IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 23, 2022, accepted May 6, 2022, date of publication May 16, 2022, date of current version May 25, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3175194*

# DPRL: Task Offloading Strategy Based on Differential Privacy and Reinforcement Learning in Edge Computing

**PEIYING ZHANG[1,2], PENG GAN[1], LUNJIE CHANG[3], WU WEN[4], M. SELVI[5], AND GODFREY KIBALYA[6]**

[1]College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China
[2]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[3]Research Institute of Petroleum Exploration and Development, PetroChina Tarim Oilfield Company, Korla 841000, China
[4]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China
[5]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India
[6]Department of Network Engineering, Technical University of Catalonia (UPC), 08034 Barcelona, Spain

Corresponding author: Wu Wen (wenwu@gzhu.edu.cn)

This work was supported in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2020MF006; in part by the Industry-University Research Innovation Foundation of Ministry of Education of China under Grant 2021FNA01001 and Grant 2021FNA01005; in part by the Major Scientific and Technological Projects of the China National Petroleum Corp. (CNPC) under Grant ZD2019-183-006; and in part by the Open Foundation of State Key Laboratory of Integrated Services Networks, Xidian University, under Grant ISN23-09.

**ABSTRACT** Mobile edge computing has been widely used in various IoT devices due to its excellent computing power and good interaction speed. Task offloading is the core of mobile edge computing. However, most of the existing task offloading strategies only focus on improving the unilateral performance of MEC, such as security, delay, and overhead. Therefore, focus on the security, delay and overhead of MEC, we propose a task offloading strategy based on differential privacy and reinforcement learning. This strategy optimizes the overhead required for the task offloading process while protecting user privacy. Specifically, before task offloading, differential privacy is used to interfere with the user's location information to avoid malicious edge servers from stealing user privacy. Then, on the basis of ensuring user privacy and security, combined with the resource environment of the MEC network, reinforcement learning is used to select appropriate edge servers for task offloading. Simulation results show that our scheme improves the performance of MEC in many aspects, especially in security and resource consumption. Compared with the typical privacy protection scheme, the security is improved by 7%, and the resource consumption is reduced by 9% compared with the typical task offloading strategy.

**INDEX TERMS** Mobile edge computing, task offloading, differential privacy, reinforcement learning.

## I. INTRODUCTION

The rapid development of the Internet of Things (IoT) and 5G technology has promoted the combination of various smart applications with mobile devices, such as face recognition, smart medical care, etc [1]. These smart applications are gradually becoming more complex and diverse. However, the size and weight of mobile devices, computing power, storage capacity and network transmission capacity of users are limited, which makes them unable to meet the quality of service requirements of users. Therefore, in order to solve the above problems, related scholars proposed mobile edge computing technology (MEC) [2]. FIGURE 1 reveals the application of MEC in an intelligent transportation system. The core technology of MEC is to deploy service nodes at the edge of the network, and offload the complex computing tasks of user to edge servers for computing [3]. Compared with local computing and cloud computing, MEC has stronger computing power and lower latency. However, while MEC provides users with efficient computing services, the privacy of mobile users is also facing a huge threat [4]. In order to satisfy the mobile user's experience, the previous task offloading strategies mostly aim to reduce the delay and network resource overhead, while ignoring the privacy

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Maaz Rehan.

