A Reward Response Game in the Federated Learning System

Suhan Jiang and Jie Wu
Department of Computer and Information Sciences, Temple University
{suhan.jiang, jiewu}@temple.edu

***Abstract*—The emergence of federated learning and the increasingly powerful mobile devices lead to a mobile-crowd machine learning paradigm. In this paper, we consider a mobile-crowd federated learning system that includes a central server and a set of mobile devices. As the model requester, the server motivates all devices to train an accurate model by paying them based on their individual contributions. Each participating device needs to balance between the training rewards and costs for profit maximization. A Stackelberg game is proposed to model interactions between the server and devices. To match with reality, our model takes the training deadline and the device-side upload time into consideration. Based on different definitions of individual contribution, two reward policies, *i.e.*, the size-based policy and accuracy-based policy, are compared. The existence and uniqueness of Stackelberg equilibrium (SE) under both definitions are analyzed, according to which algorithms are proposed to achieve the corresponding SE(s). We show that there is a lower bound of 0.5 on the price of anarchy in the proposed game. We extend our model by considering the uncertainty in the upload time, where each device's upload time is subject to a normal distribution due to its unstable channel. Numerical evaluations are presented to verify the proposed models.**

***Index Terms*—Federated learning, game theory, incentive mechanism, mobile-crowd machine learning, price of anarchy.**

I. INTRODUCTION

Federated learning [1] has enabled model(s) to be collaboratively trained across multiple devices using decentralized data samples without actual data exchange, and therefore protecting data privacy and security. Meanwhile, the growth of mobile devices also get machine learning at the end of the network for real. Therefore, mobile-crowd federated learning has emerged as a new business trend. Fig. 1 shows a typical federated learning system, consisting of a central server as a model requester and a set of mobile devices as model trainers. In such a system, the server distributes a global model to the devices. The devices train the model on locally available data. All updated models, instead of the data, are then sent back to the server, where they are averaged to produce a new global model. This new model now acts as the primary model and is again distributed to the devices. This process is repeated forever or until the global model achieves a satisfactory result from the server side. Usually, the newly aggregated global model should get a little better than it already was. Obviously, model training moves to the edge of the network so that the data never leaves the device, while it is still under the central server's orchestration.

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

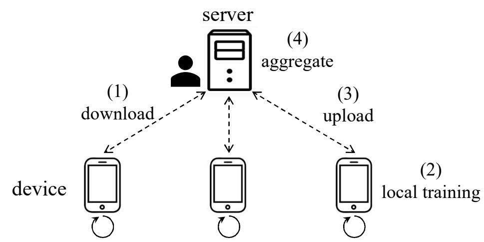


Fig. 1: Federated learning system: (1) server sends current global model to all devices; (2) each device trains its model using the local data; (3) all devices upload their updated models to server; (4) server aggregates all local models into a new global model.

The fact that model training consumes resources makes it impossible for mobile devices to voluntarily participate in the federated learning task. In most cases, monetary incentive is a necessity for any mobile-crowd federated learning system. That is, the server has to motivate participating devices with enough rewards in order to obtain a satisfactory model. The model accuracy can be used to measure how satisfied the server is with the obtained model. Usually, the model accuracy is positively related to the size of overall trained data. Thus, the server wants devices to train more data in local training round, which inevitably will increase each device's cost. To cover their losses, more rewards should be provided. However, existing works also confirm that the model accuracy and the data size show a concave down increasing trend, indicating the principle of diminishing marginal return. Thus, to balance its utility, *i.e.*, the difference between its satisfaction of the obtained model and the reward it offers to all devices, it is important for a server to decide a suitable reward amount. In some case, the server may set a deadline by which all devices should complete the training and upload step, and ignore any update submitted after the deadline.

Devices participate in the federated learning aiming for the training rewards. Due to the different network environments, devices may vary over their own upload time. Each device must pay attention to its training time to avoid missing the deadline if one is given. Similarly, each device has its own computation power, indicating a specific training speed. The higher computation power a device has, the more data it can train in a given time. Usually, a device will get rewarded based on its contribution to the global model. There exist two common ways to measure a device's individual contribution. One is using the size of its trained data, and the other is using the device's local model accuracy. As we mentioned before, the model accuracy is a concavely increasing function in terms of the training data size. Thus, the more rewards a device wants to obtain, the more data it has to train, and the more time it has to spend in the training step. However, a long training time also leads to a high computation cost. Obviously, the training time brings about a tradeoff between the reward and the cost to the device. Thus, optimizing the training time is essential for each device, as it wants to maximize its utility, *i.e.*, the difference between its reward and its cost.

We exploit game theory to analyze the complex interactions between the server and mobile devices. To solve the reward-based resource management problem, we leverage a Stackelberg game, which includes two steps for the server (as a leader) and then devices (as followers), respectively. In the first step, the server announces its deadline and sets a reward for a training round by anticipating the devices' responses. In the second step, the devices decide their training time according to the observed reward and deadline as well as their individual upload times. Moreover, we investigate how the reward policy applied by the server will affect the devices' decisions as well as the whole system. As we mentioned before, a common reward policy can be paying each device either in proportion to its data-size-based contribution or to its local-accuracy-based contribution.

All previous studies assume that each device's upload time is fixed as a common knowledge in the proposed game. In practice, a mobile device in the wifi environment experiences an unstable network speed, indicating its upload time may change due to the time-varying network condition. In this paper, we also discuss the impact of upload time uncertainty on the devices' strategies by modeling each device's upload time as a random variable. That is, we assume that each device's upload time follows a Normal distribution with fixed values of mean and variance. The major contributions of this paper are as follows:

- We propose a Stackelberg game to solve a reward-based computation resource management problem in a federated learning system.
- We study the proposed Stackelberg game in two practical reward sharing policies, *i.e.*, size-based policy and accuracy-based policy, where the existence and uniqueness of Stackelberg equilibrium (SE) are analyzed.
- We show that our proposed game is a valid utility game, thereby it has a lower bound of 0.5 on the price of anarchy.
- We investigate the impacts of upload time uncertainty, which incurs longer training time on the device side.
- We perform numerical evaluation based on real-world data and the results are consistent with all the theoretical results.

II. Model and Problem

A. Model Description

As shown in Fig. 1, we consider a cooperative federated learning system. The model consists of a number of mobile devices associated with a central parameter server. The whole system is in a universal mobile network with wireless communication infrastructures. We consider a quasi-static state where no devices are joining or leaving. Corresponding notations are shown in Table I.

TABLE I: Summary of Notations.

