Task-Load-Aware Game-Theoretic Framework for Wireless Federated Learning

Jiawei Liu, Guopeng Zhang, Kezhi Wang, and Kun Yang

***Abstract*—Federated learning (FL) can protect data privacy but has difficulties in motivating user equipment (UE) to engage in task training. This paper proposes a Bertrand-game based framework to address the incentive problem, where a model owner (MO) issues an FL task and the employed UEs help train the model by using their local data. Specially, we consider the impact of time-varying *task load* and *channel quality* on UE's motivation to engage in the FL task. We adopt the finite-state discrete-time Markov chain (FSDT-MC) to predict these parameters during the FL task. Depending on the performance metrics set by the MO and the estimated energy cost of the FL task, each UE seeks to maximize its profit. We obtain the Nash equilibrium (NE) of the game in closed form, and develop a distributed iterative algorithm to find it. Finally, simulation result verifies the effectiveness of the proposed approach.**

***Index Terms*—Machine learning, federated learning, resource allocation, Bertrand game, Nash equilibrium.**

I. Introduction

FEDERATED learning (FL) [1] is a distributed machine learning (ML) framework that allows multiple clients to collaboratively train an ML model without exposing their raw data. In FL framework, a model owner (MO) first sends a global model to edge or mobile clients. Then the clients can train the model with their local data and transmit the updated model parameters to the MO for a new aggregated model. This process iterates until the required accuracy is achieved.

Although FL protects data privacy, it causes additional energy cost of clients for model training and parameter transmission, thus limiting their motivation to engage in FL [2]. To address this issue, a Stackelberg game is proposed in [3] to encourage clients to reduce the completion time of FL tasks. In [4], a reinforcement learning-based incentive mechanism is designed to find the optimal pricing and training strategies for the clients. In [5], a Stackelberg game is proposed to motivate the MO and clients to build a high-quality FL model. However, several strong assumptions were made in the above works, e.g., the clients have invariant *task loads* and *channel quality* during an FL task. These assumptions apply only to situations where the clients are computationally powerful edge servers connected with the wired networks. When the clients are mobile UEs, the energy and computation power are normally limited, thus the influence of time-varying *task load* and *channel quality* on their motivation to engage in FL may be considered.

Against the above background, this paper proposes a Bertrand-game [6] based framework to motivate UEs to engage in FL [7] [8], which considers the changes of *task load* and *channel quality* of UEs during an FL task. Firstly, given the performance metrics set by the MO, we show that the energy cost of a UE depends not only on the *FL-task load* but also on its *existing-task load*. We characterize the dynamic of user's *future task load* as an FSDT-MC [9]. Then, the energy cost of a UE caused by multiple rounds of local training can be predicted. Furthermore, we employ another FSDT-MC to describe the *channel fading* between a UE and the MO, as it has been widely used to simulate Rayleigh fading channels [10]. Then, the energy cost of the UE caused by multiple rounds of parameter transmission can also be estimated. Depending on the predicted overall energy cost for model training and parameter transmission, the UEs can charge the MO independently for the resource usage. At the NE of the game, the MO achieves a set of specified performance metrics with the minimum total payment, while each UE seeks the best price to maximize its own profit. The contribution of this paper is as follows:

1) By integrating the FSDT-MC based *task load* and *channel quality* prediction method, a Bertrand game is proposed to help motivate UEs to engage in FL based on the estimated overall energy cost.

2) The NE of the game is addressed in closed form, and an effective iterative algorithm is designed to obtain the NE in a distributed manner.

II. System Model

The considered FL system consists of one MO and a set $\mathcal{K}$ of K mobile UEs. The MO can communicate with the UEs via direct wireless links. Each UE k ($\forall k \in \mathcal{K}$) stores a local dataset $\mathcal{D}_k$ of size $|\mathcal{D}_k|$, and the i^{th} data sample in set $\mathcal{D}_k$ is represented by $d_{k,i} = \{\mathbf{x}_{k,i}, y_{k,i}\}$, $\forall d_{k,i} \in \mathcal{D}_k$. Let $\phi\left(d_{k,i};\mathbf{w}\right)$ denote the global loss function of the MO, where $\mathbf{w}$ is the model parameter. To minimize $\phi\left(d_{k,i};\mathbf{w}\right)$ without sharing the data of the UEs, the MO can use the FL algorithm `FedAVG` [1] as given below.

Step 1) The MO broadcasts the initial model parameter $\mathbf{w}$ to all the UEs in set $\mathcal{K}$.

Step 2) Each UE k can continue to train $\mathbf{w}$ on the local dataset $\mathcal{D}_k$ and solve the local optimal model $\mathbf{w}_k$ by addressing the following problem [11]

$$\min_{\mathbf{w}_k} \frac{1}{|\mathcal{D}_k|} \sum_{i=1}^{|\mathcal{D}_k|} \phi_k\left(d_{k,i};\mathbf{w}_k\right), \ \forall d_{k,i} \in \mathcal{D}_k. \tag{1}$$

Jiawei Liu and Guopeng Zhang are with the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China (e-mail: jwliu@cumt.edu.cn; gpzhang@cumt.edu.cn).

Kezhi Wang is with the Department of Computer and Information Science, Northumbria University, Newcastle NE1 8ST, U.K. (e-mail: kezhi.wang@northumbria.ac.uk).

Kun Yang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: kunyang@essex.ac.uk).

On condition that $\phi_k(d_{k,i};\mathbf{w}_k)$ is convex, problem (1) can be solved by using an iterative approach. Let $\mathbf{w}_k^*$ denote the optimal solution to problem (1). From [1], we know that UE k can use arbitrary optimization algorithms (such as the Stochastic Gradient Descent (SGD)) to attain a relative accuracy θ_k for problem (1), such that $\mathbb{E}\left[\phi_k\left(d_{k,i};\mathbf{w}_k\right)-\phi_k(d_{k,i};\mathbf{w}_k^*)\right]\leq\theta_k$. To achieve θ_k, the general upper bound on the required number of local iterations I_k is given by [1]

$$I_k(\theta_k)=\eta_k\log\left(1/\theta_k\right),\ \forall k\in\mathcal{K}, \tag{2}$$

where $\eta_k\geq 0$ is a parameter set by UE k.