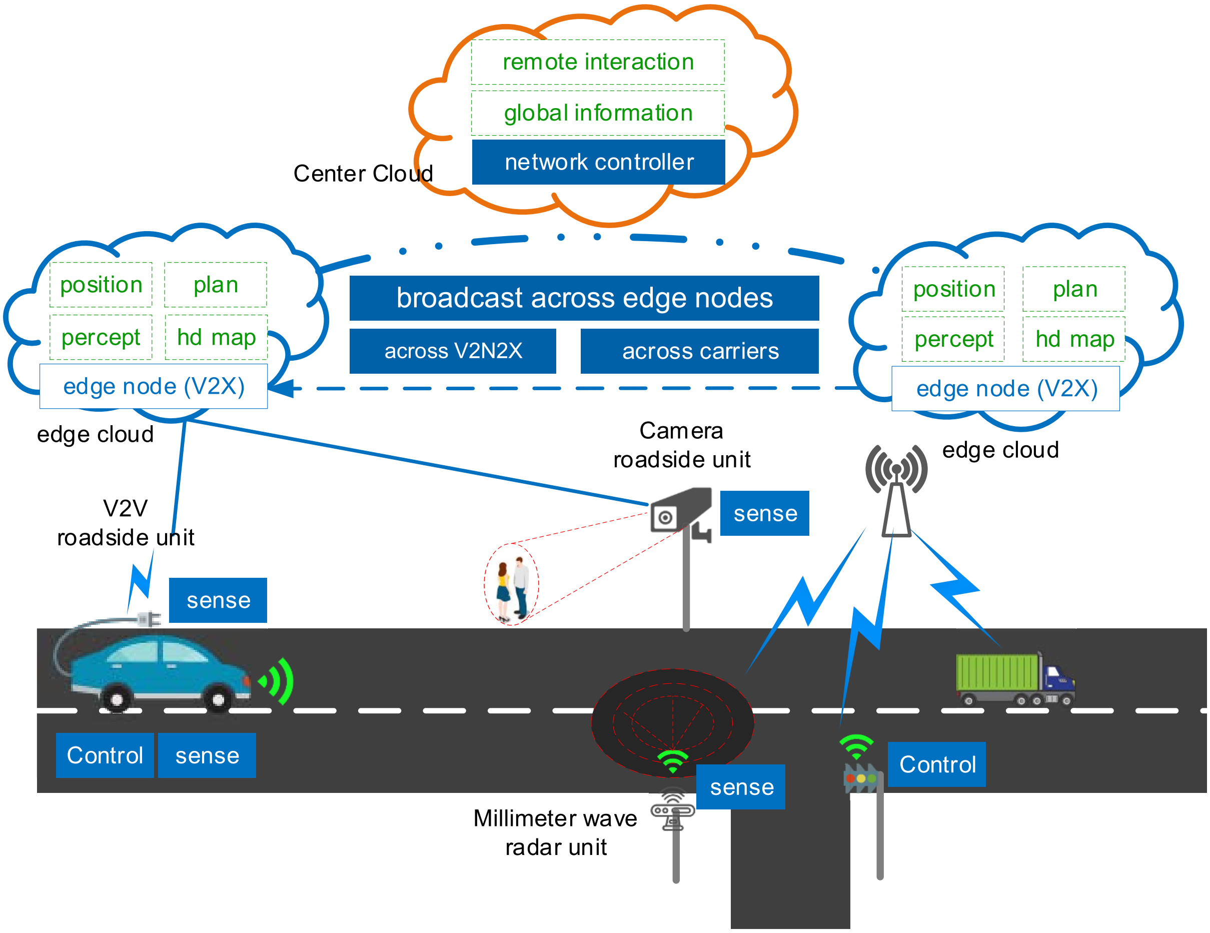


**FIGURE 1. The application of MEC in intelligent transportation system.**

protection of the user. For example, a malicious edge server can infer the user's location, name, interests and other private information by monitoring the user's wireless channel information. Therefore, when using MEC to offload computing tasks of user, his private information must be protected. In addition, it should be noted that excessive privacy protection may lead to high communication overhead and delay, which will affect the user's quality of service experience. Therefore, how to reduce the overhead of task offloading while protecting user privacy is still an urgent problem that needs to be prepared.

To solve the above problems, this paper proposes a task offloading strategy (DPRL) based on differential privacy (DP) [5], [6] and reinforcement learning (RL) [7]–[9]. Specifically, before the computing task is unloaded, to avoid leakage of user privacy, we first encrypt the user's private information with DP. Then, to reduce communication overhead, we select appropriate edge servers to offload tasks in line with the network resource environment and RL.

As a privacy protection technology, DP has been widely used in many fields [10], [11]. It protects user privacy by introducing sufficiently small interference information into user information. Since the introduced interference is small enough and can protect privacy, the edge server in the MEC can still analyze the computing tasks through the disturbed information. In addition, RL, as a typical machine learning technique, has been widely used in many fields. For example, content caching and D2D offloading [12]. Recent studies have also shown its superior performance in edge computing [13] and edge intelligence [14]. Therefore, inspired by the above research, it is feasible to use RL to reduce the communication overhead in MEC. The main innovations of this paper are as follows:

(1) Firstly, before task offloading, to prevent malicious edge servers from stealing user's privacy, we use DP to protect user's location information. User's location information often contains a lot of user privacy.

(2) Secondly, we construct a four-layer policy model as a learning agent for RL, which interacts with the MEC network environment to provide a task offloading scheme that minimizes communication overhead.

(3) Finally, to further protect users' privacy, we measure the trust of each edge server according to the intimacy between the edge server and users, and then take it as one of the characteristics of the edge server. In addition, to ensure that the RL agent can be trained in the real MEC network

environment, we extract the network resources of MEC and form a feature matrix as the input of the policy model.

The rest of this paper is organized as follows. Section II analyzes the research status of MEC. Section III describes the DP-based user privacy protection strategy. Section IV presents an RL-based edge server selection strategy. Section V analyzes the performance of the DPRL strategy. Section VI summarizes the full text and presents future work.

## II. RELATED WORK

As a hot topic in network research in recent years, MEC has been widely used in intelligent transportation systems, Internet of Vehicles and other fields. The existing researches on MEC mainly focus on two aspects: resource optimization and privacy protection during MEC task offloading. This section will analyze the research status of task offloading in MEC from the above two aspects, and compare the task offloading strategy proposed in this paper with these researches.

### A. RESOURCE OPTIMIZATION IN TASK OFFLOADING

According to the analysis of existing research, the overhead generated by MEC mainly includes the delay (DL), bandwidth (BW) and computing (CPU) resources required for the calculation. The main goal of resource optimization in the task offloading process is to minimize the BW and CPU resources consumed in the MEC process with low DL [15]–[17].

Liu *et al.* [18] adjust the task offloading strategy according to the queuing buffer of computing tasks, the available CPU resources of user and edge servers to achieve the minimum time consumption. Literature [19] first proves that minimizing the average DL during task offloading is an NP-hard problem, and then uses Lyapunov to optimize the problem to achieve a near-optimal task offloading strategy. However, it is worth noting that [18], [19] only consider the DL in the task offloading process, but do not pay attention to the overhead in the task offloading process. Therefore, the literatures [20]–[22] try to optimize the computational cost within a small DL. Among them, Kamoun *et al.* [20] propose a resource allocation strategy that takes into account the user's local CPU resources and the wireless network resources between the user and the edge server, which can minimize the average overhead of the user while satisfying the pre-defined DL constraints. Huang *et al.* [21] propose a dynamic task offloading decision algorithm that combines dynamic task offloading decision with computing resource allocation to minimize overhead while ensuring queue stability. Literature [22] studies long-term task offloading and resource scheduling. Through Lyapunov optimization technology, the task offloading problem is decomposed into multiple subproblems, so as to achieve the goal of minimizing the average power consumption of the system. Besides, to minimize the task duration while satisfying the energy budget constraints, literature [23] proposed an online task offloading algorithm based on asynchronous advantage actor-critic. The algorithm can learn good offloading strategies to obtain near-optimal task assignments.

Although the above studies improve the resource consumption of MEC to a certain extent, it has the following limitations. On the one hand, most of these schemes only reduce the unilateral resource consumption and do not comprehensively consider the CPU, DL and BW required by MEC. On the other hand, most of these schemes use heuristic algorithms to solve MEC problems, which is easy to fall into local optimization. In order to solve the above problems, based on the inspiration of literature [23], we propose a DRRL scheme. DPRL extracts the MEC network resource state as the RL training environment, so that the performance of the whole MEC is balanced. In addition, the RL problem is optimized to avoid falling into local optimization.

### B. PRIVACY PROTECTION IN TASK OFFLOADING

The privacy problem in task offloading usually consists of two aspects, namely the privacy problem caused by data interaction and the privacy problem caused by the task offloading feature. The privacy problem caused by data interaction is mainly due to the malicious MEC server stealing the user's privacy based on the calculation data. To solve this problem, Xu *et al.* [24] divide computing tasks into different types of data, and increase privacy entropy by enhancing the uncertainty of computing tasks to protect user privacy. Xu *et al.* [25] propose a two-stage task offload optimization strategy to maximize resource utilization and minimize time cost in the first stage, and balance task offload performance and privacy effect in the second stage. Aiming at the privacy problem caused by the characteristics of task offloading, Min *et al.* [26] propose a privacy-aware offloading scheme to improve the privacy protection level of users. The scheme selects the task offloading ratio and local processing ratio of user according to information such as wireless channel status, sensor data, size and priority of computing tasks, which reduces the DL and saves the computational overhead of user. Nguyen *et al.* [27] propose a MEC-based mobile network user privacy model, where user selects an efficient task offloading scheme through constrained Markov decision.

