ITC 4/53

Information Technology and Control
Vol. 53 / No. 4/ 2024
pp. 1139-1151
DOI 10.5755/j01.itc.53.4.34907

Robust Incentive Mechanism of Federated Learning for Data Quality Uncertainty

Received 2023/08/22 | Accepted after revision 2023/10/26

HOW TO CITE: Wang, C., Li, B., Yang, Y. (2024). Robust Incentive Mechanism of Federated Learning for Data Quality Uncertainty. *Information Technology and Control,* 53(4), 1139-1151. https://doi.org/10.5755/j01.itc.53.4.34907

Robust Incentive Mechanism of Federated Learning for Data Quality Uncertainty

Chao Wang

School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China

Bingze Li, Yang Yang

School of Management Engineering and Business, Hebei University of Engineering, Handan 056038, China

Corresponding author: li.bingz@outlook.com

In order to make the incentive mechanism more suitable for the actual training situation and improve the efficiency of the model, the robust incentive mechanism of federated leaning is proposed to deal with uncertainty of the data quality. (1) Firstly, the incentive mechanism of federated learning is constructed by the use of Stackelberg game to optimize the central server and data owner utilities, respectively. (2) Secondly, the uncertainty of data quality of the data owners is present by two robust uncertainty sets, and the corresponding incentive mechanism of the robust Stackelberg game is given. (3) Thirdly, the existence of equilibrium solution of the game is proved and the equilibrium solution of the whole game is derived. (4) Finally, the feasibility and robustness of the model are verified, and in the comparative experiments, the central server can select the optimal combination of perturbation ratio and uncertainty level according to the preference for uncertainty risk to obtain the optimal incentive mechanism. The incentive mechanism designed in this article not only considers the uncertainty in actual training, but also has a good incentive effect on model training under different risk preferences.

KEYWORDS: Federated learning; Stackelberg game; Robust uncertainty sets; Nash equilibrium; Data quality uncertainty.

1. Introduction

Over the past few years, machine learning (ML) has expanded into a variety of applications [19, 31]. In most applications, the training data for machine learning is obtained by aggregating datasets, and the performance of a machine learning model is heavily influenced by the size and quality of the dataset used for training. Nonetheless, many valuable data sets are often privately held and spread across different individuals or institutions. These data owners are often unwilling to share their data, leading to the formation of isolated data islands. Federated learning is introduced as a novel approach to distributed machine learning [25], and multiple data owners of federated learning can cooperate to use their private data sets to train the same learning model provided by the center server, while ensuring the privacy of these data sets. Hence, federated learning is recognized as a powerful machine learning paradigm that has garnered Signiant attention for its potential to overcome the limitations of "Isolated Data Islands." Recently, federated learning mainly focusses on privacy protection [21, 22], wireless networks [3, 26] and algorithm improvement [1].

As we all known, the participants of federated learning model training are composed of the task publisher and multiple data owners [20]. Because of the difference in data quality of the data owners, the data owners are unwilling to put themselves in an unfair position when participating in federated learning [7]. Then, the incentive mechanism is introduced as an important means to improve the willingness of data owners to participate in federated learning.

In the design of incentive mechanism of federated learning, the data owner wants to get more rewards, while the task publisher wants to pay less budget with higher training utility in the design of incentive mechanism of federated learning. Then this is a game behavior between the task publisher and the data owners, and there have been many researches on the incentive mechanism designed by game theory. For example, Khan et al. [14] proposed a Stackelberg game approach that allows data owners to strategically set the number of local iterations to maximize its utility, whose results prove that this approach is effective in modeling the interaction between the task publisher and edge device; Hu et al. [11] used a two-stage Stackelberg game approach to obtain the utility maximization strategy between the server and the user by solving the Stackelberg equilibrium; Zhan et al. [29] analyzed the uniqueness of the two phases of Stackelberg equilibrium and Nash equilibrium in the Stackelberg game, effectively solved the problem of how incentives affect the utility of task publishers, and proposed an incentive mechanism for solving the non DRL incentive mechanism faced by shared information. The comparison between this paper and the existing federated learning incentive mechanism is shown in Table 1.

At present, research on federated learning has entered the application stage, such as medical imaging [6], the Internet of Things [12], intelligent traffic control [9], etc. Its real incentive effect and contract always has errors. This is because there are many factors in the training of federated learning models that cannot be determined by both parties, such as the quality of data provided by data owners, noise in parameter transmission, and the gap in model training iteration cycles. These are particularly important in the practical training of federated learning, but currently there are few considerations, which also pose significant obstacles to the application of incentive mechanisms in federated learning. Moreover, due to the randomness [17] and unpre- dictability of these parameters, obtaining accurate probability distributions is difficult, and robust optimization is very effective in solving such parameter uncertainty problems [10, 24]. It can adjust the robustness of the model by setting uncertainty levels based on the decision-maker's risk pref-

Table 1

Literature review table of federated learning incentive mechanism

References	Critical technology	Uncertainty handing	Sub-problem
Ng [16]	Contract theory	Not consider	Model training accuracy
Chen [4]	Multi-dimensional contract theory	Not consider	Model training budget
Kang [13]	Reputation+Contract theory	Not consider	Model utility
Zhang [30]	Reinforcement Learning	Not consider	Model training accuracy
Sarikaya [28]	Stackelberg game	Not consider	Model performance
This paper	Stackelberg game	Consider	Model utility

erence [2]. Therefore, considering the use of robust optimization to characterize parameter uncertainty, introducing different sets of uncertainty into the incentive mechanism of Stackelberg games can not only ensure the robustness of federated learning training, but also effectively stimulate the training effect. Our key contributions include:

1 We use the Stackelberg game to design a new incentive mechanism for asynchronous federated learning, which aims to enhance the model's training rate while motivating data owners to enhance their efficiency.

2 We establish the existence and uniqueness of the equilibrium solution for the Stackelberg game, validating the effectiveness of the incentive mechanism we have devised.

By introducing the box uncertainty sets and polyhedral uncertainty sets to characterize data quality uncertainty, the new game models are constructed and equilibrium solutions are derived separately.

Simulation experiments show that the robust Stackelberg game enables the task publisher to choose the appropriate uncertainty level and perturbation ratio according to its risk preference, thus obtaining more appropriated equilibrium solutions.

The rest of the paper is organized as follows: Section 2 presents the federated learning system model, outlining its key components. In Section 3, we develop the federated learning incentive mechanism using the Stackelberg game, and provide a comprehensive proof for the equilibrium solution of the mechanism. Section 4 constructs the robust Stackelberg games with different uncertainty sets and gives the equilibrium solution method. Section 5 presents the analysis of experimental results, while Section 6 derives conclusions based on the findings.

