EIHDP: Edge-Intelligent Hierarchical Dynamic Pricing Based on Cloud-Edge-Client Collaboration for IoT Systems

Tian Wang, Yucheng Lu, Jianhuang Wang, Hong-Ning Dai, Xi Zheng, and Weijia Jia

Abstract—Nowadays, IoT systems can better satisfy the service requirements of users with effectively utilizing edge computing resources. Designing an appropriate pricing scheme is critical for users to obtain the optimal computing resources at a reasonable price and for service providers to maximize profits. This problem is complicated with incomplete information. The state-of-the-art solutions focus on the pricing game between a single service provider and users, which ignoring the competition among multiple edge service providers. To address this challenge, we design an edge-intelligent hierarchical dynamic pricing mechanism based on cloud-edge-client collaboration. We introduce an improved double-layer Stackelberg game model to describe the cloud-edge-client collaboration. Technically, we propose a novel pricing prediction algorithm based on double-label Radius K-nearest Neighbors, thereby reducing the number of invalid games to accelerate the game convergence. The experimental results show that our proposed mechanism effectively improves the quality of service for users and realizes the maximum benefit equilibrium for service providers, compared with the traditional pricing scheme. Our proposed mechanism is highly suitable for the IoT applications (e.g., intelligent agriculture or Internet of Vehicles), where there are multiple competing edge service providers for resource allocation.

Index Terms—Internet of Things, cloud-edge-client collaboration, dynamic pricing, stackelberg game

1 Introduction

The evolution of Internet of Things (IoT) greatly expands people's ability to access information, while cloud computing endows IoT with strong information-processing capability and improves the efficiency of the IoT systems [1]. In the IoT systems, the sensor nodes located in the underlying network are responsible for collecting various perceptive data and uploading them to cloud service platform, which analyzes and processes the data to provide users with personalized services [2]. In the IoT applications such as medical monitoring, smart city and smart agriculture etc., a large amount of monitoring data is constantly generated. However, due to the local computational resource limit, other resources such as cloud computing resources are needed to process these computing tasks. Generally, users initiate service requests to the IoT platform according to their own needs and pay on demand [3]. Meanwhile, the user's goal is to get satisfactory Quality of Service (i.e., QoS).

However, in the existing service pricing mechanism of the IoT systems, users oblige to increase their service budget if they want to achieve better QoS [4]. The IoT systems can provide resource-centralized services for users [5]. But when the amount of data services sharply increases, it is difficult to meet the heterogeneous, low-latency, intensive network access and service requirements [6]. The IoT system with cloud-edge-client collaboration is proposed to achieve such goal [7]: the edge computing node is closer to the data source and users, and the edge note can assist the cloud in processing part of the computing tasks, which effectively reduces network bandwidth and workload of the cloud data center [8]. Meanwhile, the edge layer can optimize the respond latency, thereby improving users' QoS satisfaction without increasing their service budget [9]. Whereas, the traditional service pricing method mainly focused on cloud services, which only considers the interaction between the cloud and users. Obviously, this approach cannot be directly applied to this novel IoT paradigm due to the impact of the edge computing paradigm being igored [10]. Specifically, there are multiple edge service providers competing with each other, whose interactions directly affect uers' QoS [11].

The core problem solved in this paper is to design a differential service pricing model based on the three-layer

- *Tian Wang is with the BNU-UIC Institute of Artificial Intelligence and Future Networks, Beijing Normal University (BNU Zhuhai), Zhuhai, Guangdong 519085, China, with the Guangdong Key Lab of AI and Multi-Modal Data Processing, BNU-HKBU United International College, Zhuhai, Guangdong 519087, China, and also with the College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian 361021, China. E-mail: cs_tianwang@163.com.*
- *Yucheng Lu and Jianhuang Wang are with the College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian 361021, China. E-mail: luyucheng@stu.hqu.edu.cn, w_jianhuang@163.com.*
- *Hong-Ning Dai is with the Faculty of Information Technology, Macau University of Science and Technology, Taipa, Macau 999078, China. E-mail: hndai@ieee.org.*
- *Xi Zheng is with the Department of Computing, Macquarie University, Macquarie Park, NSW 2109, Australia. E-mail: james.zheng@mq.edu.au.*
- *Weijia Jia is with the BNU-UIC Institute of Artificial Intelligence and Future Networks, Beijing Normal University (BNU Zhuhai), Zhuhai, Guangdong 519085, China, and also with the Guangdong Key Lab of AI and Multi-Modal Data Processing, BNU-HKBU United International College, Zhuhai, Guangdong 519087, China. E-mail: weijiaj@gmail.com.*

Manuscript received 31 Aug. 2020; revised 20 Dec. 2020; accepted 13 Feb. 2021. Date of publication 19 Feb. 2021; date of current version 9 July 2021. (Corresponding author: Tian Wang.) Recommended for acceptance by A. Rahimi, L. Benini, S. Benatti, and T. Jang. Digital Object Identifier no. 10.1109/TC.2021.3060484

0018-9340 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

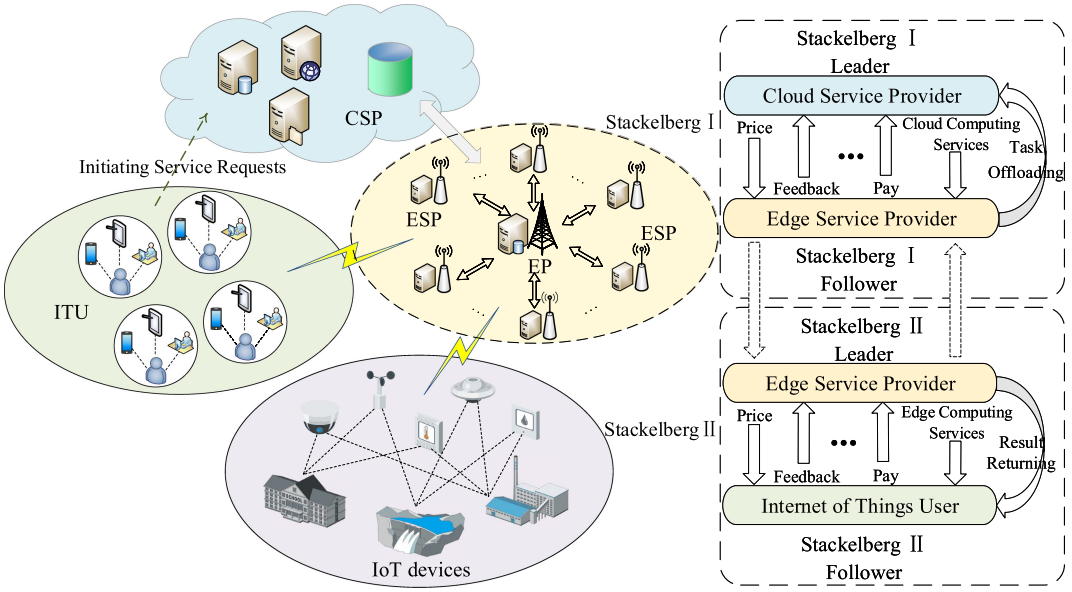


Fig. 1. Architecture of edge-intelligent hierarchical dynamic pricing mechanism.

cloud-edge-client architecture for different IoT users. In contrast to our work, existing studies mainly consider an identical service pricing model for the cloud-edge-client architecture. This identical service pricing model is so idealistic that it cannot be used for practical applications during the interactions between the cloud, edge, and IoT users. The goals of this paper are 1) to optimize the quality of service for different types of users in terms of price and time, and 2) to make the whole pricing process to converge quickly. Both these two goals also fulfill the requirements of the real application of the cloud-edge-client framework. The challenge to solve the problem is that all game participants are in an incomplete information state, which is also a more practical consideration rather than the idealistic assumption on the complete information state of all the participants. Although the CSP and ESPs are clear about the users' type in the game, they cannot know the users' detailed preference parameters for price and time. The price and time preference parameters represent the user's concern with the actual payment and system service time, and the sum of the two is 1. The price preference parameter is larger than the time preference parameter, indicating that the user cares more about the price and vice versa. For example, there is a price-oriented user. Then, the user's price preference parameter may be 0.6, 0.65, 0.7, or 0.85, while the corresponding time preference parameter should be 0.4, 0.35, 0.3, or 0.15. However, the CSP and ESPs cannot determine the exact value of the user's preference parameters. Therefore, it is hard to determine the optimal pricing scheme for users directly. Another challenge is the competitive relationship between multiple ESPs based on non-cooperative games. We also consider the differences between the computing power, storage capacity, and communication capability of individual ESPs; this consideration is also close to the real scenario. Meanwhile, the game is freely accessible; in other words, any ESP that meets the system requirements can freely choose to join or leave the system.

To address this challenge, we propose an edge-intelligent hierarchical dynamic pricing (EIHDP) mechanism to support cloud-edge-client collaboration as shown in Fig. 1, which comprehensively considers the interactions between the cloud service provider (CSP), edge service providers (ESPs), and IoT users (ITUs). We use the improved Stackelberg game model to maximize the service provider's profit and user's benefit. The proposed EIHDP model can be formulated as a double-layer Stackelberg dynamic pricing game, where Stackelberg I and Stackelberg II represent the dynamic game between CSPs/ESPs and that between ESPs/ITUs, respectively. Moreover, since ESPs adopt the non-cooperative game for pricing, the specific strategies of other competitors cannot be known in the pricing process. Therefore, the process of ESPs competition can be modeled as a static game. Although the pricing strategies of ESPs are private to other participants of the game, historical game records of other ESPs can be obtained to help to make decisions. We deploy machine-learning-based algorithms on the edge to assist ESPs in pricing scheme prediction to determine the most reasonable price, which improves the ESP's game success rate and system pricing efficiency.

To summarize, we make the following contributions in this paper:

- To the best of our knowledge, this is the first work of the service pricing problem in *cloud-edge-client collaboration* for IoT systems. We regard edge computing service as one of the main bases for pricing, thereby reducing the workload of cloud computing platform and optimizes QoS for users.
- We propose the EIHDP mechanism to achieve the maximum utility equilibrium among CSPs, ESPs and ITUs, accounting for the dynamic interactions between all players. We propose a *double-label Radius KNN (DL-RKNN) algorithm* to filter out pricing schemes with a high success rate and boost system pricing efficiency.
- We consider the scenario that *multiple ESPs* lie on edge nodes and each ESP can use machine learning algorithms to assist analysis and prediction for

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

pricing. In this case, a positive competitive process between multiple ESPs can be formed to effectively avoid the situation that a single adversarial maliciously disrupts the service pricing market.

