Reinforcement Learning Based Incentive Mechanism for Federated Meta Learning: A Game-Theoretic Perspective

Shenglv Zhang[a], Yuren Zhou[b], Haohao Qu[a], Yiting Zhu[a], Linlin You[a*]
[a] School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, China
[b] Engineering Product Development, Singapore University of Technology and Design, Singapore
{zhangshlv, quhaoh}@mail2.sysu.edu.cn, {zhuyt25, youllin}@mail.sysu.edu.cn, {yuren_zhou}@outlook.com

2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3503-9744-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTAI56018.2022.00176

***Abstract*—Federated learning (FL) is a novel decentralized machine learning mechanism, which bridges data silos to train a global model by utilizing data and computation power of local clients in a privacy-preserving way. Moreover, to handle heterogeneous data across domains, tasks and partities, federated meta-learning (FML) has been proposed, which leverages the fast adaptation of meta-learning for transferable and customizable models. Nevertheless, most of the existing studies focus on providing personalized models for different users, leaving other important issues not well solved, especially, incentive mechanisms, which are used as a common foundation for FML to attrack and maintain high-quality and reputable clients. To fill the gap, this paper proposes a learning-based incentive mechanism for FML to motivate local clients to participate in the data federation. First, we propose to reward clients according to the amount of data they contribute to the model training. Then, to analyze the behaviors of model owner and local clients, we formulate the incentivized training task as a Stackelberg game, and design a method based on reinforcement learning (RL) to learn optimal pricing and participating strategies for the task publisher and the local clients, respectively. Lastly, extensive experiments are conducted to demonstrate the efficiency and effectiveness of the proposed RL-based incentive mechanism, which assists multiple parties to reach the equilibrium of the game designed for FML.**

***Index Terms*—Reinforcement Learning, Stackelberg Game, Federated Meta-Learning, Incentive Mechanism**

I. Introduction

Along with the rapid development of Internet of Things (IoT), massive data are sensed and preserved at the network edge [1] [2]. Besides the big data, the ever-growing computation power enables the shifting of conventional methods that need to transfer data to the cloud for centralized model training to decentralized approaches, called federated learning (FL), which can bridge the data silos caused by regulations about data security and use privacy, and harness idle resources at the edge to reduce the burden of the central server [3] [4] [5]. In general, FL can create a collaborative community to foster the development and deployment of artificial intelligence in various IoT systems and services.

Moreover, within FL, instead of transmitting raw data to the server, each learning participant, called FL client, can train the local model by using its local data and only upload the model parameters for model aggregation [6]. Since FL was proposed, it has been used in various fields, such as medical imaging [7], monitoring of students' learning status [8], smart home services [9], and autonomous transportation systems [10] [11]. However, these applications still face the challenge of dispersed and low-sample data generated in the distributed and isolated environment [12] [13]. To address this issue, meta-learning that can enable fast knowledge adaptation is combined with FL, and thus, federated meta-learning (FML) starts to be discussed.

Even though current studies can assist the learning of transferable and customizable models for different tasks and users, it is not well discussed about how to maintain a stable and reputable learning community for FML, e.g., by studying related incentive mechanisms, as the model requester needs to give a sufficient reward for participants, otherwise rational local clients will have no motivation to join the model training due to additional computing and communication costs. To address such an issue, game theory as a powerful framework has been used to analyze the interactions among multi-players who act in their own interests to determine the optimal strategies for the learning publisher and participants [14] [15]. Especially, Stackelberg game has been extensively discussed to design related incentive mechanisms [16] [17] [18]. But unfortunately, the equilibrium analysis of Stackelberg game requires the condition of perfect and complete information, which does not match the actual scenario of FML and makes it inappropriate to use directly. Moreover, another problem that needs to be solved is the fair distribution of rewards among clients. In the context of FML, an intuitive idea is to distribute rewards according to the contributions of clients, e.g., assigning the reward based on data quality by using Shapley values [19]. However, it is difficult to apply the Shapley value method in practice owing to its high computational complexity. As a tradeoff, a contribution index as an approximation of Shapley value is proposed [20], but it is still computational complex.

In this article, we propose a crowdsourced incentivized FML framework by distributing rewards among clients according to the amount of data they contribute to model training. As FML can work on data in non-independent and non-identical data distribution, distributing rewards according to data quantity becomes a fitting method. In our proposed framework, there

* Corresponding author: Linlin You, e-mail: youllin@mail.sysu.edu.cn

2375-0197/22/$31.00 ©2022 IEEE
DOI 10.1109/ICTAI56018.2022.00176

are three main types of participants: 1) A task publisher, who wants to train an AI model, publishes FML tasks with rewards. 2) A coordinator, who accepts the task, charges to organize model training and distributes rewards fairly. 3) A set of clients, each of which has redundant computation power and task-related data. In the framework, the task publisher and clients both have their own interests in maximizing the utility. We formulate the system model as a two-stage Stackelberg game and focus on proving the uniqueness of game equilibrium in each sub-game. Then to address the challenge of the incomplete information problem in actual scenarios, we design a method based on deep reinforcement learning (DRL) and multi-agent reinforcement learning (MARL), both of which are powerful tools to learn the optimal strategy from historical training records.