Step 3) After each round of local training, each UE k transmits the updated $\mathbf{w}_k$ to the MO via wireless link. The MO can update the global model as $\mathbf{w}=\sum_{k=1}^{K}\frac{|\mathcal{D}_k|}{|\mathcal{D}|}\mathbf{w}_k$, where $\mathcal{D}$ = $\bigcup_{k=1}^{K}\mathcal{D}_k$, and then broadcasts the updated $\mathbf{w}$ to all the UEs to repeat the above process.

After several rounds of the global update, the MO can find the model parameter $\mathbf{w}$ to minimize the global loss function $\phi\left(d_{k,i};\mathbf{w}\right)$. Let $\mathbf{w}^*$ denote the optimal solution to the global problem. We use ϵ $(0\leq\epsilon\leq 1)$ to represent the global relative accuracy which characterizes the quality of the solution $\mathbf{w}$ to the global problem as it produces a random output satisfying $\mathbb{E}[\phi\left(d_{k,i};\mathbf{w}\right)-\phi\left(d_{k,i};\mathbf{w}^*\right)]\leq\epsilon$. Due to the heterogeneity of the UEs, the upper bound of the number of global iterations I^{g} to implement ϵ is given by [1]

$$I^{\mathrm{g}}(\epsilon,\theta_k)=\frac{\zeta\log(\frac{1}{\epsilon})}{1-\max_k\theta_k},\ \forall k\in\mathcal{K}, \tag{3}$$

where $\zeta>0$ is a constant specified by the MO.

A. Energy cost for model training

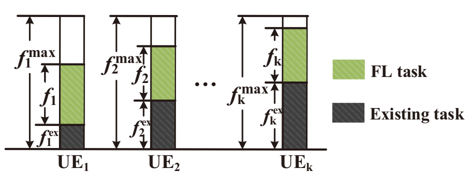

Fig. 1. The computational resource allocation of the UEs.

Before engaging in an FL task, each UE k may process some *existing tasks* (termed as EX-tasks in the following analysis) as shown in Fig. 1. We use $f_{k,t}^{\mathrm{ex}}$ to represent the CPU frequency of UE k to process the EX-task, and assume that $f_{k,t}^{\mathrm{ex}}$ remains constant during the local training session t $(1\leq t\leq I^{\mathrm{g}})$ whose duration is T^{trn}. To admit the FL-task, UE k requires an additional $C_k=c_k|\mathcal{D}_k|I_k$ CPU cycles during each session t, where c_k is the number of CPU cycles required for training per data sample $d_{k,i}$. Therefore, UE k should raise the CPU frequency by $f_k=C_k/T^{\mathrm{trn}}$ Hz in each session to complete the newly carried FL-task. According to [11], the additional energy cost of UE k during session t is given as

$$E_{k,t}^{\mathrm{F}}=\nu_k\left(f_{k,t}^{\mathrm{ex}}+f_k\right)^2T^{\mathrm{trn}}-\nu_k\left(f_{k,t}^{\mathrm{ex}}\right)^2T^{\mathrm{trn}}, \tag{4}$$

where ν_k is the effective switched capacitance of the CPU. Let $f^{\max}$ denote the maximum CPU frequency of the UEs. The following constraint should be satisfied.

$$0<f_{k,t}^{\mathrm{ex}}+f_k\leq f^{\max},\ \forall k\in\mathcal{K},\ \forall t\in\mathcal{T}. \tag{5}$$

From eq. (4), we know that $E_{k,t}^{\mathrm{F}}$ depends not only on the FL-task load f_k but also on the EX-task load f_k^{ex}, which causes the UEs to produce the same amount of CPU cycles, i.e., C_k but consume different amount of energy $E_{k,t}^{\mathrm{F}}$. It will affect the pricing strategy of the UEs in the game. It is noted that the fluctuation of the task load in the UE over a short period (e.g., a few tens of seconds) has some kind of randomness, but the task loads in the adjacent time periods may have a strong correlation [12]. Next, we propose an FSDT-MC based method [9] to predict the dynamic of $f_{k,t}^{\mathrm{ex}}$ during the FL period. Therefore, each UE can estimate its energy cost for model training and set the price for the resource usage.

1) From ref. [12], we know that the workload fluctuation of a machine over a longer period (e.g., a few minutes) has a strong periodicity, while the workload fluctuation over a shorter period (e.g., a few tens of seconds) has a certain randomness, but the workloads in the adjacent time periods have a strong correlation.
2) In this paper, the time spent for a single round of local training is $T^{\mathrm{trn}}=2$s, the time spent for a single round of parameter transmission is $T^{\mathrm{com}}=0.2$s, and the total number of local training sessions is $I^{\mathrm{g}}=10$. Therefore, the total time required to complete a federated learning (FL) task is $T=I^{\mathrm{g}}(T^{\mathrm{trn}}+T^{\mathrm{com}})=22$s.
3) In the proposed game, the UEs must determine the price of the consumed resource during a FL task. In order to capture the short-term dynamic of users' workload during the task, we set the *observation interval* and *update period* of Q_k^{F} to $T=22$s (as stated in the response to your last question), and Q_k^{F} can be timely updated by the workload changes collected at runtime.

According to [9], we discretize the value space of $f_{k,t}^{\mathrm{ex}}$ $(0\leq f_{k,t}^{\mathrm{ex}}\leq f^{\max})$ into M equal-width bins as $\mathbf{F}=\{F_1,\ \cdots,\ F_M\}$, where $F_m=\frac{m-1}{M-1}f^{\max}$ $(1\leq m\leq M)$ represents a state of the FSDT-MC. Any UE k can learn and update the state transition probability (STP) matrix $Q_k^{\mathrm{F}}=\left(\alpha_{m,m'}\right)_{M\times M}$ every T seconds by using the statistical method , where Q_k^{F} is an $M\times M$ matrix and $\alpha_{m,m'}$ is the conditional probability of transitioning from state m to m'. To reflect the short-term dynamics of $f_{k,t}^{\mathrm{ex}}$ during an FL period, we set the update period of Q_k^{F} to $T=I^{\mathrm{g}}(T^{\mathrm{trn}}+T^{\mathrm{com}})$. The method to obtain and update Q_k^{F} as follows.