The rest of the paper is organized as follows. We discuss related work in Section 2 and introduce the Stackelberg game and formulate the problem in Section 3. Then, we propose the EIHDP mechanism in Section 4. Section 5 evaluates the performance of our model. Finally, our conclusions are presented in Section 6.

2 Related Work

At present, the research on IoT pricing can be divided into three aspects: task offloading and resource scheduling strategy in the IoT systems, CSP-based pricing strategy, and ESP-based pricing strategy.

2.1 Task Offloading and Resource Scheduling Strategy in IoT Systems

In order to maximize the benefits of CSP and ESP, the cloud needs to schedule resources for computing tasks in the system. In general, the cost of processing perceived data in the cloud is smaller than the edge node, but the delay is considerable [12]. Therefore, the IoT systems need to offload computing tasks concentrated in the cloud server to the edge node and perform reasonable scheduling of computing resources [13]. Edge-based task offloading is a new model of providing mobile users with nearby network edge computing. However, without strong incentives, local edge servers may be reluctant to help offload computing [14]. In order to stimulate the CSP and the ESP to participate in the computational task unloading, Liu *et al.* developed the interaction between the CSP and the ESP [15]. They used the Stackelberg game model to maximize the benefits of CSP and ESP, and designed two computational task-unloading algorithms with low latency and low complexity [16]. In [17], the energy-saving resource allocation problem of the multi-user mobile edge computing system is investigated. Guo *et al.* [18] proposed a task model that uses three parameters (e.i., the size of the pre-calculation task, the workload, and the size of the calculation result) to specify each task, and established two computational offload models with negligible and non-negligible execution times of base stations [19]. In [20], considering the inclusion of renewable energy into mobile edge computing, an effective resource management algorithm based on reinforcement learning was proposed [21]. The online learning algorithm used both offline value iterative decomposition and online reinforcement learning. Compared with standard reinforcement learning algorithms such as Q-learning, the learning speed and runtime performance of this online learning algorithm are significantly improved.

Although these task offloading schemes and resource scheduling strategies in the IoT system have investigated the maximum benefits of CSP and ESP, they fail to consider the benefits of users and even ignore the critical role played by users in the pricing process.

2.2 CSP-Based Pricing Strategy

In the IoT systems, users use physical sensors and cloud infrastructure according to their needs and pay for the CSP. Therefore, it is necessary to develop a pricing plan for cloud services [22]. The profits generated from the user's payment are shared not only by the CSP but also by the sensor owners registered in the IoT systems. Zhu *et al.* reviewed the pricing strategy of the IoT systems and introduced five models of IoT pricing [23]. Specifically, in order to ensure profit maximization, the CSP will consider the following factors in pricing: IoT user's lease period; the IoT systems required working time; users using IoT resources; the amount of sensory data obtained by the IoT user; the path from which the sensor network transmits the sensory data to the IoT user [24]. Chatterjee *et al.* proposed a new pricing model consisting of two parts: hardware pricing (pH) and infrastructure pricing (pI). pH solves the pricing problem of physical sensor nodes, which depends on changes in user demand and utility. It maximizes the profit per sensor owner while taking into account the user's utility. pI focuses on pricing issues brought about by resource virtualization, which considers the cost of IoT resources, including the cost of maintaining virtualization within the IoT systems. pI maximizes the perceived CSP profit by considering user satisfaction [25].

However, the current CSP-based pricing strategies, which mainly focus on the interaction between the CSP and users, neither fully consider the impact of incomplete information nor consider the economic benefits of ESPs, resulting in a waste of edge resources.

2.3 ESP-Based Pricing Strategy

Mobile edge computing serves sensors near mobile devices, and transmitting data from the IoT systems to edge servers for pre-processing has become a key technology [26]. Mocnej *et al.* [27] proposed a quality-enabled decentralized IoT architecture for monitoring the IoT environment at runtime, expressing the overall quality of the system, and helping to utilize the available resources efficiently [28]. The edge devices are used to define a feature set that describes the best expectations of decentralized IoT platforms. In [29], Lin *et al.* proposed an IoT-based irrigation and fertilization management system in which both long-term management and short-term management are considered. The system can utilize the edge nodes to support the decision-making of managers and offer more economical and environmental benefits. Kumar *et al.* [30] proposed a dynamic management model to improve health monitoring by indulging IoT assisted wearable sensor platform. Since edge computing is a distributed intermediate layer between the edge network and the cloud environment, latency can be reduced to a remarkable level, and reliable communication can be achieved with the help of edge computing. These researches aim to use edge devices to assist in runtime management for IoT systems but ignore the advantages of cloud-edge collaborative management. In contrast, our method can take the advantage of cloud-edge-client collaboration to manage the runtime status for IoT systems.

The function of the ESP is to dynamically allocate resources to tasks at the edge based on the request. However, this is a challenging task, because the ESP needs to make real-time decisions without prior knowledge of future conditions, requested task satisfaction, and resource utilization. Zhang *et al.* [31] studied data transmission from mobile

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

devices to mobile edge servers and proposed a pricing scheme based on alliance games. The joint algorithm is used to sort the data of the mobile device, and the data processing relationship between the mobile device and the mobile edge server is described. In [32], a single cloud multi-user mobile edge computing offloading system with orthogonal frequency division multiple access is introduced, and an optimal pricing scheme is proposed for each mobile user's resource requirements. Kim *et al.* [32] established a single-leader multi-user Stackelberg game model and used Stackelberg equilibrium to optimize the utility of mobile users and edge clouds. Jie *et al.* proposed an online scheduling method based on repeated Stackelberg games, mapping various tasks to a given related resource [33]. This problem is first modeled as a long-term and short-term repeated Stackelberg game. Especially in each round of games, as a short-term leader, the user first decides to process the unit price of the task within the relevant budget to maximize the overall satisfaction of the current task. The ESP then maximizes the long-term profit that is obtained from the user based on the price offered by the different users in different rounds. The strategy is to match the resources to the tasks and assign those tasks to different edge centers that have different types of resources [34].

However, these ESP-based pricing strategies fail to take into account the competition among multiple ESPs. The above approaches fail to solve the service pricing problem in cloud-edge-client collaboration for IoT systems.

3 Stackelberg Game and Problem Formulation

3.1 Stackelberg Game

Game theory is used to study the predictive and actual behavior of different individuals in a game, and how to optimize their respective strategies [35]. A complete game consists of players, game information, game strategies, game order, and game rewards. Among them, game information refers to the information that players have; it is helpful to their strategies. The game strategies represent a set of all behaviors or strategies that participants can choose. The game order represents the order in which players take action, and game rewards refer to the incomes that the decision-maker gains in the game. Therefore, game theory can be subdivided based on the differences of the above five parts. For example, it can be subdivided into complete information games and incomplete information games according to game information; then it can be further divided into complete information cooperative games and incomplete information cooperative games [36].

Among these branches, static game and dynamic game formed according to game order are the two most common types of games. Static game refers to the participants in the game taking actions simultaneously; or there is a sequence of actions, but the latter party does not know the actions of the first party, which leaves all participants with no control over the outcome [37]. In practice, players tend not to act simultaneously, so dynamic games with different game sequences are more common. In dynamic games, players are divided into leaders and followers according to their order. The leader's actions will be observed by followers who can adjust their action plans based on the observation results. In different game scenarios, the dominant players of dynamic games are also different. For example, leaders have advantages in production competition, while followers have more choices in price competition [38].

In dynamic game, players determine their strategies based on known strategies, so that the strategies are no longer independent of each other while having strong interaction. The controllability of game results is also stronger than that of static game [39]. It can be considered that in such a game, the final strategies given by each participant maximize their benefits under the current circumstance, then the set of these strategies constitutes a Nash equilibrium. There are many different models in dynamic game to find the Nash equilibrium in different ways. In this paper, the Stackelberg game is a model with adding a feedback mechanism. The feedback mechanism means that the follower's strategy in this round will be fed back to the leader when the game is played multiple times, thus influencing the next round strategy. With the introduction of this mechanism, the interaction between leaders and followers is no longer limited to the observation of strategies, and the mutual constraints are strengthened at the same time. Hence, the Stackelberg game performs well in a specific dynamic game.

The Stackelberg game is a process, in which participants compete for limited resources. Further understanding, the Stackelberg game is a strategy for allocating limited resources through the level of competitive strength. Leaders are not only the first movers, but also represent their stronger competitiveness. For example, large firms have more capital than small companies, allowing them to compete at lower prices without losing money. In this article, such competitiveness can be reflected in the fact that the cloud servers have more computing resources than the edge servers. On the other hand, higher competitiveness means that the participant often needs to acquire more resources to maintain this state. For instance, large firms may also break their funds if their prices are low for a long time in the game. This paper will design the experimental model based on this understanding, so that all participants are in the normal game state.