The main contributions of this paper are summarized as follows:

- We design a crowdsourcing incentive framework for FML to enable long-term cooperation. The reward distribution in the framework is designed based on the assumption that the contribution to the global model is directly related to the amount of local data and computing resources utilized by the clients.
- We formulate the proposed system model as a two-stage Stackelberg game and derive the unique equilibrium through a rigorous game-theoretic analysis under the complete information condition.
- To tackle the problem of incomplete information in actual scenarios, we propose the RL-based incentive mechanism, which can learn optimal strategies for the task publisher and clients without any prior information. As shown by the evaluation, the proposed RL-based method can make decisions for the equilibrium consistent with the theoretical analysis

The remainder of the article is organized as follows. Section II summaries related works. Then, Section III presents the system model and the problem formulation, which is analyzed by a two-stage game discussed in Section IV. Accordingly, Sections V and VI present and evaluate the proposed RL-based incentive mechanism, respectively. Finally, Section VII concludes this article and sketches the future works.

II. Related Works

A. Federated Meta Learning

There has been some research focused on providing personalized models for different users through FML to solve the problem of data heterogeneity encountered in FL. Chen *et al.* [21] have proposed FedMeta where a parameterized algorithm is shared instead of a global model to handle the statistical and systematic challenges in FL. Jiang *et al.* [22] have proposed inserting a meta-learning fine-tuning phase after the FedAvg algorithm phase to provide a reliable initialization model. Zhang *et al.* [23] have proposed FedFomo which gets the best model combination through calculating the interaction between different clients to better adapt to clients. Acar *et al.* [24] have proposed gradient correction methods, and explicitly de-bias the meta-model in the distributed heterogeneous data setting to learn personalized device models.

B. Incentive Mechanism Based On Stackelberg Game

Incentive mechanisms based on Stackelberg game have been extensively studied. Sarikaya *et al.* [25] considered the CPU energy consumption of participants, obtained Stackelberg equilibrium(SE), and set an incentive mechanism to reduce the time spent in global model training. Khan *et al.* [26] proposed a Stackelberg game approach, which enables participants to strategically set local iterations to maximize their utility. Pandey *et al.* [27] proposed a crowdsourcing framework and constructed an incentive mechanism based on Stackelberg game to maximize the interests of both the coordinator and the participants. Lee *et al.* [28] design a novel market model of the distributed learning resource management mechanism, give a unique SE point as a closed form, and reveal that the SE solution maximizes the utility of all market participants. However, the above work is based on the perfect and complete information condition, which makes it not directly applicable to FML due to unshared decisions. In this paper, we use DRL and MARL to solve the problem.

III. System Model And Problem Formulation

A. System Model

Fig.1 demonstrates the studied system, and there are three main types of components.

- **FML task publisher:** It can be an organization or company that needs FML models but without enough data. Accordingly, it can publish the task with a sufficient reward r to hire a coordinator to assist the model training and motivate local clients to participate.
- **FML coordinator:** The coordinator is considered as a parameter server that resides in the cloud, who receives the task and takes charge of organizing a group of local clients for model training. In this model, the coordinator charges a fixed fee for its efforts.
- **FML clients:** It utilizes its local resources, i.e., data and computing powers, to perform local training on the demand of coordinator, and delivers the encrypted information, i.e., gradients trained from data. In this model, we consider a set of local clients, denoted as $\mathbb{N} = \{1, 2, ..., N\}$, each of which maintains a data set D_n and $D_n \cap D_m = \emptyset \ \forall m \in \mathbb{N}, m \neq n$.

In the following, we present the mathematical model for the utility of the task publisher and local clients in the system. Since the coordinator takes a fixed charge, which means it does not participate in the game, we do not discuss its utility.

1) Utility Function of Task Publisher: Our design of the utility function for the task publisher considers two terms. The first term represents the economic benefits attained from the FML model training task. The second term represents the total reward that the task publisher pays for the assistance of the coordinator and the participation of the clients. In summary, the utility function is designed to balance the benefit and the

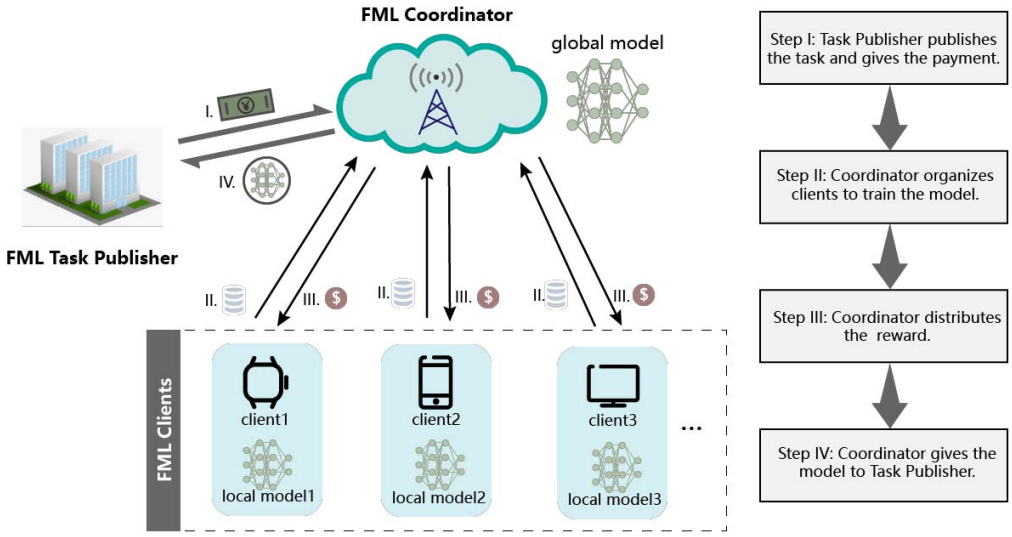


Fig. 1. The proposed system model.