Symbol	Description
R	total reward offered by the server in a global round
N	number of participating devices
β_i	unit-time computation speed of device i
c_i	unit-time computation cost of device i
α_i	device i's contribution
α / α_{-i}	sum of all devices' contributions with/without i
t_i	local training time decided by device i
τ_i	expected upload time of device i
T	server's pre-announced deadline

The server aims to build a global machine learning model by employing N devices. Firstly, the server shares the current global model parameters with all devices. All devices will train their local models using their own data. Then, each device uploads its updated local model parameters to the server. Finally, the server facilitates the computation of the parameters aggregation and obtains a new global model. We consider that these four steps form a global update round. In a global round, each device experiences lots of local training iterations, depending on its training data size. The global rounds continue repeatedly until meeting some specific requirements, *e.g.* a certain accuracy level or a deadline.

Since the final global model is obtained through lots of training rounds, here, we only consider one round, in which, the server wants to make its model as improved as possible. According to the existing works, the accuracy of a machine learning model depends on the training data size. The relation between them can be captured by a concavely increasing function (an example is given in Fig. (2)), indicating a decreasing marginal gain. For simplicity, we assume that all training data in each edge device has the same quality and is independently and identically distributed (IID). Based on this assumption, the more data trained by the devices, the better global model the server will obtain at the end of a round. To motivate all devices' participation level, the server will announce a total reward R at the beginning of a round as an incentive. All devices will share this reward based on their individual contributions to the global model.

Each device should train its local model and upload its local parameters to the server before the round deadline T. Here, we assume all devices simultaneously start local training at the time of 0. Let t_i and τ_i represent device i's local model training time and its parameter upload time. Obviously, $t_i + \tau_i \leq T$. We define β_i as device i's unit-time computation speed, indicating that the number of its trained data is β_i in a unit time. Thus, in a training time t_i, device i trains $\beta_i t_i$ data in total. Let c_i represent the unit-time computation cost of device i. Then, its total training cost will be $c_i t_i$. Obviously, a longer training time brings a higher data contribution ratio while also incurring more computation cost. Since all devices aim to make a profit, they should balance the contribution and

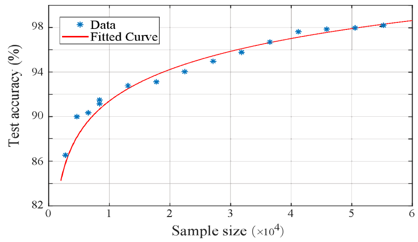


Fig. 2: Relation between the accuracy of the global model and the trained data size: $f(x) = 3.98log(9.68 \times 10^5 x - 3.69 \times 10^8)$.

the cost by carefully determining the training time t_i.

B. Stackelberg Game Formulation

Game theory provides a natural paradigm to model the interactions between the server and the devices in this network. The server sets the total reward and announces it to the devices. The devices respond to the reward by deciding an optimal training time. Since the server acts first and then the devices make their decision based on the reward, the two events are sequential. Thus, we model the interactions between the server and the devices using a Stackelberg game. It is a single-leader multi-follower Stackelberg game, where the server is the leader and the devices are the followers. In the first stage, the server optimizes the reward R it is willing to offer in a global round by predicting the devices' reactions. It also informs all devices of a deadline T. Devices that fail to upload their local models will not be rewarded. In the second stage, after observing R and T, each device i responds with a suitable training time, by considering its computation speed β_i, computation cost c_i, and upload time τ_i, as well as other devices' decisions. Since decisions are generated for individual utility maximization, a non-cooperative follower subgame is formed.

1) Device Side Utility: We define $\alpha_i(t_i)$ as device i's single round contribution. We will consider two different ways to define $\alpha_i(t_i)$, and under both definitions, the value of $\alpha_i(t_i)$ always depends on its training time t_i. In the rest of the paper, we use α_i for the simplicity of writing. With the system model, we formulate the following optimization problem for maximizing the overall profit in each round:

Problem 1 ($\text{OP}_{\text{DEVICE}}$).

$$\textit{maximize} \qquad u_i(t_i, \boldsymbol{t}_{-i}) = R\frac{\alpha_i}{\alpha} - c_i t_i, \tag{1a}$$

$$\textit{where} \qquad \alpha = \sum\nolimits_{j=1}^{N} \alpha_j, \tag{1b}$$

$$\textit{subject to} \qquad t_i + \tau_i \leq T. \tag{1c}$$

Each device i aims to maximize its utility and constraint (1c) ensures that i's local model can be uploaded within the deadline, thereby avoiding the worst case of zero-reward.

2) Server Side Utility: The objective of the server is to optimize its utility by determining the corresponding reward. Let V denote the server's utility, which is the difference between its satisfaction about the newly aggregated global model and the reward R it has to pay all qualified devices. We assume that the server's satisfaction is caught by the estimated accuracy of the new global model, which is a concavely increasing function over the amount of the data trained by all devices. Thus, we use a log function to characterize the relationship between the model accuracy and the trained data. The optimization problem $\text{OP}_{\text{SERVER}}$ on the server side is thus defined as below.

Problem 2 ($\text{OP}_{\text{SERVER}}$).

$$\textit{maximize} \qquad V = \theta \log\left(1 + \lambda \sum\nolimits_{i=1}^{N} \beta_i t_i\right) - R \tag{2}$$

3) Stackelberg Equilibrium: $\text{OP}_{\text{SERVER}}$ and $\text{OP}_{\text{DEVICE}}$ together form the proposed Stackelberg game. To achieve equilibrium in this game where neither the leader (server) nor the followers (devices) have incentive to deviate, we need to find its subgame perfect Nash equilibrium (NE) in the follower stage first, and then apply backward induction to achieve the leader side equilibrium.

III. Device-Side Equilibrium in the Fixed-Upload-Time Setting

We start with a relatively simple setting, where each device has a stable channel connecting to the server. That is, the model upload time τ_i is a pre-known constant. This assumption allows us to focus on how different reward policies applied by the server will affect devices' strategies and thereby influence the result of the proposed Stackelberg game.

A. Size-based Reward Policy

1) Follower Subgame Equilibrium: It is natural to consider the size of the trained data to measure the individual contribution. The corresponding device side optimization problem can be rewritten as below.

