685514

***Abstract*—As a compelling collaborative machine learning framework in the big data era, federated learning allows multiple participants to jointly train a model without revealing their private data. To further leverage the ubiquitous resources in end-edge-cloud systems, hierarchical federated learning (HFL) focuses on the layered feature to relieve the excessive communication overhead and the risk of data leakage. For end devices are often considered as self-interested and reluctant to join in model training, encouraging them to participate becomes an emerging and challenging issue, which deeply impacts training performance and has not been well considered yet. This paper proposes an incentive mechanism for HFL in end-edge-cloud systems, which motivates end devices to contribute data for model training. The hierarchical training process in end-edge-cloud systems is modeled as a multi-layer Stackelberg game where sub-games are interconnected through the utility functions. We derive the Nash equilibrium strategies and closed-form solutions to guide players. Due to fully grasping the inner interest relationship among players, the proposed mechanism could exchange the low costs for the high model performance. Simulations demonstrate the effectiveness of the proposed mechanism and reveal stakeholder's dependencies on the allocation of data resources.**

***Index Terms*—End-Edge-Cloud Systems, Federated Learning, Incentive Mechanism, Data Trading, Game Theory**

I. Introduction

With the rapid development of the Internet of Things and its network applications, massive data will be generated every second. In recent years, big data contains rich information and knowledge of massive data resources, which has gradually attracted the attention of researchers. As an indispensable universal service for big data processing [1], deep learning has completely revolutionized many application fields, such as image processing, video analysis and natural language processing. The surging trend of deep learning stems from a huge amount of training data and massive computation capacity. However, the training data are generated by distributed end devices (a.k.a. end workers) owned by different entities. If such data are disclosed or used for purposes other than those initially intended, the individuals' privacy will be compromised. With this in mind, many data owners are reluctant to share their private data for the training model. Moreover, it becomes challenging to aggregate these data to a single computing node for centralized training due to transmission constraints. The concept of federated learning (FL) [2] has been proposed to alleviate these issues.

Unfortunately, communication overhead remains a bottleneck in a large-scale FL scenario. Hundreds to thousands of rounds of communication are required to reach the desired model accuracy. The long propagation latency will degrade model performance if with a limited training time budget. Directly communicating with the parameter server also brings inestimable big data transmission and saturates the backbone network. This means that the existing cloud-based and edge-based FLs have some limitations. Therefore, hierarchical federated learning (HFL) is proposed to solve these problems. Liu *et al.* [3] proposed the HFL architecture to reduce the communication rounds between users and the cloud by introducing the mobile edge computing platform as an intermediary structure. Abad *et al.* [4] employed gradient sparsification and periodic averaging to increase the communication efficiency of this HFL framework.

Compared with cloud-based FL, end-edge-cloud HFL will significantly reduce the costly communication supplemented by efficient end-edge updates, resulting in a significant reduction in both the runtime and the number of local iterations. However, existing studies have an optimistic assumption that all the end devices participate in training and contribute their all resource unconditionally [5], [6]. Without well-designed financial compensation, the self-interested end devices are reluctant to participate in model training [7], [8]. Therefore, it is necessary to develop an effective approach for stimulating end devices to participate in model training [9], [10]. Several works have studied the incentive design for FL [11], [12], but none considers the end-edge-cloud HFL. To the best of our knowledge, **we are the first to exploit a multi-layer game-theoretical model with an incentive mechanism to explore the HFL in end-edge-cloud systems**.

This paper addresses how to design an incentive mechanism for HFL in end-edge-cloud systems to encourage the end devices to participate in model training and have efficient performance. As illustrated in Fig. 1, we consider a typical end-edge-cloud hierarchical architecture. Particularly, edge servers act as **intermediaries** between the cloud and end devices. The cloud and edge servers motivate end devices to join in training tasks by selecting different payments to maximize the utilities. We map the hierarchical structure into sub-games that are interconnected by utility functions. Players are guided through the use of the multi-layer game and the method of reverse Stackelberg game theory. We provide the unique Nash equilibrium strategy and closed-form solutions to maximize the expected profit and obtain the expected data resources.