cost of the learning task. We define the utility function of the task publisher as $u(r)$:

$$u(r) = \omega\left(1 - \gamma e^{-\beta \sum_{n=1}^{N} d_n}\right) - r \tag{1}$$

where r is the total reward given by the task publisher to train the FML model, ω is a conversion parameter from model performance to profits, d_n donates the data quantity used by client i to train the model, $\gamma\,(>0)$ and $\beta\,(>0)$ are weight factors. We use $\left(1 - \gamma e^{-\beta \sum_{n=1}^{N} d_n}\right)$ to represent the global model performance by assuming that the model performance gain has diminishing returns with respect to data quantity. Intuitively, there exists a limit to model performance where $\lim_{\sum_{n=1}^{N} d_i \to \infty}\left(1 - \gamma e^{-\beta \sum_{n=1}^{N} d_n}\right) = 1$

2) Utility Function of Clients: The design of the utility function for clients also considers two terms. The first term represents the revenue gotten through participating in the learning task. The second term represents the cost of model training due to additional computation and communication, which is proportional to the amount of data used for training. For $i \in \{1, 2, ..., N\}$, we use $u_i(d_i, d_{-i})$ to donate the utility of local client i:

$$u_i(d_i, d_{-i}, r) = \frac{d_i}{\sum_{n=1}^{N} d_n} r(1-\mu) - d_i\left(c_i^{cmp} + c_i^{com}\right) \tag{2}$$

where $d_{-i} = \{d_1, d_2, ..., d_{i-1}, d_{i+1}, ..., d_N\}$ is the training data size of others except client i, $\mu \in [0, 1]$ denotes the rate of the fixed fee charged by the coordinator over the total reward, c_i^{cmp} and c_i^{com} represent the computation cost and communication cost per unit of data sample, respectively.

B. Problem Formulation

We formulate the incentive mechanism for FML as a two-stage Stackelberg game [29] in each training round. In Stackelberg game, the leader needs to predict the influence of his decision on the followers and make the decision first, then the followers make their own decision according to the leader's decision. In the proposed system model, the task publisher is the leader, and the clients are the followers. Accordingly, there are two main stages in this mechanism.

In the first stage, the task publisher, i.e., the leader, determines its optimal payment strategy to maximize its utility. The optimization problem is defined as follows:

$$\text{Sub} - \text{game } G_I : r^* = \arg\max_{r>0} u(r) \tag{3}$$

In the second stage, given the reward $r(1-\mu)$, the local clients determine their optimal training strategies to maximize the utility. Accordingly, the optimal training strategy can be obtained by solving the following optimization problem:

$$\text{Sub} - \text{game } G_{II} : d_i^* = \arg\max_{d_i \geq 0} u(d_i, d_{-i}, r) \tag{4}$$

Note that the second stage sub-game can be considered as a non-cooperative game where each client seeks to maximize their own interests selfishly. Intuitively, for any given reward $r(1-\mu)$ and other clients' training strategies d_{-i}, client i determines an optimal strategy d_i to maximize its utility by considering the revenue and cost for the model training.

Such that, a Nash equilibrium exists among clients, while the strategy profile is stable and no participant can improve utility through change strategy. We define the Nash equilibrium of the second-stage sub-game in our proposed system model in Definition 1.

Definition 1: A set of strategies $d^* = (d_1^*, d_2^*, ..., d_n^*)$ is a strict Nash equilibrium of the second-stage sub-game if for any client i and $d_i \neq d_i^*$

$$u_i(d_i^*, d_{-i}^*, r) > u_i\left(d_i, d_{-i}^*, r\right) \tag{5}$$

for any $d_i \geq 0$

The aforementioned two sub-games form the two-stage Stackelberg game of FML. The objective of the game is to find a Stackelberg equilibrium. When the best responses of followers, i.e., the Nash equilibrium, is adopted, the leader maximizes its utility. In the next section, we derive the solution to the game.

IV. Analysis of the Two-stage Stackelberg Game

In this section, we focus on proving the uniqueness of equilibrium in each sub-game by using the backward induction to derive the analytical solutions of the Stackelberg game with complete information. First, we prove that for any given r, the second-stage sub-game has a unique Nash equilibrium in Section IV-A. Then, in Section IV-B, we prove that for task publisher, there is a unique r^* to maximize its utility as defined in Formula (3), showing that the game exists a unique Stackelberg equilibrium.

A. The Analysis of Second Stage Game for Clients

Theorem 1: For given r by task publisher, there exists a Nash equilibrium, and the optimal training strategy for client $i \in \mathbb{M} \subseteq \mathbb{N}$ is

$$d_i^* = \frac{(M-1)(1-\mu)r}{\sum_{n\in\mathbb{M}}(c_n^{com}+c_n^{cmp})}\left(1-\frac{(M-1)\left(c_i^{com}+c_i^{cmp}\right)}{\sum_{n\in\mathbb{M}}(c_n^{com}+c_n^{cmp})}\right) \tag{6}$$

where $\mathbb{M}$ donates a set of clients that contribute $d_i > 0$ and M donates the number of set elements.

