FedParking: A Federated Learning Based Parking Space Estimation With Parked Vehicle Assisted Edge Computing

Xumin Huang , Peichun Li , *Student Member, IEEE*, Rong Yu , *Member, IEEE*, Yuan Wu , *Senior Member, IEEE*, Kan Xie , and Shengli Xie , *Fellow, IEEE*

Abstract—As a distributed learning approach, federated learning trains a shared learning model over distributed datasets while preserving the training data privacy. We extend the application of federated learning to parking management and introduce FedParking in which Parking Lot Operators (PLOs) collaborate to train a long short-term memory model for parking space estimation without exchanging the raw data. Furthermore, we investigate the management of Parked Vehicle assisted Edge Computing (PVEC) by FedParking. In PVEC, different PLOs recruit PVs as edge computing nodes for offloading services through an incentive mechanism, which is designed according to the computation demand and parking capacity constraints derived from FedParking. We formulate the interactions among the PLOs and vehicles as a multi-lead multi-follower Stackelberg game. Considering the dynamic arrivals of the vehicles and time-varying parking capacity constraints, we present a multi-agent deep reinforcement learning approach to gradually reach the Stackelberg equilibrium in a distributed yet privacy-preserving manner. Finally, numerical results are provided to demonstrate the effectiveness and efficiency of our scheme.

Index Terms—Federated learning, parked vehicle assisted edge computing, deep reinforcement learning and Stackelberg game.

I. INTRODUCTION

NOWADAYS, federated learning has been envisioned as a distributed learning framework in which devices cooperate to build and update a globally shared learning model by supporting training the global model over distributed datasets [1], [2]. Federated learning enables each device as a data owner to locally train the global model with individually collected data. This approach exchanges model parameters instead of the actual training data and preserves data privacy of the devices. Due to the significant privacy-friendly characteristic, federated learning is applied to diverse domains, e.g., mobile keyboard prediction [3], wearable activity recognition [4] and content caching placement [5]. The promising approach is also integrated into vehicular networks to forecast traffic information such as traffic flow [6] and traffic speed [7] while guaranteeing reliable data privacy preservation during the forecast. Motivated by the current works, we extend the application of federated learning to parking management in smart cities, and develop a federated learning based parking space estimation scheme named by FedParking.

Fedparking allows multiple Parking Lot Operators (PLOs) to collaboratively learn a Long Short-Term Memory (LSTM) model for parking space estimation while keeping their training data secure and private. On one hand, a PLO should predict the number of free parking spaces in real time for traffic management at a parking lot. To achieve parking space estimation, researchers have presented a variety of applicable methods. Since prediction of parking space availability including parking occupancy and duration of occupancy can be regarded as a time-series problem, time series prediction is utilized as a promising solution to make full use of prior knowledge from historical data and obtain accurate prediction results. Furthermore, the time-series prediction problem on parking space estimation is characterized by the long-term trend and cyclical fluctuations, and LSTM as a powerful variant of the typical recurrent neural network is particularly suitable to tackle it [8]. Thus, we adopt a

Xumin Huang is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the State, Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China (e-mail: huangxu_min@163.com).

Peichun Li is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Key Laboratory of Intelligent Detection and Internet of Things in Manufacturing, Ministry of Education, Guangzhou 510006, China (e-mail: peichun@mail2.gdut.edu.cn).

Rong Yu is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong-Hong Kong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, China (e-mail: yurong@ieee.org).

Yuan Wu is with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China, and also with Department of Computer and Information Science, University of Macau, Macau, China (e-mail: yuanwu@um.edu.mo).

Kan Xie is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the 111 Center for Intelligent Batch Manufacturing Based on IoT Technology, Guangzhou 510006, China (e-mail: kxie@gdut.edu.cn).

Shengli Xie is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong Key Laboratory of IoT Information Technology, Guangzhou 510006, China (e-mail: shlxie@gdut.edu.cn).

Digital Object Identifier 10.1109/TVT.2021.3098170

typical LSTM model as the global model in Fedparking, which is shared to each PLO to achieve an accurate estimation of the parking spaces.

On the other hand, the global model can simply feed off the training data extracted from parking space information of all the PLOs. In this regard, a naive learning approach usually requires each PLO to upload the training data and utilizes the collected data for centralized training. However, the traditional learning approach puts individual privacy of vehicles at risk, since the parking space information records basic profiles of daily parking requests such as arrival and departure time of the vehicles. The parking space information also records time-varying parking price and occupancy of the parking lots. For commercial purposes, PLOs would not like to reveal the parking space information to the others to prevent malicious entities which infer the operating mode of the parking lots. This means that a secure and distributed learning approach is necessitated such that different PLOs collaborate to train the global model without potential security vulnerabilities. We pay attention to the recently emerged federated learning that does not rely on sharing training data among the clients and propose FedParking. In FedParking, different PLOs collaborate to train the global model by the local datasets and only feedback the updated model parameters for parameter aggregation. To summarize, we are motivated to study an integration of the federated learning and LSTM to facilitate the FedParking design with efficiency and security guarantee.

Building upon the FedParking for predicting the parking spaces at different PLOs, we devise an incentive mechanism for Parked Vehicle assisted Edge Computing (PVEC). As an application of mobile edge computing in Internet of vehicles, PVEC schedules stochastic idle communication and computing resources from Parked Vehicles (PVs). It can significantly establish an available and cost-effective computing resource pool, which alleviates computational workloads and augments system capacity [9]. A PLO acts as an offloading service provider to receive computation-intensive but not delay-sensitive tasks from nearby offloading users, afterwards stimulates vehicles to enter the parking spaces and utilize the on-board resources of the PVs to accomplish the given workloads. To realize the goal, an appropriate incentive mechanism is necessitated for the PLO to provide vehicles with monetary rewards according to the parking capacity and computation demand. Through FedParking, we can estimate the number of available parking spaces of every parking lot which leads to the corresponding parking capacity constraint for each PLO.

After that, we formulate the strategic interactions among multiple PLOs and vehicles as a two-stage and non-cooperative game, namely, a multi-leader multi-follower Stackelberg game. A PLO is a game leader that provides monetary rewards to attract the vehicles to enter the parking lot and share the idle computing resources in the first stage. In turn, the vehicles are the game followers which act in response to the PLOs' strategies. The Stackelberg game aims to improve the utilities of the PLOs while simultaneously guaranteeing utility maximization for each vehicle. By theoretical analysis, we prove the existence and uniqueness of the Stackelberg equilibrium while neglecting the parking capacity constraints. Furthermore, we apply Deep Reinforcement Learning (DRL) as a solution methodology to reach the Stackelberg equilibrium in practical scenarios where both arrivals of the vehicles and parking capacity constraints of the PLOs are varying over time. In our DRL approach, each PLO is regarded as an agent to learn a near-optimal decision by referring to the historical strategies of the others. The approach depends on iterative interactions among the players and supports distributed decision making instead of holding any prior knowledge of all the players in advance.

The main contributions of the paper can be summarized as follows.

- We adopt federated learning and LSTM to promote the collaboration among the PLOs in parking space estimation. In FedParking, each PLO trains a shared LSTM model to forecast available parking spaces by the local data while preserving the training data privacy. Technique details including system model, network entities, workflow and LSTM model design are provided.

- We study the incentive mechanism design of PVEC by FedParking, which puts forward a parking capacity constraint for each PLO. A PLO needs to recruit PVs to complete offloading workloads and an appropriate incentive mechanism is necessitated to maximize the expected profit and simultaneously take into account the parking capacity constraint. After introducing a Stackelberg game model among the PLOs and vehicles, we provide the equilibrium analysis to validate the existence and uniqueness of Stackelberg equilibrium by studying the sub-game perfect equilibrium in each stage.

- We present a DRL based incentive mechanism for the PLOs since the closed-form Stackelberg equilibrium is difficult to acquire in the non-static environment. A multi-agent DRL approach is introduced to gradually reach the Stackelberg equilibrium in a distributed yet privacy-preserving manner. Our approach avoids collecting the private information of the players of the Stackelberg game and achieves the almost same convergence performance as the centralized approach.

The rest of this paper is organized as follows. Section II presents the related work. We introduce the detailed FedParking design based on federated learning and LSTM in Section III. Section IV studies the Stackelberg game based incentive mechanism. Section V provides the theoretical Stackelberg equilibrium analysis without considering the parking capacity constraints. In Section VI, we propose the DRL approach to gradually reach the Stackelberg equilibrium with the given parking capacity constraints. Numerical results are shown in Section VII. Finally, Section VIII concludes this paper and discusses the future direction.

II. RELATED WORK

A. Federated Learning for Vehicular Networks

As a privacy-preserving learning approach, federated learning enables the collaborative training of a globally shared learning model without exchanging raw data among the clients. Recently,

the promising tool has been applied to vehicular networks for different purposes. For example, federated learning was combined with a gated recurrent unit neural network to encourage multiple organizations to cooperatively predict future traffic flow without privacy leakage [6]. The distributed learning technique was exploited to provide robust privacy-preserving traffic speed forecasting and protection of topological information [7]. To eliminate the vehicular data leakage, the authors in [10] presented a federated learning based collaborative data leakage detection scheme. Besides, federated learning was adopted among mobile vehicles to improve vehicular communications with ultra reliability and low latency [11], and support AI applications such as image classification [12] in vehicular edge computing environment.