Problem 3 ($\text{OP}_{\text{DEVICE}}$).

$$\textit{maximize} \qquad u_i(t_i, \boldsymbol{t}_{-i}) = R\frac{\alpha_i}{\alpha} - c_i t_i, \tag{3a}$$

$$\textit{where} \qquad \alpha_i = \beta_i t_i, \tag{3b}$$

$$\textit{subject to} \qquad t_i + \tau_i \leq T. \tag{3c}$$

Theorem 1. *At least one Nash equilibrium exists in Problem 3.*

Proof. Obviously, each device's strategy space $[0, T - \tau_i]$ is a non-empty, compact, and convex subset of the Euclidean space, and the utility function $u_i(t_i, \mathbf{t}_{-i})$ is continuous and twice differentiable over $[0, T - \tau_i]$. In order to show the existence of Nash equilibrium, we need to prove that u_i is concave with respect to t_i. According to Eq. (4), the second order derivative of u_i is less than 0 over $[0, T - \tau_i]$.

$$\frac{\partial^2 u_i}{\partial t_i^2} = \frac{-2R\beta_i^2 \alpha_{-i}}{\alpha^3} < 0 \tag{4}$$

where $\alpha_{-i} = \sum_{j \neq i} \alpha_j$. Therefore, we can conclude that there exists at least one Nash equilibrium in $\text{OP}_{\text{DEVICE}}$. □

Lemma 1. $\sqrt{\frac{R}{\beta_i c_i}} - \frac{2}{\beta_i}\sqrt{\alpha_{-i}} > 0$ *always holds if the following condition holds.*

$$2(N-1)\frac{c_i}{\beta_i} < \sum\nolimits_{j=1}^{N} \frac{c_j}{\beta_j} \tag{5}$$

Proof. Given the domain $[0, T-\tau_i]$ and the first order derivative Eq. (6) of u_i, device i's the best response strategy can be obtained in Eq. (7), which is a function over $\mathbf{t} = [t_1, \cdots, t_N]$.

$$\frac{\partial u_i}{\partial t_i} = R\beta_i \frac{\alpha_{-i}}{\alpha^2} - c_i \tag{6}$$

$$t_i^* = g(\mathbf{t}) = \begin{cases} 0 & \frac{\sqrt{R\beta_i c_i \alpha_{-i}} - c_i\alpha_{-i}}{\beta_i c_i} < 0 \\ \frac{\sqrt{R\beta_i c_i \alpha_{-i}} - c_i\alpha_{-i}}{\beta_i c_i} & 0 < \frac{\sqrt{R\beta_i c_i \alpha_{-i}} - c_i\alpha_{-i}}{\beta_i c_i} \leq T - \tau_i \\ T - \tau_i & \frac{\sqrt{R\beta_i c_i \alpha_{-i}} - c_i\alpha_{-i}}{\beta_i c_i} > T - \tau_i \end{cases} \tag{7}$$

Let Eq. (6) be equal to 0. Then, we obtain the following equation.

$$\frac{\alpha_{-i}}{\alpha^2} = \frac{1}{R}\frac{c_i}{\beta_i} \tag{8}$$

By summing up on the both sides of Eq. (8), we obtain $\sum_{j=1}^{N} \beta_j t_j = R(N-1)/\sum_{j=1}^{N} \frac{c_j}{\beta_j}$. According to Eq. (7), we obtain $c_i \left(\sum_{j=1}^{N} \beta_j t_j\right)^2 = R\beta_i \sum_{j\neq i} \alpha_j$. Combining these two equations, we obtain the following result.

$$\sum\nolimits_{j\neq i} \beta_j t_j = \left(\frac{N-1}{\sum_{j=1}^{N} \frac{c_j}{\beta_j}}\right)^2 \frac{Rc_i}{\beta_i} \tag{9}$$

When Eq. (5) holds, we can easily prove that $\sqrt{\frac{R}{\beta_i c_i}} - \frac{2}{\beta_i}\sqrt{\sum\limits_{j\neq i} \alpha_j} > 0$ always holds. ☐

Definition 1. *The function $g(\boldsymbol{t})$ is standard if for all $\mathbf{t} \geq 0$, the following properties are satisfied.*

(1) Positivity: $g(\boldsymbol{t}) > 0$,
(2) Monotonicity: if $\mathbf{t} \geq \mathbf{t}'$, then $g(\boldsymbol{t}) \geq g(\boldsymbol{t}')$.
(3) Scalability: $\forall\lambda > 1$, $\lambda g(\boldsymbol{t}) \geq g(\lambda \boldsymbol{t}')$.

Theorem 2. *There exists a unique Nash equilibrium in* $\text{OP}_{\text{DEVICE}}$ *if Eq. (5) holds.*

Proof. If Eq. (7) is a standard function, the proposed game has a unique Nash equilibrium.

The positivity is obviously satisfied by $g(\mathbf{t})$. We prove the monotonicity of $g(\mathbf{t})$ under the condition Eq. (5) by showing $g(\mathbf{t}) - g(\mathbf{t}') \geq 0$ given $\mathbf{t} \geq \mathbf{t}'$.

$$\begin{aligned} g(\mathbf{t}) - g(\mathbf{t}') &= \sqrt{\frac{R}{\beta_i c_i}}\left(\sqrt{\sum_{j\neq i}\alpha_j} - \sqrt{\sum_{j\neq i}\alpha_j'}\right) - \frac{1}{\beta_i}\left(\sum_{j\neq i}\alpha_j - \sum_{j\neq i}\alpha_j'\right) \\ &= \left(\sqrt{\sum_{j\neq i}\alpha_j} - \sqrt{\sum_{j\neq i}\alpha_j'}\right)\left[\sqrt{\frac{R}{\beta_i c_i}} - \frac{1}{\beta_i}\left(\sqrt{\sum_{j\neq i}\alpha_j} + \sqrt{\sum_{j\neq i}\alpha_j'}\right)\right] \\ &\geq \left(\sqrt{\sum_{j\neq i}\alpha_j} - \sqrt{\sum_{j\neq i}\alpha_j'}\right)\left[\sqrt{\frac{R}{\beta_i c_i}} - \frac{2}{\beta_i}\sqrt{\sum_{j\neq i}\alpha_j}\right] \end{aligned} \tag{10}$$

According to Lemma 1, the monotonicity property is proved.

Finally, to show the scalability property, we prove that $\forall\lambda > 1, \lambda g(\mathbf{t}) \geq g(\lambda\mathbf{t}')$ based on Eq. (11).

$$\lambda g(\mathbf{t}) - g(\lambda\mathbf{t}') = \lambda\sqrt{\frac{R\sum_{j\neq i}\alpha_j}{\beta_i c_i}} - \sqrt{\frac{R\sum_{j\neq i}\lambda\alpha_j'}{\beta_i c_i}} > 0 \tag{11}$$

Therefore, the proposed game always possesses a unique Nash equilibrium. ☐

Algorithm 1 Best Response Algorithm

Output: $\boldsymbol{t} = \{t_1, \cdots, t_N\}$
Input: Initialize k as 1 and pick a feasible starting point $\boldsymbol{t}^{(0)}$
for round k **do**
  for device i **do**
    Decide $t_i^{(k)} = t_i^{(k-1)} + \Delta \frac{\partial U_i\left(t_i, t_{-i}^{(k-1)}\right)}{\partial t_i}$
    Send the local model the server
  Server aggregates all models received before T into a new global model and send back to devices.
  if $\boldsymbol{t}^{(k)} = \boldsymbol{t}^{(k-1)}$ **then** Stop
  else set $k \leftarrow k+1$

This naturally gives a distributed iterative algorithm, allowing each device to iteratively update its strategy, given the strategies of other devices. We summarize the distributed iterative algorithm in Algorithm 1. Algorithm 1 is applicable to find the unique NE point, where each device is engaged in a gradient ascent process to maximize its utility.