Proof 1: First, We proof the existence of Nash equilibrium by using Lemma 1 [29].

Lemma 1: There is a Nash equilibrium, if the following conditions are satisfied:

1) The player set is finite.
2) The strategy sets are closed, bounded, and convex.
3) The utility functions are continuous and quasi-concave in strategy space.

Due to $u\left(d_i, d_{-i}, r\right)$ must be positive, based on (2), we obtain

$$d_i < \frac{r\left(1-\mu\right)}{c_i^{com}+c_i^{cmp}} \tag{7}$$

The first-order derivative of $u\left(d_i, d_{-i}, r\right)$ with respect to d_i is

$$\frac{\partial u_i\left(d_i, d_{-i}\right)}{\partial d_i} = \frac{r\left(1-\mu\right)\left(\sum_{n\neq i} d_n\right)}{\left(\sum_{n=1}^{N} d_n\right)^2} - \left(c_i^{cmp}+c_i^{com}\right) \tag{8}$$

The second-order derivative of $u\left(d_i, d_{-i}, r\right)$ with respect to d_i is

$$\frac{\partial^2 u_i\left(d_i, d_{-i}\right)}{\partial d_i^2} = -\frac{2r\left(1-\mu\right)\sum_{n\neq i} d_n}{\left(\sum_{n=1}^{N} d_n\right)^3} < 0 \tag{9}$$

Therefore, based on Lemma 1, according to Formulas (7), (9) and the finite set of local clients in the actual scenario, we can derive that there exists a Nash equilibrium in the second-stage sub-game.

Then, to obtain the unique optimal training strategies of each client i, we set the first-order derivative of $u_i\left(d_i, d_{-i}, r\right)$ to be 0.

By solving $u_i^{'}\left(d_i, d_{-i}, r\right) = 0$, we obtain:

$$d_i = \sqrt{\frac{r\left(1-\mu\right)\sum_{n\neq i} d_n}{c_i^{cmp}+c_i^{com}}} - \sum_{n\neq i} d_n \tag{10}$$

According to the transposition of Formula (10), we can derive:

$$\sum_{i=1}^{M} d_i = \sqrt{\frac{r\left(1-\mu\right)\sum_{n\neq i} d_n}{c_i^{cmp}+c_i^{com}}} \tag{11}$$

Setting $\xi = \sum_{n=1}^{M} d_n^*$, we obtain:

$$\begin{cases} d_1^* = \xi - \frac{\xi^2\left(c_1^{com}+c_1^{cmp}\right)}{r(1-\mu)} \\ d_2^* = \xi - \frac{\xi^2\left(c_2^{com}+c_2^{cmp}\right)}{r(1-\mu)} \\ \quad\quad\cdot \\ \quad\quad\cdot \\ \quad\quad\cdot \\ d_M^* = \xi - \frac{\xi^2\left(c_M^{com}+c_M^{cmp}\right)}{r(1-\mu)} \end{cases} \tag{12}$$

Based on Formula (12), we can obtain:

$$\xi = M\xi - \frac{\xi^2\sum_{i=1}^{M}\left(c_i^{com}+c_i^{cmp}\right)}{r\left(1-\mu\right)} \tag{13}$$

Then, we derive:

$$\xi = \frac{\left(M-1\right)\left(1-\mu\right)r}{\sum_{i=1}^{M}\left(c_i^{com}+c_i^{cmp}\right)} \tag{14}$$

By plugging the equation in Formula (14) into Formula (12), we finally have:

$$d_i^* = \frac{(M-1)(1-\mu)r}{\sum_{n\in\mathbb{M}}(c_n^{com}+c_n^{cmp})}\left(1-\frac{(M-1)\left(c_i^{com}+c_i^{cmp}\right)}{\sum_{n\in\mathbb{M}}(c_n^{com}+c_n^{cmp})}\right)$$

■

B. The First Stage Game for Task Publisher

Theorem 2: The optimal payment strategy $r^* > 0$ for the task publisher is unique by knowing the existence of a unique Nash equilibrium among clients under any given r.

Proof 2: By setting $D = \sum_{i=1}^{M} d_i^*$ and $p(D) = \omega\left(1-\gamma e^{-\beta D}\right)$, we have:

$$u\left(r\right) = p\left(D\right) - r \tag{15}$$

The first-order derivative of $u(r)$ is:

$$\begin{aligned} \frac{\partial u\left(r\right)}{\partial r} &= p^{'}\left(D\right)\frac{\partial D}{\partial r} - 1 \\ &= p^{'}\left(D\right)\sum_{i=1}^{M}\frac{\partial d_i^*}{\partial r} - 1 \\ &= p^{'}\left(D\right)\sum_{i=1}^{M}\frac{(M-1)(1-\mu)}{\sum_{i\in\mathbb{M}}(c_i^{com}+c_i^{cmp})} \\ &\times\left(1-\frac{\left(M-1\right)\left(c_m^{com}+c_m^{cmp}\right)}{\sum_{i\in\mathbb{M}}(c_i^{com}+c_i^{cmp})}\right) - 1 \end{aligned} \tag{16}$$