The main contributions of this paper are listed as follows:

- We propose an incentive mechanism for HFL in end-edge-cloud systems. It maps the hierarchical training process in end-edge-cloud systems into sub-games interconnected through utility functions that describe realistic price responses at each layer.
- We introduce a multi-layer game and the reverse Stackelberg game-theoretic approach to analyze the game process. We derive the Nash equilibrium and the closed-form solutions and guide players to obtain the desired data resources and profit effectively.
- The simulations demonstrate that the proposed mechanism can effectively reduce the costs and improve the model performance.

The remainder of this article is organized as follows. Section II presents the system model. In Section III, we present the proposed incentive mechanism based on the game theory. Extensive performance evaluations are presented in Section IV. Section V concludes this paper.

II. System model

We consider the HFL framework in end-edge-cloud systems with one cloud and L edge servers. The cloud publishes a training task and wants to improve the model performance by motivating a set $\mathcal{N} = \{1, 2, 3, ..., N\} = \cup_{i=1}^{L} S_l$ of end devices to participate in the model training, where S_l is the set of end devices connected to the edge sever l. We assume that these end devices are owned by different users. Each end device has a local dataset and contributes them to gain the payment of the training model, while still suffering from training time and energy consumption.

Considering the cost and payment of training, each end device determines how much data budget contributes to the training task and submits its plan to the edge server. After receiving all the data budget plans, the edge server computes the payment for end devices and sends the results to the cloud. The cloud will send the payment for edge servers, and then edge servers will pay for the end devices. The chosen end devices (whose payment is positive) will conduct the HFL training process. The process mentioned above completes the whole process of data trading for the training task. Next, we specify the detail of the key steps.

Task description: The cloud publishes the description of a training task, including 1) the type of this task, e.g., image classification task; 2) the relationship between the amount of contributed data and model performance; 3) the reward of data per unit ($P > 0$) correlated with the performance.

Payment allocation: The edge servers will receive their reward from the cloud according to the amount of training data for the end devices it connects to. Each edge server computes

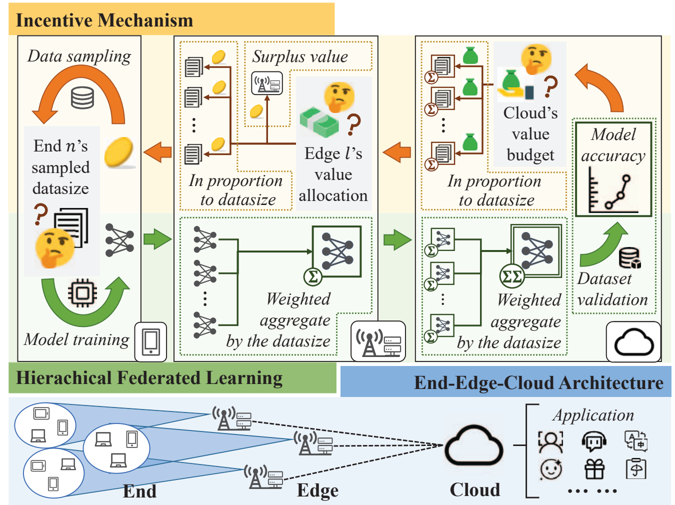


Fig. 1. An illustration of the proposed incentive mechanism for hierarchical federated learning in end-edge-cloud systems.

the payment of connected end devices according to the reward from the cloud and the amount of their training data. In the rest of this paper, we utilize the data compensation function to describe assigning payments to end devices proportionally to their data budgets, i.e.,

$$\text{payment}_n = \frac{x_n}{\sum_{i \in S_l} x_i} R_l,$$

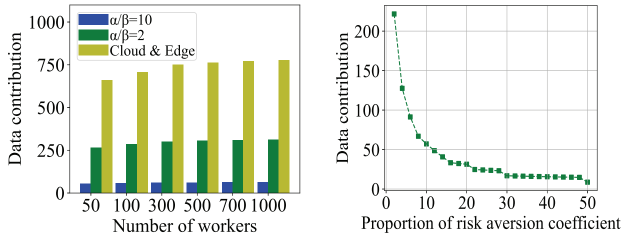
where R_l is the total payment of edge server l allocated to end devices connected to it. On this basis, **risk preference is introduced innovatively**, and the details will be shown in the next section.