1) Define the *observation interval* or *update period* of Q_k^{F} as $T=I^{\mathrm{g}}(T^{\mathrm{trn}}+T^{\mathrm{com}})$, where I^{g} represents the number of local training sessions, and T^{trn} (T^{com}) represent the time spent for a single round of local training (parameter transmission). It means that T the *observation interval* or *update period* of Q_k^{F} is the total time required to complete a federated learning task.
2) We divide the *observation interval* T into a large number of (e.g., K) small *time slots* with equal length $\tau=T/K$. We can use the operating system of a UE to monitor and record the workload, i.e. the operating frequency of the CPU in each slot.
3) Let N_m^T denote the number of times that the CPU frequency is in state m during the *observation interval*, $N_{m,i}^T$ the number of times that the CPU frequency

changes from state m to state i during the *observation interval*. According to the following literature [13], we can use the observed records to obtain the frequency of the change from state m to state i during the *observation interval* as

$$\hat{p}_{m,i} = \frac{N_{m,i}^{T}}{N_{m}^{T}},\ 1 \leq i, m \leq M,\ \text{and}\ m \neq i,$$

Then, we can take $\hat{p}_{m,i}$ as the probability of the change from state m to state i during the *observation interval*, that is

$$\alpha_{m,i} = \hat{p}_{m,i} = \frac{N_{m,i}^{T}}{N_{m}^{T}},\ 1 \leq i, m \leq M,\ \text{and}\ m \neq i.$$

4) According to the principle of Markov chain, i.e. the next state of transition only depends on the current state and is independent on the previous states, the transition matrix Q_k^{F} of UE k during local training session t is constructed as

$$Q_k^{\mathrm{F}} = \left(\alpha_{m,i}\right)_{M \times M},\ 1 \leq i, m \leq M,\ \text{and}\ m \neq i.$$

Suppose that $f_{k,t-1}^{\mathrm{ex}}$ is in state m during session $(t-1)$. UE k can predict that $f_{k,t}^{\mathrm{ex}}$ is in state m_t during session t by extracting the most probable future state m' (i.e., the one with the largest STP) from Q_k^{F} at row m, which is

$$m_t = \underset{\forall m' \in \{1,\cdots,M\}}{\arg\max}\ \alpha_{m,m'},\ \forall t \in \mathcal{T}. \tag{6}$$

By substituting eq. (6) into eq. (4), UE k can estimate its energy cost in training session t as

$$\tilde{E}_{k,t}^{\mathrm{F}} = \nu_k \left(F_{m_t} + f_k\right)^2 T^{\mathrm{trn}} - \nu_k \left(F_{m_t}\right)^2 T^{\mathrm{trn}},\ \forall t \in \mathcal{T}. \tag{7}$$

B. Energy cost for parameter transmission

After each round of local training, the MO assigns UE k a time period T^{com} and an orthogonal channel of W Hz to upload the updated model parameter $\mathbf{w}_k$. Let L denote the size of $\mathbf{w}_k$, $p_{k,t}$ the transmit power of UE k, and $g_{k,t}$ the channel gain from UE k to the MO during the t^{th} parameter transmission. Then, the achievable data rate of UE k under a given bit error rate (BER) level can be approximated as

$$r_{k,t} = W \log_2\left(1 + \Delta p_{k,t} g_{k,t}/\sigma^2\right), \tag{8}$$

where σ^2 is noise power and $\Delta = \frac{1.5}{-\ln(5\mathrm{BER})}$ is a constant related to BER. Since $r_{k,t}T^{\mathrm{com}} \geq L$ should be satisfied, the minimum transmit power of UE k is derived as

$$p_{k,t} = \left(2^{\frac{L}{WT^{\mathrm{com}}}} - 1\right)\sigma^2/(g_{k,t}\Delta). \tag{9}$$

and the resultant energy cost of UE k is

$$E_{k,t}^{\mathrm{C}} = p_{k,t}T^{\mathrm{com}} = \left(2^{\frac{L}{WT^{\mathrm{com}}}} - 1\right)\sigma^2 T^{\mathrm{com}}/(g_{k,t}\Delta). \tag{10}$$

Next, we construct an FSDT-MC [10] to predict $g_{k,t}$ ($\forall t \in \mathcal{T}$) for the UEs to develop pricing strategies.

Let T_c denote the channel correlation time, over which the channel response is invariant. For instance, when the system operates on the 900 MHz of 4G (wavelength $\lambda = 0.34$ m) and the UEs move slowly (speed $v \approx 2$ m/s), $T_c = \frac{\lambda}{v} \approx 0.2$s [10]. We set $T^{\mathrm{com}} = T_c$ and $T^{\mathrm{trn}} = \delta T_c$, where $\delta \gg 1$ is a positive integer. Divide the time horizon of the system into time slots with equal length T_c and index the slots by τ. Then, $g_{k,\tau}$ is exponentially distributed over τ. We represent the lower and upper bounds of $g_{k,\tau}$ by $\check{g}$ and $\hat{g}$, respectively, and can then discretize the value space of $g_{k,\tau}$ into N equal-width levels as $\mathbf{L} = \{L_1,\ \cdots,\ L_N\}$, where $L_n = \frac{n-1}{N-1}(\hat{g} - \check{g})$, $1 \leq n \leq N$, represents a state of the FSDT-MC [10].

Define the STP matrix of $g_{k,\tau}$ over τ as $Q^{\mathrm{C}} = \left(\beta_{n,n'}\right)_{N \times N}$, where $\beta_{n,n'}$ is the probability that $g_{k,\tau}$ changes from state n to n' as the time goes from τ to $\tau+1$. We can obtain Q^{C} by using the statistical method. The main idea is to take the frequency of $g_{k,\tau}$ changing from state n to n' during an observation/update period as the STP $\beta_{n,n'}$. The update period can be adjusted adaptively to reflect the change rate of the channel.

1) Let $\check{g}$ and $\hat{g}$ denote the lower and upper bounds of $g_{k,\tau}$, respectively. We discretize the value of $g_{k,\tau}$ into N equal-width levels as $\mathbf{L} = \{L_1,\ \cdots,\ L_N\}$, where

$$L_n = \frac{n-1}{N-1}(\hat{g} - \check{g}), 1 \leq n \leq N$$

represents a state of the FSDT-MC. The state transition of $g_{k,\tau}$ over τ can be described by matrix $Q^{\mathrm{C}} = \left(\beta_{n,j}\right)_{N \times N}$, where $\beta_{n,j}$ is the probability that $g_{k,\tau}$ changes from state n to state j as the time goes from τ to $\tau + 1$.