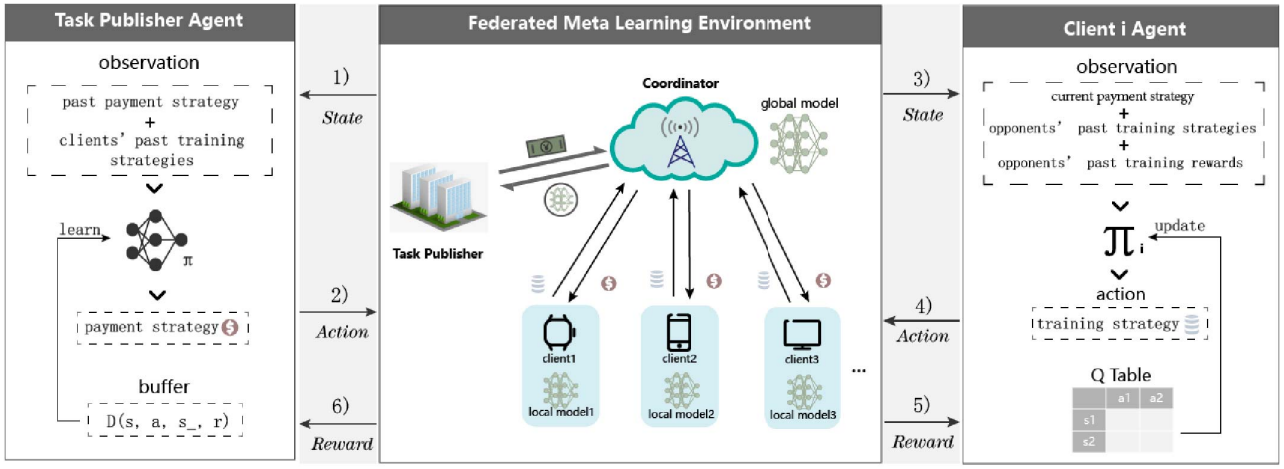


Fig. 2. Workflow of the RL-based method.

Based on the first-order derivative, we derive the second-order of $u(r)$ derivative as:

$$\frac{\partial u^2(r)}{\partial^2 r} = p''(D)\sum_{i=1}^{M}\frac{(M-1)(1-\mu)}{\sum_{i\in\mathbb{M}}(c_i^{com}+c_i^{cmp})} \times\left(1-\frac{(M-1)(c_m^{com}+c_m^{cmp})}{\sum_{i\in\mathbb{M}}(c_i^{com}+c_i^{cmp})}\right) \tag{17}$$

Note that $\frac{(M-1)(1-\mu)}{\sum_{n\in\mathbb{M}}(c_n^{com}+c_n^{cmp})}\left(1-\frac{(M-1)(c_m^{com}+c_m^{cmp})}{\sum_{n\in\mathbb{M}}(c_n^{com}+c_n^{cmp})}\right)=\frac{d_i^*}{r}>0$. The first-order derivative of $p(D)$ is:

$$p'(D)=\omega\gamma\beta e^{-\beta D} \tag{18}$$

The second-order derivative of $p(D)$ is:

$$p''(D)=-\omega\gamma\beta^2 e^{-\beta D}<0 \tag{19}$$

Therefore, as the second-order derivative $\frac{\partial^2 u(r)}{\partial^2 r}<0$, the utility function of task publisher $u(r)$ is a concave function of r for $r>0$. In addition, the value of $u(r)$ is 0 for $r=0$ and goes to $-\infty$ when r goes to ∞. Hence, there exists a unique optimal payment strategy for task publisher r^* for a unique Stackelberg equilibrium. ■

V. RL-Based Incentive Mechanism

In this section, we study the RL-based incentive mechanism designed to solve the challenge of incomplete information in FML. We describe how to transform the aforementioned Stackelberg game into a learning task. Specifically, we model the FML training task as a Markov decision process.