Model training: After each end device received its reward, it will start the training process under the coordination of the edge server. Each end device maintains a data set $\mathcal{X}_n$ and wants to collaboratively learn a global model ω using their data. Specifically, the global model ω is learned by minimizing the overall empirical risk of the loss $F(\omega)$ on the union of all local datasets. After every τ local updates on each end device, each edge server aggregates its end devices' models. For every σ edge model aggregations, the cloud aggregates all the edge servers' models, which means that the communication with the cloud happens every $\tau\sigma$ local updates. Denote $\omega_n(t)$ as the local model parameters after the t-th local update. The evolution of local model parameters $\omega_n(t)$ is as follows:

$$\omega_n(t) = \begin{cases} \omega_n(t-1) - \eta_t \nabla F_n(\omega_n(t-1)), & t|\tau \neq 0, \\ \dfrac{\sum_{n \in S_l} x_n [\omega_n(t-1) - \eta_t \nabla F_n(\omega_n(t-1))]}{\sum_{n \in S_l} x_n}, & \begin{matrix} t|\tau = 0 \\ t|\tau\sigma \neq 0 \end{matrix}, \\ \dfrac{\sum_{n \in \mathcal{N}} x_n [\omega_n(t-1) - \eta_t \nabla F_n(\omega_n(t-1))]}{\sum_{n \in \mathcal{N}} x_n}, & t|\tau\sigma = 0. \end{cases}$$

A. Utility of End Device

Before starting HFL training, end devices receive a total reward R_l from the edge server l to motivate them. Each end device $n \in \mathcal{N}$ decides whether to participate in the training and how much data to contribute. If $x_n = 0$, end device n



(a) Number of end devices (b) Proportion of risk aversion coefficient

Fig. 2. The dependence of the total data contribution on the number of end devices and risk aversion coefficient.

does not participate in the model training during this round. Each end device's goal is to determine the best amount of data contribution x_n to maximize its utility. Computational and communication costs are incurred when participating in HFL training. Denote the training costs of end device n as a function of $c^E(c_n^{cmp}, c_n^{com}, x_n)$, where c_n^{cmp} and c_n^{com} are unit computational cost and communication cost, respectively. The utility of end device n can be modeled as:

$$\begin{aligned} &\textbf{P1:} \quad \arg\max_{x_n} U_n^E(x_n, x_{-n}), \ \forall n \in S_l, \ l = 1, ..., L, \\ &s.t. \quad U_n^E(x_n, x_{-n}) = \frac{x_n}{\sum_{i \in S_l} x_i} R_l - \tau\sigma \cdot c^E(c_n^{cmp}, c_n^{com}, x_n), \\ &\qquad 0 \le x_n \le |\mathcal{X}_n|, \end{aligned} \tag{1}$$

where $|\cdot|$ denotes the cardinality of a set. Note that the training costs are proportional to computational and communication costs. Natural choices of such training costs function can be linear functions, exponential functions or others. In this paper, we assume all end devices use the linear training costs function, which takes the form $c^E(c_n^{cmp}, c_n^{com}, x_n) = (c_n^{cmp} + c_n^{com})x_n$.

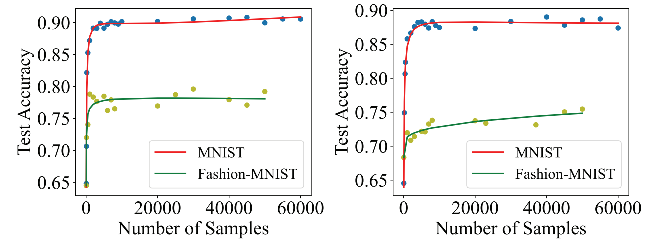
B. Utility of Edge Server

Before starting HFL training, the cloud announces a unit reward P to motivate. Each edge server l will decide the total payment allocated to the end devices to which it connects. Its goal is to determine the optimal total reward R_l that maximizes its utility. Like end devices, computational and communication costs are also incurred when participating in training and have the form as $c^S(c_l^{cmp}, c_l^{com}, S_l')$, where c_l^{cmp} and c_l^{com} are unit computational cost and communication cost, respectively. $S_l' \subseteq S_l$ is the set of end devices that participate in. For the edge server l, the utility is defined as a risk-averse model:

$$\begin{aligned} &\textbf{P2:} \quad \arg\max_{R_l} U_l^S(R_l), \ \forall l = 1, ..., L, \\ &s.t. \quad U_l^S(R_l) = \text{In}(\alpha_l + \sum_{i \in S_l'} x_i^{ne} P) - \frac{R_l}{\beta_l} \\ &\qquad - \sigma \cdot c^S(c_l^{cmp}, c_l^{com}, S_l'), \ R_l \ge 0. \end{aligned} \tag{2}$$

where α_l is a risk aversion parameter (which the edge server sets to reflect its costs), and β_l is a reward scaling coefficient. The costs are proportional to the number of the end devices connected, so we assume all edge servers use the linear costs function like end devices, namely $c^S(c_l^{cmp}, c_l^{com}, S_l') = (c_l^{cmp} + c_l^{com})|S_l'|$.