Motivated by the above works, we propose an integration of federated learning and LSTM to design FedParking and achieve the PLOs' collaborative parking space estimation. According to a federated learning based framework, various PLOs cooperate with a parameter server to jointly train the LSTM model in an efficient and secure manner. Under the privacy restrictions, FedParking only collects model parameters updated from the PLOs and does not require sharing the datasets of the PLOs with the parameter server. Moreover, owing to the flexibility of the federated learning based framework, FedParking is computation efficient and also compatible with the other machine learning models in addition to the proposed LSTM model for parking space estimation.

B. Parked Vehicle Assisted Edge Computing

Nowadays, many research efforts have been devoted to optimizing the performance of PVEC with different objectives. For example, a contract based incentive mechanism was designed to incentivize mobile and parked vehicles with different types to contribute their on-board resources for task processing [13]. A contract was correlated with an offloading task and a contract item specifies the required number of computing resources and promising rewards for a vehicle type. The contract assignment between various tasks and vehicles was determined by using matching theory. The similar work was found in [14]. Besides, mobile and parked vehicles cooperate with nearby mobile edge computing servers for admission control [15] and workload processing [16], [17]. The vehicles also receive offloading tasks from unmanned aerial vehicles in smart city [18] and help a local cloudlet cope with traffic packets and establish a distributed city-wide traffic management system [19]. In addition to contract theory, game and auction theories have been applied for the incentive mechanism design in PVEC. A Stackelberg game was formulated to jointly employ a set of PVs and a mobile edge computing server for cooperative task processing [20]. A multi-round multi-item parking reservation auction scheme was introduced to guide passing vehicles to specific parking spaces and make use of their processing capability to support proximal offloading services [21].

In spite the above studies, there still exist several important issues to be addressed for the success of PVEC. First, a PLO needs to dynamically predict the number of available parking spaces over time and adjust the incentive mechanism according to a parking capacity constraint and computation demand. Second, previous works have not considered the competition effect among different PLOs, which provide different monetary rewards for arriving vehicles. Due to selfishness in resource usage, the PLOs compete against each other to recruit the PVs and reserve their idle computing resources during the parking time. In turn, parking choices of the vehicles will be influenced by the reward policies of the PLOs. The vehicles are rational to decide which parking lot to join and optimize the computing resources shared to the corresponding PLO. Last but not the least, it is straightforward in most the previous works to collect private information of each PV and PLO for centralized decision making. This violates privacy of the participants. Alternatively, a learning based incentive mechanism could be helpful for a PLO to seek a suboptimal solution under incomplete information.

Compared with the previous works, we propose a joint federated learning and DRL approach to promote the management of PVEC. A flowchart is illustrated in Fig. 1 to present the comprehensive methodology of the proposed FedParking, which consists of two basic parts: parking space estimation and incentive mechanism design. Considering the parking management in smart cities, we extend the application of federated learning to parking space estimation and introduce FedParking. In the scheme, multiple PLOs collaborate to train a globally shared learning model to predict the number of free parking spaces in real time, without exchanging the raw data. Moreover, we utilize a LSTM model as the global model in FedParking to achieve an accurate estimation of the parking spaces. Building upon the above scheme for estimating the parking spaces at different PLOs, we further facilitate the incentive mechanism design for PVEC. According to the computing demand and parking capacity constraint, each PLO determines how to stimulate vehicles to enter the parking spaces and schedule idle resources of the PVs to support local offloading services. We formulate the strategic interactions among multiple PLOs and vehicles as a multi-leader multi-follower Stackelberg game. The competition effect of different PLOs is considered. We temporarily neglect the parking capacity constraints, and provide the Stackelberg equilibrium analysis under the static conditions. We realize that the closed-form Stackelberg equilibrium is difficult to acquire in the practical scenarios where both arrivals of the vehicles and parking capacity constraints of the PLOs are varying over time. Thus, a DRL approach is applied, in which each PLO plays as an agent to learn a near-optimal decision, such that we can reach the Stackelberg equilibrium in a distributed manner.

III. FEDPARKING

For traffic management at a parking lot, a PLO predicts the number of available parking spaces. To this end, a learning model is developed and utilized as a global model that is shared to multiple PLOs after iteratively training. To avoid the centralized data collection and centralized model training, we investigate the use of federated learning on parking space
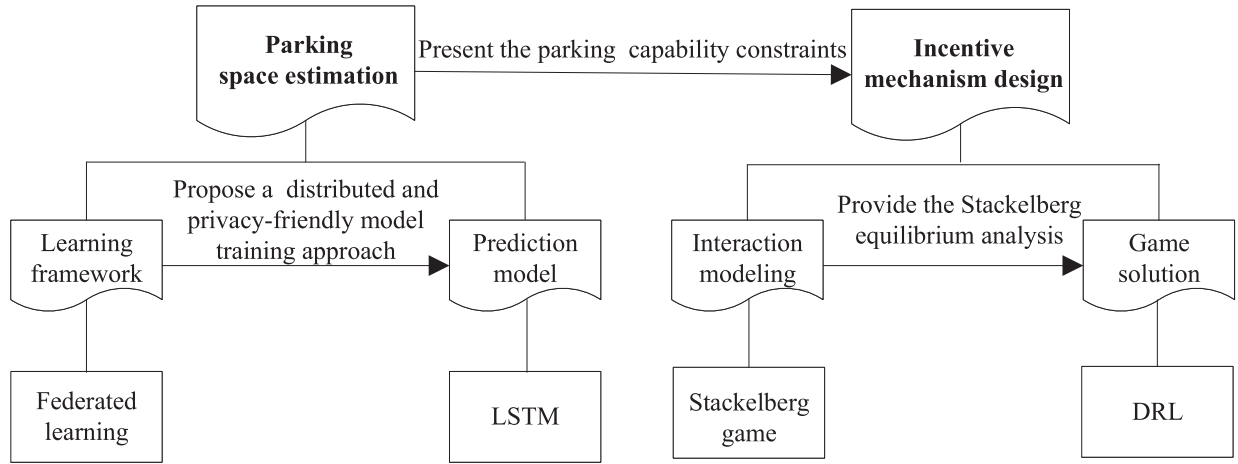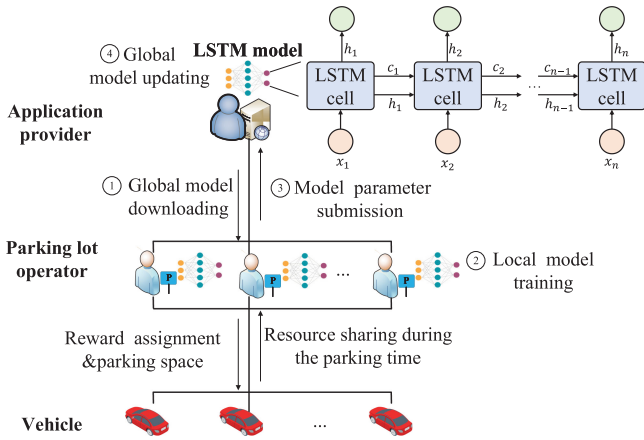
Fig. 1. Methodology of FedParking.

Fig. 2. System design of FedParking.

estimation, and design FedParking where many PLOs keep their raw data local but jointly train the global model by only sharing the updated model parameters with a parameter server. We first describe essential network entities of FedParking, and present the motivations and basic workflow of FedParking. Moreover, a LSTM model is adopted as the global model in FedParking, thereby improving the accuracy performance of parking space estimation.

A. System Model and Network Entity

In this work, we consider a system model shown in Fig. 2. Specifically, for parking demand, vehicles on the roads send their parking requests to an application provider of local parking services. The application provider notifies PLOs in the region to serve the vehicles. Through PVEC, PLOs act as offloading service providers to attract the vehicles to enter their parking spaces by monetary rewards, and incentivize them to receive and complete given computation tasks during the parking time. We describe the essential network entities of FedParking as follows.

- Parking Lot Operator: Each PLO manages a parking lot with limited parking spaces and specifies that an arriving vehicle occupies a parking space. The PLO adopts a learning model to estimate the number of free parking spaces and would like to participate in training the model by using the local data. In addition, when the vehicles are parked in a parking lot, their under-utilized on-board resources can be exploited to provision offloading services. Thus, there is great potential for tackling the computation demand of nearby users and service providers by offloading computation-intensive parts of mobile applications to the parking lot. Consequently, the PLO acts as an employer to reserve computing resources from the PVs to complete the offloading workloads. Here, an incentive mechanism is required to provide economic compensations to the PVs.

- Parked Vehicle: A vehicle submits a parking request for parking space reservation. The parking request contains auxiliary information of the parking service, such as reservation time and original preferences over different parking lots. By knowing current parking lots with their reward policies and workload requirements, the vehicle decides whether to enter a parking lot and share the computing resources to the PLO. For the vehicle, monetary rewards could compensate for parking fee and mainly influence the parking choice. After promising to accept the offloading workloads, the vehicle in the parking lot will receive input data from the PLO through a nearest roadside unit. The vehicle virtualizes the on-board computing resources by the containerization technology to create the task instances and improve the task processing efficiency [9].

- Application Provider: The application provider is a totally trusted coordinator of the parking services in FedParking. It develops a client for vehicles that enables users with entrance to submit the parking requests and choose a suitable parking lot. A client is also developed for PLOs to receive the parking requests and publish their reward policies. With the help of the clients, a legitimate two-way

communication channel between the vehicles and PLOs is built.