B. Accuracy-based Reward Policy

Another simple and intuitive way for the server to measure the individual contribution and distribute its reward is based on each device's local model accuracy. As we mentioned before, the relationship between the model accuracy and the training time can be characterized by a log function. As we will show in the below, when using model accuracy to measure device's contributions, α_i is still a strictly increasing concave function with respect to t_i. In this case, the new problem facing each device is shown as follows.

Problem 4 ($\text{OP}_{\text{DEVICE}}$).

$$\begin{aligned} &\textit{maximize} && u_i(t_i, \boldsymbol{t}_{-i}) = R\frac{\alpha_i}{\alpha} - c_i t_i, && \text{(12a)} \\ &\textit{where} && \alpha_i = \theta\log\left(1 + \lambda\beta_i t_i\right), && \text{(12b)} \\ &\textit{subject to} && t_i + \tau_i \leq T, \quad \theta > 0, \quad \lambda > 0. && \text{(12c)} \end{aligned}$$

Theorem 3. *At least one Nash equilibrium exists in Problem 4.*

Proof. The existence of Nash equilibrium can be confirmed by showing that its second derivative is negative based on Eq. 13.

$$\frac{\partial^2 u_i}{\partial t_i^2} = -\frac{R\theta\lambda^2\beta_i^2\alpha_{-i}}{\left(1 + \lambda\beta_i t_i\right)^2\alpha^2}\left(2\frac{\theta}{\alpha} + 1\right) < 0 \tag{13}$$

Since the objective function is concave, we can conclude that there exists at least one Nash equilibrium in Problem 4. ☐

Lemma 2. *Let $\Gamma = \left\{\mathbb{N}, (A_i)_{i\in\mathbb{N}}, (\pi_i)_{i\in\mathbb{N}}\right\}$ be an N-player non-zero-sum game in normal form, where $\mathbb{N}$ represents the player set, A_i represents the i-th player's feasible strategies, which is a non-empty compact convex subset of the Euclidean space, and π_i represents the i-th player's utility function. If the utility functions $(\pi_1, \cdots, \pi_N)$ are diagonally strictly concave for $(A_i)_{i\in\mathbb{N}}$, then the game has a unique pure strategy Nash equilibrium [2].*

Lemma 3. *Given $\bigtriangledown\pi(\mathbf{x}) = \left[\frac{\partial\pi_i}{\partial x_1}, \cdots, \frac{\partial\pi_N}{\partial x_N}\right]^\intercal$ as the game's pseudo-gradient function and let $\Pi(\mathbf{x})$ denote the Jacobian of*

$\bigtriangledown\pi(\mathbf{x})$, *if the symmetric matrix* $\Pi(\mathbf{x})+\Pi^{\intercal}(\mathbf{x})$ *is negative definite for* $\mathbf{x}\in(A_i)_{i\in\mathbb{N}}$, *then the utility functions* $(\pi_1,\cdots,\pi_N)$ *are diagonally strictly concave for* $(A_i)_{i\in\mathbb{N}}$ *[2].*

Theorem 4. *A unique Nash equilibrium exists in Problem 4.*

Proof. To prove the uniqueness of Nash equilibrium in Problem 4, we need to show that $U(\mathbf{t})+U^{\intercal}(\mathbf{t})$, where U is given in Eq. (15), is negative definite.

We start with constructing the pseudo-gradient function $\bigtriangledown u(\mathbf{t})=\left[\frac{\partial u_1}{\partial t_1},\cdots,\frac{\partial u_N}{\partial t_N}\right]^{\intercal}$, where $\frac{\partial u_i}{\partial t_i}$ is shown in Eq. (14).

$$\frac{\partial u_i}{\partial t_i}=\frac{R\theta\lambda\beta_i\alpha_{-i}}{(1+\lambda\beta_i t_i)\,\alpha^2}-c_i,\quad \forall i\in[1,N] \tag{14}$$

Thus, we have the Jacobian of $\bigtriangledown u(\mathbf{x})$ as below.

$$U(\mathbf{t})=\begin{bmatrix}\frac{\partial^2 u_1}{\partial t_1^2} & \frac{\partial^2 u_1}{\partial t_1 t_2} & \cdots\\ \frac{\partial^2 u_2}{\partial t_2 t_1} & \ddots & \\ \vdots & & \end{bmatrix} \tag{15}$$

where $\frac{\partial^2 u_i}{\partial t_i^2}$ is given in Eq. (13) and $\frac{\partial^2 u_i}{\partial t_i t_j}$ can be referred in Eq. (16).

$$\frac{\partial^2 u_i}{\partial t_i t_j}=\frac{R\theta^2\lambda^2\beta_i\beta_j\left[\alpha-2(\alpha_i+\alpha_j)\right]}{(1+\lambda\beta_i t_i)\,(1+\lambda\beta_j t_j)\,\alpha^3} \tag{16}$$

Obviously, $U(\mathbf{t})+U^{\intercal}(\mathbf{t})$ is symmetric. Due to the complexity of this matrix, we use Matlab to check its eigenvalues, which are all negative, indicating it is negative definite. ☐

C. Stackelberg Equilibrium

While Algorithm 1 achieves a unique pure strategy for the devices' game, our final goal is to obtain the Stackelberg equilibrium of the entire system. For this purpose, we leverage Algorithm 1 to construct the corresponding SE. The Stackelberg equilibrium of the game can be found by solving the following non-linear optimization problem. Let $\mathbf{t}^*(R)$ be the unique NE obtained by the followers when the server offers a reward of R. The server needs to solve the following optimization problem a priori to find its unique optimal reward R^* and announce it to the devices.

Problem 5 (OP$_{\text{SERVER}}$)**.**

$$\textit{maximize}\quad V=\theta\log\left(1+\lambda\sum\nolimits_{i=1}^{N}\beta_i t_i^*(R)\right)-R \tag{17}$$

The corresponding SE can be achieved by solving the Problem 5. We analyze the effects induced by these two different contribution definitions in the simulation part. Specially, we compare the server utility, the total utility of all devices, and the social welfare achieved under these two definitions. Meanwhile, we also compare the social welfare computed under the proposed Stackelberg game with the optimal social welfare to see the price of anarchy caused by selfishness.