Similar to the literature [26], [27], the proposed scheme is mainly used to solve the privacy problem caused by the task offloading feature. The difference is that most of the privacy protection methods in previous task offloading only pre-set the privacy protection level and optimize it as an indicator. This kind of methods lacks clear privacy protection means, which may lead to privacy disclosure. And due to the lack of protection of the location information of the user, a malicious MEC server can steal the private information of the user according to the location information of the user. The scheme proposed in this paper takes the user's location information as the entry point, and interferes with the distance between the user and the edge server through DP, which provides clear privacy protection for the user.

### C. RESEARCH ON MEC SECURITY BASED ON DIFFERENTIAL PRIVACY

At present, DP-based MEC research is mainly divided into centralized DP and localized DP. Centralized DP refers to uploading the initial data to the data center for storage, and then the data center uses the DP mechanism to perform privacy protection processing on the data, and finally publishes the protected privacy information. Literature [28] proposed a new data encryption technology based on Trusted Third Party (TTP), which can maintain privacy protection in cloud environment. Literature [29] uses DP to build an optimal model based on TTP, in which TTP can obtain some information from public datasets to better understand private datasets without compromising its privacy. The work of Wang *et al.* [30] is to propose a DP-based location perturbation scheme to protect location privacy in third-party geolocation services, which uses a TTP architecture to perform the protection. However, it is worth noting that although the centralized DP mechanism can ensure that the user's private information will not be leaked. However, since third-party data collectors are often untrustworthy, they can pose a threat to users' private information without their knowledge.

To solve the above problems, a localized DP mechanism is proposed. On the basis of protecting user privacy, localized DP also considers the personalized protection of each user's private information. The difference between it and centralized DP is that each user will obfuscate the private data locally, and then publish the perturbed private data. The representative local DP studies are in [31] and [32]. Wang *et al.* [31] proposed a method to protect user location based on localized DP preference. In order to avoid the privacy problems caused by excessive reliance on third parties, Zhang *et al.* [32] proposed a contract signing protocol, which uses the decentralized features of block-chain technology to avoid excessive reliance on third parties and protect user privacy.

Similar to the literatures [31] and [32], this paper uses localized DP technology to solve the security problem of MEC, and on this basis, uses RL to reduce the communication overhead of MEC.

## III. LOCATION PRIVACY PROTECTION BASED ON DP

During the task offloading process of MEC, the user will choose different task offloading strategies according to the wireless channel between the user and the edge server. If the channel conditions are good, user will offload computing tasks to edge servers to save overhead. On the contrary, if the channel condition is poor, the user tends to perform operations locally. It is worth noting that when the channel conditions are good, the malicious edge server will infer the location of the user by analyzing the wireless channel between the user and the edge server. Therefore, in this section, we will focus on the location privacy leakage problem of user during task offloading.

### A. DESIGN OF INTERFERENCE DISTANCE PROBABILITY DENSITY FUNCTION

DP is widely used in data mining and deep learning privacy protection because of its superior privacy protection performance. However, due to the limited coverage of edge servers, traditional DP methods are difficult to directly apply to task offloading in MEC. Therefore, this paper proposes a new distance obfuscation probability density function (PD), which can directly obfuscate the distance between the user and the edge server to prevent the leakage of the user's location information. It is calculated as follows:

$$PD\left(l^*|l\right) = \begin{cases} \dfrac{\epsilon}{2\Delta l} e^{-\frac{\epsilon|l^*-l|}{\Delta l} + \frac{e^{\frac{\epsilon(l1-l)}{\Delta l}} + e^{\frac{\epsilon(l2-l)}{\Delta l}}}{2\Delta l}}, \\ \qquad if\ l^* \in [l_1, l_2] \\ 0, \qquad otherwise \end{cases} \tag{1}$$

among them, $l$ is the distance between the user and the edge server, the distance after obfuscation is $l^*$. $l_1$ and $l_2$ represent the previous and next term of the confusion range, where $l_1 < l_2, l_1, l_2 \in [0, l_{max}]$ and $\Delta l = l_2 - l_1$. This function takes into account the restriction of the confusion range and ensures that $l^*$ is within the confusion range with probability 1.

The flow of using PD to interfere with probability is shown in FIGURE 2. It can be revealed from Fig.2 that the key to using the PD to interfere with the distance between the user and the edge server lies in the selection of the interference range, that is, the determination of $l_1$ and $l_2$. Generally speaking, when the real distance $l$ is relatively small, the distance $l^* \in [l_1, l_2]$ after interference is relatively small, so as to ensure that users can offload more computing tasks to the edge server to save costs. Therefore, in order to ensure that the task offloading decision made by the user based on the interference distance $l^*$ has the same utility as the decision made at the distance $l$, the value of $l^*$ must be within the smallest possible range on the left and right sides of $l$. The closer $l_1$ and $l_2$ are to $l$, the better the utility of the task offloading strategy after privacy protection.