B. Federated Learning for FedParking

Learning-based time series prediction model is envisioned as a promising solution to tackle the problems of parking space estimation. The development of a time series prediction model inherits the methodology from traditional learning tasks, including data processing, model building, model training and inference. In [22], the authors have developed a data cleaning algorithm and trained various time series models for different parking lots. Given a training dataset with M time series $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, we regard $\mathcal{X} = \{X_1, X_2, \ldots, X_M\}$ as the inputs and $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_M\}$ as the corresponding ground truth predictions. Let F_Θ represent the time series model parameterized by weight Θ. To train the time series prediction model, we minimize the Mean Square Error (MSE) between the estimation and the ground truth, i.e.,

$$\min_\Theta \mathcal{L}(\Theta, \mathcal{D}) \leftrightarrow \min_\Theta \frac{1}{M} \sum_m (F_\Theta(X_m) - Y_m)^2. \quad (1)$$

In this paper, we consider a feasible scenario where multiple PLOs train and adopt a shared time series prediction model to forecast their available parking spaces.

With the privacy considerations, we introduce FedParking to facilitate the collaboration among the PLOs in parking space estimation. In FedParking, there exist several communication rounds for the PLOs. In each communication round, each PLO firstly trains a global model with local data, and then submits the updated model parameters to aggregate a new global model. According to the concept of federated learning, knowledge of the local models can be shared without revealing the training data to each other. The distributed learning approach finally preserves the data privacy of the participants.

We consider that there exist I vehicles with parking demand and J parking lots. Each vehicle $i, 1 \leq i \leq I$ can choose whether to enter a parking lot $j, 1 \leq j \leq J$. Each PLO has obtained the historical data of parking space information. Let $\mathcal{D}_j$ represent a whole dataset of training data samples of PLO j, and Θ_0 indicate the weight of the global model. Under the federated learning based framework, the goal of FedParking is expressed by

$$\min_{\Theta_0} \frac{\|\mathcal{D}_j\|}{\sum_{1 \leq j \leq J} \|\mathcal{D}_j\|} \sum_{1 \leq j \leq J} \mathcal{L}(\Theta_0, \mathcal{D}_j), \quad (2)$$

where $\|\mathcal{D}_j\|$ denotes the training data size of PLO j, and $\mathcal{L}(\cdot)$ is the loss function in Eqn. (1).

To optimize the above Problem (2), we refer to the original FedAVG algorithm in [1]. As shown in Fig. 2, the overall workflow of FedParking is described as follows.

- Global Model Downloading: An application provider deploys a parameter server to build and iteratively update a time series predict model as the global model for parking space estimation. When the global model is initialized, each PLO is notified to download it.
- Local Model Training: As required by the parameter server, each PLO performs a local model training task by the local

data. After that, they upload the updated model parameters to the parameter server.
- Model Parameter Submission: All the model parameters are collected and aggregated by the parameter server to update the weight Θ_0 and improve the accuracy of the global model.
- Global Model Updating: A new global model is acquired and transmitted to all the PLOs. The iterations of local model training and aggregation repeats until the global model converges.

Note that the whole training procedure of the global model uses the distributed training datasets of all the PLOs, and is executed through peer-to-peer collaboration without exchanging the raw data.

C. LSTM for Fedparking

We apply a LSTM model as the important global model in FedParking. Compared with the original recurrent neural network, LSTM has advantages in remembering and forgetting the information in the time sequence. This enables us to achieve an accurate estimation of the parking spaces.

A LSTM network consists of two data gates to capture the sequence information, i.e., update gate and forget gate. In the following, we provide more details of the procedure of local model training. Take PLO j as an example. By data collection, parking space information (e.g., parking occupancy rate) of a parking lot over different time slots is recorded. We apply a sliding window approach to extract the input time series $X_j = \{x_1, x_2, \ldots, x_{\|\mathcal{D}_j\|}\}$ of the long short-term memory model from the data source. When all the original data samples consist of y time slots, we obtain $\|\mathcal{D}_j\| = y - z + 1$ input time series if the sliding window size is z. Let $H_j = \{h_1, h_2, \ldots, h_{\|\mathcal{D}_j\|}\}$ indicate the hidden states, and $O_j = \{o_1, o_2, \ldots, o_{\|\mathcal{D}_j\|}\}$ represent the output states. We use $W^{\cdot h}$ and $W^{\cdot x}$ to denote the weights, and use $b^\cdot$ to denote the bias.

At the time step t, the value of forget gate Γ_t^f is calculated by

$$\Gamma_t^f = \sigma(W^{fh}h_{t-1} + W^{fx}x_t + b^f), \quad (3)$$

where x_t is the input vector at the time step t and h_{t-1} is the hidden state at the previous time step, $\sigma(\cdot)$ denotes a Sigmoid function that normalizes the value into the range $[0, 1]$. The forget gate determines what information will be eliminated from the cell state. When the value of the forget gate $\Gamma_t^f = 0$, it throws away all the information, and $\Gamma_t^f = 1$ means that it keeps all the information. As for the update gate, it decides what information can be stored in the cell state. The value of update gate is calculated by

$$\Gamma_t^u = \sigma(W^{uh}h_{t-1} + W^{ux}x_t + b^u). \quad (4)$$

Next, we calculate the information of candidate memory cell $\tilde{c}_t$ at the time step t by

$$\tilde{c}_t = \tanh(W^{ch}h_{t-1} + W^{cx}x_t + b^c), \quad (5)$$

where $\tilde{c}_t$ represents the candidate information that should be stored in the cell state. Until now, we obtain $\{\Gamma_t^f, \Gamma_t^u, \tilde{c}_t\}$ to

update the cell state by

$$c_t = \Gamma_t^{\mathrm{f}} \odot c_{t-1} + \Gamma_t^{\mathrm{u}} \odot \tilde{c}_t, \tag{6}$$

where $\odot$ is an operator of executing the point-wise multiplication of two vectors. The output and hidden states are computed by

$$\begin{cases} o_t = \sigma(W^{\mathrm{oh}}h_{t-1} + W^{\mathrm{ox}}x_t + b^o) \\ h_t = o_t \odot \tanh(c_t). \end{cases} \tag{7}$$

Finally, we take o_t as the input for the multilayer perceptron and compute the time series prediction $\hat{x}_{t+1}$ of the next time step. After the local model training, the weights $W^{\cdot\mathrm{h}}$ and $W^{\cdot\mathrm{x}}$, and the bias $b^{\cdot}$ are shared to the parameter server.

With the help of FedParking, each PLO can predict the number of available parking space over time, which is utilized as a parking capacity constraint to adjust the following incentive mechanism design when necessary.

IV. TWO-STAGE STACKELBERG GAME FORMULATION FOR INCENTIVE MECHANISM DESIGN

Within a time period, a batch of parking requests are gathered and the PLOs are notified to process them. In this paper, we model the interactions among the PLOs and vehicles as a Stackelberg game, where each PLO plays as a game leader and provides monetary rewards for the vehicles in Stage I, and each vehicle plays as a game follower and adopts a probabilistic strategy to decide the parking choice and the number of computing resources shared to a PLO in Stage II.

A. Vehicles in Stage II

Vehicle i is incentivized to enter a parking lot j and share its idle computing resources during the parking time. Let r^j represent the crucial incentive parameter of PLO j that indicates the monetary rewards per unit computing resource and unit time. According to the distance between the parking lot and destination, vehicle i may have an original preference p_i^j over parking lot j and the reserved parking duration is d_i. We express the utility gain of vehicle i by $p_i^j r^j f_i^j d_i$, where f_i^j is the decision variable of vehicle i referring to the number of computing resources shared to PLO j. To support offloading services, vehicle i is required by PLO j to undertake workloads w^j. When processing the workloads, a majority of energy is consumed in the CPU execution. Similar to [23], we neglect the energy consumed in the data transmission. For vehicle i, the energy consumption cost of CPU execution is $\kappa_i f_i^{j2} w^j$, where κ_i denotes the energy coefficient of the vehicle and represents the effective switched capacitance relying on the chip architecture [24], [25]. As a consequence, vehicle i acquires a following utility function if accepting the employment of PLO j,

$$U_i^j = p_i^j r^j f_i^j d_i - \kappa_i f_i^{j2} w^j. \tag{8}$$

Motivated by the incentive mechanism design in the previous work [26], we consider that vehicle i makes a probabilistic decision on the parking destination. Particularly, we consider that monetary incentives have a relatively strong positive impact on the parking decision of the vehicle since the monetary rewards can compensate for the parking fee after it shares the idle computing resources to the PLO. We formulate the pairing probability between the vehicle and PLO by only the monetary rewards. Clearly, a vehicle is more likely to enter a parking lot with more monetary rewards. Thus, we set the pairing probability between vehicle i and PLO j as follows

$$\rho_i^j = \frac{r^j}{\sum_{1 \le k \le J} r^k}. \tag{9}$$

For vehicle i, we have $\sum_{1 \le j \le J} \rho_i^j = 1$. Considering J PLOs, the expected utility of vehicle i is

$$\mathcal{U}_i(\mathbf{f}_i, \mathbf{r}) = \sum_{1 \le j \le J} \rho_i^j U_i^j, \tag{10}$$

where $\mathbf{f}_i = \{f_i^1, \ldots, f_i^J\}$ and $\mathbf{r} = \{r^1, \ldots, r^J\}$. Each vehicle i decides the optimal strategy vector $\mathbf{f}_i$ to maximize $\mathcal{U}_i$ according to the reward policy vector $\mathbf{r}$.