In the cloud-edge system, an edge server is used as an intermediary to assist the cloud in completing the training task. Generally, the edge server can be a service from a third party, which does not belong to the same alliance with the cloud. In this situation, the edge server may be relatively rational and complete certain tasks according to the reward. However, the edge server can also be greedy for the reward. The risk aversion coefficient in the system model reflects the edge server's characteristics. As the α_l/β_l goes up, edge servers get greedier and demand higher compensation. In reality, there are also some cases where the cloud will use its own edge servers to complete the task. In this case, edge servers have the same goal as the cloud, which can be reflected by converted their utilities to $\text{In}(\alpha_l + \sum_{i \in S_l'} x_i P) + \frac{R_l}{\beta_l} - \sigma \cdot (c_l^{cmp} + c_l^{com})|S_l'|$. Fig. 2 shows the incentive effect of different edge servers on the amount of data contribution.



(a) IID Data Set (b) Non-IID Data Set

Fig. 3. Using HFL to test accuracy with varying samples on the independent identically distributed (IID), Non-IID MNIST and Fashion-MNIST dataset.

C. Utility of Cloud

For the cloud, P is the unit service price paid by the cloud. The goal of the cloud is to choose the optimal P that maximizes its utility. Its utility relies on the training result, i.e., the performance of the trained model and the payment. We analyzed the impact of the amount of data used for the training task on the expected performance of the training model. Some experiments measure the model accuracy under different amounts of training data and are shown in Fig. 3. We observe that the test accuracy of the training model can be regarded as a concave function with respect to the amount of training data [13]. Therefore, the reward of cloud can be defines as $\lambda \cdot g(\sum_{i \in \mathcal{N}} x_i)$, where $\lambda > 0$ is a system parameter and $g(\sum_{i \in \mathcal{N}} x_i)$ is a concave function with respect to the amount of training data, which is shown as:

$$\begin{aligned} &\textbf{P3:} \quad \arg\max_{P} U^C(P), \\ &s.t. \quad U^C(P) = \lambda \cdot g(\sum_{i \in \mathcal{N}} x_i^{ne}) - \sum_{i \in \mathcal{N}} x_i^{ne} P, \ P \ge 0. \end{aligned} \tag{3}$$

III. An Incentive Mechanism for Hierarchical Federated Learning

The edge servers collaborate between the cloud and multiple end devices. Their goal is to maximize their own utilities. They play a reverse Stackelberg game to determine their payment via the Nash equilibrium [14]. The equilibrium strategies for followers in this Stackelberg game can be

Algorithm 1 The process of incentive mechanism in a round

```
Initial all the strategies as $x_n^0,\ R_l^0\ ,P^0,\ \epsilon,\ k=0$.
while $\forall\ n \in \mathcal{N},\ |x_n^{k+1} - x_n^k| > \epsilon$ do
    Each end device adopts the strategy $x_n^k$ in Theorem 1,
    Cloud adjusts its pricing strategy $P^k$ according to (12),
    Each edge server adjusts strategy $R_l^k$ according to (11),
    $k = k+1$.
end while
Results finally obtained are the equilibrium solutions.
```

defined as the strategies that establish an optimal response to the offers of the leader [15]. The algorithm is shown in Algorithm 1. Assume that the system is in a quasi-static state during a round, which means no end devices join or leave.

Definition 1. ***Nash equilibrium:*** *The strategy profile* $X^{ne} = (x_1^{ne}, x_2^{ne}, ..., x_N^{ne}) = (x_n^{ne}, x_{-n}^{ne})$ *denotes the strategy profile of all end devices and is a Nash equilibrium in the end devices' sub-game if for any end device* n

$$U_n^E(x_n^{ne}, x_{-n}^{ne}) \geq U_n^E(x_n, x_{-n}^{ne}), \tag{4}$$

for any $x_n > 0$.