2) Let $O = X\tau$ denote the *observation period* or *update period* of Q^{C}, where X is a sufficiently large positive integer. We use Z_n and $Z_{n,j}$ to represent the numbers of times per observation period that $g_{k,\tau}$ stays in level L_n and changes from level L_n to level L_j, respectively. Following the priciple of Markov chain, the transition probabilities from state L_n to state L_j, i.e., $\beta_{n,j}$, can be approximated by the ratio

$$\beta_{n,j} = \frac{Z_{n,j}}{Z_n},\ 1 \leq n, j \leq N,\ n \neq j.$$

As a result, the transition matrix $Q^{\mathrm{C}} = \left(\beta_{n,j}\right)_{N \times N}$ is obtained.

3) When a user moves not so fast, i.e, the channel state is relatively stable, we can increase the update period of Q^{C} by increasing X. However, when the user moves faster and the channel state changes more dramatically, we can reduce the update period of Q^{C} by reducing X, thus reflecting the rapid change of the channel state. In this paper, we set $X = 100$ considering that walking users move at a low speed of 2 m/s.

Assuming that the initial distribution of $g_{k,\tau=0}$ over the N states is known as $\pi(g_{k,\tau=0})_{1\times N}$, UE k can predict that $g_{k,t}$ is in state n_t during the t^{th} parameter transmission by extracting the most probable future state n' from Q^{C} at row n, which is

$$n_t = \underset{\forall n' \in \{1,\cdots,N\}}{\arg\max}\ \pi(g_{k,\tau=0})(Q^{\mathrm{C}})^{t(\frac{T^{\mathrm{trn}}}{T_c}+1)},\ \forall t \in \mathcal{T}. \tag{11}$$

By substituting eq. (11) into eq. (10), UE k can estimate its energy cost for the t^{th} parameter transmission as

$$\tilde{E}_{k,t}^{\mathrm{C}} = (2^{\frac{L}{WT^{\mathrm{com}}}} - 1)\sigma^2 T^{\mathrm{com}}/(L_{n_t}\Delta),\ \forall t \in \mathcal{T}. \tag{12}$$

where L_{n_t} is the n_t^{th} state of the FSDT-MC of the channel.

III. Game Formulation

To motivate the UEs to engage in FL, an effective way is to enable the UEs to profit from the energy consumption of model training and data uploading. Game theory is a powerful tool to analyze the optimal or equilibrium strategies of the UEs in such a competitive situation. In general, strategic games can be divided into the following two categories [6]: (1) the production-based games, such as *Cournot* and *Stackelberg games*, which take the *supply quantity* of a commodity as a strategy to maximize the players' profits, and (2) the price-based games, such as *Bertrand games*, which take the *price* of a commodity as a strategy to maximize the players' profits. In the FL system, the UEs have no access to the amount of resources that other UEs contribute, but they can observe the pricing information of others to make decisions. Thus we adopt the Bertrand game to solve the incentive problem. While the UEs fight for market share through price competition, the MO can adjust its purchase according to the prices offered by the UEs, so as to achieve its performance with a minimum total investment. Next, we model the price competition between the UEs and their interaction with the MO as a Bertrand game. The *game execution cycle* is the time to complete an entire FL task with a continuous I^{g}-round global iterations.

1) Strategic games can be divided into the following *quantity*-based games and *price*-based games [6].
   a) Games based on *commodity production*, such as *Cournot games* and *Stackelberg games*, in which competing monopolists take the supply quantity of a commodity as a strategy to maximize their profits.
   b) Games based on *commodity price*, such as *Bertrand games*, in which competing monopolists take the price of a commodity as a strategy to maximize their profits.

   In general, the monopolists cannot obtain each other's production information, but they can easily observe the rivals' product pricing from the market, which helps the monopolists to make decisions. The price of a commodity is a signal to coordinate the supply and demand of the commodity in the market.
2) In this paper, we motivate the UEs to participate in FL through pricing the consumed resources, and, therefore, the *Bertrand game* is adopted to solve problem (14).
   a) In problem (14), multiple competitive UEs produce homogeneous products (computing and communication resources) and set the prices based on their specific production costs (determined by their respective energy consumption during FL) and others' prices to meet the market demand (the FL performance ordered by the model server).
   b) Similar to the monopolists in a oligopoly market, the UEs have no access to the amount of resources that others contribute to the FL task, but they can obtain the price-information of others to make decisions. Therefore, the proposed game belongs to the price-based games and we adopt the *Bertrand game* to analyze the incentive problem (14).

1) The objective of UEs: We represent the price asked by UE k for training session t by $\varrho_{k,t}$ (in *Joule/iteration*), which is the compensation required by UE k for each local iteration over the dataset $\mathcal{D}_k$. Denote the overall energy cost of UE k for completing a round of local training and parameter transmission by $\psi_{k,t} = \tilde{E}^{\mathrm{F}}_{k,t} + \tilde{E}^{\mathrm{C}}_{k,t}$, and the payment of the MO to UE k by $\xi_{k,t} = \varrho_{k,t} I_k$. The aim of UE k is to maximize its profit in the total I^{g} rounds by choosing the optimal price profile $\Gamma_k = (\varrho_{k,1}, \cdots, \varrho_{k,I^{\mathrm{g}}})$. This can be formulated as the following problem.

$$\max_{\Gamma_k} U_k = \sum_{t\in\mathcal{T}} (\xi_{k,t} - \psi_{k,t}) = \sum_{t\in\mathcal{T}} (\varrho_{k,t} I_k - (\tilde{E}^{\mathrm{F}}_{k,t} + \tilde{E}^{\mathrm{C}}_{k,t})) \tag{13}$$

$$\text{s.t.} \quad \varrho_{k,t} \geq 0,\ \forall k \in \mathcal{K}, \forall t \in \mathcal{T}. \tag{13.1}$$