B. Parking Lot Operators in Stage I

PLO j collects a certain number of computing resources from the PVs and is able to provision offloading services. After satisfying the computation demand of offloading users, PLO j obtains revenue g^j per unit computing resource and unit time. Considering the total monetary cost for I PVs, PLO j obtains the expected utility in terms of profit as follows

$$\mathcal{V}^j(r^j, \mathbf{r}^{-j}, \mathbf{f}^j) = \sum_{1 \le i \le I} \rho_i^j (g^j - r^j) f_i^j d_i, \tag{11}$$

where $\mathbf{r}^{-j}$ is a reward policy vector released by all the PLOs except PLO j and $\mathbf{f}^j = \{f_1^j, \ldots, f_I^j\}$. Each PLO j decides the optimal strategy r^j to maximize $\mathcal{V}^j$ according to the strategies of all the other PLOs (i.e., $\mathbf{r}^{-j}$) and the strategies of all the vehicles (i,e., $\mathbf{f}^j$). Note that there exists a non-cooperative game among the J PLOs in the incentive mechanism. Moreover, two essential constraints of r^j are presented. The first constraint named by parking capacity constraint is expressed by $\sum_{1 \le i \le I} \rho_i^j \le n^j$, where n^j is acquired from the above FedParking scheme and represents the predicting number of available parking spaces in the parking lot. This means that PLO j should adjust r^j well to control the expected number of arriving vehicles with respect to the given parking capacity constraint. Considering a budget constraint, r^j is constrained within an upper limit $r_{\max}^j \le g^j$.

Each player (i.e., each PLO and vehicle) of the Stackelberg game has its own interest. We aim to find a unique Stackelberg equilibrium at which each leader (PLO) obtains the maximal expected utility given the best responses from the followers (vehicles). At the time, no single PLO or vehicle will have any motivation to unilaterally change its decision. Given the multiple-leader multi-follower Stackelberg game, we define the Stackelberg equilibrium as follows.

Definition 1 (Stackelberg Equilibrium): There exist an optimal computing resource vector denoted as $\mathbf{f}_i^*$ and an optimal monetary reward r^{j*} for vehicle i and PLO j, respectively. $(\mathbf{f}^* = \{\mathbf{f}_i^*\}, \mathbf{r}^* = \{r^{1*}, \ldots, r^{J*}\}, \forall i$ and $j)$ is the Stackelberg

equilibrium if and only if the following conditions are satisfied.

$$\begin{cases} \mathcal{V}^j(r^{j*}, \mathbf{r}^{-j*}, \mathbf{f}^{j*}) \geq \mathcal{V}^j(r^j, \mathbf{r}^{-j*},), \forall j, \\ \mathcal{U}_i(\mathbf{f}_i^*, \mathbf{r}^*) \geq \mathcal{U}_i(\mathbf{f}_i, \mathbf{r}^*), \forall i. \end{cases}$$

We find the subgame perfect Nash equilibrium to achieve the Stackelberg equilibrium, which ensures that the expected utility of each PLO is maximized subject to the best responses from the vehicles. In the proposed Stackelberg game, there exists a subgame among the PLOs in the upper stage and they strictly compete in a non-cooperative fashion. The Nash equilibrium of the subgame is denoted as $(r^{j*}, \mathbf{r}^{-j*})$, where the expected utility of each PLO j cannot be further improved by adopting a different strategy other than the given strategies of the other PLOs $\mathbf{r}^{-j*}$. In the bottom stage, the best response of vehicle i is acquired as $\mathbf{f}_i^*$ by maximizing the expected utility with respect to $\mathbf{r}^*$. To reach the Stackelberg equilibrium, we adopt the backward induction method in the following.

V. STACKELBERG EQUILIBRIUM ANALYSIS UNDER COMPLETE INFORMATION

We temporarily neglect the parking capacity constraints and provide the theoretical analysis to prove the existence and uniqueness of the above Stackelberg equilibrium.

A. Best Response of a Vehicle

We first pay attention to the best response of a follower. Given the reward policy vector $\mathbf{r}$ published by all the PLOs, each vehicle i determines $\mathbf{f}_i$ to maximize its $\mathcal{U}_i$. We take the first-order and second-order derivatives of $\mathcal{U}_i$ with respect to f_i^j,

$$\frac{\partial \mathcal{U}_i}{\partial f_i^j} = \rho_i^j(p_i^j r^j d_i - 2\kappa_i f_i^j w^j),$$

$$\frac{\partial^2 \mathcal{U}_i}{\partial f_i^{j2}} = -2\rho_i^j \kappa_i w^j < 0.$$

The negative second-order derivative of $\mathcal{U}_i$ indicates that $\mathcal{U}_i$ is strictly concave with respect to f_i^j. As a consequence, for vehicle i, the best response f_i^{j*} can be calculated by

$$f_i^{j*} = \frac{p_i^j d_i}{2\kappa_i w^j} r^j. \tag{12}$$

The above result means that more monetary rewards are provided to the vehicle, more computing resources will be shared to the PLO, which is consistent with the intuition.

B. Nash Equilibrium Analysis for PLOs

We realize that each PLO is selfish to employ the vehicles with idle computing resources and maximize the payoffs. The PLOs play a non-cooperative game because of the competition effect. In the non-cooperative game, the player set consisting of J PLOs is finite and strategy spaces of the PLOs are nonempty, convex, and compact subsets of the Euclidean spaces. The Nash equilibrium exists in the non-cooperative game if we can prove the concavity of the utility functions in the strategy spaces.

Theorem 1: A Nash equilibrium exists in the non-cooperative game among the PLOs.

Proof: By substituting f_i^{j*} into $\mathcal{V}^j$, we update

$$\mathcal{V}^j = \frac{g^j r^j \sum_{1 \leq i \leq I} \lambda_i^j d_i - r^{j2} \sum_{1 \leq i \leq I} \lambda_i^j d_i}{\sum_{1 \leq k \leq J} r^k} = \frac{g^j \mu^j r^j - \mu^j r^{j2}}{\sum_{1 \leq k \leq J} r^k}, \tag{13}$$

where $\mu^j = \sum_{1 \leq i \leq I} \lambda_i^j d_i$. Furthermore, we take the first-order and second-order derivatives of $\mathcal{V}^j$ with respect to r^j,

$$\frac{\partial \mathcal{V}^j}{\partial r^j} = \frac{(g^j \mu^j - 2\mu^j r^j) \sum_{1 \leq k \leq J} r^k - g^j \mu^j r^j + \mu^j r^{j2}}{\left(\sum_{1 \leq k \leq J} r^k\right)^2}$$

$$= \frac{g^j \mu^j \sum_{k \neq j} r^k - \mu^j r^{j2} - 2\mu^j r^j \sum_{k \neq j} r^k}{\left(\sum_{1 \leq k \leq J} r^k\right)^2},$$

$$\frac{\partial^2 \mathcal{V}^j}{\partial r^{j2}} = -2 \frac{g^j \mu^j \sum_{k \neq j} r^k - \mu^j r^{j2} - 2\mu^j r^j \sum_{k \neq j} r^k}{\left(\sum_{1 \leq k \leq J} r^k\right)^3}$$

$$- \frac{2\mu^j r^j + 2\mu^j \sum_{k \neq j} r^k}{\left(\sum_{1 \leq k \leq J} r^k\right)^2}.$$

When $\partial \mathcal{V}^j / \partial r^j = 0$, $\partial^2 \mathcal{V}^j / \partial^2 r^{j2} < 0$. Utility function of PLO j is quasi-concave. So we conclude that the Nash equilibrium exists in this non-cooperative game. ∎

Theorem 2: The Nash equilibrium of the non-cooperative game among the PLOs is unique.

Proof: Based on the first-order optimality condition $\partial \mathcal{V}^j / \partial r^j = 0$, we have

$$r^{j*} = -\sum_{k \neq j} r^k + \sqrt{\left(\sum_{k \neq j} r^k\right)^2 + g^j \sum_{k \neq j} r^k} = \phi^j(\mathbf{r}). \tag{14}$$

Considering the upper limit, we update

$$r^{j*} = \begin{cases} \phi^j(\mathbf{r}), & 0 \leq \phi^j(\mathbf{r}) < r_{\max}^j, \\ r_{\max}^j, & \phi^j(\mathbf{r}) \geq r_{\max}^j. \end{cases} \tag{15}$$

Next, we pay attention to the case that $r^{j*} < r_{\max}^j$. We prove the uniqueness of the Nash equilibrium by validating that the best response function $\phi^j(\mathbf{r})$ is a standard function, which satisfies three properties as follows

- *Positivity:* $\phi^j(\mathbf{r}) > 0$.
- *Monotonicity:* if $\mathbf{r}' > \mathbf{r}$, then $\phi^j(\mathbf{r}') > \phi^j(\mathbf{r})$.
- *Scalability:* for any $\alpha > 1$, $\alpha\phi^j(\mathbf{r}) > \phi^j(\alpha\mathbf{r})$.

First, due to $g^j > 0$ and $\mathbf{r} > \mathbf{0}$, $\phi^j(\mathbf{r}) > 0$ is always satisfied. Second, we know $(\sum_{k \neq j} r^k + 0.5g^j)^2 > (\sum_{k \neq j} r^k)^2 + g^j \sum_{k \neq j} r^k$. Thus,

$$\frac{\partial \phi^j(\mathbf{r})}{\partial(\sum_{k \neq j} r^k)} = -1 + \frac{0.5 * (2\sum_{k \neq j} r^k + g^j)}{\sqrt{(\sum_{k \neq j} r^k)^2 + g^j \sum_{k \neq j} r^k}} > 0$$

When $\mathbf{r'} > \mathbf{r}$, $\sum_{k \neq j} r^{k\prime} > \sum_{k \neq j} r^k$ and we have $\phi^j(\mathbf{r'}) > \phi^j(\mathbf{r})$. Finally, $\forall \alpha > 1$,

$$\alpha \phi^j(\mathbf{r}) = -\alpha \sum_{k \neq j} r^k + \alpha \sqrt{\left(\sum_{k \neq j} r^k\right)^2 + g^j \sum_{k \neq j} r^k}$$

$$= -\alpha \sum_{k \neq j} r^k + \sqrt{\left(\sum_{k \neq j} \alpha r^k\right)^2 + \alpha g^j \sum_{k \neq j} \alpha r^k}$$

$$> -\alpha \sum_{k \neq j} r^k + \sqrt{\left(\sum_{k \neq j} \alpha r^k\right)^2 + g^j \sum_{k \neq j} \alpha r^k}$$

$$= \phi^j(\alpha \mathbf{r})$$

Note that we have proved that the best response function of PLO j satisfies the above three properties in this case. When $r^{j*} = r^j_{\max}$, these three properties are still satisfied. Then we conclude that standard best response functions of all the PLOs have the unique Nash equilibrium. ∎

Theorem 3: There exists a unique Stackelberg equilibrium in the Stackelberg game.