IV. Robust Price of Anarchy

Consider that the federated learning system operates in a centralized control. That is, all devices follow the server's instruction to train their local models. Then the objective of this whole system should be maximizing the social welfare, denoted by W, *i.e.*, the difference between the global model accuracy and the training cost among all devices, which is given in Eq. (18).

$$W=\theta\log\left(1+\lambda\sum\nolimits_{i=1}^{N}\beta_i t_i\right)-\sum\nolimits_{i=1}^{N}c_i t_i \tag{18}$$

Centralized control achieves a better performance than decentralized (game theoretic) control solutions in terms of the social objectives being met. The concept of price of anarchy, which is caused by the devices' selfish behaviors, is used to quantify the loss of efficiency in decentralized game solutions as compared to the optimal centralized control.

In the following, we prove that the non-cooperative game played among all devices is a valid monotone utility game. As a result, we obtain a lower bound on the PoA of value 0.5.

Definition 2. *Let* $\Gamma=\left\{\mathbb{N},(A_i)_{i\in\mathbb{N}},(U_i)_{i\in\mathbb{N}}\right\}$ *be an N-player non-zero-sum game in normal form, where* $\mathbb{N}$ *represents the player set,* A_i *represents the i-th player's feasible strategies, which is a non-empty compact convex subset of the Euclidean space, and* U_i *represents the i-th player's utility function. Assume that the objective function* $W((A_i)_{i\in\mathbb{N}})$ *where* $w:2^{(A_i)_{i\in\mathbb{N}}}\mapsto\mathbb{R}$, *is a general function defined over all subsets of* $(A_i)_{i\in\mathbb{N}}$. *Game* Γ *is called a valid utility game if it satisfies the following three properties*

(1) W *is submodular,*
(2) The objective value of a player is at-least his added value for the societal objective,
(3) The total value for the players is less than or equal to the total societal value.

And Γ *is called a monotone game if for all* $S\subseteq S'\subseteq(A_i)_{i\in\mathbb{N}}, W(S)\le W(S')$ *[20].*

Lemma 4. *If a game* Γ *is a valid monotone utility game, then its lower bound on the PoA is* 0.5.

Theorem 5. *Our proposed Stackelberg game has a lower bound on the PoA, which is* 0.5.

Proof. We show that our proposed Stackelberg game has the following three properties:

1) W *is submodular:* Assume there exists one set $\mathbf{a}\subseteq\mathbf{t}$ and two elements $t_p,t_q\in\mathbf{t}-\mathbf{a}$. We define set $\mathbf{a}'=\mathbf{a}\cup\{t_p\}$, indicating that $\mathbf{a}\subseteq\mathbf{a}'\subseteq\mathbf{t}$. Therefore, we have

$$\begin{aligned}&(W(\mathbf{a}\cup\{t_q\})-W(\mathbf{a}))-(W(\mathbf{a}'\cup\{t_q\})-W(\mathbf{a}'))\\ =&\theta\log\left(\frac{1+s+\lambda\beta_q t_q}{1+s}\right)-\theta\log\left(\frac{1+s+\lambda\beta_p t_p+\lambda\beta_q t_q}{1+s+\lambda\beta_p t_p}\right)\\ =&\theta\log\left(\frac{(1+s+\lambda\beta_q t_q)\,(1+s+\lambda\beta_p t_p)}{(1+s)\,(1+s+\lambda\beta_p t_p+\lambda\beta_q t_q)}\right)>0,\end{aligned} \tag{19}$$

where $s=\lambda\sum_{i\in\mathbf{a}}\beta_i t_i$. Thus, we can conclude that W is a submodular function.

2) Device i's utility u_i *is at-least its added value for the societal objective:* To prove this property, we need to show that $u_i(t_i,\mathbf{t}_{-i})\ge W(t_i,\mathbf{t}_{-i})-W(\mathbf{t}_{-i})$.

$$u_i(t_i,\mathbf{t}_{-i})-(W(t_i,\mathbf{t}_{-i})-W(\mathbf{t}_{-i}))$$

$$=R\frac{\alpha_i}{\alpha}-\theta\log\left(\frac{1+\lambda\sum_{j=1}^{N}\beta_j t_j}{1+\lambda\sum_{j=1}^{N}\beta_j t_j-\lambda\beta_i t_i}\right)$$
$$=R\frac{\alpha_i}{\alpha}-\theta\log\left(1+\frac{\lambda\beta_i t_i}{1+\lambda\sum_{j=1}^{N}\beta_j t_j-\lambda\beta_i t_i}\right)>0 \quad (20)$$

3) The total value for the devices is less than or equal to the total societal value: Here, we show the difference of $\sum_{i=1}^{N}u_i$ and W as below.

$$\sum_{i=1}^{N}u_i-W$$
$$=\sum_{i=1}^{N}\left(R\frac{\alpha_i}{\alpha}-c_i t_i\right)-\left(\theta\log\left(1+\lambda\sum_{i=1}^{N}\beta_i t_i\right)-\sum_{i=1}^{N}c_i t_i\right)$$
$$=R-\theta\log\left(1+\lambda\sum_{i=1}^{N}\beta_i t_i\right)>0 \quad (21)$$

Thus, our proposed game is a valid monotone utility game. Thus, it has a lower bound of 0.5 on the PoA. ☐

V. Unstable Communication Chanel

A. Model Stochastic Upload Time

As we mentioned before, local updates have to be transferred to the server. Every such update is of the same size as the trained model, which can be in the range of gigabytes for modern architectures with millions of parameters [3, 4]. Nevertheless, the devices typically employed in federated learning are communication-constrained, for example IoT devices or smartphones are generally connected to Wifi networks. Obviously, our previous assumption that each device has a fixed upload time τ_i is not realistic due to the mobility of devices and instability of wifi connection. In the following, we further consider a complex setting, where each device i's upload time is stochastic and subject to a normal distribution $\mathcal{N}(\mu_i,\sigma_i^2)$ for $\forall i\in[1,N]$. In the following, we will focus on equilibrium analysis in the size-based-policy setting, while it also holds in the accuracy-based-policy setting.

B. Problem Formulation

Since device i's upload time follows a normal distribution $\mathcal{N}(\mu_i,\sigma_i^2)$, the probability that i successfully uploads its model within a specific time τ_i can be expressed as Eq. (22).

$$F_i(\tau_i)=\int_{-\infty}^{\tau_i}\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\}\mathrm{d}x \quad (22)$$

After a device i decides on its training time t_i, it has a time period of $T-t_i$ for uploading. Therefore, its model can be successfully uploaded with the probability of $F_i(T-t_i)$. Any update after the deadline T will not be accepted by the server. Thus, with a probability of $F_i(T-t_i)$, device i can contribute a set of data $\alpha_i=\beta_i t_i$ to the global model, otherwise, its data contribution will be 0. Since all other devices follow the same principle to participate in this game as well, device i can estimate that the total data contributed by other devices would be $\sum_{j\neq i}\beta_j t_j F_j(T-t_j)$ in expectation, which we denote as $\hat{\alpha}_{-i}$ for simplification. Thus, if device i successfully uploads its model, then the system-wide data contribution is $\hat{\alpha}_{-i}+\beta_i t_i$. Based on the analysis above, we reformulate the optimization problem for an individual device i.