2. System Model

A typical federated learning system consists of two entities, including the federated learning center server and the data owners, as shown in Figure 1. Each individual data owner possesses their own distinct private data, which is stored locally. Each data owner, following the guidelines set by the central server, utilizes their local data to train a model. Upon the completion of local model training, the data owner trans- mits the trained model parameters back to the central server within the federated learning system. The central server aggregates and integrates these updated parameters to update the global model. Afterwards, the central server communicates new iteration requirements to the data owners. This iterative process persists until the de- sired performance level is achieved or a predefined number of iterations, as determined by the central server, is reached:

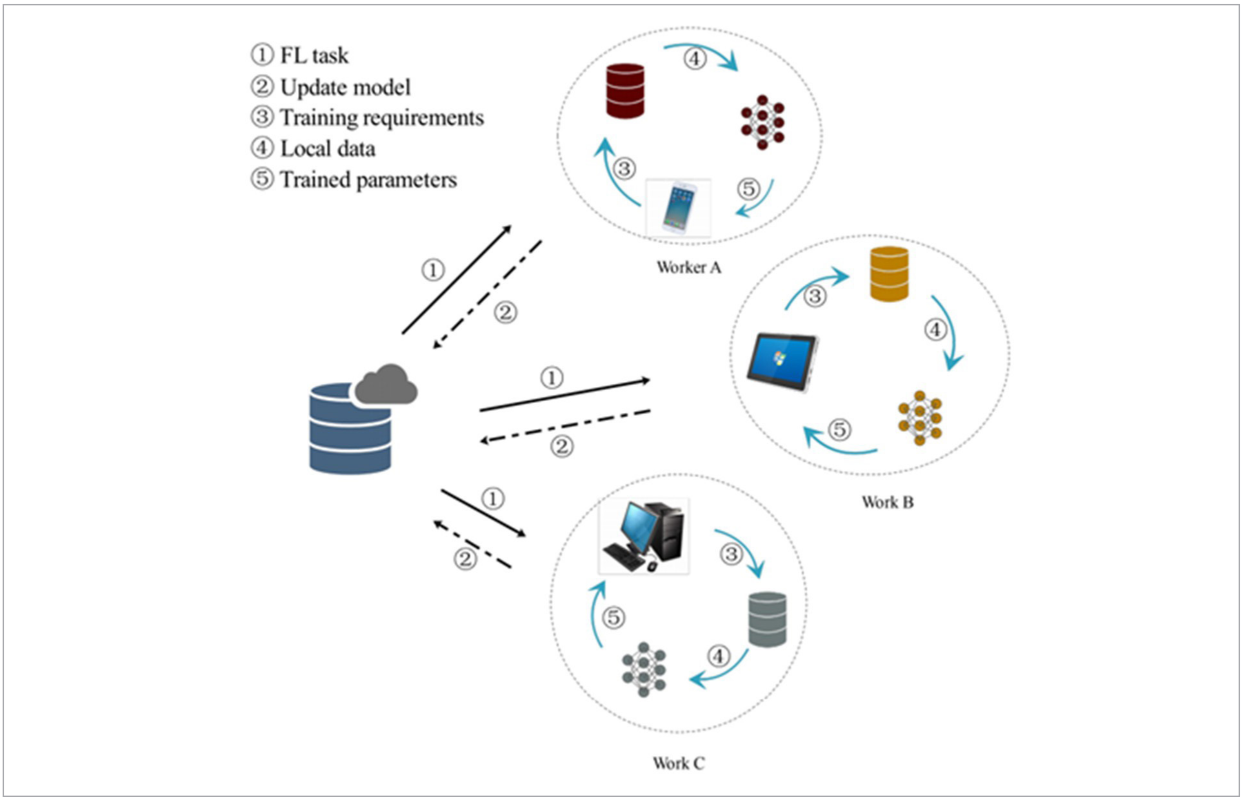
Synchronous stochastic gradient descent (SGD) is considered in most federated learning, but high latency updates can affect global parameters, which in turn leads to less efficient algorithm operation. The federated learning scenario being discussed here pertains to an asynchronous federated learning framework, which involves a task publisher and multiple data owners [5, 15]. In this scenario, it is assumed that every data owner n ∈ N has an identical local data sample size denoted as s for their participation in federated learning. For each data owner, represented as n, they utilize a distinct CPU cycle frequency denoted as fn during the training of their local model. The number of CPU cycles needed to complete one round of data training is denoted as cn. Therefore, the computation time required for local model training for data owner n can be calculated as Tn = scn/fn. Furthermore, the CPU energy consumed by the data owner in completing a local data training can be given by the equation:

$$D_n(f_n) = \varepsilon s c_n f_n^2. \quad (1)$$

In the context of federated learning, each data owner is tasked with updating the global model by training their local data and contributing the updated parameters. The training efficiency of the data owners is influenced by the quality and accuracy of their local data, denoted as λn. The difference in data quality is mainly reflected in reliability and accuracy. Specifically, when the data quality is poor, federated learning needs to run more global iterations to achieve a certain model training accuracy. The lower bound on the number of iterations is usually used in federated learning algorithms log(1/λn) to represent local training data [30], for computational convenience, most applicants make it straightforward to denote the number of local training iterations. Based on this premise, the computation time for model updates during global iterations can be expressed as $T_n^c = \log(1/\lambda_n)T_n$.

Figure 1

Federated learning work flow



During the process of data transmission, it is common for the downlink bandwidth to be significantly greater than the uplink bandwidth. Thus, the downlink transmission time between the task publisher and the data owner can be deemed insignificant when compared to the uplink time. The total time spent during the global iteration is thus the sum of the iterative computation time and the uplink communication time. In terms of the communication time for the model updates [8], the transmission rate of the data owners can be expressed as rn = B ln (1 + (ρnhn/N0)), where B represents the transmission bandwidth, 7n is the transmission power of the data owner, and hn is the channel gain between the data owner and the task publisher. N0 denotes the background noise. With a fixed sample data size, the transmission time for the local model update is given by $T_n^t = s/(B\ln(1+\rho_n h_n/N_0))$. Therefore, the total time required for a global model iteration can be calculated as:

$$T_n^{total} = \log(1/\lambda_n)T_n + T_n^t. \quad (2)$$

According to [27], the energy consumption of data owner n for transmitting updated parameters in a global iteration can be expressed as Dt/N0)). The total energy consumption of the data owner n in a global iteration can be calculated as:

$$D_n^{total} = \log(1/\lambda_n)D_n + D_n^t. \quad (3)$$

Due to the information asymmetry between the two parties involved in federated learning, task publishers offer different data quality and CPU frequency for federated learning model training to generate profits. The task publisher, in turn, provides pay- offs based on the different payoff levels of the data owners. Assume that the data owner receives paid as Rn=qnfn, where qn represents the price per unit for the data owner to utilize the CPU frequency fn. It is important to highlight that a greater contribution of computational resources by the data owner results in a faster training of the local model, leading to a higher payoff. The data owner can choose to sign the contract to complete the model training, but if the corresponding workload in the contract is not completed, the data owner cannot get the predetermined payment.