Proof: Until now, we have proved that the best response of a follower is unique in the second stage and the Nash equilibrium among the leaders is also unique in the first stage. We can conclude that the Stackelberg equilibrium is unique. The proof is completed. ∎

To reach the Stackelberg equilibrium, we utilize the following rule to update the monetary rewards in each iteration k to gradually approach the Nash equilibrium in the first stage,

$$r^{j,k} = r^{j,k-1} + \lambda^j r^{j,k-1} \frac{\partial \mathcal{V}^j}{\partial r^j}, \quad \forall j, \tag{16}$$

where λ^j denotes a learning rate, and $\partial \mathcal{V}^j / \partial r^j$ is approximately calculated by the central difference method,

$$\frac{\partial \mathcal{V}^j}{\partial r^j} \approx \frac{\mathcal{V}^{j,k}_+ - \mathcal{V}^{j,k}_-}{2\Delta r}, \tag{17}$$

where $\mathcal{V}^{j,k}_+ = \mathcal{V}^j(r^{j,k-1} + \Delta r, \mathbf{r}^{-j,k-1})$, $\mathcal{V}^{j,k}_- = \mathcal{V}^j(r^{j,k-1} - \Delta r, \mathbf{r}^{-j,k-1})$ and Δr is a small constant indicating the minor change of the reward policy. Given the updating rule of the monetary rewards, two different approaches can be considered by the PLOs to iteratively adjust their strategies, including Gauss-Seidel and Jacobi based algorithms. In the former algorithm, the PLOs update their strategies sequentially. In the latter algorithm, they update the strategies in parallel. In this paper, we adopt the Jacobi based algorithm.

VI. A LEARNING BASED INCENTIVE MECHANISM BY DRL

In Section V, we focus on analyzing the Stackelberg equilibrium under complete information, without taking into account the parking capacity constraints. However, both the arrivals of the vehicles and number of available parking spaces in each parking lot are varying over time. It is difficult to obtain the closed-form Stackelberg equilibrium in practical scenarios. In addition, the above theoretical analysis is based on the centralized decision making which requires to collect private information of all the players to derive the solution of the Stackelberg
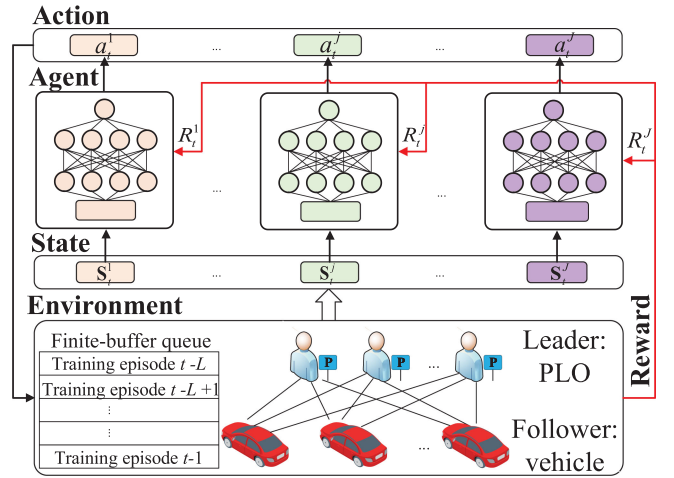

Fig. 3. A learning based incentive mechanism by multi-agent DRL.

game. This violates the individual privacy of the players. Under the complicated decision-marking conditions and for the fulfillment of privacy-related requirements, advanced artificial intelligence technologies can be potential tools to facilitate the incentive mechanism design [27], [28]. Hence, we are motivated by [29] to apply DRL as an online algorithm to tackle the incentive mechanism design problem of the multi-leader multi-follower Stackelberg game under incomplete information. The extensively concerned DRL technique has outstanding ability in adapting to the non-static environment and dynamically resolving the complicated decision-making problems according to environment variations [30]. Our DRL approach enables each PLO as an agent to quickly learn a near-optimal decision in a distributed and privacy-friendly manner. We first outline a learning framework of the whole DRL based incentive mechanism. After that, we formulate the incentive design problem as a DRL learning task and provide more details of the neural network configuration and optimization method based on the proposed learning framework.

A. Overview of the Learning Framework

To reach the Stackelberg equilibrium, each PLO becomes an agent in the DRL approach reacting to the environment consisting of all the vehicles, as shown in Fig. 3. PLO j is agent j, which refers to the historical strategies of the other agents and determines an action according to the current state.

Specifically, PLO j in each training episode $t(1 \leq t \leq T)$ interacts with the environment to know the current state $\mathbf{S}^j_t$, which consists of reward policy history of the other PLOs and its profit parameter. PLO j plays as a game leader that decides the action a^j_t and releases the reward policy p^j_t. After that, a vehicle with parking demand in the environment plays as a follower that confirms the optimal strategy based on Eqn. (12). By collecting the strategies of all the vehicles, the environment computes the learning rewards R^j_t for each PLO. At the same time, the environment saves all the historical data of the actions into a finite-buffer queue. Let L denote the size of the finite-buffer queue. Here, an application provider of the parking services

in the region could become such an essential coordinator to establish the learning environment. The application provider is convenient to collect historical strategies of each player and accordingly update the finite-buffer queue. It extracts the related information from the finite-buffer queue to generate new states for the corresponding PLOs. Hence, the next training episode is conducted.

Note that during the training procedure, each player cannot obtain any privacy-related information of the other players that influences their decisions. As a consequence, we solve the Stackelberg game without leaking private information and causing privacy threats to the players.

B. Design Details

In the following, we describe a DRL learning task under the proposed learning framework. For each player, the decision-making procedure is performed independently according to the others' past strategy information and its own property. We further formulate the interactions among the PLOs as a multi-agent markov decision process.

a) State space: We focus on a case where state space of each player is partially observable. For PLO j, the state is formed by the past strategies of the other PLOs and its own properties, which is denoted as $\{\mathbf{r}_{t-L}^{-j}, \mathbf{r}_{t-L+1}^{-j}, \ldots, \mathbf{r}_{t-1}^{-j}\}$ and g^j, respectively. In the training episode t, the PLO interacts with the environment to know the current state, which is represented by $\mathbf{S}_t^j = \{\mathbf{r}_{t-L}^{-j}, \mathbf{r}_{t-L+1}^{-j}, \ldots, \mathbf{r}_{t-1}^{-j}, g^j\}$.

b) Action space: We use $\mathcal{A} = \{a_t^j, \forall j\}$ to represent the actions of all PLOs. In the training episode t, PLO j determines the action a_t^j with respect to the state $\mathbf{S}_t^j$. The action refers to the reward policy. To improve the learning efficiency, we set lower and upper bounds $[r_{\min}^j, r_{\max}^j]$ for the action. Since $a_t^j \in [r_{\min}^j, r_{\max}^j]$, the action space of a_t^j is continuous and we simply normalize the range of a_t^j to $[0, 1]$ by letting $a_t^j = (r_t^j - r_{\min}^j)/r_{\max}^j$.

c) Reward function: The reward space is defined as $\mathcal{R} = \{R_t^j, \forall j\}$. We consider a time-varying parking capacity constraint for each agent. In the training episode t, n_t^j is the available parking capacity of agent j. This means that for all the I vehicles, the expected number of the vehicles choosing PLO j is limited by n_t^j. With respect to the parking capacity constraint, we are motivated to express the reward function of agent j as

$$R_t^j =$$

$$\begin{cases} \sum_{1 \leq i \leq I} \rho_i^j (g^j - r_t^j) f_i^j d_i, & \sum_{1 \leq i \leq I} \rho_i^j \leq n_t^j \\ \frac{n_t^j}{I} \sum_{1 \leq i \leq I} (g^j - r_t^j) f_i^j d_i - \mathcal{P}(\sum_{1 \leq i \leq I} \rho_i^j), & \text{otherwise.} \end{cases}$$
(18)

where $\mathcal{P}(\cdot)$ is a penalty function to control the expected number of arriving vehicles. The function is predefined by

$$\mathcal{P} = \frac{\iota^j}{n_t^j} \max \left(\sum_{1 \leq i \leq I} \rho_i^j - n_t^j, 0 \right),$$
(19)

where ι^j is a penalty factor.

d) Learning Objective: We first denote the reward policy of PLO j parameterized by θ^j as π_θ^j, which is defined as $\pi_\theta^j : \mathbf{S}^j \to a^j$. Let $V_{\pi_\theta}^j(\mathbf{S}^j)$ indicate the value function for the given state, and $Q_{\pi_\theta}^j(\mathbf{S}^j, a^j)$ become the value function for the given state and action. The learning objective of agent j is shown by

$$\theta_*^j = \arg \max_{\theta^j} L_j(\pi_\theta^j)$$

$$= \arg \max_{\theta^j} \mathbb{E} \left[V_{\pi_\theta}^j(\mathbf{S}_0^j) \right]$$

$$= \arg \max_{\theta^j} \mathbb{E} \left[Q_{\pi_\theta}^j(\mathbf{S}_0^j, a_0^j) \,\middle|\, \pi_\theta^j \right],$$
(20)

where

$$V_{\pi_\theta}^j(\mathbf{S}^j) = \mathbb{E} \left[\overline{R}_t^j \,\middle|\, \mathbf{S}_t^j = \mathbf{S}^j, \Pi \right],$$

$$Q_{\pi_\theta}^j(\mathbf{S}^j, a^j) = \mathbb{E} \left[\overline{R}_t^j \,\middle|\, \mathbf{S}_t^j = \mathbf{S}^j, a_t^j = a^j, \Pi \right],$$

$$\overline{R}_t^j = \sum_{k=t}^{T} \gamma^{k-t} R_t^j.$$
(21)

Here, $\Pi = \{\pi_\theta^j, \forall j\}$ represents the whole policy set of all PLOs, $\overline{R}_t^j$ is the expected discounted reward for PLO j in all the T training episodes, and $\gamma \in [0, 1]$ indicates the discounted factor.