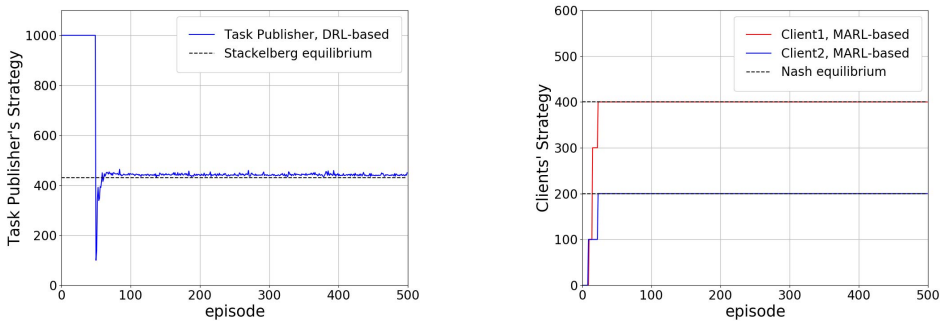
A. Learning Mechanism

The proposed RL-based incentive mechanism adopts a DRL model based on the deep deterministic policy gradient (DDPG) [30] for the task publisher and a MARL model based on Nash Q-Learning [31] for local clients. The workflow of the DRL-based incentive mechanism is illustrated in Fig. 2. At each training period t: 1) The agent representing the task publisher observes the interaction histories with clients; 2) According to the histories, the publisher agent determines the payment strategy to motivate clients; 3) After the payment is given, to determine how to participate, client agents learn optimal strategies in a non-cooperative game simulation environment. In the simulation environment, each client agent observes the current payment and the past training strategies of opponents; 4) Client agents determine their training strategies based on the observations; 5) After the action, client agents receive a simulated reward from the environment and continues step 3), 4), and 5) to update their policies until they get the Nash equilibrium. When the client agents learn the Nash equilibrium, they finally determine the amount of data used for the training model. Until the final action is done, clients receive the real reward; 6) The publisher obtains its reward. Till now, the t-th training period ends and the publisher agent learns from the interaction histories with clients.

B. RL Model Design

1) State Space of Task Publisher: Under the incomplete information environment, the task publisher can only make a decision based on the observation of past interaction records with local clients. Therefore, the observation of task publisher consists of two components, including the past payment strategies $R_L=\{r_{t-L},r_{t-L+1},...,r_{t-1}\}$ and the clients' total data quantity used for model training $T_L=\{T_{t-L},T_{t-L+1},...,T_{t-1}\}$. Hence, we define the state input of the publisher as a union of above two parts $s_t=\{R_L,T_L\}$ where L represents the number of historical interactions observed by the publisher.

2) State Space of Clients: By observing the given reward, each client needs to determine its data quantity used to train the model. To make the optimal strategies, clients need to learn the Nash equilibrium of the second-stage sub-game in a simulation environment first. In the



(a) (b)

Fig. 3. Convergence of RL-based incentive mechanism. (a) Pricing Strategy of task publisher. (b) Training Strategies of clients.

simulated second-stage game, each client could observe the current payment $r_t(1-\mu)$ announced by coordinator, the past training strategies of other clients $D_{-i}^{t,k-L} = \left\{d_{-i}^{t,k-L}, d_{-i}^{t,k-L+1}, ..., d_{-i}^{t,k-1}\right\}$, and rewards of other clients $U_{-i}^{t,k-L} = \left\{u_{-i}^{t,k-L}, u_{-i}^{t,k-L+1}, ..., u_{-i}^{t,k-1}, \right\}$. Hence, the state input of client i is $s_i^{t,k} = \left\{D_{-i}^{t,k-L}, U_{-i}^{t,k-L}, r_t(1-\mu)\right\}$ where k represents the k-th second-stage sub-game in the simulation environment in the t-th training round.

3) Reward of Task Publisher: After the clients make their optimal training strategies, the publisher would receive the reward from the environment. As formulated in Section III-A, we define the reward of task publisher as $u\left(r_t\right)$.

4) Reward of Clients: In the k-th simulation game of the t-th training period, each client determines its training strategy $d_i^{t,k}$ and receives the reward from the simulation environment. While considering the model formulation in Section III-A, we define the reward of client i as $u\left(d_i^{t,k}, d_{-i}^{t,k}, r_t\right)$.

5) Policy of Task Publisher: In each training period t, the publisher needs to give a sufficient $r_t \in (0, \infty)$ to motivate clients to participate in the model training. The payment strategy r_t is a continuous value. To make the optimal payment strategy, the task publisher maintains an actor network $\pi\left(r_t|s_t, \theta\right)$ and a critic network $v\left(s_t, \delta\right)$, where θ is the parameters of actor network and δ is the parameters of critic network. The policy of task publisher can be represented as $\pi\left(r_t|s_t, \theta\right) \rightarrow (0, R]$, where R denotes the maximum budget.

6) Policy of Clients: In the simulation environment of each training period t, client i needs to determine its training strategy $d_i^{t,k}$ based on the state input $s_i^{t,k}$. Since the clients' training strategy is finite and discrete, to determine the optimal training strategies, clients keep updating the Q table in the simulation environment until they reach the Nash equilibrium. The policy of client i can be marked as $\pi_i\left(d_i^{t,k}|s_i^{t,k}, Q_i\right) \rightarrow \{1, 2, ..., D\}$, where D denotes the total local data of client i.

VI. Performance Evaluation

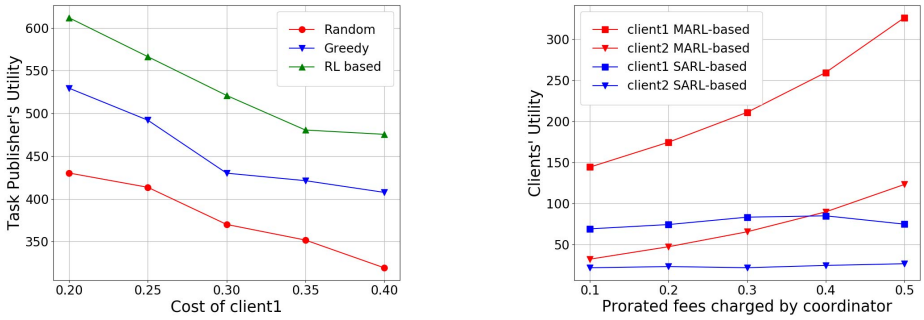
In this section, we evaluate the performance of the proposed RL-based incentive mechanism.