3.2 Problem Description

In this section, we describe the pricing problem of our proposed EIHDP framework. In each round of game, the participants in game are CSPs, ESPs and ITUs from top to bottom. In the game process of vertical direction, multiple ESPs are regarded as a whole. The decisions of each layer are in order, that is, the CSP makes the decision first, then the ESP decides the strategy and finally the ITU makes the choice, which constitutes a double-layer Stackelberg dynamic game. In the game process of historical direction, multiple ESPs are independent of each other, and there is no order in the decision-making. Each ESP can refer to the historical game data of other competitors at the same time for predictive analysis, thus the game among ESPs belongs to static game. Table 1 defines the notations used in this paper. Therefore, a double-layer Stackelberg game model is established as

$$S = < L; \Omega; I >, \tag{1}$$

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Definition of Notations

Parameter	Definition
Q_{SUM}	Number of computing tasks
T_U	Expected service time of the user
T_P	Predicted service time given by the edge
Q_{CSP}	Number of tasks processed at the cloud
Q_{ESP}	Number of tasks processed at the edge
V_{CSP}	Speed of tasks processed at the cloud
V_{ESP}	Speed of tasks processed at the edge
V_B	Bandwidth speed for tasks offloading
P^h_{CSP}	Unit price of task handling at the cloud
P^b_{CSP}	Unit price of bandwidth using
P^h_{ESP}	Unit price of task handling at the edge
P_{ESP}	Service price for the user
P_{ITU}	Actual payment of the user
R_{ITU}	Dissatisfaction degree with service time
Ω_{CSP}, Ω_{ESP}, Ω_{ITU}	Strategy space of the cloud, edge, user
I_{CSP}, I_{ESP}, I_{ITU}	Utility function of the cloud, edge, user

which contains three basic elements of Stackelberg game, namely, game player L, game strategy Ω and utility of game participants I. As shown in Table 2, game participant L includes the CSP, N ESPs, and the ITU. In the game model, the CSP's strategy space Ω_{CSP} is (P^h_{CSP}, P^b_{CSP}). Furthermore, the CSP adjusts P^b_{CSP} and P^h_{CSP} in each round of game according to the voting result by ESPs. In the game intermediate layer, the ESP's strategy space Ω_{ESP} includes the CSP-oriented strategies (Q_{CSP}, Q_{ESP}, V_B) and ITU-oriented strategies $(P^h_{ESP}, P_{ESP}, T_P)$. In fact, Ω_{ESP} can be described as the strategies $(Q_{CSP}, Q_{ESP}, V_B, P^h_{ESP})$ to determine other strategies (P_{ESP}, T_P). The ITU's strategy space Ω_{ITU} is (P_{ITU}, R_{ITU}), implying that the ITU needs to consider the utility I_{ITU} from both cost and time.

Definition 1. (Price Acceptance Rate) *Given Ω_{CSP}, the rate that ESPs accept the CSP's price strategy is defined as*

$$\Theta(\Omega_{CSP}) = \frac{NUM_{agree}(l)}{NUM_{agree}(l) + (1-\delta) * NUM_{reject}(l)} \tag{2}$$

$$l = 1, 2, \ldots, \left\lceil \frac{NUM_{reject}(l)}{NUM_{agree}(l)} \right\rceil \tag{3}$$

where l is the number of pricing rounds for the CSP; $NUM_{agree}(l)$ and $NUM_{reject}(l)$ are number of votes for ESPs agreement and rejection to the price, respectively; and $\delta(0 \le \delta \le 1)$ is the filtering percentage of ESPs for refusing to the price.

The CSP determines the value of parameter δ according to the initial number and computing power of ESPs participating in the vote. The purpose of the voting game between the CSP and ESP on the CSP's pricing strategy is to choose ESPs that meet basic requirements such as computing power and cost as much as possible and reduce the loss of system resources for ESPs with low success rate. When the initial number of ESP is too large, or the computing power gap is small, the CSP chooses to set $\delta \ge 0.5$ for accelerating the overall game convergence. When the initial number of ESP is small, or the computing power gap is large, the CSP can choose to set $\delta < 0.5$. In order to make the percentage of ESPs that accept prices greater than or equal to 50 percent, the system filters out δ percentage of ESPs that reject the price in the following order. a) The ESP can choose to voluntarily withdraw from the current game. b) The CSP removes ESPs with significantly weaker computing power. c) The system randomly deletes ESPs that reject prices.

TABLE 2
The Basic Elements of EIHDP Game

L (game player)	Ω (strategy space)	I (utility function)
CSP	$\Omega_{CSP}(P^h_{CSP}, P^b_{CSP})$	$I_{CSP}(\Omega_{CSP})$
ESP	$\Omega_{ESP}(Q_{SUM}, V_B, P^h_{ESP})$	$I_{ESP}(\Omega_{ESP})$
ITU	$\Omega_{ITU}(P_{ITU}, R_{ITU})$	$I_{ITU}(\Omega_{ITU})$

We take the user demand as a game precondition $D = \{Q_{SUM}, T_U\}$, and $\Omega_{CSP} \sim D$ means that Ω_{CSP} is constrained by D. The constraint relation between the CSP, ESP and ITU can be expressed as $\Omega_{ESP} \sim \Omega_{CSP}$ and $\Omega_{ITU} \sim (\Omega_{ESP}, \Omega_{CSP})$. First, the upper CSP analyzes the historical data based on the ITU's requests to initialize pricing strategy space $\Omega_{CSP}(P^h_{CSP}, P^b_{CSP})$ to give the two prices for participating in the current round of game, where P^h_{CSP} and P^b_{CSP} are strategy variables. Then $NUM_{initial}$ ESPs vote on the initial strategy for feedback. Since ESPs cannot know the voting information between each other, this process belongs to the static game of ESPs in the first stage. By Definition 1, the remaining $N \leftarrow NUM_{agree}(l) + (1-\delta) * NUM_{reject}(l)$ ESPs continue the game when $\Theta(\Omega_{CSP}) \ge 0.5$. Otherwise, the CSP needs to readjust the price and $(1-\delta) * NUM_{reject}(l)$ ESPs continue the $l+1$th vote until l reaches the critical value. The ESP uses the DL-RKNN algorithm to make scheme prediction based on historical game data to get a group of pricing schemes $\langle Q_{CSP}, Q_{ESP}, V_B, P^h_{ESP}\rangle \rightarrow \langle P_{ESP}, T_P\rangle$ arranged from large to small on the basic of the utility value I_{ESP}. In each game, the ESP chooses a pricing scheme in the order of utility value, thus N schemes are sent to the ITU after being summarized by the edge platform (EP). Assume that ESP t is the game winner with the strategy space $\Omega_{ESP_t}(Q_{CSP_t}, Q_{ESP_t}, V_{B_t}, P^h_{ESP_t}), t \in [1, N]$. We define a double-layer Stackelberg equilibrium in Definition 2.

Definition 2. (Double-Layer Stackelberg Equilibrium) *Given Ω^*_{ITU}, Ω^*_{ESP} and Ω^*_{CSP}, the double-layer stackelberg equilibrium for EIHDP is defined as*

$$I_{ITU}\left(\Omega^*_{ITU}, \Omega^*_{ESP}, \Omega^*_{CSP}\right) \ge I_{ITU}\left(\Omega_{ITU}, \Omega^*_{ESP}, \Omega^*_{CSP}\right), \tag{4}$$

$$I_{ESP}\left(\Omega^*_{ITU}, \Omega^*_{ESP}, \Omega^*_{CSP}\right) \ge I_{ESP}\left(\Omega^*_{ITU}, \Omega_{ESP}, \Omega^*_{CSP}\right), \tag{5}$$

$$I_{CSP}\left(\Omega^*_{ITU}, \Omega^*_{ESP}, \Omega^*_{CSP}\right) \ge I_{CSP}\left(\Omega^*_{ITU}, \Omega^*_{ESP}, \Omega_{CSP}\right), \tag{6}$$

where Ω^*_{ITU} and Ω^*_{CSP} are the user's equilibrium strategy and cloud's equilibrium strategy, respectively. Ω^*_{ESP} represents equilibrium strategies for all ESPs, and $\Omega_{ESP} = (\Omega_{ESP_t}, \Omega^*_{ESP_{-t}})$. Ω_{ESP_t} and $\Omega^*_{ESP_{-t}}$ indicate non-winning strategy (i.e., non-equilibrium strategy) of the winner ESP t and the equilibrium strategy of other ESPs except ESP t, respectively. The total number of ESPs is denoted by N.

The ITU's strategy space includes the tuple of P_{ITU} and R_{ITU}. The preference coefficient for cost and time is

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

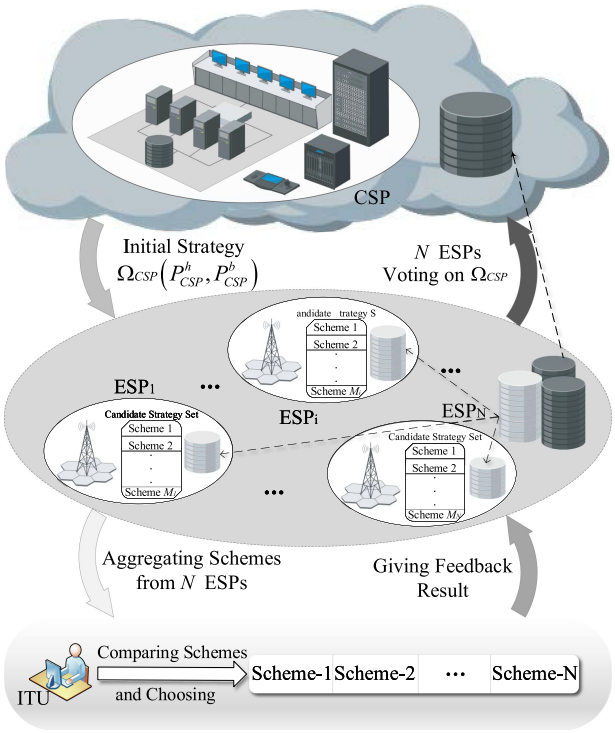


Fig. 2. The double-layer stackelberg games.

Algorithm 1. The EIHDP Model Algorithm

Input: Game precondition D; The number of ESPs N.
Output: Equilibrium utilities $(I_{ITU}, I_{ESP}, I_{CSP})$.