C. Actor-Critic Networks and Policy Optimization

We train the above DRL model based on the policy-based methods and resorts to the typical actor-critic model. We design a decentralized actor network and a decentralized critic network for each agent. For agent j, it maintains actor and critic networks parameterized by θ^j and ω^j for approximating the policy and value function, respectively. We apply the policy gradient method to optimize the learning objective defined in Eqn. (20). Referring to the policy optimization theorem proposed in [31], the policy gradient is computed by

$$\nabla_{\theta^j} L^j = E_{\pi_\theta^j(\mathbf{S}^j)} [\nabla_{\theta^j} \log \pi_\theta^j(\mathbf{S}^j, a^j) A_{\pi_\theta}^j(\mathbf{S}^j, a^j)]$$

$$\approx E_{\pi_\theta^j(\mathbf{S}^j)} [P^j \nabla_{\theta^j} \log \pi_\theta^j(\mathbf{S}^j, a^j) A_{\pi_\theta}^j(\mathbf{S}^j, a^j)], \quad (22)$$

where $P^j = \frac{\pi_\theta^j(\mathbf{S}^j|a^j)}{\hat{\pi}_\theta^j(\mathbf{S}^j|a^j)}$, $A_{\pi_\theta}^j(\mathbf{S}^j, a^j) = Q_{\pi_\theta}^j(\mathbf{S}^j, a^j) - V_{\pi_\theta}^j(\mathbf{S}^j)$. $A_{\pi_\theta}^j(\mathbf{S}^j, a^j)$ is the advantage function for state and action and $\hat{\pi}_\theta^j(\mathbf{S}^j, a^j)$ is the policy for importance sampling. Furthermore, we design the training procedure according to the proximal policy optimization algorithm in [32], which is the state-of-the-art policy gradient method with excellent stability. Therefore, the policy gradient is further clipped as

$$\nabla_{\theta^j} L^j \approx E_{\pi_\theta^j(\mathbf{S}^j)} [\nabla_{\theta^j} \log \pi_\theta^j(\mathbf{S}^j, a^j) \mathcal{C}_{\pi_\theta}^j(\mathbf{S}^j, a^j)]$$
(23)

where

$$\mathcal{C}_{\pi_\theta}^j(\mathbf{S}^j, a^j) = \min[P^j A_{\pi_\theta}^j(\mathbf{S}^j, a^j), \mathcal{F}(P^j) A_{\pi_\theta}^j(\mathbf{S}^j, a^j)],$$
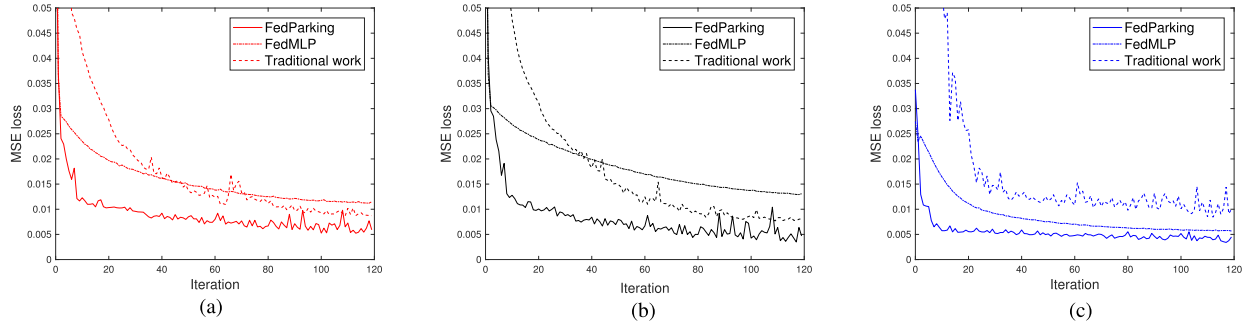(24)

Fig. 4.   Comparison of training loss under different approaches. (a) Parking lot operator 1. (b) Parking lot operator 2. (c) Parking lot operator 3.

$$\mathcal{F}(P^j) = \begin{cases} 1 + \varepsilon, & P^j > 1 + \varepsilon \\ P^j, & 1 - \varepsilon \le P^j \le 1 + \varepsilon \\ 1 - \varepsilon, & P^j < 1 - \varepsilon \end{cases} \qquad (25)$$

and ε is an adjustable hyper-parameter. After that, we update the actor and critic models by the mini-batch stochastic gradient ascent and descent methods, respectively.

In Fig. 3, the actor network of a PLO is the multilayer perceptron with two hidden layers, consisting of 200 neurons and 50 neurons, respectively. For the critic network, we combine connected-layer and differentiable neural computer model in [33] to build the whole model for value estimation. Differentiable neural computer model is introduced as a type of recurrent neural network with internal memory module, and has been successful applied by the previous works to solve the partially observable markov decision process problem. The agents gradually learn the optimal strategies with the training procedure continuing. When the training procedure converges, they determine the strategies based on the outputs of the actor networks.

VII. NUMERICAL RESULTS

A. Parameter Setting

We evaluate the proposed FedParking by numerical studies. First, we study the overall performance of FedParking on the real-world dataset named by Birmingham parking dataset.[1] The dataset includes the parking history operated by National Car Parks in Birmingham, United Kingdom, and is updated every 30 minutes from 8:00 AM to 5:00 PM. In our experiments, we select 3 parking lots, referring to "BHMEURBRD01," "BHMEURBRD02" and "Bull Ring," to collaboratively learn a shared LSTM model for parking space estimation. A local dataset of a PLO is partitioned using a 80/20 training/testing split. For a LSTM model as the global model in FedParking, the size of hidden state is 256. To extract the input time series of the LSTM model, we set the sliding window size $z = 15$.

A PLO acts as an offloading service provider and employs PVs as edge computing nodes. When a PV is effectively stimulated to share its idle computing resources, the PV continuously receives offloading tasks within a specific time period whose duration

[1]https://archive.ics.uci.edu/ml/datasets/Parking+Birmingham
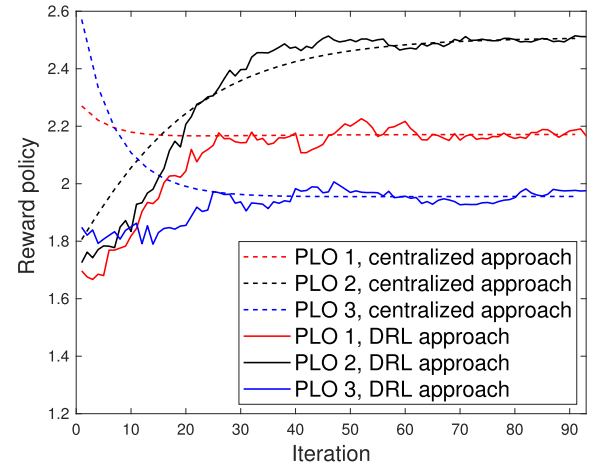

Fig. 5.   Performance comparison among different approaches without the parking capacity constraints.

ranges from 5 to 20 minutes. The arrival process of the tasks is a typical Poisson process, where the average arrival rate and workload of the tasks are chosen from $U[1, 3]$ per minute and $U[2, 5]$ giga CPU cycles, respectively. We assume that there are 35 vehicles with parking demand. For a PV, reserved parking time d, computing capability f and hardware parameter κ are uniformly distributed in the range of $U[20, 100]$ minutes, $U[0.5, 3.5]$ GHz and $U[1, 10]$e-28, respectively. The preference parameter p over different parking lots is randomly generated within $(0,1)$. For a PLO, profit parameter g follows the uniform distribution $U[3, 5]$. As for the DRL experiments, we conduct them using Tensorflow 1.14 on Ubuntu 20.04 LTS with CUDA 10.1. We set the parameters $L = 5, r_{\min} = 0.2, r_{\max} = 3, \iota = 2, T = 20, \gamma = 0.95$ and $\varepsilon = 0.1$ by default.