### B. DESIGN OF PRIVACY LEAKAGE DEGREE FUNCTION

According to equation 1, the user can interfere with the real distance between the user and the edge server before the task is offloaded. Malicious edge servers are then unable to deduce the user's true location. However, it is worth noting that since the process of using DP to interfere with the real distance is random, the real distance $l$ and the interfered distance $l_*$ may be equal. The greater the equal probability, the higher the probability of user privacy leakage. This privacy leakage caused by the randomness of DP interference data is inevitable, so we also need to introduce the user privacy leakage degree function. We first use the Kullback-Leibler divergence (KLD) to measure the fit between the true distance $l$ with and without privacy protection. The specific calculation method is as follows:

$$KLD\,(P||Q) = \int_{l_1}^{l_2} Q\left(l^*|l\right) \log \frac{Q\,(l^*|l)}{P\,(l^*|l)} dl^*, \tag{2}$$

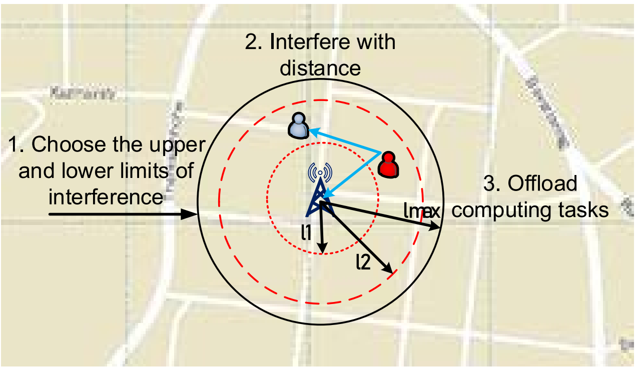

2. Interfere with distance
1. Choose the upper and lower limits of interference
3. Offload computing tasks
lmax
l1
l2

**FIGURE 2.** DP-based privacy protection process.

among them, $Q(l^*|l)$ represents the probability distribution of the user's task offloading according to the real distance, and $P(l^*|l)$ represents the probability distribution of the user's task offloading according to the distance after interference. According to the definition of KLD, we can know that the smaller the value, the higher the degree of fit and the greater the probability of user privacy leakage. Conversely, when the value of KLD is larger, the fitting degree is lower and the probability of user privacy leakage is smaller. Therefore, the degree of privacy leakage (PLD) can be defined as the inverse of equation 3.

$$PLD = -\int_{l_1}^{l_2} Q\left(l^*|l\right) \log \frac{Q\left(l^*|l\right)}{P\left(l^*|l\right)} dl^* \tag{3}$$

## IV. RL-BASED EDGE SERVER SELECTION ALGORITHM

After using the DP to interfere with the user's location information, the computing task needs to be offloaded. Before offloading computing tasks, to further protect user privacy and ensure user QoS requirements, we need to select a safe and high-performance edge server for task offloading. Therefore, in this section, we first set a trust attribute for each edge server according to the interaction frequency between the edge server and the user. Then we design an edge server selection algorithm based on RL, which comprehensively considers the resource environment and security conditions of the entire MEC network. Finally provides users with an optimal task offload scheme through the interaction between the resource environment and the RL agent. This scheme can optimize the resource overhead of task offloading while protecting user privacy [15].

### *A. FEATURE EXTRACTION OF MEC NETWORK RESOURCES*

Edge server has strong social attributes because it interacts with users. Interaction frequency is an important aspect of the social attribute of edge server. The higher the interaction frequency, the more reliable the edge server is. Before feature extraction, we set the trust attribute for each edge server by calculating the interaction frequency between the edge server and users. The calculation of trust is as follows.

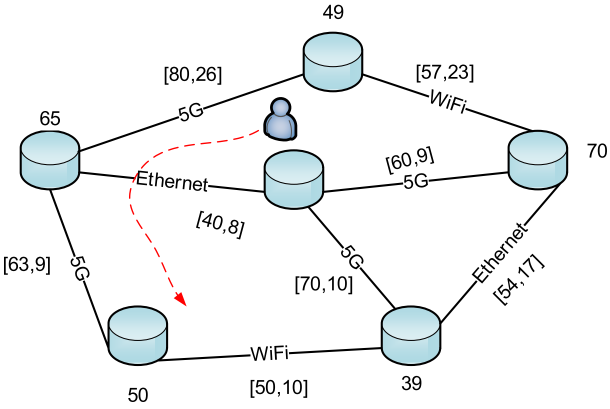


**FIGURE 3.** MEC network model.

$$Trust = \frac{\sum_{i=1}^{n} t_i}{T}, \tag{4}$$

where, $T$ represents a time period, $n$ represents the number of interactions between the edge server and the user in the $T$ time period, and $t_i$ represents the time of each interaction.

To facilitate the extraction of network features, as shown in FIGURE 3, we model the MEC network as a weighted undirected graph $G^{es} = \{N^{es}, L^{es}\}$. $N^{es}$ represents the set of edge servers, and $L^{es}$ represents the set of communication links. The numbers on the edge server nodes represent the available CPU of the edge server, and the numbers on the links represent the BW and DL, respectively. The key to using RL to select edge servers for task offloading lies in the understanding of the MEC network environment, which facilitates the training of RL agents in real environments. Therefore, we extracted the following features for each edge server node:

(1) CPU: The capacity of edge server depends on its available computing resources. An edge server with higher computing capability can load more computing tasks.

(2) Degree: In the MEC network, the greater the degree of the edge server node, the more available communication links exist between it and the user.

(3) $SUM\ (n^{es})_{BW}$: Each edge server has a set of links linked to it, $SUM\ (n_{es})_{BW}$ is the sum of the bandwidth resources of these links. The larger the value, the stronger the ability to transmit data between the edge server and the user.

(4) $SUM\ (n^{es})_{DL}$: $SUM\ (n_{es})_{DL}$ is the sum of the delays of the links connected to the edge server. The smaller the value, the shorter the interaction time between the edge server and the user, and the better the user QoS experience.

(5) $Trust(n^{es})$: Each node has a trust attribute. The higher the trust, the smaller the probability of user privacy disclosure.

In fact, there are many network features that can be extracted from MEC networks. The more features extracted, the more realistic the training environment of the RL agent, but the complexity of the algorithm increases accordingly.

Therefore, after comprehensively considering the actual situation of MEC task offloading, we extract the above features, which are also the main factors determining the task offloading overhead.

After extracting the features of the edge servers, we normalize them to feature vectors, and construct the feature vectors of all edge servers into a feature matrix, which is used as the input environment for the RL agent. The feature matrix is shown in equation 5, as shown at the bottom of the next page.

### B. DESIGN OF POLICY NETWORKS FOR RL

We design a four-layer policy network as the RL agent, which is the input layer, the convolutional layer, the softmax layer and the output layer. We take the feature matrix of the MEC network as the input of the agent, and operate it in the convolution layer. The specific convolution operation is as follows:

$$Output_i = w.v_i + b, \tag{6}$$

among them, $Output_i$ is the i-th output of the convolution layer, $w$ is the weight vector of the convolution kernel, $v_i$ is the i-th input of the convolutional layer, and b is the deviation.