Define V_{CSP}, V_{ESP}, $E(D)$, δ, τ, λ and μ
repeat
Vote on the strategy space Ω_{CSP}
until $\Theta(\Omega_{CSP}) \geq 1/2$ or $l \geq \max\{l\}$
for each ESP **do**
  for $i = 1$ to M **do**
    if $T^d_{CSP} < T_{ESP} \leq (1+\tau)T_U$ or
      $T_{ESP} < T^d_{CSP} < (1+\tau)T_U$ **then**
      Update the set of candidate pricing schemes
    end if
  end for
  The DL-RKNN classifier for service pricing schemes
end for
Calculate the lowest pricing P_m with R_M and the smallest dissatisfaction R_m with P_M from N pricing schemes
for $k = 2$ to N **do**
  if $E(D) \leq I_{ITU}(k-1) \leq I_{ITU}(k)$ **then**
    Update the selected scheme according to (15)-(18)
  end if
end for
Calculate I_{ESP} and I_{CSP} according to (10)-(14)
Return $(I_{ITU}, I_{ESP}, I_{CSP})$

only valid locally, implying that the utility function of ITU is unknown to the CSP and ESPs. The client has an evaluation function $E(D)$ as the criterion for accepting the pricing scheme, which is also only valid locally. If this thing $\exists I_{ITU} \geq E(D)$ is satisfied, the largest I_{ITU} is selected, and the winner is generated in ESPs. The ITU chooses to determine the scheme and makes the payment. The EP records the game results and pricing schemes of all ESPs, thus the whole game ends. The winning ESP gets the data from the EP and returns the result to the user after processing the task based on the winning scheme. If this thing $\exists I_{ITU} \geq E(D)$ cannot be satisfied, this game is invalid. Each ESP needs to select a scheme in order again and the ITU makes judgment and selection. The ESPs need to give feedback to the CSP until every ESP selects the last pricing scheme. The CSP should readjust the pricing strategy space $\Omega_{CSP}\left(P^h_{CSP}, P^b_{CSP}\right)$ at this time. Then the intermediate layer and the user layer repeat the above process until the end of the game. By Definition 2, the expected result is expressed in the form of Stackelberg equilibrium for the above game model with hierarchical decision structure. Each participant cannot independently adjust the strategy to obtain more benefits when strategies of all game participants are equilibrium strategies.

4 Edge-Intelligent Hierarchical Dynamic Pricing Mechanism

In this section, we introduce the detailed process of the proposed EIHDP mechanism as illustrated in Fig. 2, which includes two Stackelberg games. Stackelberg I is the dynamic game between CSPs and ESPs, and Stackelberg II is the dynamic game between ESPs and ITUs.

4.1 CSP-ESP Dynamic Game

In the CSP-ESP dynamic game, the CSP is the game leader and the ESP is the game follower. The CSP aims to maximize the utility I_{CSP}. The ESP makes its strategies according to Ω_{CSP} to reduce the cost of using bandwidth. The ESP can effectively split Q_{SUM} into $Q_{CSP} = \{Q_{CSP_1}, Q_{CSP_2}, \ldots, Q_{CSP_M}\}$ and $Q_{ESP} = \{Q_{ESP_1}, Q_{ESP_2}, \ldots, Q_{ESP_M}\}$, which can be formulated as

$$\begin{aligned} &Q_{SUM} = Q_{CSP_i} + Q_{ESP_i}, i = 1, 2, \cdots, M, \\ &s.t. 0 < Q_{CSP} < Q_{SUM}, 0 < Q_{ESP} < Q_{SUM}, \end{aligned} \tag{7}$$

where M represents the amount of combination modes for Q_{SUM} in the initial state. Q_{CSP} is the number of tasks that the ESP chooses to offloading to the cloud for processing, and Q_{ESP} is the number of tasks that the ESP selects to process locally. In the process of dynamic game, time is also one of the important factors that need to be considered. In the proposed EIHDP model, the time involved comprises task offloading time and task processing time. The latter includes task processing time in ESP and CSP. The task offloading time T^t_{CSP} and task processing time T^d_{CSP} for Q_{CSP} can be obtained when the tasks Q_{SUM} is grouped by the ESP. The ESP offloads some tasks to the cloud while processing computing tasks, but the CSP must wait for task offloading to complete before starting task processing. Therefore, T^t_{CSP} and T^d_{CSP} are continuous when analyzing the time, and the total time for Q_{CSP} is

$$T_{CSP} = T^t_{CSP} + T^d_{CSP} = \frac{Q_{CSP}}{V_B} + \frac{Q_{CSP}}{V_{CSP}}, \tag{8}$$

Then the ESP's time to process Q_{ESP} is $T_{ESP} = \frac{Q_{ESP}}{V_{ESP}}$. T_{ESP} and T_{CSP} overlap in time due to that the ESP can execute task offloading and task processing simultaneously, so the predicted service time for users should be the maximum

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

time of both, which can be formulated as

$$\begin{aligned} T_P &= \max\{T_{CSP}, T_{ESP}\}, \\ s.t. 0 &< T_P \leq (1+\tau)T_U, \end{aligned} \tag{9}$$

where $\tau \geq 0$ indicates the proportion coefficient exceeding the expected time, which means that the predicted service time cannot be greater than the upper limit $(1+\tau)T_U$. In the system, τ represents the user's tolerance for actual service time. Therefore, the value of τ changes for different users. According to user preferences, users can be divided into three types: time-oriented, price-oriented, and neutral users. Among them, τ takes zero as the default value for time-oriented and neutral users, and τ takes a real value greater than zero for price-oriented users. The CSP's goal is to maximize the utility function:

$$\begin{aligned} \max I_{CSP} &= \max\{Q_{CSP}P^h_{CSP} + V_B P^b_{CSP} - W_{CSP}\} \\ &= \max\{Q_{CSP}P^h_{CSP} + V_B P^b_{CSP} - G_{CSP} - G_B\}, \end{aligned} \tag{10}$$

$W_{CSP} = G_{CSP} + G_B$ represents the CSP's cost function, where G_{CSP} is the cost of equipment to process computing tasks, and G_B is the cost to maintain bandwidth. G_{CSP} is a quadratic function about the amount of processing tasks (the tasks which the ESP offloads) and is strictly convex, and the formula is expressed as follows:

$$G_{CSP} = a(Q_{CSP})^2 + b(Q_{CSP}) + c, \tag{11}$$

where a, b and c are the coefficients of CSP's equipment cost, and G_B is formulated as

$$G_B = \frac{V_B}{V_{MAX}}(w_{cost,H} + w_{cost,L}), \tag{12}$$

where V_{MAX} represents the maximum available bandwidth, $w_{cost,H}$ is the hardware cost of deploying bandwidth, and $w_{cost,L}$ is the labor cost of maintaining bandwidth. The ESP's cost function is also composed of two parts: one is the bandwidth expense $W'_{ESP} = V_B P^b_{CSP}$ paid to CSP, and the other is the equipment cost $W''_{ESP} = G_{ESP}$ to process computing tasks. The total cost for ESP is $W_{ESP} = W'_{ESP} + W''_{ESP} = V_B P^b_{CSP} + G_{ESP}$. G_{ESP} is also a quadratic function about the number of processing tasks (the tasks which the ESP processes locally) and is strictly convex, which can be formulated as

$$G_{ESP} = a'(Q_{ESP})^2 + b'(Q_{ESP}) + c', \tag{13}$$

where a', b' and c' are the coefficients of ESP's equipment cost. In the dynamic game stage with the CSP, the ESP's purpose is to reduce the bandwidth cost $W'_{ESP} = V_B P^b_{CSP}$ as much as possible under certain constraints. In the initial state, there are M grouping methods for tasks Q_{SUM}. It is necessary to make preliminary screening to reduce computing complexity in the course of the game. The predicted service time T_P must be less than or equal to $(1+\tau)T_U$ according to the request.

Theorem 1. *The ESP achieves the optimal bandwidth V_B and optimal bandwidth cost W'_{ESP} for pricing schemes.*

Proof. We fix the unit price P^b_{CSP}. Assuming that $T_{ESP} > T^d_{CSP}$. The time $T_{CSP} = T_{ESP}$ aims to make full use of the maximum time, because the ESP can offload the tasks to the CSP while processing computing tasks. Thus we get $\min\{W'_{ESP}\} \sim \min\{V_B\} \sim \max\{T^t_{CSP}\}$ when $T^t_{CSP} = T_{CSP} - T^d_{CSP}$. Assuming that $T_{ESP} \leq T^d_{CSP}$. Due to $(1+\tau)T_U \geq T_{ESP}$, we get $\min\{W'_{ESP}\} \sim \min\{V_B\} \sim \max\{T^t_{CSP}\}$ when $T^t_{CSP} = (1+\tau)T_U - T^d_{CSP}$. □

Algorithm 1 describes the detailed process of the proposed EIHDP model. The predicted service time for the ITU is the maximum value between T_{CSP} and T_{ESP}. When it comes to $T_{ESP} \geq T_{CSP}$, the ESP deletes the grouping scheme which makes $T_{ESP} > (1+\tau)T_U$. When it comes to $T_{CSP} > T_{ESP}$, the ESP can similarly delete the grouping scheme which makes $T_{ESP} > (1+\tau)T_U$, if $T_{ESP} > (1+\tau)T_U$ which also means $T_{CSP} > T_{ESP} > (1+\tau)T_U$. There are $M' \leq M$ grouping schemes in the ESP for tasks Q_{SUM} after filtering. In each grouping method, T_{ESP} is determined by Q_{ESP}, and T^d_{CSP} is determined by Q_{CSP}. The size of the bandwidth using for offloading is optional. The smaller T^t_{CSP} makes the smaller T_{CSP} when the ESP uses more bandwidth. On the contrary, The larger T^t_{CSP} makes the larger T_{CSP} when the ESP uses less bandwidth.

4.2 ESP-ITU Dynamic Game

In the ESP-ITU dynamic game, the ESP is the game leader and the ITU is the game follower. In this game process, the strategies that the ESP chooses include the ITU-oriented service pricing and predicted service time, while the ITU needs to take time dissatisfaction and actual payment into consideration when making decisions. The ESP's service pricing involves the cost to process tasks locally and the cost for the CSP to process tasks, i.e., $P_{ESP} = Q_{ESP}P^h_{ESP} + Q_{CSP}P^h_{CSP}$, where P_{ESP} represents the service pricing finally determined by ESPs for the ITU to choose. The ESP's goal is to maximize the utility function:

$$\begin{aligned} \max I_{ESP} &= \max\{P_{ESP} - W_{ESP}\} \\ &= \max\{P_{ESP} - V_B P^b_{CSP} - G_{ESP}\}. \end{aligned} \tag{14}$$

The EP sends a summary of all pricing schemes to the ITU and stores it in the local database. The ITU needs to consider price and time when choosing a pricing scheme.