As noted above, the data owners share the same size of sample data to process and the task publisher has the same accuracy requirements for model training. In this case, maximizing the federated learning utility for the task publisher is to improve the efficiency of the data owner. This paper uses game theory to characterize the incentive mechanism, mainly considering the interaction between the task publisher and each data owner. Due to the heterogeneity among the data owners in asynchronous federated learning, the development of an incentive mechanism becomes necessary. In this context, we define the utility of the task publisher as the duration of the global iteration:

$$U_T = \log(1/\lambda_n)T_n + T_n^t . \qquad (4)$$

Data owner n considers the energy loss associated with their participation in the learning process, which is determined by the CPU power level, upon receiving the corresponding payment Rn from the task publisher. The objective of each data owner is to maximize their profit:

$$\begin{aligned} U_D &= R_n - \mu D_n^{total} \\ &= q_n f_n - \mu(\log(1/\lambda_n)D_n + D_n^t). \end{aligned} \qquad (5)$$

The equation provided earlier is subject to the constraint fn ≤ fmax, where fmax represents the upper limit of CPU power of the data owner. Additionally, μ is a predefined weight parameter that governs the influence of energy consumption.

3. Incentive Mechanism of Federated Learning by Stackelberg Game

As we all known, data owners in federated learning use private data to participate in the model training initiated by the task publisher. The task publisher ranks the data owners according to their data quality, so as to facilitate reasonable incentives for the data owners. To ensure effective incentives for participants with varying data quality, the CPU power of the data owners is bounded by the offered reward, which is capped. Therefore, the data owners need to optimize their CPU power to balance the cost and the benefit. The task publisher hopes to improve model iteration efficiency or reduce model iteration time by providing rewards to the data owners. Therefore, a Stackelberg game with two-level structure is adopted to jointly optimize the utility of the upper and lower games. The Stackelberg game constructed in this paper is an interaction between task publish and multiple data owner. When the task publishers first make decisions as leaders, and data owners as followers will make their own decisions based on the task publisher's decisions. In this process, the task publisher and data owners will adjust to each other's decision until both can maximize their benefits. Finally, the compensation provided by the mission publisher is the corresponding relationship between the data provided by the data owner and the computing power.

When the quality of the data provided by the data owner is determined, a lower bound on the number of iterations it can participate in the learning process can be determined by gradient descent. For computational convenience, this is often analyzed directly as the number of iterations of the data owner. And it is assumed that the iter- ation time of the data owner satisfies the maximum duration requirement of parameter aggregation.

– Lower-level game model:

At the unit price qn for the CPU frequency of each data owner, the Lower-level subgame focuses on maximizing the profit of data owners, and the problem is defined as follows:

$$\begin{aligned} \max_{f_n} \quad & U_D = q_n f_n - \mu D_n^{total} \\ \text{s.t.} \quad & f_n \le f_{\max} \end{aligned} \qquad (6)$$

– Upper-level Subgame:

Once the relationship between the CPU frequency and the unit price of CPU usage for data owners during the learning process is established, the upper-level game focuses on minimizing the iteration time. The problem is then formulated as follows:

$$\begin{aligned} \min_{q_n} \quad & U_T = \log(1/\lambda_n)T_n + T_n^t \\ \text{s.t.} \quad & q_n f_n \le R_{\max} \end{aligned} , \qquad (7)$$

where Rmax is the maximum unit price of CPU power that the task publisher can provide.

In general, the equilibrium of the Stackelberg game is achieved by finding the optimal Nash equilibrium. In the game described in this paper, the unit price of CPU power for the data owners is predetermined. The data owners engage in a non-cooperative game environment, where the Nash equilibrium is defined as a state in which no player can improve their payoff by unilaterally changing their strategy.

In the game described in this paper, the number of participants is limited, and the optimal unit price of CPU power, as set by each central server, is restricted to a bounded closed set in the Euclidean space. The utility function of the lower-level subgame exhibits continuous variation with respect to the independent variables, and the profit function UD of the lower-level subgame displays concave properties, and its first-order derivative and second-order partial derivative with respect to CPU power can be expressed using the following estimation equation:

$$\begin{aligned}\frac{\partial U_D}{\partial f_n} &= \frac{\partial[q_n f_n - \log(1/\lambda_n)\mu\varepsilon c_n f_n^2 - D_n^t]}{\partial f_n}\\ &= q_n - 2\log(1/\lambda_n)\mu\varepsilon c_n f_n \qquad (8)\\ \frac{\partial^2 U_D}{\partial f_n^2} &= -2\log(1/\lambda_n)\mu\varepsilon c_n\end{aligned}$$

The solution demonstrates that the second-order partial derivative is negative, indicating that the profit function UD exhibits a strict concave property. Consequently, the subgame Nash equilibrium solution exists the Stackelberg game model [18].

In the game model described above, our objective is to identify the Stackelberg equilibrium solution for both the task publisher and the data owner. Considering the situation where data quality is unknown in the context of the Stackelberg game, we determine the equilibrium solutions for the upper and lower games using the backward induction method. The process starts by identifying the equilibrium solution for the lower-level subgame based on the first-order optimality condition. Subsequently, the lower equilibrium solution is incorporated into the upper-level subgame to derive the overall game solution.

4. Robust Incentive Mechanism for Stackelberg Games with Data Quality Uncertainty

In the process of contract selection and participation of data owners in learning based on their data quality, the data quality is usually inferred by the data owner from limited data and provided to the task publisher. However, the real value of data quality is influenced by various factors and cannot be accurately derived. Therefore, the impact of this uncertainty on the effectiveness of the federated learning incentives can be reduced by applying the idea of robust optimization so that the real value of data quality is disturbed within a certain range around the nominal value provided by the data owner. The data quality is mainly reflected in the number of iterations in the upper and lower games, i.e:

$$\log(1/\lambda_n) = \frac{\varphi}{\omega_n}, \qquad (9)$$

where the data quality ωn is uncertainty and ϕ is a deterministic parameter for the number of iterations influenced by the data quality. Larger ωn implies better data quality and higher accuracy and data reliability [23], which can reduce the number of local iterations for model training. In order to facilitate the solving processing the data quality, we assume

$$\frac{\varphi}{\omega_n} = \sigma_n, \qquad (10)$$

where σn is data quality parameter.