2) The objective of the MO: The MO aims to achieve its performance (including the global accuracy ϵ and the training parameters I^{g}, T^{trn} and T^{com}) with a minimum total investment. Let $\Theta = \big(\theta_1, \cdots, \theta_{|\mathcal{K}|}\big)$ denote the strategy profile of the MO, which represents the amount of resources it purchases from each UE k. Once the UEs determine their optimal prices Γ_k, $\forall k \in \mathcal{K}$, the MO can adjust Θ to achieve its objective, which is formulated as the following problem

$$\min_{\Theta} U_{\mathrm{S}} = \sum_{t\in\mathcal{T}} \sum_{k\in K} \varrho_{k,t} I_k + \frac{1}{2}\Big(\sum_{k\in K} (\theta_k)^2 + 2v \sum_{k\neq j} (\theta_k \theta_j) \Big), \tag{14}$$

$$\text{s.t.} \quad 0 < \theta_k \leq \theta^{\max} = 1 - \frac{\zeta \log(\frac{1}{\epsilon})}{I^{\mathrm{g}}},\ \forall k \in \mathcal{K}, \tag{14.1}$$

where constraint (14.1) indicates the value space of θ_k. From eq. (3), we can get $\theta^{\max} = \max_k \theta_k = 1 - \frac{\zeta \log(\frac{1}{\epsilon})}{I^{\mathrm{g}}}$. The meaning of U_{S} is given as follows.

1) *The meaning of the first term* $\sum_{t\in\mathcal{T}} \sum_{k\in K} \varrho_{k,t} I_k$.
   a) Parameter I_k represents the required number of local iterations in each k^{th} training session to achieve local accuracy θ_k.
   b) Parameter $\varrho_{k,t}$ represents the price asked by UE k for the t^{th} training session.
   c) Therefore, $\sum_{t\in\mathcal{T}} \sum_{k\in K} \varrho_{k,t} I_k$ represents the total payment of the server to all the UEs in the total I^{g} rounds of local training.
2) The second term $\frac{1}{2}\Big(\sum_{k\in K} (\theta_k)^2 + 2v \sum_{k\neq j} (\theta_k \theta_j) \Big)$ considers the *Resource Substitutability* (RS) in a Bertrand game.
   a) We note that the Bertrand game model is first proposed in [14]. The content related to the RS is in the second formula, Part 2, page 547 of ref. [14]. Next, we introduce it briefly.
   b) According to ref. [15], the RS is an ability of a buyer to substitute one product (hold by some sellers) with other products (hold by other sellers) of similar functionality. The RS of a buyer is described by parameter $v \in [0, 1]$. When $v = 0$, there is no substitutability between products sold by different sellers, when $v = 1$, the products sold by different sellers are completely homogeneous with each other. Please refer to it for more details.

c) In what follows, we explain how the RS is applied to our game framework for federated learning. The function of term $\frac{1}{2}\Big(\sum_{k\in K}(\theta_k)^2 + 2v\sum_{k\neq j}(\theta_k\theta_j)\Big)$ can be explained from the economic aspect, as it tasks the RS into account through parameter v.
It is known that a major challenge to implement FL is to solve the non independent-and-identical distribution of the training data on different UEs. In the considered FL system, the model server faces a limited number of UEs to perform a FL task. If the server needs special datasets owned by certain UEs to train the learning model, the RS of different UEs is weak, namely v tends to 0; otherwise, the learning model of the server can be trained on any dataset owned by the UEs, namely, the datasts owned by different UEs have a strong RS, so v tends to 1.

d) The issue of statistical heterogeneity of training data is not discussed here as it is beyond the scope of this paper. In the simulation, we set $v = 0.5$ as in [16] to represent a general case. In [16], the authors presented the similar RS-based utility function to address the spectrum allocation problem in multi-user cognitive radio networks. Readers can refer to the first formula in Sec. IV.A, page 4275 of [16] for detail.

IV. Solution Of The Game

We first solve problems (13) and (14), respectively, and obtain the NE of the game. Then, an iterative algorithm is developed to find the NE in a distributed manner.

1) Solve the MO problem (14)*:* In the Bertrand game, each UE k first gives the resource price Γ_k. Then, the MO can find the optimal resource purchase Θ according to Γ_k, $\forall k \in \mathcal{K}$, by solving problem (14). Through Taylor expansion, one can get $-\log(1-x) = x + \mathcal{O}(x)$. Then, Eq. (2) can be rewritten as

$$I_k = -\eta_k \log(1-(1-\theta_k)) = \eta_k\,(1-\theta_k) + \mathcal{O}(1-\theta_k). \quad (15)$$

By substituting eq. (15) into eq. (14), U_{S} can be rewritten as

$$\tilde{U}_{\mathrm{S}} = \sum_{t\in\mathcal{T}}\sum_{k\in K} \varrho_{k,t}\eta_k\,(1-\theta_k) + \frac{1}{2}\Big(\sum_{k\in K}(\theta_k)^2 + 2v\sum_{k\neq j}(\theta_k\theta_j)\Big). \quad (16)$$

Since $\tilde{U}_{\mathrm{S}}(\theta_k)$ is convex with respect to θ_k, one can differentiate $\tilde{U}_{\mathrm{S}}(\theta_k)$ with respect to θ_k and set the results to 0. Then, the optimal solution of problem (14) is obtained as

$$\theta_k = A\sum_{t\in\mathcal{T}}\varrho_{k,t} + B\sum_{j\in\mathcal{K},j\neq k}\sum_{t\in\mathcal{T}}\varrho_{j,t}, \quad (17)$$

where $A = \frac{-(1-2v+Kv)}{(1-v)(Kv+1-v)}$ and $B = \frac{v}{(1-v)(Kv+1-v)}$ are constants.

2) Solve the UE problem (13)*:* After solving problem (14), the resource purchase Θ of the MO is known. Then, any UE k can update its price Γ_k by solving problem (13). Note that the pricing of UE k for session t, i.e., $\varrho_{k,t}$ is affected not only by the accuracy θ_k but also by the pricing of the other UEs, which is represented by $\varrho_{-k,t} = \{\varrho_{x,t}|\forall x \in \mathcal{K} \text{ and } x \neq k\}$. By substituting eq. (17) into problem (13) and expanding it with Taylor. The expression of U_k can be rewritten as

$$\tilde{U}_k = \sum_{t\in\mathcal{T}} U_{k,t}, \quad (18)$$

where $U_{k,t} = \varrho_{k,t}(1 + A\sum_{t\in\mathcal{T}}\varrho_{k,t} - BV) - C_t(1 + A\sum_{t\in\mathcal{T}}\varrho_{k,t} - BV) - (D(1 + A\sum_{t\in\mathcal{T}}\varrho_{k,t} - BV)^2 + \tilde{E}^{\mathrm{C}}_{k,t})$.