Given x_{-n}, if a strategy maximizes $U_n^E(x_n, x_{-n})$ over all $x_n > 0$, it is the best strategy of end device n, denoted as $x_n^*(x_{-n})$. To study the best strategy for each end device, we first compute the first-order and second-order derivatives of utility $U_n^E(x_n, x_{-n})$ with respect to x_n:

$$\frac{\partial U_n^E(x_n, x_{-n})}{\partial x_n} = \frac{\sum_{k \in S_l \backslash \{n\}} x_k}{(\sum_{k \in S_l} x_k)^2} \cdot R_l - L_n,$$

where $L_n = \tau\sigma \cdot (c_n^{cmp} + c_n^{com})$ and

$$\frac{\partial^2 U_n^E(x_n, \mathbf{x}_{-n})}{\partial {x_n}^2} = -\frac{2R_l \sum_{k \in S_l \backslash \{n\}} x_k}{(\sum_{k \in S_l} x_k)^3} < 0.$$

Due to the utility $U_n^E(x_n, x_{-n})$ is concave, the best strategy $x_n^*(x_{-n})$ is unique if exists when edge servers' strategy R_l and other end devices' strategy x_{-n} are given. By setting the first-order derivative to zero, the best strategy $x_n^*(x_{-n})$ denotes as:

$$x_n^*(x_{-n}) = \begin{cases} 0, & R_l < L_n \cdot \sum_{i \in S_l \backslash \{n\}} x_i, \\ \sqrt{\frac{R_l \cdot \sum_{i \in S_l \backslash \{n\}} x_i}{L_n}} - \sum_{i \in S_l \backslash \{n\}} x_i, & otherwise. \end{cases} \tag{5}$$

In general, the number of participants is greater than 0. Next, we will discuss how to choose the participants from the following three aspects. First, we discuss the case when the number of participants is 0, namely, $|S_l'| = 0$. Then a non-participating end device can increase its utility from zero to $R_l/2$ if the end device unilaterally changes x_n from zero to $R_l/2L_n$. This contradicts the assumption of Nash equilibrium. So $|S_l'| = 0$ does not hold. Second, $|S_l'| = 1$, a participating end device n has $x_n > 0$ and other end devices' strategies $x_i = 0, i \in S_l \backslash \{n\}$. Then the utility of end device n equals $(R_l - L_n x_n)$. In this situation, end device n can increase the utility by changing its data contribution x_n to $x_n/2$. This contradicts the uniqueness of the Nash equilibrium. So $|S_l'| = 1$ also does not hold. Third, for the situation $|S_l'| \geq 2$, we have the following theorem. Therefore we can obtain the best strategy profile for end devices.

Theorem 1. *For at least two players in the end devices sub-game, they can be sorted in ascending order according to their costs in the set* S_l*, then a set of participants* $S_l' \subseteq S_l$ *can be found, satisfy*

$$L_n < \frac{\sum_{i \in S_l'} L_i}{|S_l'| - 1}, \tag{6}$$

and ensure

$$x_n^{ne}(x_{-n}) = \begin{cases} 0, & n \notin S_l', \\ \frac{R_l(|S_l'|-1)}{\sum_{i \in S_l'} L_i}\left(1 - \frac{L_n(|S_l'|-1)}{\sum_{i \in S_l'} L_i}\right), & n \in S_l'. \end{cases} \tag{7}$$

Proof. The details are shown in the supplement material. [1] □

Further, using the best strategy $x_n^{ne}(x_{-n}),\ n \in S_l'$ provided by end devices, edge servers need to determine the optimal rewards R_l to maximize their own utilities $U_l^S,\ l = 1, .., L$. The best strategy $x_n^{ne},\ n \in S_l'$ in (7) can be rewritten as

$$x_n^{ne} = Y_n R_l, \tag{8}$$

where

$$Y_n = \frac{|S_l'| - 1}{\sum_{i \in S_l'} L_i}\left(1 - \frac{L_n(|S_l'| - 1)}{\sum_{i \in S_l'} L_i}\right). \tag{9}$$