4.1. Robust Incentive Mechanisms for Stackelberg Games Under Boxed Uncertainty in Data Quality

We first consider the worst-case uncertainty model, the introduction of a boxed un- certainty set, where the data quality parameters σn uncertainty variables are assumed to be a given uncertainty set. The boxed uncertainty set is introduced to model the data quality uncertainty parameters. Then

_en _ [_n ◎ _bn; _n + _bn],

where _n is the data quality parameter in the nominal model and bn is its perturbation. When the data quality is a boxed uncertainty set, the lower-level subgame is:

$$\max_{f_n} U_D = q_n f_n - (\sigma_n + \sigma_n)\mu\varepsilon s c_n f_n^2 + s\rho_n / (B\ln(1+\rho_n h_n \qquad (11)$$

$$\text{s.t.} \quad f_n \le f_{\max}$$

The upper-level subgame is:

$$\min_{q_n} U_T = (\overline{\sigma}_n + \widehat{\sigma}_n)(sc_n / f_n) + s/(B\ln(1+(\rho_n \mathrm{h}_n / \mathrm{N}_0))) \qquad (12)$$

$$\text{s.t.} \quad q_n f_n \le R_{\max}$$

In the constructed Stackelberg game with data quality uncertainty as mentioned above, backward induction is employed to determine the equilibrium solutions for both the upper and lower games. Initially, the equilibrium solution for the lower game is determined by considering the first-order optimality condition. Subsequently, the lower equilibrium solution is incorporated into the upper game to find the overall solution of the game.

The lower game solving:

To find the equilibrium solution of the data owner in the lower-level game, the first order derivative of the profit function with respect to the CPU power fn in the lower game can be specified as follows:

$$\begin{aligned}\frac{\partial U_D}{\partial f_n} &= \frac{\partial(q_n f_n - (\overline{\sigma}_n + \widehat{\sigma}_n)\mu\varepsilon s c_n f_n^2 + s\rho_n / (B\ln(1+\rho_n h_n / N_0)))}{\partial f_n} \\ &= q_n - 2(\overline{\sigma}_n + \widehat{\sigma}_n)\mu\varepsilon s c_n f_n\end{aligned} \qquad (13)$$

If the above equation is zero, the CPU power of the data owner is as follows:

$$f_n^* = \begin{cases} \dfrac{q_n}{2\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)} & if \ \dfrac{q_n}{2\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)} \le f_{\max} \\ f_{\max} & otherwise \end{cases} \qquad (14)$$

The upper game solving:

Once an equilibrium solution is reached in the lower-level game, the optimal CPU power of the data owner is substituted into the utility maximization problem of the upper-level game. As the constraints of the upper-level game problem are linear, the Lagrangian function can be expressed as follows:

$$L(q,\alpha) = (\overline{\sigma}_n + \widehat{\sigma}_n)T_n + T_n^t + \alpha\left(\frac{q_n^2}{2\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)} - R_{\max}\right) \qquad (15)$$

where α denotes the Lagrangian multiplier.

The first order derivative of the above Lagrangian function is derived as follows:

$$\frac{\partial L(q,\alpha)}{\partial q_n} = \frac{\partial U_T}{\partial q_n} + \alpha\left(\frac{q_n}{\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)}\right) \qquad (16)$$

Then, the first derivative given above is equal to zero, the value of Lagrange multi- plier value α, at the optimal point can be obtained by solving the Lagrange optimality conditions:

$$\alpha = -\frac{\partial U_T}{\partial q_n}\frac{\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)}{q_n} \qquad (17)$$

In order to find the range of values of α, the first order derivative of the utility function of the upper game yields:

$$\frac{\partial U_T}{\partial q_n} = \frac{\partial U_T}{\partial f_n}\frac{\partial f_n}{\partial q_n} = \frac{\partial((\overline{\sigma}_n + \widehat{\sigma}_n)(sc_n / f_n))}{\partial f_\mathrm{n}} \cdot \frac{\partial\left(\dfrac{q_n}{2\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)}\right)}{\partial q_n} \qquad (18)$$

In the above equation, it is clear that the upper-level game utility function is negatively correlated with the CPU power fn provided by the data owner. Furthermore, based on the equation, there exists a positive correlation between the price per unit of CPU power and the CPU power provided by the data owner. The first term on the right-hand side of the equation is positive, indicating a positive influence. Additionally, the second term is always positive, further contributing to the positive relationship. Therefore, it can be inferred that the Lagrange multiplier α > 0.

In combination with the above KKT conditions:

$$\alpha\left(\frac{q_n^2}{2\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)} - R_{\max}\right) = 0 \qquad (19)$$

Therefore, the equilibrium solution of the upper level game is:

$$q_n^* = \sqrt{2\mu\varepsilon s c_n(\overline{\sigma}_n + \widehat{\sigma}_n)R_{\max}} \qquad (20)$$

4.2. Robust Incentive Mechanisms for Stackelberg Games Under Polyhedron Uncertainty in Data Quality

In order to take into account the robustness of data quality uncertainty while maximizing the benefits for both sides of the upper and lower level games, a polyhedron uncertainty set is therefore considered to characterize data quality uncertainty.

The uncertainty type variables of the data quality parameters σn are assumed to be a given uncertainty set, and the polyhedron uncertainty set is introduced, then σnψn], where σn is the data quality parameter in the nominal model, σn is its perturbation. Its uncertainty set is Ψ={ψ : ψn ≤ Γn, 0 ≤ ψn ≤ 1}, n∈N; where smaller Γn denotes the level of uncertainty in this uncertainty set and is used as an objective measure of how conservative the uncertainty in data quality is, thus reflecting the degree of risk appetite of the task publisher, with smaller Γn values indicating a higher degree of risk-seeking preference by the task publisher.

The lower level game constructed under this condition is:

$$\begin{aligned} &\max_{f_n} \quad U_D = q_n f_n - (\overline{\sigma}_n + \widehat{\sigma}_n \psi_n)\mu\varepsilon sc_n f_n^2 - D_n^t \\ &\text{s.t.} \quad f_n \le f_{\max} \end{aligned} \tag{21}$$

The upper level game is:

$$\begin{aligned} &\min_{q_n} \; U_T = (\overline{\sigma}_n + \widehat{\sigma}_n \psi_n)(sc_n / f_n) + s/(B\ln(1+(\rho_n \mathrm{h}_n/\mathrm{N}_0)) \\ &\text{s.t.} \quad q_n f_n \le R_{\max} \end{aligned} \tag{22}$$

Based on the robust construct proposed by Bertsimas, the robust optimization model is transformed into an equivalent optimization model that is more easily solvable for the upper-level subgame. The transformed model is formulated as follows:

$$\begin{aligned} &\min_{q_n, c_n, \sigma_n, v_n} \quad \omega \\ &s.t. \quad sc_n / f_n + s/(B\ln(1+(\rho_n \mathrm{h}_n/\mathrm{N}_0))) - \Gamma u' - \sum_{n=1}^{N} v_n \le \omega \\ &\qquad u + v_n \ge s\sigma_n c_n / f_n + s/(B\ln(1+(\rho_n \mathrm{h}_n/\mathrm{N}_0))) \\ &\qquad u \ge 0 \\ &\qquad v_n \ge 0 \\ &\qquad q_n f_n \le R_{\max} \end{aligned} \tag{23}$$