The softmax layer mainly outputs the probability of each edge server being selected according to the available resource vector of each edge server. The higher the probability, the higher the priority of selecting the edge server for task offloading. The specific calculation method of softmax is as follows:

$$P_i = \frac{e^{Output_i}}{\sum_j e^{Output_j}}. \tag{7}$$

During the training of an RL agent, the agent relies on the reward signal to decide what action to take next. In order to motivate the agent to act correctly and optimize the resource overhead during the MEC task offloading process, this paper uses the total resource consumption of the task offloading process as the reward signal. If the current total resource consumption is low, it means that the agent's current behavior is correct. Conversely, if the current total resource consumption is high, it means that the agent needs to adjust the current behavior. The equation for calculating total resource consumption is as follows:

$$RC\left(s,a\right) = \omega_1 E\left(s,a\right) + \omega_2 T\left(s,a\right) + \omega_3 \varGamma\left(s,a\right), \tag{8}$$

among them, $E\left(s,a\right)$ represents the total computing resources required to offload the current task. $T\left(s,a\right)$ represents the total delay in offloading the current task. $\varGamma\left(s,a\right)$ represents the task loss probability, which is calculated as follows:

$$\varGamma\left(s,a\right) = \frac{n_{loc}}{N_t}, \tag{9}$$

where $n_{loc}$ represents the number of tasks executed locally, and $N_t$ represents the total number of currently submitted tasks unloaded. In addition, $s$ and $a$ represent the current state and behavior of RL, respectively, where $s$ is determined by the input feature matrix, and $a$ is the offloading of the current computing task.

**TABLE 1.** The parameters of the MEC network.

| Parameter | Value | Unit |
|---|---|---|
| Channel bandwidth | 10 | MHZ |
| Background noise | -174 | dBM/HZ |
| User CPU frequency | 1 | GHZ |
| Edge server CPU frequency | 10 | GHZ |
| Calculate density | 1000 | cycles/bit |
| Task size | 0.1-0.4 | MB |
| Transmission power | 500 | mW |

## V. EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

In this section, we will verify the effectiveness of the proposed scheme in this paper. We divided the experimental part into two parts. On the one hand, we verify the feasibility of the DP-based location privacy protection strategy. On the other hand, we verify the superior performance of the RL-based task offloading strategy in reducing resource overhead. In addition, we also analyze the scalability and time complexity of the algorithm.

### A. EXPERIMENTAL ANALYSIS OF LOCATION PRIVACY PROTECTION STRATEGY BASED ON DP

In order to simulate the scenario of task offloading in MEC as realistically as possible, this paper selects the real dataset EUA to simulate the relationship between edge servers and users. We select a base station with a longitude of 144.96 and a dimension of -37.81 in the EUA as the edge server, its coverage is 400 meters, and there are about 70 mobile users within the coverage. Then we set the parameters of the MEC network. For the specific configuration, we refer to Chen *et al.* [33]. For the convenience of viewing, the network parameter settings are shown in TABLE 1.

To highlight the effectiveness of the DP-based user location privacy protection strategy, we selected two strategies for comparison with the strategy proposed in this paper, namely the Basic task offloading strategy without considering user privacy and the NDR task offloading strategy without considering the interference range. At the same time, we determined two indicators to measure the effectiveness of the strategy, namely the probability of privacy leakage, and the user's security level.

The security level of mobile users in each period of task offloading can be calculated by the following equation:

$$USL_n = PLD_n + \zeta USL_{n-1}, \tag{10}$$

where $\zeta$ is the security and privacy compromise factor of the user's continuous task offloading. $PLD_n$ is the probability of user privacy leakage, which can be calculated by equation 3.

FIGURE 4 indicates the privacy leakage probabilities of the three strategies. As revealed in the FIGURE 4, the probability of privacy leakage of the three strategies gradually

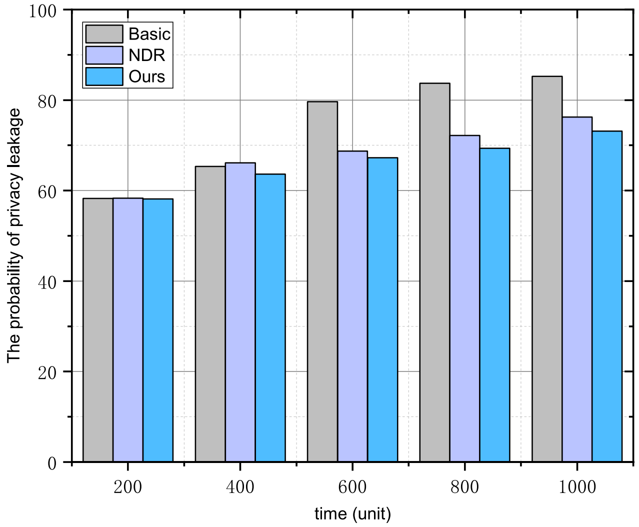


**FIGURE 4.** **The privacy leakage probability of three strategies.**

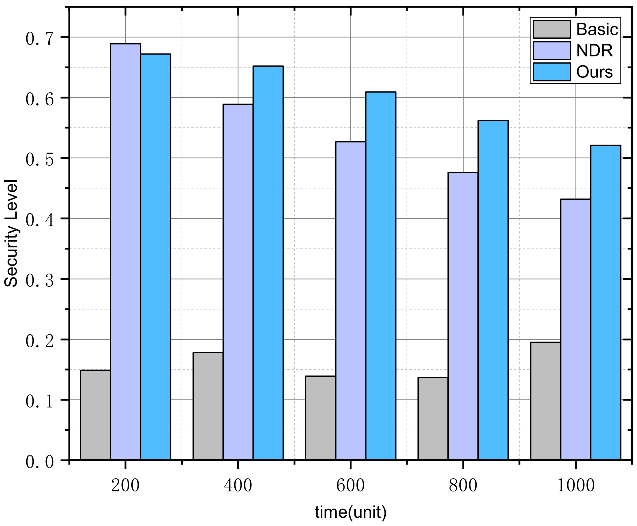


**FIGURE 5.** **The security level of three strategies.**

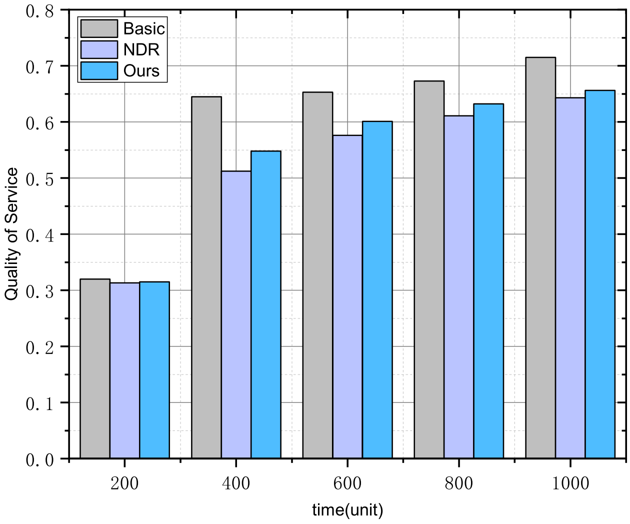


**FIGURE 6.** **The quality service of three strategies.**

increases, because as time goes on, users continue to send task offloading information to the edge server, and the edge server can infer user privacy from this information. Therefore, the probability of user privacy leakage increases with time. Compared with the Basic strategy without privacy protection, NDR and the strategy proposed in this paper are at a lower level, which means that interfering with user location information can effectively avoid user privacy leakage. In addition, it is worth noting that since the NDR does not limit the interference range, the edge server still has a certain probability to infer the user's location information, so the probability of privacy leakage is higher than the scheme proposed in this paper.

FIGURE 5 reveals the change in user security level over time. It can be clearly seen from the FIGURE 5 that the Basic task offloading strategy without privacy protection is relatively low in security. Privacy protection can significantly improve the security level of users and reduce the probability of privacy leakage. At the same time, compared with the NDR strategy that does not limit the interference range, the scheme proposed in this paper can stabilize the user's security level above 0.5, and the security level is improved by about 7%.

In addition, it is worth noting that privacy protection will affect the quality of service for users to a certain extent. Because privacy protection is invisible to users, the intuitive performance of privacy protection is the degradation of service quality. Therefore, we compare the service quality of the three strategies over time. As shown in FIGURE 6, the quality of service of the Basic strategy without privacy protection is higher than that of NDR and the task offloading strategy proposed in this paper, because privacy protection will bring additional delay overhead, such as transmission delay and privacy protection delay. Furthermore, because the NDR does not limit the interference range, the real distance of the user may be blurred too far, so the delay and bandwidth overhead will also increase accordingly.

$$\begin{bmatrix} CPU\left(n_1^{es}\right) & DEG\left(n_1^{es}\right) & SUM\left(n_1^{es}\right)_{BW} & SUM\left(n_1^{s}\right)_{DL} & Tru\left(n_1^{es}\right) \\ CPU\left(n_2^{es}\right) & DEG\left(n_2^{s}\right) & SUM\left(n_2^{es}\right)_{BW} & SUM\left(n_2^{s}\right)_{DL} & Tru\left(n_2^{es}\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ CPU\left(n_k^{es}\right) & DEG\left(n_k^{s}\right) & SUM\left(n_k^{es}\right)_{BW} & SUM\left(n_k^{es}\right)_{DL} & Tru\left(n_k^{s}\right) \end{bmatrix}. \tag{5}$$

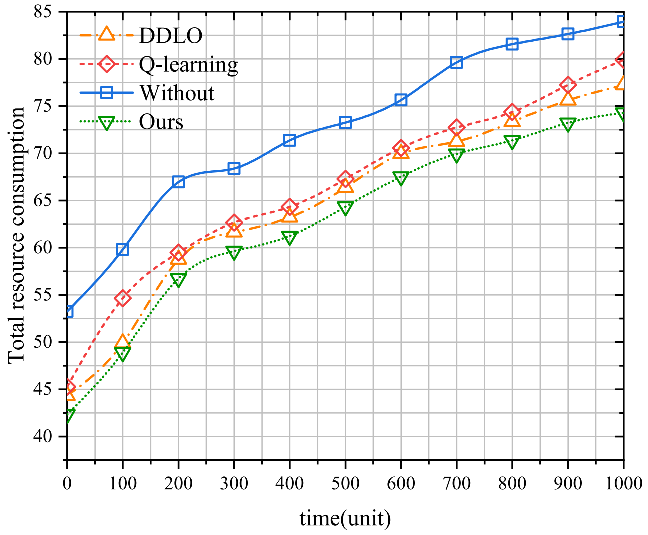


**FIGURE 7.** The resource consumption of the four task offloading strategies.

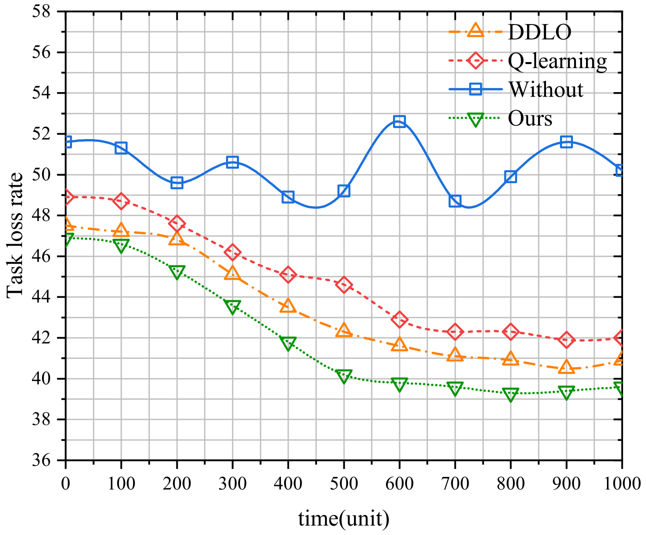


**FIGURE 8.** The task loss rate of four strategies.

**TABLE 2.** Scalability and time complexity analysis.

| Task offloading strategy | Security | Time complexity | Scalability |
|---|---|---|---|
| Ours | high | high | excellent |
| Q-learning | low | medium | medium |
| DDLO | low | medium | medium |
| Without | low | low | low |

### *B. RL-BASED EDGE SERVER SELECTION STRATEGY*

To verify the performance of the RL-based edge server selection strategy during task offloading. We compare it with the DDLO algorithm proposed in [34], the classical task offloading algorithm using Q-learning and the task offloading algorithm without resource optimization in terms of total resource consumption and task loss rate.