As UE k can estimate $f^{\mathrm{ex}}_{k,t}$ and $g_{k,t}$ in each session t by using eqs. (6) and (11) respectively, the parameters $C_t = 2\nu_k c_k|\mathcal{D}_k|f^{\mathrm{ex}}_{k,t}$, $D = \frac{\nu_k(c_k)^2|\mathcal{D}_k|^2}{T^{\mathrm{trn}}}$, $V = \sum_{x\in\mathcal{K},x\neq k}\sum_{t\in\mathcal{T}}\varrho_{x,t}$ and $\tilde{E}^{\mathrm{C}}_{k,t}$ in eq. (16) can be treated as constants by UE k.

Eq. (18) shows that any UE k can decompose the pricing problem into I^{g} sub-problems, which can be solved in parallel to get $\varrho_{k,t}$ in each session t. Since $U_{k,t}(\varrho_{k,t})$ given in eq. (18) is concave with respect to $\varrho_{k,t}$, one can differentiate $U_{k,t}(\varrho_{k,t})$ with respect to $\varrho_{k,t}$ and set the results to 0. Then, the solution of $\varrho_{k,t}$ is obtained as

$$\varrho_{k,t} = \frac{1-AC_t-BV-2AD+2ABDV}{2A^2D-2A} - \sum_{i\in\mathcal{T},i\neq t}\varrho_{k,i}. \quad (19)$$

3) Find the NE of the game: At the NE of the game, neither the MO nor each of the UEs can get higher profit from changing the strategy profile, i.e., Θ or Γ_k. Denote the NE strategy profile of the MO and the UEs by $\{\hat{\Theta}, \hat{\Gamma}_1, \cdots, \hat{\Gamma}_k\}$, where $\hat{\Theta} = \Big(\hat{\theta}_1, \cdots, \hat{\theta}_{|\mathcal{K}|}\Big)$ and $\hat{\Gamma}_k = (\hat{\varrho}_{k,1}, \cdots, \hat{\varrho}_{k,I^g})$. We propose an iterative algorithm to find the NE.

Let $i = 1, 2, \cdots$ denote the iteration numbers. In the i^{th} iteration, we use $\Gamma_k[i]$, $\Gamma_{-k}[i]$ and $\Theta[i]$ to represent the pricing of UE k, the pricing of the UEs in set $\mathcal{K}$ other than UE k, and the resource purchase of the MO, respectively. Given that the pricing of all UEs are observable in a market, the following distributed algorithm is designed to find the NE.

Algorithm 1 Find the NE of the proposed Bertrand game.

The MO specifies and broadcasts its performance parameters: ϵ, I^{g}, T^{trn} and T^{com}.
Let $i = 1$. Each UE k initialize the price as $\Gamma_k[i]$.
The MO obtain the resource purchase $\Theta[i]$ using eq. (17).

repeat
  For each UE k, $\forall k \in \mathcal{K}$, after collecting $\theta[i]$ of the MO and $\Gamma_{-k}[i]$ of the other UEs in set $\mathcal{K}$, it can update the optimal pricing $\Gamma_k[i]$ by using eq. (19).
  For the MO, after collecting all $\Gamma_k[i]$, it can update the optimal resource purchase $\Theta[i]$ by using eq. (17).
  Each UE k derives the gradient $\nabla U_k(\varrho_{k,t})[i]$.
  Update $i = i + 1$.
until $\|\nabla U_k(\varrho_{k,t})[i]\| \leq \Xi\,\|\nabla U_k(\varrho_{k,t})[i-1]\|$, $\forall k \in \mathcal{K}$.

Because $U_k(\varrho_{k,t})$ is strictly concave with respect to $\varrho_{k,t}$, the upper bound on the iteration number of **Algorithm 1** to reach the convergence threshold Ξ is $\mathcal{O}(\log\frac{1}{\Xi})$ [17] and the computational complexity of **Algorithm 1** is $\log\frac{1}{\Xi}(K + I^g)$

[18]. Although it is assumed that all the UEs are willing to engage in the FL task, some of them cannot be selected because their computing power fails to meet the requirement of local relative accuracy θ_k. Then we apply constraint (14.1) to the outcome of **Algorithm 1** and determine which UEs are qualified for the FL task. Finally, we give the execution process of the game in the following **Algorithm 2**, which is termed as the *task-load-aware game-theoretic scheme* (TLA-GTS)

Algorithm 2 TLA-GTS

Input:
The set of the potential client UEs $\mathcal{K}$ for the MO.
repeat
  Perform **Algorithm 1** and obtain the solution $\{\hat{\Theta}, \hat{\Gamma}_k\}$.
  if $\theta_k \leq 0$ **or** $\theta_k > \theta^{\max}$ (i.e., constraint (14.1) is not satisfied) **then**
    Remove the k^{th} UE from set $\mathcal{K}$.
  end if
until $0 < \theta_k \leq \theta^{\max}$ for $\forall k \in \mathcal{K}$, or $\mathcal{K} = \varnothing$.
Output: The remaining UEs in set $\mathcal{K}$.

In **Algorithm 2**, we initialize $\mathcal{K}$ as all the potential client UEs for the MO. After performing **Algorithm 1**, one can make the following choices according to the obtained solution. If the solution satisfies constraint (14.1), the algorithm terminates and the solution is taken as the NE of the game. Otherwise, the UE with the highest price is removed from set $\mathcal{K}$, and **Algorithm 1** is performed again until constraint (14.1) is satisfied or set $\mathcal{K}$ is empty. An empty UE set indicates that the performance set by the MO (e.g., ϵ, I^{g}, T^{trn} and T^{com}) cannot be achieved by the UEs. The MO needs to lower the performance metrics and restart the game. **Algorithm 2** is implemented by performing **Algorithm 1** for K times. Therefore, its computational complexity is $\log\frac{1}{\Xi}(K^2 + KI^g)$.