The lower subgame is:

$$\begin{aligned} &\max_{f_n, c_n, \sigma_n, v_n} \quad D \\ &s.t. \quad q_n f_n - \mu(\sigma_n \varepsilon sc_n f_n^2 - D_n^t) - \Gamma u - \sum_{n=1}^{N} v_n \ge D \\ &\qquad u + v_n \ge q_n f_n - \mu(\sigma_n \varepsilon sc_n f_n^2 - D_n^t) \\ &\qquad 0 \le f_n \le f_{\max} \\ &\qquad u \ge 0 \\ &\qquad v_n \ge 0 \end{aligned} \tag{24}$$

The Lagrangian function corresponding to the lower level game is:

$$\begin{aligned} &L(f_n, \lambda_{1n}, \lambda_{2n}, \lambda_{3n}, \lambda_{4n}) \\ &\quad = D + \lambda_{1n}(q_n f_n - \mu(\sigma_n \varepsilon sc_n f_n^2 - D_n^t) - \Gamma u - \sum_{n=1}^{N} v_n - D) \\ &\qquad + \lambda_{2n}(u + v_n - \widehat{\sigma}_n \mu\varepsilon sc_n f_n^2) + \lambda_{3n}(f_{\max} - f_n) + \lambda_{4n} f_n \end{aligned} \tag{25}$$

The KKT condition for the lower subgame can be expressed as:

$$\begin{cases} \lambda_{1n}(q_n - 2\sigma_n \mu\varepsilon sc_n f_n) - 2\lambda_{2n} \widehat{\sigma}_n \mu\varepsilon sc_n f_n - \lambda_{3n} + \lambda_{4n} = 0 \\ q_n f_n - \mu(\sigma_n \varepsilon sc_n f_n^2 + D_n^t) - \Gamma u - \sum_{n=1}^{N} v_n - D = 0 \\ u + v_n - \widehat{\sigma}_n \mu\varepsilon sc_n f_n^2 = 0 \\ f_n - f_{\max} \le 0 \\ \lambda_{3n}(f_{\max} - f_n) = 0 \\ \lambda_{4n} f_n = 0 \\ f_n \ge 0, n = 1, 2, \ldots.N \\ \lambda_{in} \ge 0, i = 1, 2, 3, 4, n = 1, 2, \ldots.N \end{cases} \tag{26}$$

From the optimization principle it follows, the optimal solution of the lower subgame satisfies the set of inequalities formed by the KKT conditions above:

$$f_n = \begin{cases} q_n / 2\mu(\sigma_n \mu\varepsilon \mathrm{sc}_n + D_n^t) + N\widehat{\sigma}_n \; \mu\varepsilon \mathrm{sc}_n \\ \qquad if \;\; q_n / 2\mu(\sigma_n \mu\varepsilon \mathrm{sc}_n + D_n^t) + N\widehat{\sigma}_n \; \mu\varepsilon \mathrm{sc}_n \le f_{\max} \\ f_{\max} \\ \qquad \text{otherwise} \end{cases} \tag{27}$$

Using backward induction, the calculated frequencies fn of the model from the lower game are brought into the upper game, and the equilibrium solution is obtained by analyzing the solution of the upper subgame.

The Lagrangian function for the upper subgame is:

$$\begin{aligned} L(q_n, \phi_{1n}, \phi_{2n}, \phi_{3n}, \phi_{4n}) = \Delta + \phi_{1n}(&\frac{2\mu s\sigma_n c_n(\sigma_n \varepsilon \mathrm{sc}_n + D_n^t) + N\widehat{\sigma}_n \; \mu\varepsilon \mathrm{sc}_n}{q_n} \\ &+ s/(B\ln(1 + \rho_n h_n / N_0) - \Gamma u - \sum_{n=1}^{N} v_n - \Delta)) \\ &+ \phi_{2n}(\frac{s\widehat{\sigma}_n c_n(2\mu(\sigma_n \varepsilon sc_n + D_n^t) + N\widehat{\sigma}_n \mu\varepsilon sc_n}{q_n} - u - v_n) \\ &+ \phi_{3n}(\frac{q_n^2}{2\mu(\sigma_n \varepsilon \mathrm{sc}_n + D_n^t) + N\widehat{\sigma}_n \; \mu\varepsilon \mathrm{sc}_n)} - R_{\max}) - \phi_{4n} q_n \end{aligned} \tag{28}$$

The upper game is solved by applying the KKT conditional transformation, and since the solution function of the upper game is convex in the range of independent variables, the feasible solution is the optimal solution when discussing the values of the Lagrange multipliers.

Combined with the KKT condition, it follows that:

$$\phi_{1n}\frac{sc_n\sigma_n}{q_n^2}+\phi_{2n}\frac{sc_n\sigma_n}{q_n^2}-2\phi_{3n}\frac{q_n^2}{(2\mu(\sigma_n\varepsilon \mathrm{sc}_n+D_n^t)+N\widehat{\sigma}_n\mu\varepsilon \mathrm{sc}_n)^2}+\phi_{4n}=0 \quad (29)$$

The derivative of the Lagrangian function with respect to the dyadic variables v_n is given by:

$$(N+1)\phi_{1n}\frac{sc_n\sigma_n}{q_n^2}=2\phi_{3n}\frac{q_n^2}{(2\mu(\sigma_n\varepsilon \mathrm{sc}_n+D_n^t)+N\widehat{\sigma}_n\ \mu\varepsilon \mathrm{sc}_n)^2}-\phi_{4n} \quad (30)$$

If the above equation _1n = 0, a feasible solution can be found, at which point the solution is:

$$q_n^*=\sqrt{((2\sigma_n\mu\varepsilon sc_n+D_n^t)+N\widehat{\sigma}_n\varepsilon sc_n)R_{\max}} \quad (31)$$

5. Experimental Results

In the simulation experiments, we assess the proposed incentive scheme by utilizing the MNIST dataset and the widely used TensorFlow software environment for numerical classification tasks. In the simulation experiments, we consider a federated learning model training scenario involving 10 task publishers and 100 data owners. The data owners are classified into 10 groups based on the nominal value of data quality, while also considering the uncertainty associated with the provided data quality. To accurately represent the heterogeneity among data owners during local model training, we randomly select participating data owners, considering the number of training repetitions. For the simulation experiments, we set a maximum CPU power of 15 for each data owner fmax, and a maximum CPU frequency unit price of 100 for the task publisher qmax. Other parameters used in the simulation experiments are detailed in Table 2.