Definition 3. (Time Dissatisfaction Degree). *Given T_U and T_P, the ITU's dissatisfaction degree for the service time is defined as*

$$R_{ITU} = \beta\left[\alpha^{\sigma\left(1-\frac{T_U}{T_P}\right)} - 1\right] \tag{15}$$

where α, β and σ are the relevant parameters of the time dissatisfaction function, and $\alpha > 1$, $\beta > 0$, $\sigma > 1$. β is the preset value, σ is related to the time-scale elasticity of service demand, and α is related to the amplitude of R_{ITU}. T_U is the determined value and T_P is given by different ESPs.

By Definition 3, the smaller T_P makes the smaller R_{ITU}, and the larger T_P makes the larger R_{ITU}. The ITU model is a double-objective optimization function. Since the dimension of the two targets is different, simple weighted linear

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

processing cannot be carried out directly. Here, the fuzzy solution method is adopted to deal with the double-objective optimization problem, and the steps are as follows:

1) The optimization calculation is carried out with the goal of the lowest P_{ITU} to obtain the lowest service pricing P_m in all pricing schemes, and the dissatisfaction function value is recorded as R_M at this time.
2) The optimization calculation is carried out with the goal of the smallest R_{ITU} to obtain the smallest dissatisfaction value R_m in all pricing schemes, and the actual payment cost is recorded as P_M at this time.
3) The optimal attributes of the two objective function values are fuzzified, and the mapping from the single objective function value to the membership degree is established. The membership degree is determined according to the linear rule, and the distribution is semi-trapezoidal as

$$\varphi(P_{ITU}) = \begin{cases} 1 & P_{ITU} \le P_m \\ \frac{P_M - P_{ITU}}{P_M - P_m} & P_m < P_{ITU} < P_M \\ 0 & P_{ITU} \ge P_M \end{cases} \tag{16}$$

$$\varphi(R_{ITU}) = \begin{cases} 1 & R_{ITU} \le R_m \\ \frac{R_M - R_{ITU}}{R_M - R_m} & R_m < R_{ITU} < R_M \\ 0 & R_{ITU} \ge R_M \end{cases}. \tag{17}$$

4) The fuzzy membership of the two target functions is linearly weighted, then the utility function of ITU of the fuzzy double targets is obtained, which can be formulated as

$$\max I_{ITU} = \max\{\lambda\varphi(P_{ITU}) + \mu\varphi(R_{ITU})\}, \tag{18}$$

where λ and μ are the weight coefficients of P_{ITU} and R_{ITU} respectively, namely, the preference coefficients of ITU for cost and time. These two coefficients are only valid locally, which indicates that the ITU's utility function is transparent to the CSP and ESP. The termination condition of the game between the ESP and ITU is that the ITU uses the evaluation function to select an optimal pricing scheme from a collection of pricing schemes. At this time, a winner is generated from multiple ESPs. When the candidate set of pricing schemes for an ESP is filtered to the last one, the ESP continues to send this scheme to the ITU for selection. The ESP withdraws from the current game if it fails the competition. When all ESPs' candidate sets are filtered to the last one, the ESPs send the pricing schemes to the ITU while giving feedback to the CSP, requiring the CSP to readjust the pricing strategy and vote again.

However, the utility functions of the ESP and ITU are not the same, so the edge utility value and user utility value corresponding to the same pricing scheme are also not equal. In order to improve the game success rate of ESPs, reduce unnecessary pricing schemes and the number of invalid games, this paper adopts an improved DL-RKNN algorithm based on KNN at the edge. The DL-RKNN algorithm improved from KNN is a prediction algorithm for double screening of training samples, where R refers to the selection of the training set range, and K refers to the filter of the training set. For the user sample label $\langle Q_{SUM}, T_U\rangle$, the predictor needs to filter out enough invalid samples and ensure the number of remaining valid samples. Therefore, R is the radius of the effective sample area centered on the user demand label. Theoretically, the shape of the effective sample area can be arbitrary, but the linear relationship between two labels of random sample points is difficult to be precisely determined. At the same time, a random selection of valid areas may lead to ignoring the importance of a certain label, and the number of effective samples selected may not be optimal. Accordingly, the sample area with R as the radius can not only effectively avoid the omission of key labels, but also make the selected samples more accurate under the premise of ensuring fairness.

Algorithm 2. The DL-RKNN Classifier Algorithm

Input: Candidate scheme $\langle P_{ESP}, T_P\rangle$.
Output: The predicted label for candidate scheme.
Define R and K for the effective sample area
Set the main label $\langle Q_{SUM}, T_U\rangle$
Calculate the euclidean distance Z between the historical data items and $\langle Q_{SUM}, T_U\rangle$
if $Z \le R$ **then**
  Update the training sample set
end if
Set the secondary label $\langle P_{ESP}, T_P\rangle$
Calculate and sort Z between $\langle P_{ESP}, T_P\rangle$ and the training sample by distance increasing
Select the first K items for the judging set $\{J(K)\}$
if There is successful scheme in $\{J(K)\}$ **then**
  Return the forecasted label with possible deal
else
  Return the forecasted label with impossible deal
end if

Algorithm 2 is designed to allow ESPs to make predictions based on historical transaction data and filter strategies according to the forecast results, that is, to categorize strategy set to determine which strategies are likely to be ultimately selected by ITUs. Each data in the historical transaction data set consists of seven dimensions $\langle Q_{SUM}, T_U, P_{ITU}, T_P, V_B, Q_{ESP}, Q_{CSP}\rangle$. Because these dimensions do not include user utility values and utility functions, the ESP cannot predict user utility values by linear regression. The ESP needs to classify each strategy in the current strategy set into two labels of "possible deal" and "impossible deal", and select the data from the label of possible deal for the subsequent game. The distribution of historical data collected in the center of the edge is wide and discrete. Therefore, in order to reduce the training sample size, DL-RKNN first selects the training set range. First, the current user demand is taken as the main label $\langle Q_{SUM}, T_U\rangle$, and the historical game records at the EP are filtered. The history records which meet the range requirements of $\langle Q_{SUM} \pm R, T_U \pm R\rangle$ are added to the candidate training set. After that, all records with euclidean distance greater than those in the candidate set are calculated and deleted, so as to select the training set for the pricing scheme filtering. Next, the ESP's pricing scheme is used as the secondary

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

TABLE 3
Experiment Environment

Parameters	CSP	ESP
Operation System	Ubuntu 16.01	Ubuntu 16.01
CPU	Intel xeon E-2186G	Intel Core i7-8700
GPU	Leadtek Tesla M40	NVDIA 1080
Memory	48GB	16GB
Memory (GPU)	24GB	8GB
Storage (SSD via NVMe)	500GB	500GB
Programming Language	Python	Python
Compiler	Python 3.7	Python 3.7

label $\langle P_{ESP}, T_P \rangle$, and the euclidean distances between the records in the training set and the current scheme to be predicted are calculated. Since each record includes all strategies for the successful round of the game, and only one of these strategies is finally selected by ITUs, the "successful" game scheme is far less than the "unsuccessful" game scheme. Therefore, unlike traditional algorithms which take high-frequency category as the prediction result, "possible deal" will be taken as the prediction result of the current pricing scheme as long as the "successful" scheme is contained in the K most adjacent records.

5 Performance Evaluation

In this section, we validate the effectiveness of our proposed EIHDP algorithm through the experimental results of the number of games and prediction time experiments.

5.1 Experimental Setup

The experimental environment parameters are shown in Table 3. The experimental data came from the meteorological and precipitation information collected by sensors in the collection station of the TLINK industrial IoT platform and OneNET IoT platform. The TLINK and OneNET platforms can realize remote monitoring and management for IoT devices. The data dimension covers the target site, the precipitation in a certain period of time, the latitude and longitude of the target site, and the weather conditions taken at a certain time. Compared with the traditional pricing scheme without prediction (TPSP) [33], the significance of introducing the prediction algorithm in EIHDP is to judge whether the current strategy can meet the ITU requirements based on historical data, thereby reducing the number of games and improving system performance. We introduce a Logic-KNN (L-KNN) algorithm for comparison, whose basic idea is to reduce meaningless sample points based on the distribution of Q_{SUM} and T_U with the certain linear correlation. To evaluate the performance of EIHDP, we compare EIHDP with TPSP in KNN, L-KNN, and DL-RKNN under different parameters.

Both L-KNN and DL-RKNN are improved versions of KNN. The main differences between the three algorithms lie in determination methods and the selection ranges for sample points. First, in terms of the determination method, KNN selects the top K sample points with the highest correlation and then count the label value with the highest proportion as the prediction result. The label values of the sample in historical game records are only "success" and "failure," and the number of "success" values is far smaller

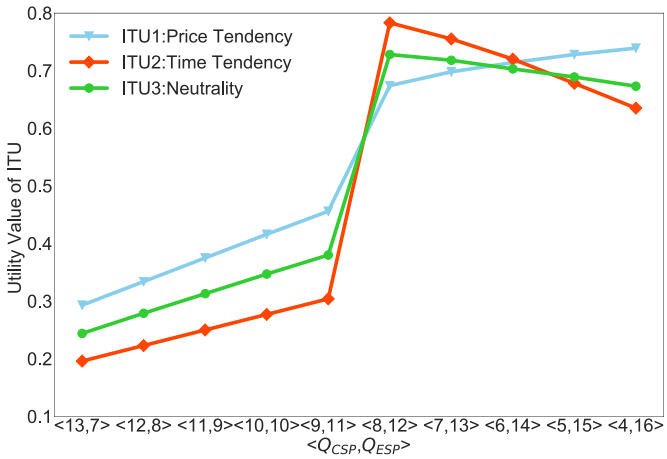


Fig. 3. The variation of ITU utility value.

than that of "failure" values. All prediction results will be "impossible" if we continue to take high-frequency categories as the prediction result due to the unbalanced sample proportion.

Therefore, the determination methods of L-KNN and DL-RKNN are both changed to the case that the prediction result is "possible" if there is a successful label in the current K sample points; otherwise, it is "impossible." The second is the selection range of sample points. KNN selects all sample points in the sample space. L-KNN first calculates the ratio k of Q_{SUM} and T_U for the prediction point and selects the sample points that are approximate to k in the sample space. The selection method of DL-RKNN is taking the user requirements as the center of the effective sample area and R as the radius of the sample area to select as many useful sample points as possible.