V. Simulation Results

One MO and four UEs are placed in wireless networks, e.g., WLAN. We set $f^{\max} = 2$ GHz [11] and discretize $f^{\text{ex}}_{k,t}$ ($0 \leq f^{\text{ex}}_{k,t} \leq f^{\max}$) of each UE k into 5 levels. By using the statistical method, each UE k can obtain a different state transition probability matrix Q^{F}_k as follows.

$$Q^{\text{F}}_1 = \begin{bmatrix} 0.3 & 0.15 & 0.25 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.1 & 0.4 & 0.2 \\ 0.1 & 0.4 & 0.1 & 0.1 & 0.2 \\ 0.4 & 0.1 & 0.1 & 0.2 & 0.2 \\ 0.1 & 0.3 & 0.1 & 0.3 & 0.2 \end{bmatrix}$$

$$Q^{\text{F}}_2 = \begin{bmatrix} 0.2 & 0.3 & 0.1 & 0.1 & 0.3 \\ 0.3 & 0.1 & 0.4 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.3 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.4 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.4 & 0.2 & 0.1 \end{bmatrix}$$

$$Q^{\text{F}}_3 = \begin{bmatrix} 0.2 & 0.1 & 0.35 & 0.15 & 0.1 \\ 0.1 & 0.15 & 0.3 & 0.25 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.4 & 0.3 \\ 0.1 & 0.1 & 0.4 & 0.1 & 0.3 \\ 0.1 & 0.1 & 0.4 & 0.3 & 0.1 \end{bmatrix}$$

$$Q^{\text{F}}_4 = \begin{bmatrix} 0.2 & 0.1 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0.15 & 0.3 & 0.25 & 0.1 \\ 0.3 & 0.1 & 0.2 & 0.1 & 0.3 \\ 0.2 & 0.4 & 0.2 & 0.1 & 0.1 \\ 0.2 & 0.4 & 0.1 & 0.2 & 0.1 \end{bmatrix}$$

The lower and upper bounds of the time-varying channel gain $g_{k,\tau}$ is set to $\check{g} = 2^{0.4} - 1$ and $\hat{g} = 2^{3.1} - 1$ [19], respectively, and the transition probability matrix Q^{C} of $g_{k,\tau}$ as follows.

The other simulation parameters include dataset size $|\mathcal{D}_k| = 8\times10^7$ [11], the number of CPU cycles for computing one data sample $c_k = 15$ [11], the size of model parameters $L = 0.1$ Mb, the global iteration number $I^{\text{g}} = 10$, the duration for each local training $T^{\text{trn}} = 2$s, the duration for each parameter transmission $T^{\text{com}} = 0.2$s, the CPU parameter $\nu_k = 10^{-28}$ [11], the bandwidth allocated to each UE $W = 1$ MHz, the noise power $\sigma^2 = 10^{-9}$, the BER requirement 10^{-3}, the substitutability factor for the MO $\upsilon = 0.5$, $\eta_k = 1$ in eq. (2), and $\zeta = 1$ in eq. (3).

First, we show the impact of different EX-task load $f^{\text{ex}}_{k,t}$ on the pricing of the UEs. For that purpose, we set $f^{\text{ex}}_{k,t=0} = 0$ for all the 4 UEs. Following eq. (6), $f^{\text{ex}}_{k,t}$ of the UEs in 3 consecutive training sessions (e.g., $t = 1, 2, 3$) are estimated as $f^{\text{ex}}_{1,t} = \{0, 0.5, 1.5\}$, $f^{\text{ex}}_{2,t} = \{0.5, 1, 1\}$, $f^{\text{ex}}_{3,t} = \{1, 1.5, 1\}$, and $f^{\text{ex}}_{4,t} = \{1, 0, 1\}$, respectively. Fig. 2 shows the convergence of the prices of the UEs in the 3 training sessions.

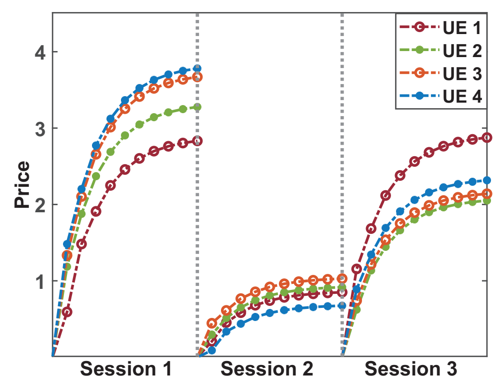

Fig. 2. Price convergence.

As shown in Fig. 2, the UEs with lower $f^{\text{ex}}_{k,t}$ have a price advantage over the UEs with higher $f^{\text{ex}}_{k,t}$ in each training session. As the energy cost for data uploading $E^{\text{C}}_{k,t}$ is negligible compared to that for model training $E^{\text{F}}_{k,t}$, the results imply that the pricing of a UE is sensitive to its available computing power. To incentive the UEs with higher $f^{\text{ex}}_{k,t}$ (i.e. lower-level of available computing power) to engage in an FL task, the MO has to pay a higher price for the resource usage. Next, we show the performance of the proposed TLA-GTS, i.e., **Algorithm 2** on a real FL task. The task is to classify handwritten digits using the `MNIST` dataset [20]. The distribution of the dataset over the UEs are balanced but non-i.i.d, and the training model is a 2-layer deep neural networks (DNN), consisting of one flatten layer, two fully-connected layers with rectified linear unit (ReLU) activation and a log-softmax output layer. For each ordered model accuracy {.65,

$$Q^{\mathrm{C}} = \begin{pmatrix} 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 \\ 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 \\ 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 \\ 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 \\ 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 & 0.016 \\ 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 & 0.032 \\ 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 & 0.064 \\ 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 & 0.128 \\ 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 & 0.256 \\ 0.256 & 0.128 & 0.064 & 0.032 & 0.016 & 0.008 & 0.004 & 0.002 & 0.001 & 0.489 \end{pmatrix}.$$

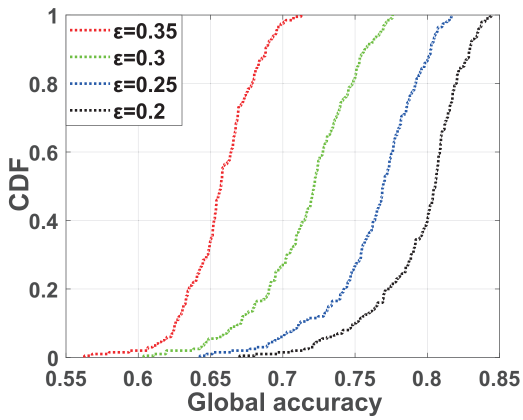

Fig. 3. The global accuracy achieved by the TLA-GTS.