Problem 6 (OP$_{\text{DEVICE}}$)**.**

$$\textit{maximize} \quad u_i(t_i,\boldsymbol{t}_{-i})=R\frac{\alpha_i F_i(T-t_i)}{\hat{\alpha}_{-i}+\alpha_i}-c_i t_i, \quad (23a)$$
$$\textit{subject to} \quad 0\le t_i<T. \quad (23b)$$

In fact, each device's utility function fails to satisfy the quasi-concavity condition in the strategy space. However, a non-concave game still possesses a pure strategy Nash equilibrium when meeting some specific conditions which is given in Lemma 5 [5].

Lemma 5. *Let $\Gamma=\left\{\mathbb{N},(A_i)_{i\in\mathbb{N}},(U_i)_{i\in\mathbb{N}}\right\}$ be an N-player non-zero-sum game in normal form, where $\mathbb{N}$ represents the player set, A_i represents the i-th player's feasible strategies, which is a non-empty compact convex subset of the Euclidean space, and U_i represents the i-th player's utility function. Assume that for each $i\in\mathbb{N}$:*

(1) A_i is some closed interval of the real line,
(2) $U_i(.)$ is continuous on A_i,
(3) For each $\boldsymbol{x}_{-i}\in\boldsymbol{A}_{-i}$, there exists a local maximum of $U_i(\boldsymbol{x}_{-i},.)$, and this local maximum is also a global maximum,

then the game Γ possesses a pure-strategy Nash equilibrium.

Theorem 6. *Nash equilibrium exists in* OP$_{\text{DEVICE}}$.

Proof. Obviously, our proposed game satisfies the conditions (1) and (2). Thus, we now prove that $\forall\mathbf{x}_{-i}$, $u_i(t_i,\mathbf{t}_{-i})$ has a local maximum over its strategy domain $[0,T)$ and that this local maximum is also a global maximum of $u_i(t_i,\mathbf{t}_{-i})$ over its feasible domain.

For each feasible $\mathbf{t}_{-i}$, we start to analyze the function $u_i(t_i,\mathbf{t}_{-i})$'s monotonicity with respect to t_i. We show the first-order derivative of u_i in the below

$$\frac{\partial u_i}{\partial t_i}=R\frac{\beta_i F_i(T-t_i)\hat{\alpha}_{-i}-\beta_i t_i f_i(T-t_i)\left(\hat{\alpha}_{-i}+\beta_i t_i\right)}{\left(\hat{\alpha}_{-i}+\beta_i t_i\right)^2}-c_i$$

Obviously, $\frac{\partial u_i}{\partial t_i}$ is continuous over its feasible domain. We show the signs of $\frac{\partial u_i}{\partial t_i}$ on the boundary points over its strategy space.

$$\frac{\partial u_i}{\partial t_i}(0,\mathbf{t}_{-i})=\frac{R\beta_i F_i(T)}{\hat{\alpha}_{-i}}-c_i \quad (24)$$
$$\frac{\partial u_i}{\partial t_i}(T,\mathbf{t}_{-i})=R\frac{\beta_i F_i(0)-\beta_i t_i f_i(0)\left(\hat{\alpha}_{-i}+\beta_i T\right)}{\left(\hat{\alpha}_{-i}+\beta_i T\right)^2}-c_i \quad (25)$$

Since Eq. (24) is positive and Eq. (25) is negative, there must exist a certain $t_i^*\in(0,T)$ so that $\frac{\partial u_i}{\partial t_i}(t^*,\mathbf{t}_{-i})=0$, and u_i increases in the domain $[0,t_i^*)$ and decreases in the domain $(t^*,T]$. Thus, u_i reaches its local maximum at the point t_i^*.

Next, we will prove that $u_i(t_i^*)$ is a global maximum in its feasible domain $[0,+\infty)$. When $t_i>=T$, the value of $F_i(t_i)=0$ always holds, meaning that $\frac{\partial u_i}{\partial t_i}(t_i,\mathbf{t}_{-i})<0$ holds, thus u_i is a decreasing function in terms of t_i in the domain of $[T,+\infty)$. Therefore, $u_i(t_i^*)$ is a global maximum as well. Now, we can conclude that the proposed game possesses a pure-strategy Nash equilibrium. ☐

T \ R	200	400	600	800	1000
100	32	64	90	90	90
120	32	64	96	110	110
160	32	64	96	128	150

(a) Size-based policy.

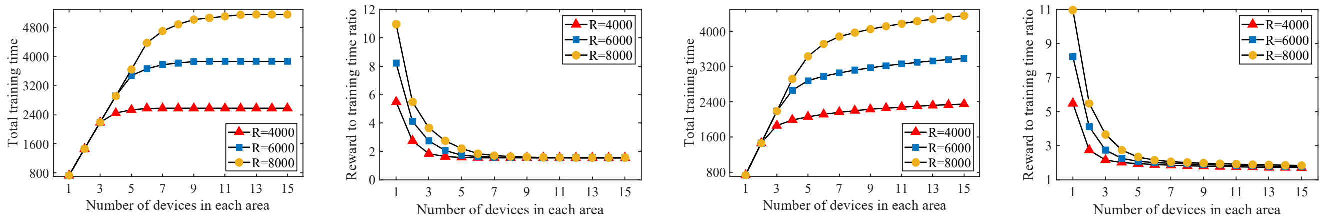
T \ R	200	400	600	800	1000
100	31	61	90	90	90
120	31	61	90	110	110
160	31	61	90	118	144

(b) Accuracy-based policy ($\theta = 10$, $\lambda = 8 \times 10^{-6}$).

T \ R	200	400	600	800	1000
100	29	54	76	90	90
120	29	54	76	97	110
160	29	54	76	97	117

(c) Accuracy-based policy ($\theta = 10$, $\lambda = 4 \times 10^{-5}$).

TABLE II: Homogeneous follower subgame Nash equilibrium under different rewards and deadlines where $(N, \beta, c, \tau) = (5, 200, 1, 10)$.



(a) Size-based policy. (b) Size-based policy. (c) Accuracy-based policy. (d) Accuracy-based policy.