5.2 Experimental Analysis

The variation of ITU utility value and the income-expenditure status of all participants are jointly shown in Figs. 3 and 4 under different strategies, when the task is set as $Q_{SUM} = 20$ G and $T_U = 25$ min, indicating that the game will reach Nash equilibrium under a certain strategy. Here, the abscissa represents the number of tasks that the CSP and ESP should process in a set of strategies, respectively. Fig. 3 shows the changes in ITU utility value I_{ITU} when the CSP and ESP each undertake different tasks. We divide users into three types: price tendency users (ITU1), time tendency users (ITU2) and neutrality users (ITU3). At the allocation strategy $\langle 8, 12 \rangle$, the utility values of the three increase significantly, and the utility values of ITU2 and ITU3 reach the highest point in this round of the game, because T_P is the lowest value in this round of the game. The ITU1 will still seek a lower price, so its highest utility value appears at the lowest price strategy $\langle 4, 16 \rangle$.

Fig. 4 shows that the income of ESP gradually increases while the price trend of CSP and ITU fluctuates. As a follow-up, Fig. 4 adds that the change in user utility value is due to changes in the income of CSP and ESP and user expenditure in the game, when the CSP and ESP undertake different amounts of tasks. In Figs. 3 and 4, the distribution schemes of different strategies are arranged from left to right on the horizontal axis. In fact, this ranking

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

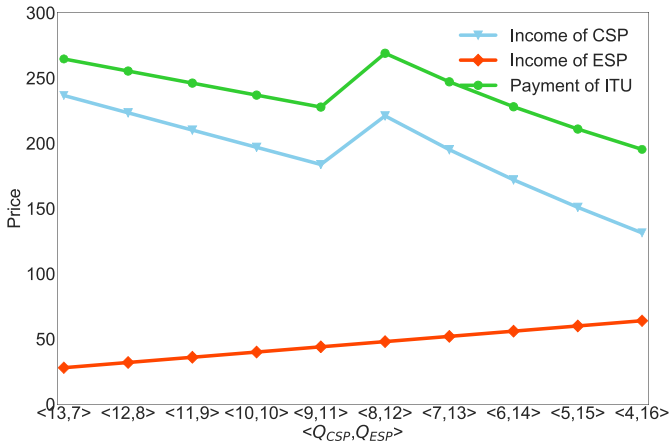


Fig. 4. The payment or income of players.

also represents the order in which different strategies are proposed in a round of the game, that is, the strategy on the left is always proposed first. This is because players in the game try to maximize their profits. In this paper, it can be further understood that the CSP and ESP need to compete for more tasks. As the first actor, the CSP makes Q_{CSP} as large as possible, and then reduces Q_{CSP} if the game fails. Therefore, Q_{CSP} decreases from left to right and Q_{ESP} increases sequentially on the horizontal axis.

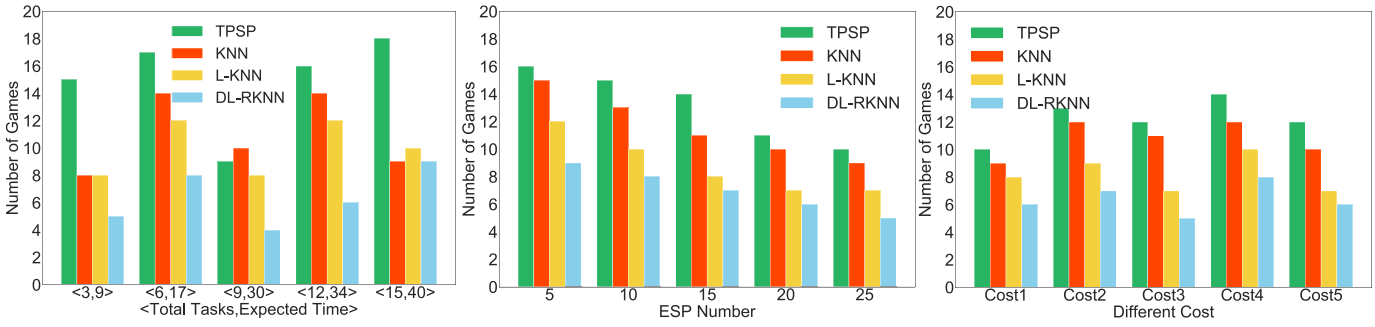
In Fig. 4, the income of CSP and ESP is between [30, 250], showing that the CSP and ESP can always make profits under a reasonable strategy, and the profitability of different strategies is different. Second, the ESP income reaches the highest point at $\langle 8, 12\rangle$, and the CSP income is also at the local highest point. Combined with the strategy $\langle 8, 12\rangle$ in Fig. 3, we can find that the ITU utility value basically reaches the highest value. The above two figures illustrate that the order of the strategies is arranged in the order of decreasing Q_{CSP} in a certain round of the game, that is, the X-axis distribution in Figs. 3 and 4. The income or expenditure amount of players changes accordingly as the game proceeds normally, then the highest ITU utility value is shown in Fig. 3, and the highest comprehensive income of ESP and CSP is shown in Fig. 4 under a certain strategy $\langle 8, 12\rangle$. At this time, the strategy is the Nash equilibrium point for the game to end successfully.

Fig. 5 shows the influence of different prediction algorithms on the number of games. In terms of the results, compared with TPSP, the three algorithms all reduce the number of games to some extent. This is because the ESP always gives the strategy with the highest current income, and such a strategy is often not a global optimal solution. Through the prediction algorithm, the ESP can eliminate these high-yield but unreasonable strategies in advance, so as to achieve the purpose of reducing the number of games. Among the three prediction algorithms, DL-RKNN requires the least number of games, L-KNN is the second, and KNN is the worst.

Fig. 5a shows the change of the number of games when the ITU demand $\langle Q_{SUM}, T_U\rangle$ gradually increases from $\langle 3, 9\rangle$ to $\langle 15, 40\rangle$ when the ESP cost is fixed and the ESP quantity is determined. When TPSP algorithm is used, the number of games is basically maintained between [15, 18]. The lowest value of 9 appears in $\langle 9, 30\rangle$ because the sample size at this point is small, and the performance of each algorithm is relatively close. After the introduction of three prediction algorithms, the number of games is significantly reduced. Among them, the number of games required by DL-RKNN is maintained between [4, 9], the range of the number of games of L-KNN is [8, 12], and the range of the number of games of KNN is [8, 14]. In addition, the increase or decrease of Q_{SUM} and T_U has no direct influence on the number of games, because the quality of the sample points around the pricing scheme to be predicted rather than the demand of a specific task determines the prediction result.

Fig. 5b illustrates the effect of the number of ESPs participating in the game on game times when the cost of ESP is fixed and the ITU demand is determined to be $\langle 6, 17\rangle$. As the number of ESPs increases from 5 to 25, the number of games of TPSP decreases from 16 to 10, and the three prediction algorithms of KNN, L-KNN, and DL-RKNN decrease from 15, 12, and 9 to 9, 7, and 5, respectively. In general, the increase in the number of ESPs can significantly reduce game times in any situation, because the increase in the ESP number enriches the strategy set and makes it easier to achieve the ITU demand. Meanwhile, we find that when the number of ESPs increases from 20 to 25, the decrease of game times is lower than the decrease from 5 to 15. This is due to the fact that when the ESP number increases to a certain scale, the strategy set tends to be saturated. Continuing to increase the number does not necessarily lead to new strategies, so the variation in the number of games tends to moderate.

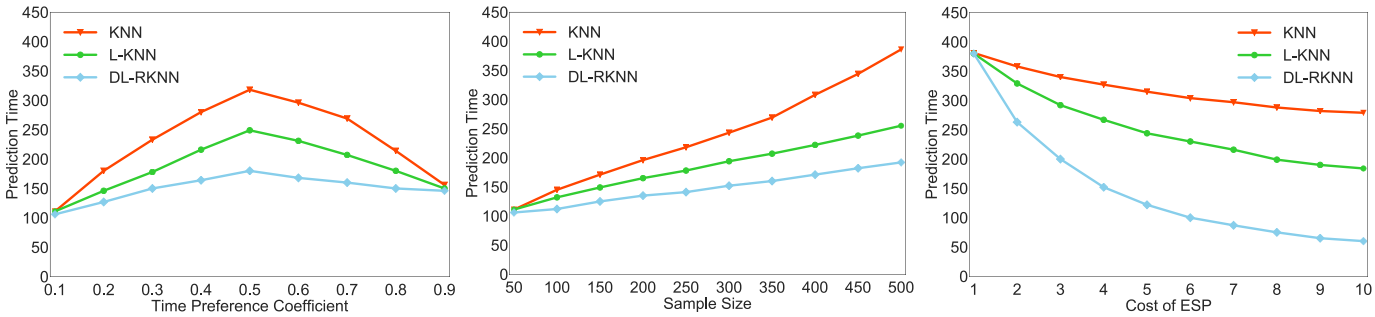
Fig. 5c shows the impact of different algorithms on game times when the requirement $\langle 6, 17\rangle$ is predicted at different



(a) Requirement of ITU (b) ESP Number (c) Different Cost

Fig. 5. Number of games comparison with different ITU requirement, ESP number, and cost parameter.

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.



(a) Time Preference Coefficient (b) Sample Size (c) Cost of ESP

Fig. 6. Prediction time comparison with different ITU preference, sample size, and ESP cost.