Taking the first-order and second-order derivatives of edge server l's utility $U_l^S(R_l)$ with respect to R_l, we have

$$\frac{\partial U_l^S(R_l)}{\partial R_l} = \frac{\sum_{n \in S_l'} Y_n P}{\alpha_l + R_l \sum_{n \in S_l'} Y_n P} - \frac{1}{\beta_l}, \tag{10}$$

and

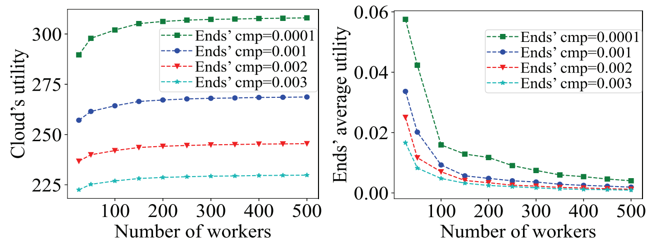
$$\frac{\partial^2 U_l^S(R_l)}{\partial {R_l}^2} = -\frac{(\sum_{n \in S_l'} Y_n P)^2}{(\alpha_l + R_l \sum_{n \in S_l'} Y_n P)^2} < 0.$$

Therefore, the utility of each edge server is strictly concave with respect to R_l for $R_l > 0$. Since the value of $U_l^S(R_l)$ approaches 0 when R_l approaches 0 and goes to $-\infty$ when R_l goes to ∞, there is a unique maximizer R_l^*. By setting the first-order derivative to zero, the best strategy of edge server l can be derived as:

$$R_l^* = \beta_l - \frac{\alpha_l}{\sum_{n \in S_l'} Y_n P}. \tag{11}$$

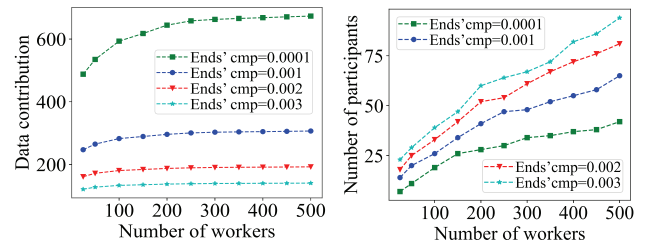
Based on the above analysis, the cloud's strategy P is determined by end devices' strategies $(x_n^{ne},\ n \in S_l')$ and edge servers' optimal strategies $R_l^*,\ l = 1, 2, ..., L$. As the leader of the entire game, the cloud knows that there is Nash equilibrium in the end devices, so it only needs to maximize its utility to find the best strategy P^*. Substituting (8) and

[1] https://github.com/YunfengZhao97/IM-BD-HFL



(a) Cloud's utility (b) Ends' average utility

Fig. 4. The performance of cloud's and end devices' average utilities under different settings of the number of end devices.



(a) Data contribution (b) Number of participants

Fig. 5. The performance of participants and data contribution under different settings of the number of end devices.

(11) into the cloud's utility function $U^C(P)$, the first-order and second-order derivatives of it denote as:

$$\frac{\partial U^C(P)}{\partial P} = \lambda \cdot g'\left(\sum_{n\in\mathcal{N}'}\beta_l Y_n - \frac{\sum_{l=1}^{L}\alpha_l}{P}\right) - \sum_{n\in\mathcal{N}'}\beta_l Y_n,$$

and

$$\frac{\partial^2 U^C(P)}{\partial P^2} = \lambda \cdot g''\left(\sum_{n\in\mathcal{N}'}\beta_l Y_n - \frac{\sum_{l=1}^{L}\alpha_l}{P}\right).$$

In general, $g(\cdot)$ is a concave function whose first-order and second-order derivatives are $g'(\cdot)$ and $g''(\cdot)$, and then we can derive that $\frac{\partial^2 U^C(P)}{\partial P^2} < 0$. Therefore, the utility of the cloud is a strictly concave function of P for $P \in [0, \infty)$. Since the value of $U^C(P)$ is zero when $P = 0$ and goes to $-\infty$ when P goes to ∞, it has a unique maximizer P^*.

For example, if $g(\cdot) = 0.5 \cdot \text{In}(1 + x)$, by setting the first-order derivative to zero, the cloud's best strategy is as follows:

$$P^* = \sqrt{\frac{0.5\lambda\sum_{l=1}^{L}\alpha_l}{m(1+m)} + \frac{\left(\sum_{l=1}^{L}\alpha_l\right)^2}{4(1+m)^2}} + \frac{\sum_{l=1}^{L}\alpha_l}{2(1+m)}, \quad (12)$$

where $m = \sum_{n\in\mathcal{N}'}\beta_l Y_n$ and $\mathcal{N}' = \cup_{i=1}^{L} S_l'$.