B. FedParking for Parking Space Estimation

In this paper, we study an integration of federated learning and LSTM to design FedParking, in which all PLOs collaborate to train a shared LSTM model for parking space estimation, ultimately improving the prediction accuracy while avoiding data transfer among the PLOs. In addition to FedParking, a general approach based on federated learning for parking space estimation is named by FedMLP, which adopts a multilayer perceptron model as the global model in federated learning.
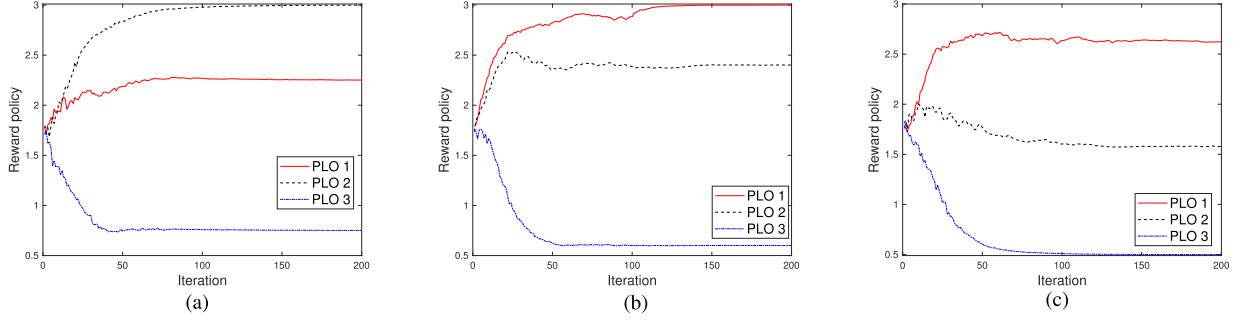
Fig. 6.   Convergence of the DRL approach with different parking capacity constraints. (a) Case 1. (b) Case 2. (c) Case 3.
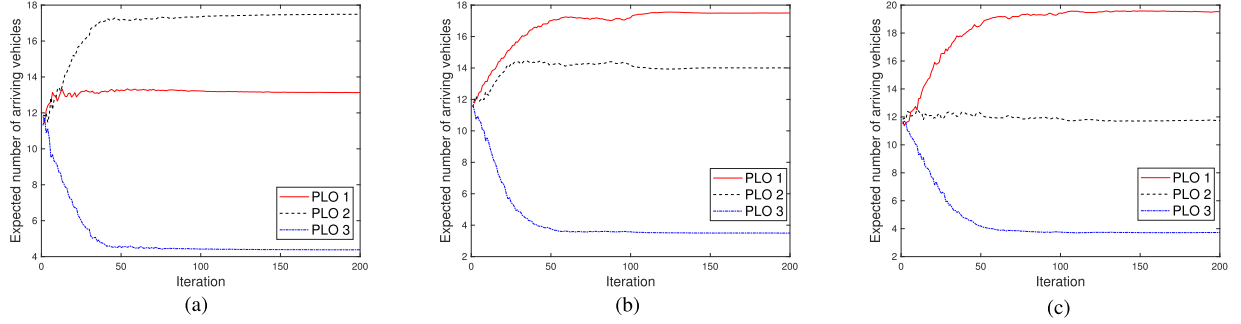

Fig. 7.   Comparison of the expected number of arriving vehicles under the DRL approach with different parking capacity constraints. (a) Case 1. (b) Case 2. (c) Case 3.

In the traditional work, each PLO only utilizes the local data for isolated training of its LSTM model. To evaluate the performance of our proposed approach in comparison with other approaches, we set the local batch size as 64. The local epoch is set as 1 for FedParking and FedMLP.

To highlight the advantage of FedParking, we compare our approach with the baseline approaches. We evaluate the training loss by using MSE. The performance regarding the MSE loss for three approaches are assessed in Fig. 4. FedParking ensures that knowledge of the local models is securely shared among the PLOs and simultaneously apply the LSTM model for accurate parking space estimation. As a consequence, FedParking converges significantly faster than the baseline approaches. For each PLO, experimental results show that compared with the baseline approaches, the proposed FedParking always achieves smaller MSE, thereby achieving a higher prediction accuracy.

C. DRL for Incentive Mechanism Design

In the following, we perform numerical analysis of the DRL based incentive mechanism under different scenarios. We first investigate the performance of our DRL approach in the ideal case, where parking capacity constraints are neglected for the PLOs. To reach the Stackelberg equilibrium, our work proposes a DRL approach to make each PLO as an agent learn a near-optimal solution in a distributed manner while the traditional work considers a centralized approach by collecting private information of all the PLOs and vehicles to execute the Jacobi based algorithm, as introduced in Section V-B.
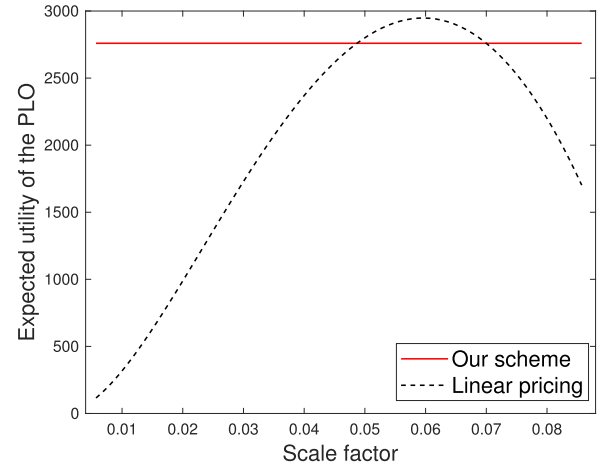

Fig. 8.   Comparison of the expected utility of a PLO under different schemes.

Here, we set the learning rate $\lambda = 1e\text{-}3$ and constant $\Delta r = 1e\text{-}2$ for the Jacobi based algorithm in the centralized approach. Without the parking capacity constraints, the penalty function is removed from reward function of each PLO in the DRL approach. As shown in Fig. 5, we compare the performance of our DRL approach with the centralized approach in the ideal case. The convergence of these two approaches is validated and the centralized approach converges relatively faster, owing to the perfect complete information condition. Beside, we observe that the learning error of the DRL approach is very minor. Our approach can finally converge to a near-optimal solution which
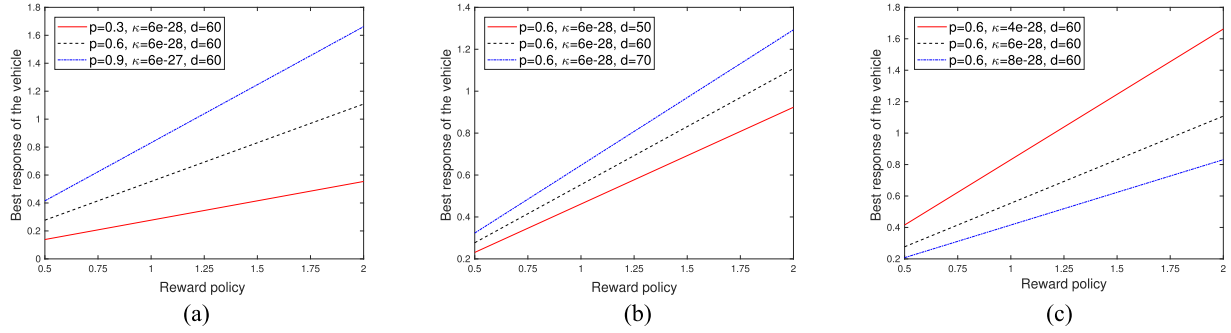
Fig. 9. Comparison of the best response of a vehicle under different external and internal parameters. (a) Comparison of the best response when varying p and r. (b) Comparison of the best response when varying d and r. (c) Comparison of the best response when varying κ and r.

can closely approximate the optimal solution in the centralized approach within the limited iterations.

In addition, we investigate the performance of our DRL approach under different parking capacity constraints. For simplicity, we consider three cases of the distribution of the parking capacity constraints $[n^1, n^2, n^3]$, i.e., Case 1: [15, 20, 5], Case 2: $[n^1, n^2, n^3] = [25, 20, 5]$ and Case 3: $[n^1, n^2, n^3] = [35, 20, 5]$ for 3 PLOs, respectively. From Fig. 6, we realize that the DRL approach converges finally under the parking capacity constraints. The expected number of the vehicles that choose to enter a specific parking lot is also controlled well by adopting an appropriate reward policy, as illustrated by Fig. 7. For a PLO, reward policy r referrings to monetary rewards per unit computing resource and unit time, roughly decreases with the reduction of the parking capacity constraint. Given a lower value of n^j, PLO j is forced to provide less incentive rewards for arriving vehicles in order to avoid congestion to the parking lot. For example, when PLO 1 has only 15 free parking spaces in Case 1, r^1 is mainly influenced by the current parking capacity constraint. At the equilibrium point, the equality constraint $\sum_{1 \leq i \leq I} \rho_i^1 = n^1$ is approximately satisfied. The similar observation is found for PLO 3, which has only 5 free parking spaces. In Case 2, n^1 is increased such that the PLO can improve r^1 on demand. At this time, PLO 1 has the maximal number of free parking spaces. To attract more vehicles, the PLO is encouraged to improve r^1, which is close to the upper limit in Case 2. By knowing the obvious improvement of r^1, PLO 2 and PLO 3 realize that more vehicles will be attracted by PLO 1. It is an alternative for them to improve their expected utilities by reducing r^2 and r^3 to save unnecessary monetary cost. Particularly, r^2 sharply decreases from 3.000 to 2.402 in Case 2. With the continuing increase of n^1, this leads to the further reduction of r^2 and r^3 in Case 3. The competitive strength of PLO 1 is enhanced when n^1 is far greater than n^2 and n^3. To effectively improve the expected utility, PLO 1 is active to reduce the addition monetary cost since r^1 with a slightly lower value still can attract sufficient vehicles, because of the simultaneous reduction of r^2 and r^3.