We use Pycharm and python to code the experiments, and build a policy network based on TensorFlow. The initial parameters of the policy network satisfy a normal distribution, and the learning rate of the agent is 0.005. Finally, we trained 100 epochs using gradient descent. The agent dynamically adjusts the current behavior according to formula 8 to obtain better training effect.

FIGURE 7 shows the total resource consumption of the four task offloading strategies. It is obvious that our proposed RL-based task offloading strategy is more advantageous than the other three algorithms in optimizing resource consumption. This is because our strategy uses the network resource environment of MEC as the input of the RL agent, and trains with the goal of minimizing resource overhead. Compared with other algorithms, it is more in line with the actual situation of task offloading in MEC, and pays attention to the change of network resources. According to the analysis of specific data, the solution proposed in this paper reduces resource consumption by about 9% compared with the Without strategy, 3.3% less than DDLO, and 6.3% less than Q-learning.

FIGURE 8 reveals the task loss rates of the four algorithms at different time periods. The task loss rate is the proportion of the number of locally executed tasks and the number of failed tasks to the total tasks, which is greatly affected by the channel state and delay. As indicated in FIGURE 8, the task loss rates of the other three strategies except the Without strategy decreased rapidly in the first 500 time points, and then gradually stabilized. This is because with the training iteration of the model, the model's understanding of the network environment gradually deepens, and then gradually stabilizes. According to the analysis of specific data, the strategy proposed in this paper reduces task offloading by 19.5% compared to Without, 6.8% lower than DDLO, and 11.4% lower than Q-learning.

### *C. SCALABILITY AND TIME COMPLEXITY ANALYSIS*

The scalability of task offloading strategy refers to the adaptability of the strategy when dealing with a large number of computing tasks. For a scalable strategy, the change of processing efficiency is stable when the amount of data increases. In addition, the scalability of task unloading strategy is closely related to its time complexity. Based on this, we analyze the scalability and time complexity of the DPRL strategy proposed in this paper. The analysis results are indicated in TABLE 2.

It can be seen from TABLE 2 that the DPRL strategy proposed by us has high time complexity, but good security and scalability. This is because compared with the other three algorithms, we first need to blur the user information and calculate the reputation of the edge server, so the time complexity is higher than other algorithms. However, in terms of scalability, DPRL has excellent performance. This is because our solution uses RL to solve the problem of task unloading, so the mobility is better. When there are new computing tasks, a good task offloading scheme can be obtained by using a simple pre-train, so it has better scalability.

According to the analysis of the above experimental results, the task offloading strategy based on DP and RL proposed in this paper can effectively solve the privacy leakage in the process of MEC task offloading and optimize the resource overhead.

## VI. CONCLUSION

The low latency and strong computing characteristics of MEC promote the development of the Internet of Things. However, due to the lack of consideration of security and resource overhead in the task offloading process, the existing task offloading strategies are gradually unable to meet the high quality of service requirements of users. To address these issues, we propose a task offloading strategy based on DP and RL. This strategy first encrypts the user's private information through DP to ensure security, and then uses RL to select appropriate edge servers for task offloading to optimize resource overhead in combination with MEC network resource status. Experimental results show that this scheme can effectively ensure the privacy of users and reduce the cost of task offloading. However, it should be noted that the network resource environment of MEC is often very complex in practical situations, and the time complexity of DPRL is high. So in future work, we will extract more reasonable network features, and strive to reduce the time complexity of DPRL. In addition, we will study how to meet the differentiated service quality requirements of users in MEC.

## REFERENCES

[1] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, "IRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7011–7024, Aug. 2019.

[2] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7635–7647, Oct. 2019.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[4] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, and G. Xu, "Constructing dummy query sequences to protect location privacy and query privacy in location-based services," *World Wide Web*, vol. 24, no. 1, pp. 25–49, Jan. 2021.

[5] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4049–4058, Jun. 2022.

[6] N. Ding, Y. Liu, and F. Farokhi, "A linear reduction method for local differential privacy and log-lift," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, VI, Australia, Jul. 2021, pp. 551–556.

[7] P. Zhang, P. Gan, G. S. Aujla, and R. S. Batth, "Reinforcement learning for edge device selection using social attribute perception in industry 4.0," *IEEE Internet Things J.*, early access, Jun. 11, 2021, doi: 10.1109/JIOT.2021.3088577.

[8] C. Wang, R. S. Batth, P. Zhang, G. S. Aujla, Y. Duan, and L. Ren, "VNE solution for network differentiated QoS and security requirements: From the perspective of deep reinforcement learning," *CoRR*, vol. abs/2202.01362, pp. 1–15, Feb. 2022.

[9] P. Zhang, C. Wang, C. Jiang, and A. Benslimane, "Security-aware virtual network embedding algorithm based on reinforcement learning," *CoRR*, vol. abs/2202.02452, pp. 1–11, Feb. 2022.

[10] S. Shen, L. Huang, H. Zhou, S. Yu, E. Fan, and Q. Cao, "Multistage signaling game-based optimal detection strategies for suppressing malware diffusion in fog-cloud-based IoT networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1043–1054, Apr. 2018.

[11] S. Shen, H. Zhou, S. Feng, L. Huang, J. Liu, S. Yu, and Q. Cao, "HSIRD: A model for characterizing dynamics of malware diffusion in heterogeneous WSNs," *J. Netw. Comput. Appl.*, vol. 146, Nov. 2019, Art. no. 102420.

[12] H. Zhou, T. Wu, H. Zhang, and J. Wu, "Incentive-driven deep reinforcement learning for content caching and D2D offloading," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2445–2460, Aug. 2021.

[13] K. Jiang, C. Sun, H. Zhou, X. Li, M. Dong, and V. C. M. Leung, "Intelligence-empowered mobile edge computing: Framework, issues, implementation, and outlook," *IEEE Netw.*, vol. 35, no. 5, pp. 74–82, Sep. 2021.

[14] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. M. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1517–1530, Jan. 2022.

[15] P. Zhang, C. Wang, G. S. Aujla, and R. S. Batth, "ReLeDP: Reinforcement-learning-assisted dynamic pricing for wireless smart grid," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 62–69, Dec. 2021.

[16] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[17] X. Xu, J. Liu, and X. Tao, "Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation," *IEEE Access*, vol. 5, pp. 16406–16415, 2017.

[18] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451–1455.

[19] T. Liu, S. Sheng, L. Fang, Y. Zhang, T. Zhang, and W. Tong, "Latency-minimized and energy-efficient online task offloading for mobile edge computing with stochastic heterogeneous tasks," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Tianjin, China, Dec. 2019, pp. 376–383.

[20] M. Kamoun, W. Labidi, and M. Sarkiss, "Joint resource allocation and offloading strategies in cloud enabled cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 5529–5534.

[21] X. Huang, K. Xu, C. Lai, Q. Chen, and J. Zhang, "Energy-efficient offloading decision-making for mobile edge computing in vehicular networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, p. 35, Dec. 2020.

[22] Y. Sun, T. Wei, H. Li, Y. Zhang, and W. Wu, "Energy-efficient multimedia task assignment and computing offloading for mobile edge computing networks," *IEEE Access*, vol. 8, pp. 36702–36713, 2020.

[23] B. Gu, M. Alazab, Z. Lin, X. Zhang, and J. Huang, "AI-enabled task offloading for improving quality of computational experience in ultra dense networks," *ACM Trans. Internet Technol.*, vol. 22, no. 3, pp. 1–17, Aug. 2022.

[24] Z. Xu, X. Liu, G. Jiang, and B. Tang, "A time-efficient data offloading method with privacy preservation for intelligent sensors in edge computing," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 236, Dec. 2019.

[25] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2622–2629, Apr. 2020.

[26] M. Min, X. Wan, L. Xiao, Y. Chen, M. Xia, Di Wu, and H. Dai, "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4307–4316, Jun. 2019.

[27] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2536–2549, Dec. 2020.

[28] A. H. Aljammal, H. Bani-Salameh, A. Qawasmeh, A. Alsarhan, and A. A. F. Otoom, "A new technique for data encryption based on third party encryption server to maintain the privacy preserving in the cloud environment," *Int. J. Bus. Inf. Syst.*, vol. 28, no. 4, pp. 393–403, 2018.

[29] M. Wang, Z. Ji, H.-E. Kim, S. Wang, L. Xiong, and X. Jiang, "Selecting optimal subset to release under differentially private M-estimators from hybrid datasets," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 573–584, Mar. 2018.

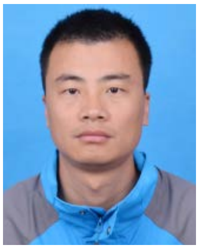
[30] Y. Wang, H. Zhang, and S. Su, "Enhancing location privacy for geolocation service through perturbation," in *Cloud Computing and Security* (Lecture Notes in Computer Science), vol. 11063, X. Sun, Z. Pan, E. Bertino, Eds. Haikou, China: Springer, Jun. 2018, pp. 502–511.

[31] J. Wang, Y. Wang, G. Zhao, and Z. Zhao, "Location protection method for mobile crowd sensing based on local differential privacy preference," *Peer-Peer Netw. Appl.*, vol. 12, no. 5, pp. 1097–1109, Sep. 2019.

[32] L. Zhang, H. Zhang, J. Yu, and H. Xian, "Blockchain-based two-party fair contract signing scheme," *Inf. Sci.*, vol. 535, pp. 142–155, Oct. 2020.

[33] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2015.

[34] L. Huang, X. Feng, A. Feng, Y. Huang, and L. P. Qian, "Distributed deep learning-based offloading for mobile edge computing networks," *Mobile Netw. Appl.*, vol. 23, pp. 1–8, Nov. 2018.



**PEIYING ZHANG** received the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, in 2019. He is currently an Associate Professor with the College of Computer Science and Technology, China University of Petroleum (East China). He has been publishing multiple IEEE/ACM TRANSACTIONS/journal/magazine articles, since 2016, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, *IEEE Network*, IEEE ACCESS, IEEE INTERNET OF THINGS JOURNAL, *ACM TALLIP*, *COMPUT COMMUN*, and *IEEE Communications Magazine*. His research interests include semantic computing, future internet architecture, network virtualization, and artificial intelligence for networking. He has served the Technical Program Committee for ISCIT 2016, ISCIT 2017, ISCIT 2018, ISCIT 2019, GLOBECOM 2019, COMNETSAT 2020, SoftIoT 2021, IWCMC-Satellite 2019, and IWCMC-Satellite 2020.

**PENG GAN** is currently pursuing the degree with the College of Computer Science and Technology, China University of Petroleum (East China). His research interests include virtual network embedding and mobile edge computing.

**LUNJIE CHANG** is currently a Professor-Level Senior Engineer at the Research Institute of Petroleum Exploration and Development, PetroChina Tarim Oilfield Company. He has published more than 40 papers in journals or conferences, such as *Xinjiang Petroleum Geology*, *Natural Gas Geoscience*, *Computing Techniques for Geophysical and Geochemical Exploration*, *Petroleum Exploration and Development*, *Earth Science Frontiers*, and *Oil and Gas Geology*. His research interests include geology, mining engineering technology, oil and gas field well development engineering, composite films, computer architecture, fracture, and other directions.

**WU WEN** received the Master of Science degree from the Huazhong University of Science and Technology, in 2009. He is currently an Associate Professor. He is engaged in teaching and scientific research at the School of Computer Science and Cyber Engineering, Guangzhou University. His main research interests include computer application, networks, and service computing.

**M. SELVI** received the B.E. degree in electronics and communication engineering and the M.E. degree in communication and network engineering from the Madras Institute of Technology and the Ph.D. degree from the Faculty of Information and Communication Engineering, College of Engineering, Anna University, Guindy Campus, Chennai, India. She is currently working as an Assistant Professor at VIT, Vellore Campus, India. She has published more than 20 papers in reputed journals and conferences. Her research interests include wireless sensor networks and data mining.

**GODFREY KIBALYA** received the B.Sc. degree in telecommunications engineering from Makerere University, Uganda, in 2010, and the M.Sc. degree in telecommunications engineering from the University of Trento, Italy. He is currently pursuing the Ph.D. degree with the Department of Network Engineering, Technical University of Catalonia (UPC). He is also an Assistant Lecturer with the Department of Electrical Engineering, Kabale University, Uganda. His research interests include network function virtualization and application of artificial intelligence in network and service management.