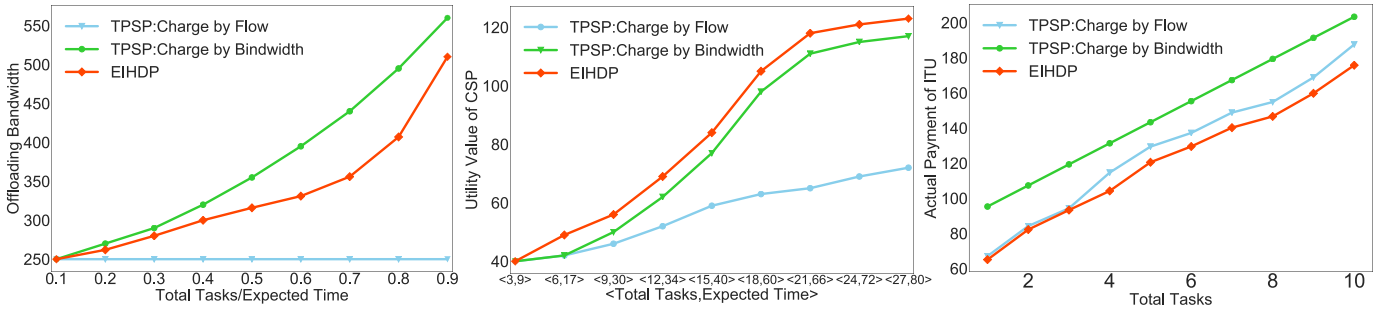
costs. Cost1 indicates that ESP cost is balanced, Cost2 and Cost3 represent the increase and decrease of ESP cost, and Cost4 and Cost5 represent the increase and decrease of CSP cost, respectively. For TPSP and KNN, the game times under Cost1 are 10 and 9, respectively, which are the lowest in their respective whole situations. In other words, the number of games still increases when the cost of ESP or CSP is decreased in Cost2 and Cost5. In comparison, L-KNN and DL-RKNN are more sensitive to the change of cost, and the game times of both keep consistent with the change of cost. The game times of DL-RKNN range between [5,8], and the performance of DL-RKNN under other tasks is almost the same comparing Fig. 5a. According to the experimental results in Fig. 5, it can be seen that the introduction of DL-RKNN can greatly reduce the number of games, thereby achieving superior performance than other other algorithms. Moreover, the reduced number of games also leads to the feasiblity in the practical deployment. Compared with the TPSP method, EIHDP improves system pricing efficiency by $50\%\sim60\%$.

In addition to decreasing the number of games, it is also necessary to effectively reduce the prediction time. In Fig. 6, we compare the influence of different prediction algorithms on prediction time in terms of the ITU time tendency, sample size recorded by the EP and ESP cost. Fig. 6a shows the change in predicted time when the ITU time tendency gradually increases from 0.1 to 0.9. The three algorithms all show a trend of rising first and then falling, and reveal the longest prediction time when the time weight is 0.5 (that is, the time weight is equal to the price weight). The reason for this trend is that although each user may have a different tendency for time and price, more users tend to reach a more balanced situation. Therefore, the sample points are the most and the prediction time is more when time weight = 0.5. Fig. 6b illustrates that the prediction time required by the three algorithms gradually increases when the number of samples recorded by the EP increases from 50 to 500. On the whole, the prediction time of DL-RKNN and L-KNN increases steadily with the increase of sample size, while the prediction time of KNN increases significantly after the sample size reaches 350. This is because both DL-RKNN and L-KNN can pre-filtrate the number of samples participating in the algorithm, and the increase of sample size has little impact on them.

Fig. 6c shows the changes in the prediction time of the three algorithms when the ESP cost gradually increases. The abscissa represents the increasing degree of EPS cost. The prediction time of the three at the lowest cost is 380ms, but the prediction time begins to decrease as the cost increases. The decrease of DL-RKNN is the most obvious, and the prediction time at cost = 10 is only 57 ms. The decrease of KNN and L-KNN is more moderate than that of DL-RKNN. In practice, the increase in ESP cost means that the ESP has more computing power and needs to pay more for electricity and equipment maintenance. Therefore, the increase of cost also represents the increase of computing power, so the prediction time of the algorithm decreases accordingly. In conclusion, the main impact on the prediction time of the three prediction algorithms is the number of samples participating in the prediction. Fig. 5 compares from different aspects, but the essence is to control the sample size. Comparing with Fig. 6, the control mechanism of DL-RKNN is better under the same circumstances so as to reduce the number of games to a large extent while consuming less time, thereby satisfying both practicability and high efficiency. DL-RKNN improves game convergence speed by $42\%\sim58\%$ compared with L-KNN. DL-RKNN improves game convergence speed by $55\%\sim67\%$ compared with KNN.

We compare the performance of EIHDP with the existing traffic charging and bandwidth charging from three different dimensions in Fig. 7. As shown in Fig. 7a, the abscissa is the ratio of the total task Q_{SUM} to the ITU's expected time T_U. The lower ratio indicates that the ITU's requirement for task time is lower, therefore the less bandwidth will be chosen. The higher ratio indicates that the ITU wants the task to be completed as soon as possible so that more bandwidth will be taken. The bandwidth usage charged based on traffic remains unchanged at 250, because this mode charges based on the amount of traffic used, and the bandwidth is fixed. When the ITU's time requirement is low, the bandwidth usage of both bandwidth charging and EIHDP is close. As the ITU's time requirement increases, the difference between the two gradually becomes obvious. This is because the task Q_{SUM} is allocated to the CSP and ESP in EIHDP. The CSP spends less time processing tasks and has more time to transmit than the CSP in the traditional mode when the computing power is fixed. Therefore, the amount of bandwidth used by EIHDP for the same Q_{SUM} is less than the bandwidth charging model. Comparing the situation of EIHDP and traffic charging model, we find that the latter is lower than the former in V_B after the ratio reaches 0.4. It

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.



(a) Bandwidth Size (b) CSP Utility Value (c) ITU Payment

Fig. 7. Our EIHDP model compared with TPSP from bandwidth size, CSP utility value, and ITU Payment.

means that no matter whether the ITU's time is urgent or not, the traffic charging model cannot provide feedback and users cannot control their own time. In summary, EIHDP is more flexible in responding to the urgency of ITU's time than traffic charging model, and EIHDP utilizes ESP to share tasks to increase transmission time, and reduce bandwidth usage compared to bandwidth charging model.

Fig. 7b compares the CSP's utility values of the three modes under different demands. It can be seen that I_{CSP} is the same when the demand is $\langle 3, 9\rangle$. However, I_{CSP} changes with the increase of tasks and time, and the traffic charging model has the worst performance. Since the intervention of ESP in EIHDP reduces the number of tasks processed by CSP, and the ITU can obtain a more favorable price under the game model. At this time, the I_{CSP} must be decreased. Judging from the performance in the figure, the CSP's utility values of EIHDP and bandwidth charging model are very close. Fig. 7c reflects the price that the ITU needs to pay in three modes as the number of tasks increases. The traffic charging mechanism appears as a straight line and is always at the highest position. The standardizations of EIHDP and bandwidth charging are approximate before Q_{SUM} reaches 4, and the ITU payment is lower when Q_{SUM} is greater than 4. Fig. 7a illustrates that EIHDP makes the system use less bandwidth for transmission. Figs. 7b and 7c illustrate that EIHDP reduces the ITU's payment while guaranteeing the CSP's utility value. Since the payment price is always fixed according to the traffic charging method, we selected ITU1 and ITU2 to observe the performance of CSP when charging by bandwidth and EIHDP, as shown in Fig. 7b. With the addition of edge layer, the utility value of CSP is greatly improved compared with the traditional model under the same computing tasks. As the total number of tasks increases, EIHDP is lower than traditional models in terms of the price paid by users of the same type in both models, as shown in Fig. 7c.

5.3 Discussion

Our method can be applied to many practical IoT applications such as smart agriculture, the Internet of Vehicles, and environmental monitoring through the above feasibility analysis. Take the smart agriculture system as an example, in which users want to speculate on the growth trends and expected yields of various crops to decide how to adjust production plans. According to users' requirements, the IoT system needs to collect monitoring data such as rainfall, temperature changes, light intensity, and soil acidity-alkalinity in diverse areas in different periods. For example, there is crop X in area 1, while crops Y and Z exist in area 2. Based on the requirements, the system needs to collect data for crop X in area 1, such as the maximum and average rainfall, the temperature change trend and maximum temperature difference, the maximum and average light intensity, and the variation of soil acidity-alkalinity in the last seven days. Meanwhile, the system also needs to collect similar data for crops Y and Z in area 2 in the last few days. Then, this large amount of IoT data are uploaded to the ESPs and CSP for complex processing and analysis. Finally, combined with the growth characteristics and current growth situation of the corresponding crops, the system provides users with prediction results. In this process, the problem of resource coordination between the ESPs and CSP and the benefit equilibrium of cloud-edge-client are involved. Therefore, our method makes full use of the advantages of cloud-edge-client collaboration to provide users with the optimal service scheme.

6 Conclusion

The emergence of IoT enables users to pay for sensor resources and cloud services on demand. In this paper, we study hierarchical dynamic pricing for cloud-edge-client collaboration under incomplete information. We use a double-layer Stackelberg game model to describe the mixed game process of CSPs, ESPs and ITUs. We propose an edge-intelligent hierarchical dynamic pricing (EIHDP) mechanism based on the DL-RKNN prediction algorithm. Through extensive performance evaluation, EIHDP improves system pricing efficiency by $50\%\sim60\%$ compared with the TPSP method. Furthermore, compared with L-KNN and KNN for pricing prediction, our DL-RKNN algorithm improves game convergence speed by $42\%\sim58\%$ and $55\%\sim67\%$, respectively. There are some possible directions to study in the future. For example, ESPs can cooperate with others to improve the game success rate and reduce the resource consumption of pricing prediction. The coalitional game based on federated learning can be a considerable solution.

Acknowledgments

This work was supported in part by Grants from Guangdong Key Lab of AI and Multi-modal Data Processing; in part by the Natural Science Foundation of Fujian Province of China under Grant 2020J06023; in part by BNU-UIC Institute of Artificial Intelligence and Future Networks funded by Beijing Normal University (Zhuhai) and AI-DS Research Hub, BNU-HKBU United International College (UIC), Zhuhai, Guangdong, China; in part by the National Natural Science

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

Foundation of China (NSFC) under Grant 61872154, Grant 61872239, and Grant 61772148; and in part by Subsidized Project for Postgraduates' Innovative Fund in Scientific Research of Huaqiao University, China under Grant 17014083012.

REFERENCES

[1] X. Liu, M. S. Obaidat, C. Lin, T. Wang, and A. Liu, "Movement-based solutions to energy limitation in wireless sensor networks: State of the art and future trends," *IEEE Netw.*, to be published, doi: 10.1109/MNET.011.2000445.

[2] T. Wang, H. Luo, X. Zeng, Z. Yu, A. Liu, and A. K. Sangaiah, "Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.2997377.