A. Experiments Settings

TABLE I
Experiment Parameters

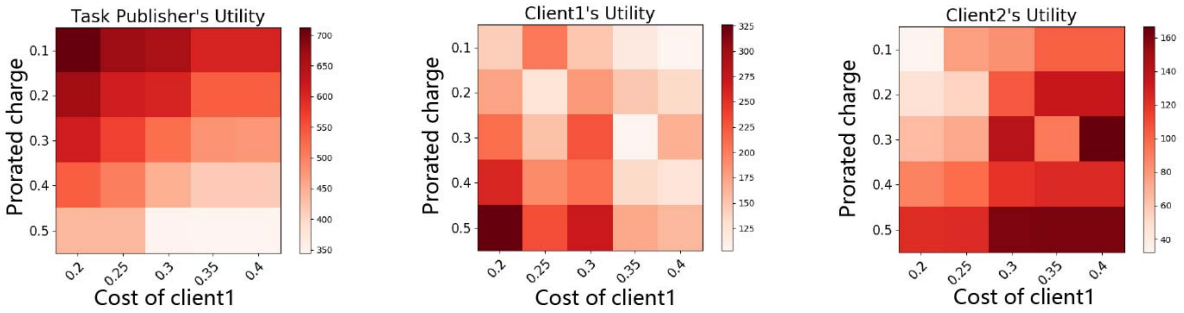
Parameter	Values
Number of task publisher	1
Number of coordinator	1
Number of local clients	2
Payment strategy of task publisher	[100, 1000]
Training strategies of local clients	$\{0, 100, 200, 300, 400, 500\}$
Weight factor γ	1
Weight factor β	500
Conversion parameter ω	1500
Prorated Fees charged by coordinator μ	[0.1, 0.5]
Client i 's communication cost, c_i^{com}	[0.1, 0.3]
Client i 's computation cost, c_i^{com}	[0.1, 0.3]
the number of interaction history observed by agent L	1
Learning rate for actor network of the task publisher agent	0.001
Learning rate for critic network of the task publisher agent	0.001
Buffer size of DDPG for task publisher agent	10000
reward discount for DDPG	0.9
Learning rate for clients agents	0.001
Discount factor for Nash Q value	0.1

The values of parameters used in the experiment are provided in Table 1. Notably, we set the task publisher' s payment strategy $r \in [100, 1000]$ and the clients' training strategies



(a) (b)

Fig. 4. Performance of the proposed incentive mechanism. (a) The utility comparison of task publishers under different methods. (b) The utility comparison of clients under different methods.



(a) (b) (c)

Fig. 5. The heatmap of utility varying the cost of client1 and chrages of coordinator. (a) Utility of task publisher. (b) Utility of client1. (c) Utility of client2.

$d_i \in \{0, 100, 200, 300, 400, 500\}$. We conduct our experiments using TensorFlow 1.9.

B. Experiments Results and Discussions

First, to demonstrate the convergence of the proposed RL-based incentive mechanism, we conduct proof experiments and derive the training curve. We set $c_1^{com} = 0.1$, $c_2^{com} = 0.3$, $c_1^{cmp} = c_2^{cmp} = 0.1$ and $\mu = 0.3$. As a result, the payment strategy of task publisher converges to Stackelberg equilibrium in Fig. 3 (A), and the training strategies of clients converge to Nash equilibrium in Fig. 3 (B), respectively. Hence, it shows that the RL-based mechanism can determine the optimal strategies for task publisher and clients through the learning.

Second, to demonstrate the efficiency and effectiveness of the proposed RL-based mechanism, we compare it with two baselines, namely:

- **The random approach**: The task publisher determines its payment strategy in each training period randomly;
- **The greedy approach**: The task publisher determines its payment strategy based on its best strategy in the past training period.

As shown in Fig. 4 (A), we can see that the proposed method can achieve the highest utility than the baselines in different conditions, i.e., client costs. Note that in the experiment, μ is set to 0.3 as the charging rate of the collaborator.

Third, as for the collaboration among clients, the MARL-based method is compared with the single-agent reinforcement learning (SARL) method in Fig. 4 (B). We can obviously find that the MARL-based method gets more utility than the SARL-based method because the SARL method cannot converge in this complex case.

Finally, we study the utility of the task publisher and clients influenced by the charging rate of the coordinator μ from 0.1 to 0.5 and the unit training cost of client1 from 0.2 to 0.4. In this experiment, we set the communication cost of client2 $c_2^{com} = 0.1$ and the computation cost of client2 $c_2^{cmp} = 0.3$. As shown in Fig. 5 (A), we find that the publisher utility decrease when the client cost and charging rate increase, e.g., when the cost is 0.2 and μ is 0.1, the publisher can obtain the utility of 711.85. However, when the cost and μ increase to 0.4 and 0.5, respectively, a much lower utility of 344.19 is obtained. In Fig. 5 (B) and (C), we observe that along with the increase in one client's cost, another client can obtain an increasing utility. In the meanwhile, along with the increase of μ, both clients can obtain more benefits, due to the collaborator tends to give more rewards to motivate FML clients.