IV. Numerical Evaluation

A. Experiment Settings

In this section, some experiments are conducted to evaluate the performance of the proposed incentive mechanism for HFL in the end-edge-cloud systems. Without loss of generality, our experimental settings refer to the setups in [13]. We set $L = 2$ by default and use $g(x) = 0.5 \cdot \ln(1 + x)$ to denote the reward of the cloud in HFL. We assume that the unit computational cost of end devices $c_n^{cmp} = 0.001$ and the communication cost of devices connected to edge servers 1 and 2 are uniformly distributed over $[0.0005, 0.002]$ and $[0.0008, 0.0025]$, respectively. We set the unit computational cost of edge servers $c_l^{cmp} = 0.0001$, $l \in \{1, 2\}$ and the communication cost of edge servers 1 and 2 are uniformly distributed over $[0.005, 0.02]$ and $[0.003, 0.015]$, respectively. For simplicity, we set $\lambda = 100$, $\alpha_l = 5$, $\beta_l = 2.5$, $l \in \{1, 2\}$ and the aggregation frequency $\tau = 2$, $\sigma = 3$, respectively.

B. Performance of Utilities

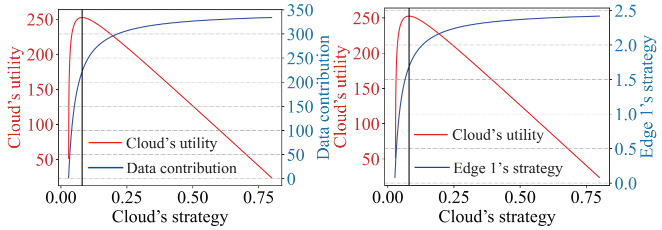
We evaluate the impact of the number of end devices on the cloud's and end devices' average utility. As shown in Fig. 4(a), the utility of the cloud increases as the number of end devices increases. It is expected to see that the influence of the number of end devices on the cloud's utility is diminishing. It is reasonable that in Fig. 4(b), the average utilities of the end devices decreases as the participants become more diverse since more competitions are involved. Meanwhile, we also evaluate the impact of the amount of unit computational cost of end devices on various values. It is reasonable that as the costs of the end devices increase, the utility of end devices decreases and the enthusiasm of contributing more data to participate in training reduces, resulting in a reduction of the utilities of end devices and the cloud. From the density of the curve, their downward trend tends to be slow with the increase of end devices' costs.

C. Performance of Participants and Data Contribution

We evaluate the number of all participants in our mechanism under different settings of all end devices. In Fig. 5(a), we can observe that more end devices are actually participating in, more data contributed to training the model. With the increase of the costs of end devices, the utilities of end devices will decrease, and the enthusiasm for contributing more data to participate in the training will be weakened. Cloud and edge servers also cannot bear too many costs, which leads to a decrease in the amount of data contribution. In Fig. 5(b), we can see that as the number of end devices increases, the number of all participants increases. This is reasonable, attributed to more end devices that will satisfy the condition of belonging to S_l' if more end devices are interested in the training task. With the increase in end devices' costs, more end devices will participate in training. However, the data contribution of each end device and the total amount of data contribution are reduced.

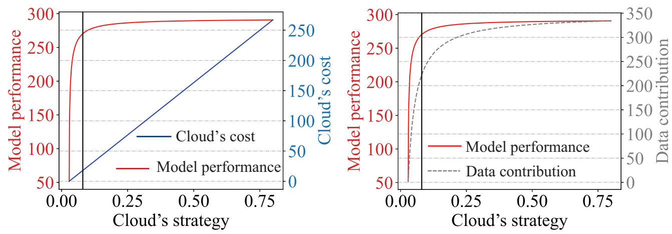
D. Impact of Deviation from Nash Equilibrium

We explore the impact on the total data contribution and the edge servers of changes in the cloud's payment P, fixing R_l at their Nash equilibrium values, as shown in Fig. 6(a). Interestingly, forcing the cloud's cost to rise beyond the Nash equilibrium value only provides a slight increase in data contribution. By contrast, data contributions are much more sensitive to the cloud's cost reductions. Comparing the impact of the cloud's utility function shows that the slight increase of data contribution requires reducing the cloud's reward, while data contribution's decline also sharply decreases it. Fig. 6(b) shows similar behaviors for the impact on the edge servers' strategy. Followers show an elasticity in price response to



(a) Cloud's utility & Data contribution (b) Cloud's utility & Edge 1's strategy

Fig. 6. The impact of the deviation from Nash equilibrium.