Table 2

Parameter setting in simulation experiment

Parameter name	Value
Transmission time of the local model T_n^t	0.5
Weight parameter for energy consumption μ	0.1
Local data sample size s	20
Data owner CPU cycles cn	5
Effective capacitance parameter for the data owner ε	2
Accuracy of the local data λn	[0.2, 0.9]
Maximum CPU power fmax	15
Maximum CPU frequency unit price qmax	100

As shown in the Figure 2. When Γ = 0, the robust Stackelberg game is equivalent to the Stackelberg game with known data quality. When Γ > 0, the optimal CPU frequency of the data owner decreases as the uncertainty level parameter Γ increases, and this decrease is further amplified with a larger perturbation ratio. This observation arises from the fact that higher levels of uncertainty result in increased uncertainty regarding the data quality provided by the data owner. To ensure the robustness of the model, the optimal CPU power offered by the data owner in the game is consequently lower.

The above experiments analyzed the impact of uncertain levels on the optimal CPU frequency of data owners under the proportion of different disturbances. According to the analysis of the above experimental results, at the same level of uncertain level, as the proportion of disturbances increases, the optimal CPU frequency of the data owner has continued to decrease, and as the level of uncertain level increases, the disturbance ratio to the optimal CPU frequency the influence is constantly increasing. The proportion of disturbances represents the data quality of each data owner around the size of the nominal value fluctuation range, and the uncertain level increases, so that the quality of data with uncertainty is more. The impact of excellent CPU frequency has continued to increase.

When the maximum value is denoted by Γ, the polyhedron uncertainty set is trans- formed into a boxed uncertainty set, representing the most robust case. The uncertainty level Γ also serves as a measure of the task publisher's risk preference to some extent. Therefore, task publishers can select the optimal combination of uncertainty level and perturbation ratio based on their preference for uncertainty risk. If a data owner exhibits a risk-seeking preference, they can opt for a higher uncertainty level and perturbation ratio. However, this choice entails bearing potential losses resulting from uncertainty. Conversely, if a data owner has a risk-averse preference, they may choose a lower uncertainty level and perturbation ratio, prioritizing a greater likelihood of efficiently feasible robust Stackelberg game while potentially sacrificing federated learning utility. Ultimately, a compromise can be made for data owners who are risk-neutral.

In this simulation experiment, the uncertainty levels of the box uncertainty set and the polyhedron uncertainty set and the model iteration lengths are compared with those of the case where the data quality is known. In Figure 3, Γ = 0 denotes the case where the data quality is known, Γ = 9 denotes the case of the iteration length for the box uncertainty set, and the middle value is taken to be the polyhedron uncertainty set. It can be seen that with the same proportion of perturbations, an increase in the uncertainty level simultaneously extends the iteration length. Since the number of data owners with possible uncertain data quality perturbations is increasing and to ensure the robustness of the model, the worst case of uncertain data should be considered. The results also indicate that compared with the polyhedron uncertainty set, the boxing uncertainty set is more robust, but it always has the longest iteration length and is less beneficial to the game sponsor.

The purpose of the above experiment is to reflect the impact of its impact on the effective use of the upper layer by the effects of the disturbance ratio on the time of the global iteration. As the proportion of disturbances continues to increase, the more uncertain level of data owners will grow faster to complete the global iteration. The

Figure 2

Variation of optimal CPU frequency with Γ for different disturbance ratios

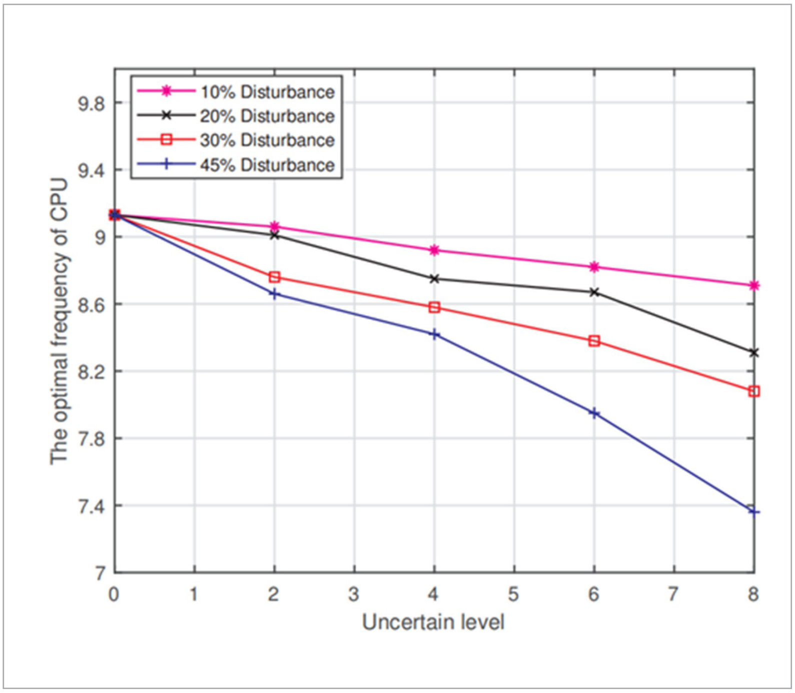
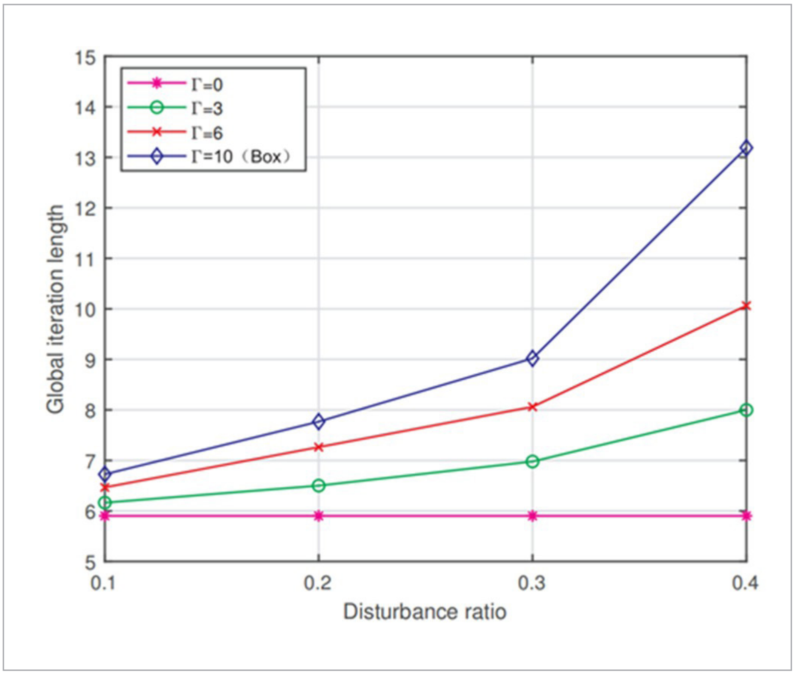


Figure 3

Variation of global iteration duration with disturbance ratio for different data owners



higher the level of data quality is allowed, the quality of the data that allows uncertainty to allow uncertainty. In order to ensure the effectiveness of the incentive, the worst case is considered. The number of data quality will inevitably increase, and with the increasing disturbance ratio, the impact on the length of the global iteration also gradually becomes larger.

In the upper-level game, the objective of the utility function is to minimize the global iteration length. As reflected in Figure 3, the iteration length may be more concentrated and robust when the perturbation ratio is small, and more scattered, less robust and more time consuming overall when the perturbation ratio is large. This is because the increase in the perturbation ratio leads to the rise of the uncertainty of data quality for each data owner. At the same time, seeking to ensure the robustness of the model, the feasible range of model training iteration length is expanded, which results in the increase of iteration time.

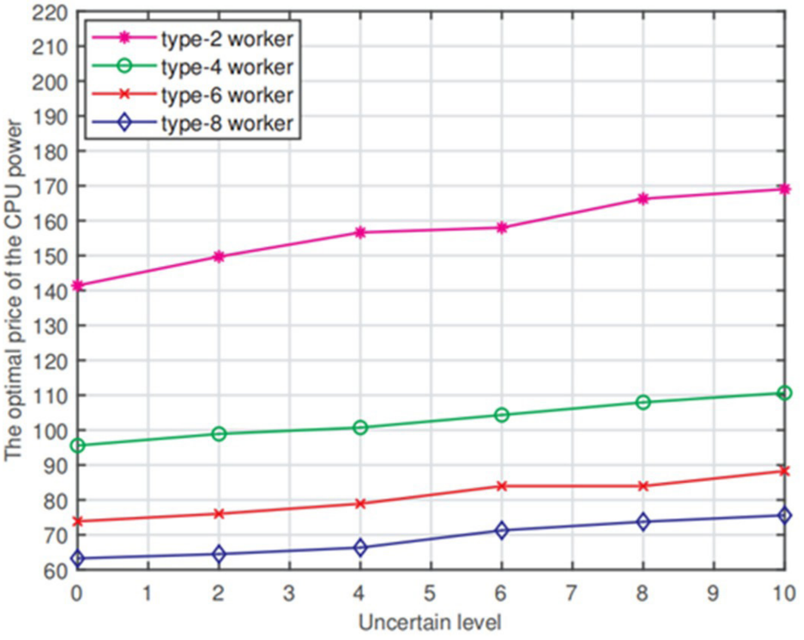
The following experiment is to explain the impact of uncertainty on the cost of model training of different data owners, so as to obtain the best unit price change that can be obtained at an uncertain level at the CPU frequency. Figure 4 displays the variation of the optimal CPU power unit price offered by the task publisher for data owners in categories 2, 4, 6 and 8, subject to the level of uncertainty. The optimal unit price for each class of data owner increases with the level of uncertainty, but at a decreasing rate. For a data owner with poor data quality, the optimal unit price is high and increasing rapidly in order to provide the owner with optimal CPU power. As can be seen from the graph, the unit price of the optimal CPU power offered by the central server is concave as the data quality increases, so for the sake of robustness, it is particularly important for the task publisher to select the data quality of the data owners and avoid participants with low data quality as much as possible.

6. Conclusions

In this work, the uncertainty of the data quality of data owner in federated learning is studied, a kind of robust incentive mechanism of federated learning based on Stackel- berg game is proposed to deal with the above

Figure 4

Variation of the optimal unit price obtained by different data owners with the level of uncertainty



uncertainty, the existence and uniqueness of robust Stackelberg equilibrium are theoretically proved, and corresponding solutions are designed for Stackelberg games with different robust uncertainty sets. Experimental results demonstrate that the robust incentive mechanism is capable of effectively capturing the risk preference of the task publisher. It enables the task publisher to select the optimal level of uncertainty

The federated learning incentive mechanism based on the robust Stackelberg game constructed in this paper not only processes the data quality of the data owner in the model, but also the different solutions of the robust Stackelberg game when the data quality is box uncertainty and polyhedron uncertainty, it has a certain reference significance for introducing the idea of optimizing the ride optimization and the uncertain parameter to the Stackelberg game. More importantly, in this paper, the robust optimization is introduced into the incentive mechanism to characterize the parameter uncertainty in the model training, and the robustness game model of federated learning is constructed. It is helpful to implement federated learning effectively in medical and financial application scenarios, and has practical significance to further expand the application fields of federated learning.

This work only considers the robust incentive mechanism for unknown data quality to improve the performance of federated learning. In practical problems, the CPU power supplied by the data owner may also change with iteration length, and even the transmission time for updating parameters may be uncertain due to the channel instability, which is also one of key factors of influencing the performance of federated learning. And this paper does not focus on the impact of malicious attackers on modeltraining.

In future research, we can consider the uncertainty of other parameters in the actual situation of federated learning model training to ensure that the incentive effect is more in line with the actual training. We can also consider introducing other methods of processing parameters into the design of incentive mechanisms for federated learning, such as random optimization and fuzzy number processing.

Acknowledgement

This research has been supported by the National Natural Science Foundation of China (Grant number 11626079), the Natural Science Foundation of Hebei Province (Grant number A2020402013).

References

1. Alferaidi, A., Yadav, K. Federated Learning Algorithms to Optimize the Client and Cost Selections. Mathematical Problems in Engineering, 2022. https://doi.org/10.1155/2022/8514562
2. Bertsimas, D., Sim, M. The Price of Robustness. Operations Research, 2004, 52, 35-53. https://doi.org/10.1287/opre.1030.0065
3. Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., Cui, S. A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks. IEEE Transactions on Wireless Communications, 2020, 20, 269-283. https://doi.org/10.1109/TWC.2020.3024629
4. Chen, Y. R., Zhang, Y. Y., Wang, S., Wang, F., Li, Y., Jiang, Y. DIM-DS: Dynamic Incentive Model for Data Sharing in Federated Learning Based on Smart Contracts and Evolutionary Game Theory. IEEE Internet of Things Journal, 2022, 9(23), 24572-24584. https://doi.org/10.1109/JIOT.2022.3191671
5. Chen, Y., Sun, X. Communication-Efficient Federated Deep Learning with Layerwise Asynchronous Model Update and Temporally Weighted Aggregations. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31, 4229-4238. https://doi.org/10.1109/TNNLS.2019.2953131
6. Darzidehkalani, E., Ghasemi-Rad, M., van Ooijen, P. M. A. Federated Learning in Medical Imaging: Part II: Methods, Challenges, and Considerations. Journal of the American College of Radiology: JACR, 2022, 19(08), 975-982. https://doi.org/10.1016/j.jacr.2022.03.001
7. Deng, Y. H., Lyu, F., Ren, J. Improving Federated Learning with Quality-Aware User Incentive and Auto-Weighted Model Aggregation. IEEE Transactions on Parallel and Distributed Systems, 2022, 33, 4515-4529. https://doi.org/10.1109/TPDS.2022.3195207
8. Dinh, C., Tran, N., Nguyen, H. Federated Learning over Wireless Networks: Convergence Analysis and Resource Allocation. IEEE/ACM Transactions on Networking, 2021, 29, 398-409. https://doi.org/10.1109/TNET.2020.3035770
9. Fadlullah, Z. M., Kato, N. HCP: Heterogeneous Computing Platform for Federated Learning Based Collaborative

Content Caching Towards 6G Networks. IEEE Transactions on Emerging Topics in Computing, 2022, 10(02), 112-123. https://doi.org/10.1109/TETC.2020.2986238

10. Golpira, H., Javanmardan, A. Robust Optimization of Sustainable Closed-Loop Supply Chain Considering Carbon Emission Schemes. Sustainable Production and Consumption, 2022, 30(3), 640-656. https://doi.org/10.1016/j.spc.2022.05.007
11. Hu, R., Gong, Y. Trading Data for Learning: Incentive Mechanism for On-Device Federated Learning. In Proceedings of the 2020 IEEE Global Communications Conference, 2020, 1-6. https://doi.org/10.1109/GLOBECOM42002.2020.9322593
12. Huang, W. X., Tiropanis, T., Konstantinidis, G. Federated Learning-Based IoT Intrusion Detection on Non-IID Data. Internet of Things, 2022, 13(14), 326-337. https://doi.org/10.1007/978-3-031-20936-9_26
13. Kang, J. W., Xiong, Z. H., Niyato, D., Zou, Y., Zhang, Y., Guizani, M. Reliable Federated Learning for Mobile Networks. IEEE Wireless Communications, 2020, 27(02), 72-80. https://doi.org/10.1109/MWC.001.1900119
14. Khan, L., Pandey, S., Tran, N. Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism. IEEE Communications Magazine, 2020, 58, 88-93. https://doi.org/10.1109/MCOM.001.1900649
15. Monhamed, R. A., Zakariya, Y. Modified Jackknife Ridge Estimator for Beta Regression Model with Application to Chemical Data. International Journal of Mathematics, Statistics, and Computer Science, 2023, 1, 15-24. https://doi.org/10.59543/ijmscs.v1i.7713
16. Ng, K. L., Chen, Z. C., Liu, Z. L., Yu, H., Liu, Y., Yang, Q. A Multi-Player Game for Studying Federated Learning Incentive Schemes. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, 5279-5281. https://doi.org/10.24963/ijcai.2020/769
17. Noura, O. On the Product and Ratio of Pareto and Erlang Random Variables. International Journal of Mathematics, Statistics, and Computer Science, 2023, 1, 33-47. https://doi.org/10.59543/ijmscs.v1i.7737
18. Reny, P. J. On the Existence of Pure and Mixed Strategy Nash Equilibria in Discontinuous Games. Econometrica, 1999, 67, 1029-1056. https://doi.org/10.1111/1468-0262.00069
19. Roberts, M., Driggs, D. Common Pitfalls and Recommendations for Using Machine Learning to Detect and Prognosticate for COVID-19 Using Chest Radiographs and CT Scans. Nature Machine Intelligence, 2021, 3, 199-217. https://doi.org/10.1038/s42256-021-00307-0
20. Toyoda, K., Zhang, A. N. Mechanism Design for an Incentive-Aware Blockchain-Enabled Federated Learning Platform. In Proceedings of the 2019 IEEE International Conference on Big Data, 2019, 395-403. https://doi.org/10.1109/BigData47090.2019.9006258
21. Wang, W., Wang, Y., Huang, Y. Privacy Protection Federated Learning System Based on Blockchain and Edge Computing in Mobile Crowdsourcing. Computer Networks, 2022, 215, 109206. https://doi.org/10.1016/j.comnet.2022.109206
22. Xiao, G., Xiao, M. Incentive Mechanism Design for Federated Learning: A Two-Stage Stackelberg Game Approach. In Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems, 2020, 148-155. https://doi.org/10.1109/ICPADS51040.2020.00028
23. Xu, J., Wang, H. Q., Chen, L. X. Bandwidth Allocation for Multiple Federated Learning Services in Wireless Edge Networks. IEEE Transactions on Wireless Communications, 2022, 21, 2534-2546. https://doi.org/10.1109/TWC.2021.3113346
24. Yang, J., Su, C. Q. Robust Optimization of Microgrid Based on Renewable Distributed Power Generation and Load Demand Uncertainty. Energy, 2021, 223(2). https://doi.org/10.1016/j.energy.2020.119472
25. Yang, Q., Liu, Y. Federated Machine Learning: Concept and Applications. ACM Transactions on Intelligent Systems and Technology, 2019, 10, 1-19. https://doi.org/10.1145/3298981
26. Yang, Z., Chen, M., Wong, K. K. Federated Learning for 6G: Applications, Challenges, and Opportunities. Engineering, 2022, 8, 33-41. https://doi.org/10.1016/j.eng.2021.09.011
27. Yang, Z., Chen, M., Saad, W., Hong, C. S., Shikh-Bahaei, M. Energy Efficient Federated Learning over Wireless Communication Networks. IEEE Transactions on Wireless Communications, 2020, 20, 1935-1949. https://doi.org/10.1109/TWC.2020.3037554
28. Yunus, S., Ozgur, E. Motivating Workers in Federated Learning: A Stackelberg Game Perspective. IEEE Networking Letters, 2020, 2(01), 23-27. https://doi.org/10.1109/LNET.2019.2947144
29. Zhan, Y., Li, P. A Learning-Based Incentive Mechanism for Federated Learning. IEEE Internet of Things Journal, 2020, 7, 6360-6368. https://doi.org/10.1109/JIOT.2020.2967772
30. Zhang, J., Guo, S. Adaptive Vertical Federated Learning on Unbalanced Features. IEEE Transactions on Parallel and Distributed Systems, 2022, 33, 4006-4018. https://doi.org/10.1109/TPDS.2022.3178443
31. Zhou, L., Fan, Q. W., Huang, X. D. Weak and Strong Convergence Analysis of Elman Neural Networks via Weight Decay Regularization. Optimization, 2022, 70, 75-100. DOI: 10.1080/02331934.2022.2048935