Fig. 3: Impact of device number under $T = 160$.

t \ R	200	400	600	800	1000
t_1	45.8	91.5	110	110	110
t_2	29.1	58.3	91.4	110	110
t_3	29.1	58.3	91.4	95	95
t_4	12.5	25	43.4	80	110
t_5	12.5	25	43.4	80	105
sum	**129**	**258.1**	**379.6**	**475**	**530**

(a) Size-based policy.

t \ R	200	400	600	800	1000
t_1	36.7	64.9	89.6	110	110
t_2	25.5	47.2	66.9	85.5	106
t_3	25.5	47.2	66.9	85.5	95
t_4	16.1	33.1	49.3	64.8	82.5
t_5	16.1	33.1	49.3	64.8	82.5
sum	**119.9**	**225.5**	**322**	**410.6**	**476**

(b) Accuracy-based policy($\theta = 10$, $\lambda = 4 \times 10^{-5}$)..

TABLE III: Heterogeneous follower subgame Nash equilibrium ($T = 120$).

VI. Simulation

Our evaluation includes three parts. First, we examine how the server and the devices (Subsection VII.A and B) decide their optimal strategies. Second, we compare the game-driven market equilibrium and the optimal social welfare to confirm our PoA lower bound (Subsection VII.C). Lastly, we analyze how the upload channel jitters influence the achieved equilibrium (Subsection VII.D). We conduct our experiments using Tensorflow 1.9 (to get fine-tuned machine-learning related parameters) and Matlab R2019b (to help the server and devices make decisions) on Ubuntu 16.04 LTS.

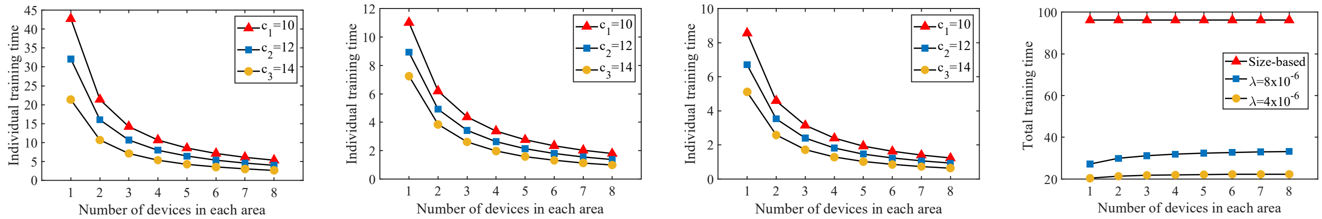
A. Follower Subgame Nash Equilibrium

In this part, we will first analyze Nash equilibrium achieved among all devices. We will discuss how different parameters will affect the devices' equilibrium strategies.

1) Parameters from the server side: The server can determine its deadline T, its reward R, and its reward policy, *i.e.*, size-based or accuracy-based. In the following, we focus on investigating how those parameters affect the device-side equilibrium. We start with a simple homogeneous-device setting, where $(N, \beta, c, \tau) = (5, 200, 1, 10)$. In Table II, we show the impact caused by different decisions on deadline T, the reward R, and the reward policy. Note that, R may not be its equilibrium value. Obviously, the increase of R's value is the main driven power for all devices to extend their training time. In the size-based reward policy, we can even observe the linear relation between the reward and the training time in some case. However, after devices reach their optimal training time, the server cannot push them to train longer by providing more rewards. This indicates that, the server should carefully determine its reward value to avoid useless monetary invest. This is an important reason why we utilize Stackelberg model since it adds the leader level to ensure the server's benefit.

By comparing Table II(a) to Table II(b) and Table II(c), we can conclude that, in most cases, the size-based reward policy leads devices to train for a longer time if other parameter values are identical, and thus, bringing more benefit to the server. By comparing Table II(b) and Table II(c), we also see the influence caused by the accuracy measurement function. Generally, an accuracy measurement function with a higher diminishing return will motivate devices for a longer time training. The accuracy measurement functions in Table II(b) and Table II(c) are transformed based on the results obtained by training the datasets Reddit and Celeba, respectively.

2) Number of Participating Devices: In this part, we investigate the impact caused by the number of participating devices N. We assume all devices are evenly distributed in 5 areas. All devices have the same computation speed and devices located in the same area enjoy the identical unit cost and upload time. The detailed settings are given as $T = 160$, $\beta = 200$, $(c_1, c_2, c_3, c_4, c_5) = (1, 1.2, 1.2, 1.4, 1.4)$, and $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5) = (10, 10, 25, 10, 15)$. We change the number of devices in each area so that the total device number ranges from 5 to 75 and we show the device number impact in Fig. 3. According to Fig. 3(a) and Fig. 3(c), we can conclude that, in the beginning, the increase on the device number can result in a longer total training time, *i.e.*, the sum of all devices' training time. When reaching some point, the increasing trend stops, meaning that the newly joining devices only split the reward with the existing devices while bringing no benefits to the system. As we can see in Fig. 3(b) and Fig. 3(d), the ratio between the reward R and the total training time converges to the same value in the end. Thus, blindly recruiting more



(a) Size-based policy. (b) Accuracy-based policy: λ=8×10^{-6}. (c) Accuracy-based policy: λ=4×10^{-6}. (d) Size vs Accuracy.

Fig. 4: Impact of device number under $T = 80$.

devices cannot increase the global model's accuracy while bringing more ordination work to the server.

3) Device Parameters: Now we study how the values of (β, c, τ) affect individual devices' equilibrium strategies. Here, we still apply the five-area setting we mentioned in the above, while we assume there is only one device in each area, *i.e.*, $N = 5$. We find each device's equilibrium strategy under different system parameters. Table III shows the results under the size-based reward policy and the accuracy-based reward policy, respectively. Based on these two tables, we can see that the training time heavily depends on a device's computation cost to speed ratio when other conditions are identical. Devices with lower cost-to-speed ratio and less upload time tend to have a longer training time under any reward policy.

B. Leader-Follower Stackelberg Equilibrium

Based on the device-side analysis, we further study the optimal strategy on the server side to obtain the desired Stackelberg equilibrium. We consider a three-area setting and the detailed settings are given as $T = 80$, $\beta = 10000$, $\tau = 15$, and$(c_1, c_2, c_3) = (10, 12, 14)$. We investigate the Stackelberg equilibrium under different reward policy. We show the device-side equilibrium strategies in Fig. 4. And we can find that the total training time in each area is almost fixed when increasing the device number in these areas. Although we change the number of devices in each area, the server's optimal reward is almost the same. The server's optimal reward is around 1731, 626, and 408, for Figs. 4(a), (b), and (c) respectively. According to Fig. 4(d), we can observe the devices' total training time is positively related to the server's reward policy. The size-based reward policy motivates devices to train for longer time as each second has the same value while the accuracy-based reward policy makes the later time less valuable, so that all devices tend to train for less time under this policy.

C. Price of Anarchy

In this section, we want to compare the social welfare created by different model designs.The definition of social welfare is the difference between the global model satisfaction and the total cost on the device side. We focus on the optimal control model and our proposed Stackelberg game. The optimal control means no money incentive: all devices are forced to follow the central server's scheduling. We consider a homogeneous-device setting. We show the obtained social welfare of these two models under different device numbers in Fig. 5. We can see that the social welfare yielded by

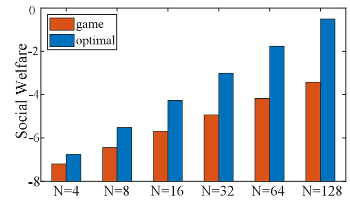


Fig. 5: Social welfare: game-driven vs optimal.

our proposed Stackelberg game is always more than half of the optimal social welfare under a central controller, which confirms our theoretical analysis on the lower bound of PoA.

D. Uncertainty in Upload Time

In this part, we investigate the impact caused by the channel instability. We still apply the homogeneous-device setting, where $(N, \beta, c, \tau) = (5, 200, 1, 10)$. We use different values of σ to reflect how unstable the communication channels are. Note that, $\sigma = 0$ is the special case, representing that the upload time is fixed as 10. According to Table IV, we can conclude that, the uncertain upload time makes devices spend more time on the local training.

VII. Related Work

1) Federated Learning: As there is more and more attention on privacy, federated learning has become one of the essential concepts in modern machine learning. Existing works in this research field can be divided into two directions. In one direction, researchers focus on solving the global model accuracy decreasing caused by device heterogeneity [6, 7] in terms of hardware, network connectivity, and battery power, and data heterogeneity among all devices [8–10]. This paper focuses on how to improve the global model accuracy by motivating all participating devices to train more data locally. Some literature also considers dealing with the coordination and operation problems in such a system, such as the communication bottleneck [11, 12] and the trust and truthfulness [13] between the server and devices.

2) Incentive Mechanism Design for Mobile Crowdsourcing: An effective incentive mechanism is indispensable in mobile crowdsourcing tasks. Most solutions integrate online auction and game theory techniques for mechanism design [14, 15]. There also exist some works dealing with incentive mechanisms in the federated learning system. In [16], the authors present an incentive mechanism called FMore with multi-dimensional procurement auction to select high-quality training nodes. [17] utilizes contract theory in order to motivate high-reputation workers to join model training. Our paper utilizes a Stackelberg game, where the server acts as a leader and

σ \ R	100	200	300	400	800
0	16	32	48	64	125
1	23	34.5	48.4	64	128
10	27	38	50	64.1	128

(a) Size-based policy.

σ \ R	100	200	300	400	800
0	15.8	31.2	46.3	61.1	117.6
1	17.8	31.3	46.3	61.1	117.6
10	24.6	35.9	46.8	61.1	117.6

(b) Accuracy-based policy ($\theta = 10$, $\lambda = 8 \times 10^{-6}$).

σ \ R	100	200	300	400	800
0	15.1	28.8	41.7	53.8	97.3
1	17.7	28.9	41.7	53.8	97.3
10	23.9	34.5	43.1	53.9	97.3

(c) Accuracy-based policy ($\theta = 10$, $\lambda = 4 \times 10^{-5}$).

TABLE IV: Homogeneous follower subgame Nash equilibrium under different rewards and deadlines where $(N, T, \beta, c, \tau) = (5, 140, 200, 1, 15)$.

provides rewards based on devices' individual contributions to motivate each device to feed its local model with more data in each iteration. We utilize utility theory, which has been widely applied to decision making [18, 19], and design suitable utility functions for both the server and the devices. Besides, our model is more practical as we take the training deadline and device upload time into consideration.

3) Price of Anarchy: In algorithmic game theory, the price of anarchy (PoA) is defined as the ratio of the social cost of a worst Nash equilibrium to that of a social optimum (*i.e.*, an assignment of strategies to players achieving optimal social cost). This highly successful and influential concept is frequently thought of as the standard measure of the potential efficiency loss due to individual selfishness, when players are concerned only with their own utility and not with the overall social welfare. Lots of works dealing with communication network problems either use this concept to measure the efficiency their methods can achieve [20, 21] or utilize this concept as the system design goal [22, 23]. In this paper, we investigate the existence of a pure strategy equilibrium in a resource management game and measure the inefficiency of equilibria by the price of anarchy. We show that the lower bound on the price of anarchy is 0.5 for our proposed solution.

VIII. Conclusion

In this paper, we utilize a Stackelberg game to model the interaction between a server and all participating devices in a federated learning system. We aim to find the server's optimal reward and each device's optimal training time for the purpose of individual utility maximization. Our model takes both the server-side deadline and the device-side upload time into consideration. We consider two different reward policies, *i.e.*, size-based and accuracy-based, and investigate how they affect the equilibrium achieved in the whole system. We prove that the proposed game is a valid utility game, which has a lower bound of 0.5 on the PoA. We also extend our model by adding uncertainty in the upload time. We show that devices spend more time on local training in the variable-upload-time setting. Our evaluation results validate the proposed models and theoretical results.

References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, 2020.

[2] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society*, 1965.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] A. Ziad, "Nash equilibria in pure strategies," *Bulletin of Economic Research*, 2003.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[7] Z. Charles and D. Papailiopoulos, "Gradient coding using the stochastic block model," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018.

[8] L. Corinzia and J. M. Buhmann, "Variational federated multi-task learning," *arXiv preprint arXiv:1906.06268*, 2019.

[9] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, "Adaptive gradient-based meta-learning methods," in *Advances in Neural Information Processing Systems*, 2019.

[10] Y. Zhang and M. Guizani, *Game theory for wireless communications and networking*, 2011.

[11] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, 2021.

[12] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[13] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, 2020.

[14] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013.

[15] G. Goel, A. Nikzad, and A. Singla, "Mechanism design for crowdsourcing markets with heterogeneous tasks." in *HCOMP*, 2014.

[16] R. Zeng, S. Zhang, J. Wang, and X. Chu, "Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec," *arXiv preprint arXiv:2002.09699*, 2020.

[17] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, 2019.

[18] P. C. Fishburn, "Utility theory," *Management Science*, 1968.

[19] J. Wu, M. Lu, and F. Li, "Utility-based opportunistic routing in multi-hop wireless networks," in *IEEE 28th International Conference on Distributed Computing Systems*. IEEE, 2008.

[20] H. Youn, M. T. Gastner, and H. Jeong, "Price of anarchy in transportation networks: efficiency and optimality control," *Physical Review Letters*, 2008.

[21] D. Ye, L. Chen, and G. Zhang, "On the price of anarchy of two-stage machine scheduling games," *Journal of Combinatorial Optimization*, 2019.

[22] Y.-J. Chen and J. Zhang, "Design of price mechanisms for network resource allocation via price of anarchy," *Mathematical Programming*, 2012.

[23] J. R. Marden and M. Philips, "Optimizing the price of anarchy in concave cost sharing games," in *2017 American Control Conference (ACC)*. IEEE, 2017.