(a) Model performance & Cloud's cost (b) Model performance & Data contribution

Fig. 7. The effect of the proposed mechanism on the model's performance.

the leader below the Nash equilibrium, nevertheless strong inelasticity above it.

E. Impact on Model Performance

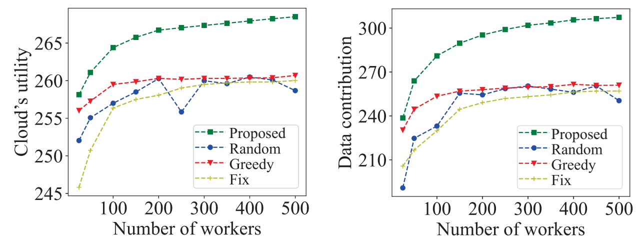
In Fig. 7(a), we investigate the effect of the proposed mechanism on the model's performance. It can be observed that when the cloud's payment increases and exceeds the equilibrium value P^{*}, the cloud's total costs continue to increase. Nevertheless, the performance of the model rises very little and remains almost unchanged. The performance of the model changes obviously when the cloud's payment mild fluctuates below the equilibrium value. Using our proposed mechanism can exchange the lowest costs for the highest model performance. A similar result is also obtained in Fig. 7(b). The least data contribution of end devices can be exchanged for the greatest model performance. While ensuring the model's performance, the data contribution is reduced to the greatest extent, thus reducing model training costs. This can promote the end devices to join in training.

F. Comparison with Baselines

Fig. 8 compares our mechanism with the Random, Greedy and Fix methods for HFL. In the Random method, the parameter server randomly determines the participants in each training round, while in the Greedy method, all the end devices participate in the training model. Only some fixed end devices are considered to participate training model in each training round in the Fix method. In Fig. 8, we can find that our mechanism is much better than baselines, has higher utility, and encourages end devices to contribute more data.

V. Conclusion

This paper has proposed an incentive mechanism for HFL in the end-edge-cloud systems. It can motivate end devices to participate in training tasks and improve the performance of HFL. A multi-layer game-theoretic approach was proposed



(a) Cloud's utility (b) Data contribution

Fig. 8. The comparison with the Random, Greedy and Fix methods under different settings of the number of end devices.

for the decision-making process by expressing utility functions as comprehensive models that described the stakeholders' price responses. We used this approach to guide players, which allowed stakeholder's dependencies on allocating data resources to be articulated in a non-iterative manner. The numerical simulations have demonstrated the effectiveness of our proposed mechanism, which can reduce the training costs and improve the model performance.

References

[1] M. A. Alsheikh, D. Niyato, S. Lin, H. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," *IEEE Network*, vol. 30, no. 3, pp. 22–29, 2016.

[2] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[3] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," *IEEE ICC*, pp. 1–6, 2020.

[4] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," *IEEE ICASSP*, 2020.

[5] H. Kim, J. Park, M. Bennis, and S. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, 2020.

[6] M. Shayan, C. Fung, C. J. Yoon, and I. Beschastnikh, "Biscotti: A ledger for private and secure peer-to-peer machine learning," *arXiv preprint arXiv:1811.09904*, 2018.

[7] Y. Li, C. Courcoubetis, and L. Duan, "Recommending paths: Follow or not follow?" in *IEEE INFOCOM*, 2019, pp. 928–936.

[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," *IEEE INFOCOM*, pp. 1387–1395, 2019.

[9] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L. Wang, "Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 5, pp. 1345–1348, 2019.

[10] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[11] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," *IEEE VTS Asia Pacific Wireless Communications Symposium*, pp. 1–5, 2019.

[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. Wirel. Commun.*, vol. 19, no. 5, pp. 3241–3256, 2020.

[13] 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, 2020.

[14] J. Fattahi, D. Wright, and H. Schriemer, "An energy internet derms platform using a multi-level stackelberg game," *Sustainable Cities and Society*, vol. 60, pp. 102–262, 2020.

[15] G. Mailath and L. Samuelson, *Repeated Games and Reputations: Long-Run Relationships*. Oxford University Press, 2006.