VII. Conclusions

In this paper, we propose an incentive mechanism for FML. First, we propose to reward clients based on the quantity of data that they use for the training task. Then, we analyze the training task as a two-stage Stackelberg game between the task publisher and local clients and prove the uniqueness of the game solution under perfect and complete information conditions. Moreover, we apply RL to address the issue of incomplete information in the actual FML scenario. Finally, we conduct numerical experiments to demonstrate the efficiency and effectiveness of our proposed RL-based incentive mechanism, which can assist the learning participants to reach the theoretical equilibrium, collaboratively and respectively. In the future, we consider adopting multi-agent deep reinforcement learning (MADRL) to support more clients and a larger action space in the system to better reflect and analyze the behaviors of clients within the FML community.

Acknowledgment

This research was funded by the National Natural Science Foundation of China (62002398) and the Collaborative Innovation Center for Transportation of Guangzhou (202206010056).

References

[1] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[2] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7):6360–6368, 2020.

[3] Linlin You, Sheng Liu, Yi Chang, and Chau Yuen. A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement. *IEEE Internet of Things Journal*, 2022.

[4] Xinghua Zhu, Jianzong Wang, Zhenhou Hong, Tian Xia, and Jing Xiao. Federated learning of unsegmented chinese text recognition model. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1341–1345. IEEE, 2019.

[5] Ashneet Khandpur Singh, Alberto Blanco-Justicia, Josep Domingo-Ferrer, David Sánchez, and David Rebollo-Monedero. Fair detection of poisoning attacks in federated learning. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 224–229. IEEE, 2020.

[6] Sheng Liu, Qiyang Chen, and Linlin You. Fed2a: Federated learning mechanism in asynchronous and adaptive modes. *Electronics*, 11(9):1393, 2022.

[7] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

[8] Xiuli Zhang and Zhongqiu Cao. A framework of an intelligent education system for higher education based on deep learning. *International Journal of Emerging Technologies in Learning*, 16(7), 2021.

[9] Tianlong Yu, Tian Li, Yuqiong Sun, Susanta Nanda, Virginia Smith, Vyas Sekar, and Srinivasan Seshan. Learning context-aware policies from multiple smart homes via federated multi-task learning. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 104–115. IEEE, 2020.

[10] Linlin You, Junshu He, Juanjuan Zhao, and Jiemin Xie. A federated mixed logit model for personal mobility service in autonomous transportation systems. *Systems*, 10(4):117, 2022.

[11] Linlin You, Junshu He, Wei Wang, and Ming Cai. Autonomous transportation systems and services enabled by the next-generation network. *IEEE Network*, 36(3):66–72, 2022.

[12] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

[13] Weiting Zhang, Dong Yang, Wen Wu, Haixia Peng, Ning Zhang, Hongke Zhang, and Xuemin Shen. Optimizing federated learning in distributed industrial iot: A multi-agent approach. *IEEE Journal on Selected Areas in Communications*, 39(12):3688–3703, 2021.

[14] Yiting Zhu, Zhaocheng He, and Guilong Li. A bi-hierarchical game-theoretic approach for network-wide traffic signal control using trip-based data. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[15] Huale Li, Xuan Wang, Shuhan Qi, Yang Liu, Haojie Wang, Fengwei Jia, and Jiajia Zhang. Solving six-player games via online situation estimation. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1795–1799. IEEE, 2019.

[16] Yang Liu, Changqiao Xu, Yufeng Zhan, Zhixin Liu, Jianfeng Guan, and Hongke Zhang. Incentive mechanism for computation offloading using edge computing: A stackelberg game approach. *Computer Networks*, 129:399–409, 2017.

[17] Peng Li and Song Guo. Incentive mechanisms for device-to-device communications. *IEEE Network*, 29(4):75–79, 2015.

[18] Kaichuan Zhao, Shan Zhang, Ning Zhang, Yuezhi Zhou, Yaoxue Zhang, and Xuemin Shen. Incentive mechanism for cached-enabled small cell sharing: A stackelberg game approach. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[19] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. Measure contribution of participants in federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2597–2604. IEEE, 2019.

[20] Tianshu Song, Yongxin Tong, and Shuyue Wei. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2577–2586. IEEE, 2019.

[21] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

[22] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

[23] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020.

[24] Durmus Alp Emre Acar, Yue Zhao, Ruizhao Zhu, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Debiasing model updates for improving personalized federated training. In *International Conference on Machine Learning*, pages 21–31. PMLR, 2021.

[25] Yunus Sarikaya and Ozgur Ercetin. Motivating workers in federated learning: A stackelberg game perspective. *IEEE Networking Letters*, 2(1):23–27, 2019.

[26] Latif U Khan, Shashi Raj Pandey, Nguyen H Tran, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine*, 58(10):88–93, 2020.

[27] Shashi Raj Pandey, Nguyen H Tran, Mehdi Bennis, Yan Kyaw Tun, Aunas Manzoor, and Choong Seon Hong. A crowdsourcing framework for on-device federated learning. *IEEE Transactions on Wireless Communications*, 19(5):3241–3256, 2020.

[28] Joohyung Lee, DaeJin Kim, and Dusit Niyato. Market analysis of distributed learning resource management for internet of things: a game-theoretic approach. *IEEE Internet of Things Journal*, 7(9):8430–8439, 2020.

[29] Roger B Myerson. On the value of game theory in social science. *Rationality and Society*, 4(1):62–73, 1992.

[30] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[31] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.