.7, .75, .8}, 100 experiments (game cycles) were performed. Fig. 3 shows the empirical cumulative distribution function (CDF) of the 100 experiments. One can see that the TLA-GTS can effectively motivate the autonomous UEs to help the MO achieve the accuracy goal of the FL task.

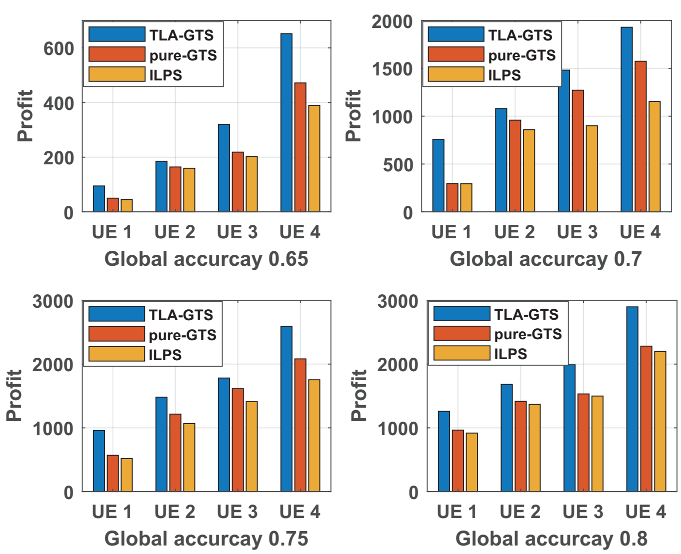

Fig. 4. The profits of the UEs using different pricing strategies.

Finally, we verify the superiority of TLA-GTS by comparing with other two schemes. Given that most of the related works do not consider the user's EX-task load, the first scheme to be compared is the *game-theoretic scheme without the task-load-aware mechanism*. It is termed as the *pure*-GTS, which still uses the proposed game framework, but the energy cost of a UE does not consider the impact of its EX-task load, i.e., $\tilde{E}^{\mathrm{F}}_{k,t} = \nu_k \left(f_k\right)^2 T^{\mathrm{trn}}$ in eq. (7). Another scheme to be compared is the *independent linear pricing strategy* (ILPS), in which, each UE prices the MO independently using a linear pricing strategy based on its available resources, regardless of the pricing of other UEs. For each ordered model accuracy, the profits of the UEs using the different schemes are shown in Fig. 4. From Fig. 4, one can see that both the game-theoretic schemes TLA-GTS and *pure*-GTS outperform the ILPS as they bring higher profits for each of the UEs. This is because the game-theoretic schemes can price the resources of UEs based on their energy cost and the pricing of other UEs in the same training session. Furthermore, one can also note that the TLA-GTS outperforms the *pure*-GTS in all cases. This is because that the energy cost of a UE in an FL task depends not only on the FL-task load but also on the EX-task load, which causes the UEs to produce the same amount of CPU cycles but consume different amount of energy. Hence, the TLA-GTS brings the UEs the highest profits in the game.

VI. Conclusion

This paper has proposed a game theoretic approach to motivate UEs to participate in FL and the NE of the game has been obtained by a distributed iterative algorithm. Simulation results have verified the effectiveness of the proposed approaches.

References

[1] J. Konen et al, "Federated optimization: Distributed machine learning for on-device intelligence," 2016.

[2] Y. Zhan, P. Li, S. Guo, and Z. Qu, "Incentive mechanism design for federated learning: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 4, pp. 310–317, 2021.

[3] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A stackelberg game perspective," *IEEE Networking Letters*, vol. 2, no. 1, pp. 23–27, 2020.

[4] Y Zhan et al, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, 2020.

[5] S. R. Pandey et al, "A crowdsourcing framework for on-device federated learning," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 5, pp. 3241–3256, 2020.

[6] D. W. Majerus, "Price vs. quantity competition in oligopoly supergames," *Econ. Lett.*, vol. 27, no. 3, pp. 293–297, 1988.

[7] W. Y. B. Lim et al, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, 2022.

[8] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Miao, and D. I. Kim, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3640–3653, 2021.

[9] S. Salamat, B. Khaleghi, M. Imani, and T. Rosing, "Workload-aware opportunistic energy efficiency in multi-fpga platforms," in *2019 ICCAD*, 2019, pp. 1–8.

[10] J. G. Ruiz et al, "On finite state markov chains for rayleigh channel modeling," in *2009 Wireless VITAE*, 2009, pp. 191–195.

[11] T. H. Thi Le, N. H. Tran, Y. K. Tun, M. N. H. Nguyen, S. R. Pandey, Z. Han, and C. S. Hong, "An incentive mechanism for federated learning in wireless cellular networks: An auction approach," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 8, pp. 4874–4887, 2021.

[12] W. Shu et al, "Resource demand prediction of cloud workloads using an attention-based gru model," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, 2021, pp. 428–437.

[13] J. Fenech, Y. Yap, and S. Shafik, "Brief technical note: A markov chain approach to measure investment rating migrations," *Australasian Accounting Business and Finance Journal*, vol. 7, no. 3, pp. 145 – 154, 2013.

[14] N. Singh and X. Vives, "Price and quantity competition in a differentiated duopoly," *RAND Journal of Economics*, vol. 15, no. 4, pp. 546–554, 1984. [Online]. Available: https://EconPapers.repec.org/RePEc:rje:randje:v:15:y:1984:i:winter:p:546-554

[15] W. S. Oh and R. Muneepeerakul, "How do substitutability and effort asymmetry change resource management in coupled natural-human systems?" *Palgrave Communications*, vol. 5, no. 1, pp. 1–8, 2019.

[16] D. Niyato and E. Hossain, "Market-equilibrium, competitive, and cooperative pricing for spectrum sharing in cognitive radio networks: Analysis and comparison," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4273–4283, 2008.

[17] C. Ma et al, "Distributed optimization with arbitrary local solvers," *Optim. Method Softw.*, 2017.

[18] R. Hunger, "Floating point operations in matrix-vector calculus."

[19] Y. Guo et al, "Adaptive bitrate streaming in wireless networks with transcoding at network edge using deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–1, 2020.

[20] Y. Lecun et al, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.