[3] T. Wang, Y. Mei, X. Liu, J. Wang, H.-N. Dai, and Z. Wang, "Edge-based auditing method for data security in resource-constrained internet of things," *J. Syst. Architecture*, 2020, doi: 10.1016/j.sysarc.2020.101971.

[4] Y. Wu, H. Huang, Q. Wu, A. Liu, and T. Wang,"A risk defense method based on microscopic state prediction with partial information observations in social networks," *J. Parallel Distrib. Comput.*, vol. 2019, no. 131, pp. 189–199, 2019.

[5] X. Liu, Z. Hu, H. Ling, and Y.-M. Cheung, "MTFH: A matrix tri-factorization hashing framework for efficient cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 964–981, Mar. 2021.

[6] Z. Li, W. Wei, T. Zhang, M. Wang, S. Hou, and X. Peng, "Online multi-expert learning for visual tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 934–946, Aug. 2019.

[7] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 15, no. 6, pp. 3661–3669, Jun. 2019.

[8] S. Long, W. Long, Z. Li, K. Li, Y. Xia, and Z. Tang, "A game-based approach for cost-aware task assignment with QoS constraints in large data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1629–1640, Jul. 2021.

[9] Q. Liu, Y. Peng, J. Wu, T. Wang, and G. Wang, "Secure multi-keyword fuzzy searches with enhanced service quality in cloud computing," *IEEE Trans. Netw. Service Manage.*, to be published, doi: 10.1109/TNSM.2020.3045467.

[10] T. Wang *et al.*, "Privacy-enhanced data collection based on deep learning for Internet of Vehicles," *IEEE Trans. Ind. Inform.*, vol. 16, no. 10, pp. 6663–6672, Oct. 2020.

[11] T. Wang, D. Zhao, S. Cai, W. Jia, and A. Liu, "Bidirectional prediction-based underwater data collection protocol for end-edge-cloud orchestrated system," *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4791–4977, Jul. 2020.

[12] Q. Liu, Y. Peng, S. Pei, J. Wu, T. Peng, and G. Wang, "Prime inner product encoding for effective wildcard-based multi-keyword fuzzy search," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2020.3020688.

[13] K. Xie *et al.*, "An efficient privacy-preserving compressive data gathering scheme in WSNs," *Inf. Sci.*, vol. 390, pp. 82–94, 2017.

[14] W. Li *et al.*, "Complexity and algorithms for superposed data uploading problem in networks with smart devices," *IEEE Internet of Things J.*, vol. 7, no. 7, pp. 5882–5891, Jul. 2020.

[15] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A stackelberg game approach," *Comput. Netw.*, vol. 129, pp. 399–409, 2017.

[16] X. Li *et al.*, "Quick and accurate false data detection in mobile crowd sensing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1339–1352, Jun., 2020.

[17] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-efficient resource allocation for multi-user mobile edge computing," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–7.

[18] Y. Luo, K. Yang, Q. Tang, J. Zhang, P. Li, and S. Qiu, "An optimal data service providing framework in cloud radio access network," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–11, 2016.

[19] Z. Liao, J. Liang, and C. Feng, "Mobile relay deployment in multi-hop relay networks," *Comput. Commun.*, vol. 112, pp. 14–21, 2017.

[20] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[21] W. Zhang *et al.*, "Adaption resizing communication buffer to maximize lifetime and reduce delay for WVSNs," *IEEE Access*, vol. 7, pp. 48 266–48 287, 2019.

[22] X. Liu, P. Lin, T. Liu, T. Wang, A. Liu, and W. Xu, "Objective-variable tour planning for mobile data collection in partitioned sensor networks," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2020.3003004.

[23] Z. Xia, Z. Fang, F. Zou, J. Wang, and A. K. Sangaiah, "Research on defensive strategy of real-time price attack based on multiperson zero-determinant," *Secur. Commun. Netw.*, vol. 2019, pp. 1–13, 2019.

[24] C. Zhu, X. Li, V. C. Leung, L. T. Yang, E. C.-H. Ngai, and L. Shu, "Towards pricing for sensor-cloud," *IEEE Trans. Cloud Comput.*, vol. 8, no. 4, pp. 1018–1029, Fourth Quarter 2020.

[25] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic optimal pricing for heterogeneous service-oriented architecture of sensor-cloud infrastructure," *IEEE Trans. Services Comput.*, vol. 10, no. 2, pp. 203–216, Mar./Apr. 2017.

[26] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IoT-based energy management in smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, Mar/Apr. 2019.

[27] J. Mocnej *et al.*, "Quality-enabled decentralized IoT architecture with efficient resources utilization," *Robot. Comput.-Integr. Manuf.*, vol. 67, 2021, Art. no. 102001.

[28] Q. Tang, K. Yang, D. Zhou, Y. Luo, and F. Yu, "A real-time dynamic pricing algorithm for smart grid with unstable energy providers and malicious users," *IEEE Internet of Things J.*, vol. 3, no. 4, pp. 554–562, Aug. 2016.

[29] N. Lin, X. Wang, Y. Zhang, X. Hu, and J. Ruan, "Fertigation management for sustainable precision agriculture based on Internet of Things," *J. Cleaner Prod.*, vol. 277, 2020, Art. no. 124119.

[30] M. S. Kumar and V. S. Dhulipala, "Fuzzy allocation model for health care data management on IoT assisted wearable sensor platform," *Measurement*, vol. 166, 2020, Art. no. 108249.

[31] T. Zhang, "Data offloading in mobile edge computing: A coalition and pricing based approach," *IEEE Access*, vol. 6, pp. 2760–2767, 2017.

[32] S.-H. Kim, S. Park, M. Chen, and C.-H. Youn, "An optimal pricing scheme for the energy-efficient mobile edge computation offloading with OFDMA," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1922–1925, Sep. 2018.

[33] Y. Jie, X. Tang, K.-K. R. Choo, S. Su, M. Li, and C. Guo, "Online task scheduling for edge computing based on repeated stackelberg game," *J. Parallel Distrib. Comput.*, vol. 122, pp. 159–172, 2018.

[34] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet of Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.

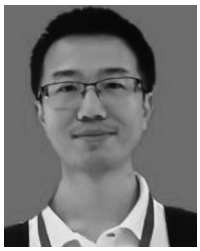
[35] W. Wei, X. Fan, H. Song, X. Fan, and J. Yang, "Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing," *IEEE Trans. Services Comput.*, vol. 11, no. 1, pp. 78–89, Jan./Feb. 2018.

[36] Q. Lu, S. Lü, and Y. Leng, "A nash-stackelberg game approach in regional energy market considering users' integrated demand response," *Energy*, vol. 175, pp. 456–470, 2019.

[37] Y. Ni, A. S. Leong, D. E. Quevedo, and L. Shi, "Pricing and selection of channels for remote state estimation using a stackelberg game framework," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 5, no. 4, pp. 657–668, Dec. 2019.

[38] B. Shen, Y. Shen, and W. Ji, "Profit optimization in service-oriented data market: A stackelberg game approach," *Future Gener. Comput. Syst.*, vol. 95, pp. 17–25, 2019.

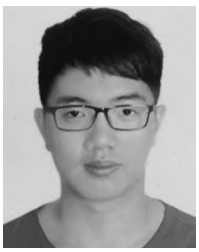
[39] C. Casorrán, B. Fortz, M. Labbé, andF. Ordóñez,"A study of general and security stackelberg game formulations," *Eur. J. Oper. Res.*, vol. 278, no. 3, pp. 855–868, 2019.



Tian Wang received the BSc and MSc degrees in computer science from the Central South University, in 2004 and 2007, respectively, and the PhD degree in computer science from the City University of Hong Kong, in 2011. Currently, he is a professor with the Institute of Artificial Intelligence and Future Networks, Beijing Normal University & UIC. His research interests include Internet of Things, edge computing and mobile computing. He has 27 patents and has published more than 200 papers in high-level journals and conferences. He has more than 6500 citations, according to Google Scholar. His H-index is 40. He has managed five national natural science projects (including two sub-projects), and four provincial-level projects.

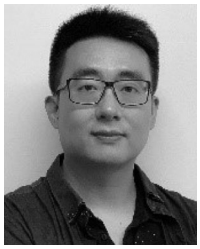
Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.

Yucheng Lu received the BS degree from the Hubei University of Technology, in 2018. He is currently working toward the master's degree with the College of Computer Science and Technology, Huaqiao University, majoring in software engineering. His research interests include wireless sensor networks, mobile computing, and edge computing.



Jianhuang Wang is working toward the College of Computer Science and Technology, Huaqiao University, majoring in network engineering. He interests in machine learning and network security.

Hong-Ning Dai received the PhD degree in computer science and engineering from the Department of Computer Science and Engineering, Chinese University of Hong Kong. He is currently with the Faculty of Information Technology, Macau University of Science and Technology as an associate professor. His research interests include Internet of Things, big data analytics, and blockchain. He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE).



Xi Zheng received the PhD degree in software engineering from UT Austin. He specialised in Service Computing, IoT Security, and Reliability Analysis. Published more than 40 high quality publications in top journals and conferences (PerCOM, ICSE, IEEE IoT Journal, ICCPS, *IEEE Systems Journal*, *ACM Transactions on Embedded Computing Systems*, *IEEE Transactions on Vehicular Technology*). Awarded the Best Paper in Australian Distributed Computing and Doctoral Conference in 2017. Awarded Deakin Research Outstanding Award in 2016. Active reviewer for top journals and conferences.

Weijia Jia received the BSc and MSc degrees in computer science from Center South University, China, in 1982 and 1984, the and master of applied science and PhD degrees in computer science from the Polytechnic Faculty of Mons, Belgium, in 1992 and 1993, respectively. For 93-95, he joined German National Research Center for Information Science (GMD) in Bonn (St. Augustine) as research fellow. From 1995–2013, he worked in City University of Hong Kong as a full professor in Computer Science Department Currently, he is a professor with the Artificial Intelligence and Future Networks, Beijing Normal University & UIC, China. His research interests include smart city; next generation IoT, knowledge graph constructions; multicast and anycast QoS routing protocols, wireless sensor networks, and distributed systems.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: Hong Kong Baptist University. Downloaded on March 17,2022 at 08:17:17 UTC from IEEE Xplore. Restrictions apply.