We compare the expected utility of a PLO under linear pricing scheme and our Stackleberg game based scheme in Fig. 8. Here, we take PLO 1 as an example and set the parking capacity constraints in Case 3. According to the number of available parking spaces n^1, it could be a simplified approach for PLO 1 to decide the reward policy r^1 by considering the

linear relationship between n^1 and r^1, namely, $r^1 = \zeta n^1$, where $\zeta \in [r^{\min}/n^1, r^{\max}/n^1]$ is a scale factor. Compared with the linear pricing scheme, our scheme derives the optimal reward policy by the Stackelberg game model with DRL approach. With respect to the varying ζ, the expected utility of PLO 1 is changing. Note that PLO 1 can adopt a fixed or dynamic reward policy by controlling the scale factor. But the determination of ζ is a challenging task and cannot guarantee the high-level expected utility for the PLO all the time. By the traversal method, we seek the optimal value of ζ to achieve the maximal expected utility. We find that the expected utility of PLO 1 calculated by our scheme significantly approximates the above maximal one. The approximation error is limited within about 6%. Moreover, the linear pricing scheme is not always feasible since the baseline scheme neglects the parking capacity constraint. In this regard, our scheme applies the Stackelberg game theoretic approach and presents a learning based solution to quickly seek the suboptimal reward policy with the given parking capacity constraint.

Fig. 9 shows the best response of a PV under the external and internal impacts. For the PV, the best response is in terms of the optimal number of computing resources shared to a PLO. Supposing that the PV is stimulated by PLO 1 and we study the best response f^*. The external impact refers to the given reward policy r, and internal factors include its preference parameter p, parking time d and hardware parameter κ. Clearly, with the increase of r, f^* is gradually improved. According to Eqn. (12), both p and d have a positive effect on the improvement of f^* while $kappa$ has a negative effect. The results in Fig. 9 are consistent with the above analysis. For example, when d is increased from 50 to 70 minutes, the vehicle would like to share more idle computing resources and earn monetary rewards during the parked time, thereby f^* increases about 40% with respect to $r = 1.5, p = 0.6$ and $\kappa = 6e - 28$.

In summary, we present a joint federated learning and DRL approach to design FedParking and ultimately facilitate the parking management in smart cities. In FedParking, a secure and distributed learning approach is applied among the PLOs for high-accuracy parking space estimation without requiring data transfer among them. The DRL approach resorts to the iterative and distributed decision making on the PLO side and may scarify the convergence speed compared with the centralized approach. But our approach guarantees the convergence accuracy, and does

not require to collect private information of all the players of the Stackelberg game.

VIII. CONCLUSION

In this paper, we introduced FedParking to study the federated learning based parking space estimation with PVEC management. In the scheme, PLOs were instructed to train a shared LSTM model for parking space estimation without exchanging the raw data among them. A parking space constraint was accordingly presented to each PLO, which acts as an incentive designer to determine how to stimulate the vehicles to enter the parking spaces and share their idle computing resources for offloading services in PVEC. We formulated the interactions among the PLOs and vehicles as a multi-leader multi-follower Stackelberg game and provided the theoretical Stackelberg equilibrium analysis under complete information. Considering dynamic arrivals of the vehicles and time-varying parking capacity constraints, a DRL approach was particularly proposed to reach the Stackelberg equilibrium in a distributed yet privacy-friendly manner. Finally, numerical results demonstrated that our scheme is effective and efficient for high-accuracy parking space estimation, and can seek a near-optimal solution under the complicated conditions.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Stat. (AISTATS)*, Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[3] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[4] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul./Aug. 2020.

[5] Z. M. Fadlullah and N. Kato, "HCP: Heterogeneous computing platform for federated learning based collaborative content caching towards 6G networks," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2020.2986238.

[6] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.

[7] C. Zhang, S. Zhang, J. J. Q. Yu, and S. Yu, "FASTGNN: A topological information protected federated learning approach for traffic speed forecasting," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8464–8474, Dec. 2021.

[8] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, Aug. 2002.

[9] X. Huang, R. Yu, S. Xie, and Y. Zhang, "Task-container matching game for computation offloading in vehicular edge computing and networks," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.2990462.

[10] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Federated learning for data privacy preservation in vehicular cyber-physical systems," *IEEE Netw.*, vol. 34, no. 3, pp. 50–56, May/Jun. 2020.

[11] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.

[12] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, 2020.

[13] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[14] X. Huang, R. Yu, D. Ye, L. Shu, and S. Xie, "Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3773–3787, Apr. 2021.

[15] Y. Qi, L. Tian, Y. Zhou, and J. Yuan, "Mobile edge computing-assisted admission control in vehicular networks: The convergence of communication and computation," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 37–44, Mar. 2019.

[16] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, 2020.

[17] L. Qian, Y. Wu, N. Yu, F. Jiang, H. Zhou, and T. Q. S. Quek, "Learning driven noma assisted vehicular edge computing via underlay spectrum sharing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 977–992, Jan. 2021.

[18] M. Dai, Z. Su, Q. Xu, and N. Zhang, "Vehicle assisted computing offloading for unmanned aerial vehicles in smart city," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1932–1944, Mar. 2021.

[19] X. Wang, Z. Ning, and L. Wang, "Offloading in Internet of Vehicles: A fog-enabled real-time traffic management system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4568–4578, Oct. 2018.

[20] X. Huang, D. Ye, R. Yu, and L. Shu, "Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 426–441, Mar. 2020.

[21] Y. Zhang, C.-Y. Wang, and H.-Y. Wei, "Parking reservation auction for parked vehicle assistance in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3126–3139, 2019.

[22] D. H. Stolfi, E. Alba, and X. Yao, "Predicting car park occupancy rates in smart cities," in *Proc. Int. Conf. Smart Cities*, pp. 107–117, 2017.

[23] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26652–26664, 2019.

[24] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, 2018.

[25] L. Qian, Y. Wu, F. Jiang, N. Yu, W. Lu, and B. Lin, "Noma assisted multi-task multi-access mobile edge computing via deep reinforcement learning for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5688–5698, Aug. 2021.

[26] Z. Xiong, J. Zhao, D. Niyato, R. Deng, and J. Zhang, "Reward optimization for content providers with mobile data subsidization: A hierarchical game approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2363–2377, Oct.-Dec. 2020.

[27] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proc. IEEE Proc. IRE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[28] Y. Zhou, F. Tang, Y. Kawamoto, and N. Kato, "Reinforcement learning-based radio resource control in 5G vehicular network," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 611–614, May 2020.

[29] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Int. Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[30] J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang, and Y. Zhang, "Smart and resilient ev charging in SDN-enhanced vehicular edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 217–228, Jan. 2020.

[31] R. S. Sutton *et al.*, "Policy gradient methods for reinforcement learning with function approximation," *NIPs*, vol. 99, pp. 1057–1063, 1999.

[32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[33] A. Graves *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

Xumin Huang received the Ph.D. degree from the Guangdong University of Technology, Guangzhou, China, in 2019. He is currently a Lecturer with the Guangdong University of Technology. His research interests include network performance analysis, simulation, and enhancement in wireless communications and networking.

Peichun Li (Student Member, IEEE) is currently working toward the M.S. degree with the School of Automation, Guangdong University of Technology, Guangzhou, China. His research interests include edge computing and deep learning, particularly in efficient algorithms for artificial intelligence applications.

Rong Yu (Member, IEEE) received the B.S. degree in communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2002 and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2007. After that, he was with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. In 2010, he joined the School of Automation, Guangdong University of Technology, Guangzhou, China, where he is currently a Professor. He is the author or coauthor of more than 120 international journals and conference papers. He is the co-inventor of more than 80 patents in China. His research interests include wireless networking and mobile computing such as edge computing, deep learning, blockchain, connected vehicles, smart grids, and Internet of Things. He was a Member of the Home Networking Standard Committee, China, where he led the standardization work of three standards.

Yuan Wu (Senior Member, IEEE) received the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2010. He is currently an Associate Professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China, and also with the Department of Computer and Information Science, University of Macau. Prior to that, he was a Full Professor with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. During 2016–2017, he was a Visiting Scholar with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include resource management for wireless networks, green communications and computing, mobile edge computing, and smart grids. He was the recipient of the Best Paper Award from the IEEE International Conference on Communications in 2016 and the Best Paper Award from IEEE Technical Committee on Green Communications and Computing in 2017. He was the Guest Editor of the *IEEE Communications Magazine*, IEEE NETWORK, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and *IET Communications*. He is currently on the Editorial Boards of the IEEE INTERNET OF THINGS JOURNAL, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and *China Communications*.

Kan Xie received the Ph.D. degree in control science and engineering from the Guangdong University of Technology, Guangzhou, China, in 2017. He joined the Institute of Intelligent Information Processing, Guangdong University of Technology, where he is currently an Associate Professor. His research interests include machine learning, nonnegative signal processing, blind signal processing, smart grids, and Internet of Things.

Shengli Xie (Fellow, IEEE) received the B.S. degree in mathematics from Jilin University, Changchun, China, in 1983, the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1995, and the Ph.D. degree in control theory and applications from South China University of Technology, Guangzhou, China, in 1997. He is currently a Full Professor and the Head of the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou, China. He has coauthored two books and more than 150 research papers in refereed journals and conference proceedings and was awarded Highly Cited Researcher in 2020. His research interests include blind signal processing, machine learning, and Internet of Things. He was awarded the Second Prize of National Natural Science Award of China in 2009. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS.