Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6622947, 16 pages
https://doi.org/10.1155/2021/6622947

Research Article

Privacy-Aware Online Task Offloading for Mobile-Edge Computing

Dali Zhu,[1,2] Ting Li,[1,2] Haitao Liu,[1,2] Jiyan Sun,[1] Liru Geng,[1] and Yinlong Liu[1,2]

[1]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Yinlong Liu; liuyinlong@iie.ac.cn

Received 1 December 2020; Revised 1 April 2021; Accepted 18 May 2021; Published 11 June 2021

Academic Editor: Yan Huang

Mobile edge computing (MEC) has been envisaged as one of the most promising technologies in the fifth generation (5G) mobile networks. It allows mobile devices to offload their computation-demanding and latency-critical tasks to the resource-rich MEC servers. Accordingly, MEC can significantly improve the latency performance and reduce energy consumption for mobile devices. Nonetheless, privacy leakage may occur during the task offloading process. Most existing works ignored these issues or just investigated the system-level solution for MEC. Privacy-aware and user-level task offloading optimization problems receive much less attention. In order to tackle these challenges, a privacy-preserving and device-managed task offloading scheme is proposed in this paper for MEC. This scheme can achieve near-optimal latency and energy performance while protecting the location privacy and usage pattern privacy of users. Firstly, we formulate the joint optimization problem of task offloading and privacy preservation as a semiparametric contextual multi-armed bandit (MAB) problem, which has a relaxed reward model. Then, we propose a privacy-aware online task offloading (PAOTO) algorithm based on the transformed Thompson sampling (TS) architecture, through which we can (1) receive the best possible delay and energy consumption performance, (2) achieve the goal of preserving privacy, and (3) obtain an online device-managed task offloading policy without requiring any system-level information. Simulation results demonstrate that the proposed scheme outperforms the existing methods in terms of minimizing the system cost and preserving the privacy of users.

1. Introduction

In the recent years, with the advent of 5G network, as well as the fast popularization of mobile devices, a myriad of new applications is emerging, such as augmented reality (AR)/virtual reality (VR) [1, 2], online 3D games [3], and connected cars [4]. Specifically, the recent Cisco Annual Internet Report expects that the number of global mobile devices will grow from 8.8 billion in 2018 to 13.1 billion by 2023 and the vast majority of mobile data traffic (99%) will originate from these mobile devices [5]. However, due to their limited computing units and battery energy, mobile devices struggle to resist to such traffic explosion and become unable to meet the stringent requirements of computing-demanding and latency-sensitive applications.

To get rid of such limitations, a novel paradigm of mobile edge computing (MEC) [6] is proposed as an extension of remote-centralized clouds [7] by the European Telecommunications Standards Institute (ETSI). The key idea beneath MEC is to deploy computing and storage resources from the core network to the radio access network (RAN) in the fifth generation (5G) networks [8]. In such computing paradigm, computation tasks will be offloaded to nearby MEC servers via wireless channels by mobile devices, which can meet the requirements of computing intensive applications and achieve ultrashort processing latency.

Despite the benefits, MEC still has shortcomings in terms of security and privacy leakage [9]. For example, the location privacy and usage pattern privacy problem [10] are investigated in this paper, which are related to the MEC task offloading feature. Intuitively, when a mobile device is to obtain optimal offloading performance, it tends to offload all its tasks to the MEC server. Accordingly, an honest-but-curious MEC server can infer the location privacy and usage

pattern privacy of users who are privacy sensitive, which may prevent these users from accessing the MEC system if not properly addressed. Although these two privacy issues have been extensively studied in other fields, one challenge still needs to be addressed in MEC systems, which is how to protect both the location privacy and usage pattern privacy while minimizing the delay and energy consumption cost.

Most existing task offloading schemes that want to achieve optimal system performance, such as [11, 12], largely ignore these privacy problems. And current privacy-preserving techniques of cloud computing are not always applicable for the MEC system, such as the works in [13, 14]. Therefore, the more challenging problem is how to prevent unintentional leakage of user's privacy while still maintaining the optimal delay and energy consumption performance. The most related works probably are [10, 15], which studied the optimization of delay and energy consumption cost while considering both location privacy and usage pattern privacy. The former scheme formulates this problem as a constrained Markov decision process (CMDP), and the latter one applies a Dyan-Q architecture based on the CMDP to achieve a better privacy-aware offloading policy. However, both of them are system-level solutions. They rely on the assumption of the wireless channel power gain that is formulated as a Markov chain model, in which some system-side information should be known in advance. Such assumption is not relevant to infrastructure-free scenarios, such as individual combat in military scenarios, forest fire rescue [16], and heterogeneous IoT [17] which are more applicable user-level schemes.

In order to minimize system cost (e.g., latency and energy consumption) and protect user's privacy without requiring any system-level information as a prior knowledge, we propose a device-level and privacy-preserving task offloading scheme for the MEC system. This scheme is based on a semiparametric contextual multi-armed bandit (MAB) problem, which can address the trade-offs inherent in the sequential decision problem and overcome the challenges of lacking system-side information. To the best of our knowledge, this user-level scheme is the first to be proposed to solve the privacy problem of MEC. The location privacy and usage pattern privacy of users will be preserved in this paper. An online-learning algorithm will be proposed to make adaptive task offloading decisions under dynamic network environment. The main contributions of this paper are summarized as follows:

(1) *MAB-based problem modelling.* We study a joint optimization problem of task offloading and privacy preservation in the MEC system. And then, this problem is transformed as a semiparametric contextual MAB problem to overcome the challenge of unknown network dynamics, which can utilize the contextual feature vector to describe user-side information for decoupling the time dependency

(2) *Privacy-aware optimal offloading decision.* We propose a privacy-aware online learning algorithm, called PAOTO (privacy-aware online task offloading), to make device-level task offloading decisions while protecting user's location privacy and usage pattern privacy. By utilizing the transformed Thompson sampling (TS) architecture, we can make adaptive task offloading decisions at the user-side perspective

(3) *Extensive simulation-based performance evaluation.* We carry out simulations to demonstrate the effectiveness of the proposed algorithm. The results show that the PAOTO algorithm performs close-to-optimal and far better than the newly proposed Dyna-Q algorithm in [15].

The remaining parts of this paper are organized as follows. In Section 2, we discuss the motivation and related works. We will formally describe the system model and problem formulation in Section 3. Next, we present the algorithm design and simulation evaluation in Sections 4 and 5, respectively. Finally, we draw some conclusions and highlight the direction for future work in Section 6.

2. Motivation and Related Work

2.1. Motivation. The problem of determining the privacy-aware and user-level task offloading decisions for mobile devices requires solving two important challenges: (1) how to best prevent leakage of user's privacy while still maintaining the optimal delay and energy consumption performance and (2) how to design a task offloading policy that can online determine the optimal execution platform (i.e., local processing unit, MEC servers, or buffer) for users at the user-side perspective?

To address the first challenge, we propose a privacy metric to jointly quantify the location and usage pattern privacy and utilize a semiparametric MAB to incorporate the privacy metric into the performance model. This can strike a balance between the privacy-preserving level and system cost (e.g., processing latency and energy consumption cost). Previous works require system-level information to design an optimal task offloading strategy, but this is not applicable to infrastructure-free scenarios (e.g., individual combat in military scenarios, forest fire rescue and heterogeneous IoT). We address this second challenge by utilizing the contextual feature vector in the contextual MAB model to describe user-side information and applying the Thompson sampling (TS) algorithm to estimate and learn the performance model based on the contextual information.

2.1.1. Characterizing Privacy Metric. Recently, MEC has been increasing in popularity but issues relating to the security and privacy in the MEC system still has shortcomings. On one hand, some security issues such as authentication, private data storage, and intrusion detection have received attentions but these security issues are inherited from the conventional cloud computing framework and are less relevant to the key technologies in the MEC system. On the other hand, based on the simulation results and considered setting in [10], we can find that the privacy problems relating to MEC unique wireless task offloading technology remains less explored, which are user's location privacy and usage pattern privacy.

According to [10], the offloading pattern can be observed as a mobile device may offload all its tasks to the MEC server when the wireless channel state is good, while it will not offload any tasks otherwise. Accordingly, the honest but curious MEC server (it may be controlled by adversary) can be based on the offloading pattern and historical statistics to obtain the number of tasks offloaded in each period. Hence, the wireless channel condition (it is only assumed as good or bad and can be extended to the multistate case) and user's actual usage pattern can be inferred by adversary. Specifically, the wireless channel condition is highly related to the distance between the user and the MEC server. If a mobile user communicates with multiple MEC servers, the location privacy may be inferred by these MEC servers based on the surveillance of the wireless channel state. The user's location privacy is leaked. Moreover, when someone's office is near the AP, its wireless channel state may be always good and it may always offload all its tasks to the MEC server. Thus, the adversary can obtain the total number of tasks offloaded that is determined by the user's actual device usage pattern. The user's usage pattern privacy may be leaked.

Particularly, it is very important for privacy-sensitive users to solve the problem of leaking location privacy and usage pattern privacy that are induced into the unique wireless task offloading feature in the MEC system. If they are note properly addressed, it may prevent these privacy-sensitive users from accessing the MEC system. Significantly, although these two privacy problems have already been studied in other system, protecting user's location privacy and usage pattern privacy while minimizing delay and energy consumption cost in the MEC system still poses a critical challenge.

Therefore, to address this challenge, it is desirable to design a metric to jointly quantify the location and usage pattern privacy. Next, we formulate the task offloading and privacy preservation problem as a contextual MAB problem with a semiparametric reward model based on processing latency, energy consumption cost, and this privacy metric. This is aiming to strike a balance between the privacy-preserving level and system cost. That is, according to problem formulation and proposed algorithm, we can obtain the optimal delay and energy consumption performance while protecting user's location and usage pattern privacy, which can be seen in Sections 3 and 4.

2.1.2. User-Level Task Offloading. With mobile data traffic growing explosively, the mobile devices with limited resources cannot meet the stringent requirements of computing-demanding and latency-sensitive applications. Therefore, designing a desirable task offloading strategy of the MEC system has attracted tremendous attention in the industry and academia. This strategy can determine the optimal task execution platform for the user, executing in the local processing unit, offloading to MEC server or queueing in the buffer.

Many previous works (e.g., [10, 15]) on task offloading generally assume that the system-side information is always available. Such assumption is more applicable to the infrastructure-assisted edge computing scenarios where the infrastructure (e.g., an access point or base station) is available for obtaining system-side information in advance [18]. However, some infrastructure-free scenarios, such as individual combat in military scenarios, forest fire rescue and heterogeneous IoT, are not suitable for previous system-level solutions, because these mobile devices in infrastructure-free scenarios are operating in a scattered manner and the system-side network information is missing for them. Especially, if they want to explore system-level information in advance, it may cause additional system cost, such as scarce bandwidth usage and additional energy consumption cost.

In this case, it is desirable to design a user-level task offloading strategy for overcoming the challenge of lacking the system-side information. In response to the challenge that the system-level information may not be readily available in some infrastructure-free scenarios, we propose an online task offloading scheme at the user perspective. In this scheme, the user-side information will be described as contextual feature vector and the Thompson sampling (TS) algorithm will be applied to estimate and learn the performance model based on the contextual information. It can adaptively decide where to execute the offloaded task for the mobile user without any system level information. It can be seen in Section 4 for details.

2.2. Related Work. In recent years, the task offloading strategies have attracted significant efforts to minimize total delay and energy consumption cost in MEC systems. For example, Xu et al. proposed an online algorithm based on Lyapunov optimization and Gibbs sampling, which jointly optimized dynamic service caching and task offloading to reduce computation latency while keeping energy consumption low [19]. Wei et al. studied the problem of task offloading and channel resource allocation based on MEC in 5G ultra-dense networks (UDN) [20]. The authors formulated task offloading as an integer nonlinear programming problem and proposed an efficient task offloading and channel resource allocation scheme based on differential evolution algorithm. Dab et al. proposed a joint radio resource allocation and task assignment strategy based on a Q-learning algorithm to minimize the energy consumption cost under both the latency and device's computation resource constraints [21]. Li and Cai discussed the incentive mechanism design for collaborative task offloading in the MEC network [22]. They proposed an online truthful mechanism integrating computation and communication resource allocation to address social welfare maximization problem by considering each task's specific requirements in terms of data size, delay, and preference. However, none of the works mentioned above considered user's privacy issues.

There are a few works considering both task offloading and privacy preservation. For example, He et al. identified a new privacy vulnerability caused by the wireless offloading feature of MEC-enabled IoT. To address this vulnerability, the authors developed an offloading strategy for MEC-enabled IoT, which can learn a good offloading strategy while protecting the devices' location privacy [23]. However, the extra prior information was required. In [24], Zhang et al. proposed a strategy that can achieve an efficient task

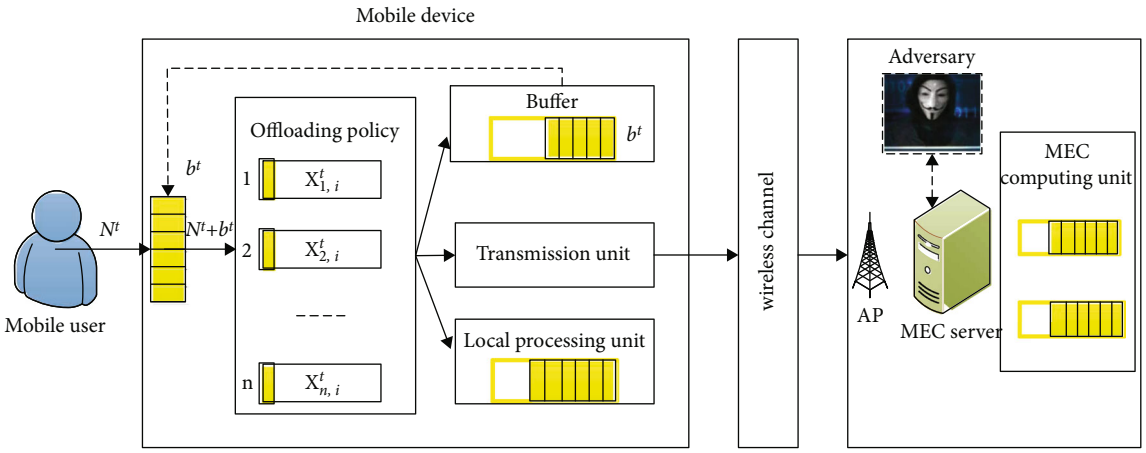

Figure 1: An illustration of task offloading in the MEC system.

scheduling policy on edge while ensuring privacy. In [25], Zhou proposed a novel context-aware task allocation framework for mobile crowdsensing in the scenario of edge computing. The task allocation was performed in both the cloud computing layer and the edge computing layer. In the cloud layer, authors proposed a privacy-preserving and contextual online learning algorithm to manage the participants' reputation. But this scheme was implemented at the system level and required a priori network information.

Besides, He et al. identified location privacy and usage pattern privacy issues, which are induced by the wireless task offloading feature of MEC [10]. To address these privacy issues, authors proposed a constrained Markov decision process- (CMDP-) based privacy-aware task offloading scheduling algorithm to achieve the best possible system performance while protecting user's privacy. Min and Wan proposed a reinforcement learning- (RL-) based privacy-aware offloading scheme, which enables the IoT device to make the task offloading decisions and protect both the user location privacy and the usage pattern privacy for the MEC system [15]. Nevertheless, both the works were implemented at the system level. They all need to explore system-level information in advance, which is difficult to obtain and may cause additional system cost, such as scarce bandwidth usage and energy consumption of network devices.

In general, none of the aforementioned works consider both task offloading and privacy protection problem at the user level. These aforementioned studies mainly face two challenges. First, they only consider the simple task offloading strategies for minimizing total delay and energy consumption cost in MEC systems. However, the privacy issues related to the task offloading pattern were ignored in their works, which may be very important for the privacy-sensitive users. Second, some works considering both task offloading and privacy preservation were all implemented at the system level. That is, these works generally assume that the system-side information is always available. However, this is applicable for the infrastructure-assisted scenarios where the infrastructure (e.g., an access point (AP)) is available for obtaining system-side information in advance. For the infrastructure-free scenarios (e.g., individual combat in military scenarios, forest fire rescue, and heterogeneous IoT), these mobile devices will operate in a decentralized manner and the system-side information is difficult to obtain and may cause additional system cost.

To conquer this challenge, we propose a novel privacy-aware task offloading scheme based on an online learning algorithm that just requires device-level information and it can achieve the best possible system performance while protecting the user's privacy.

3. System Model and Problem Formulation

3.1. System Model. In this section, the task offloading model will be presented. As illustrated in Figure 1, we consider a scenario in which the mobile user/device communicates with the MEC server through the access point (e.g., Wi-Fi or 5G base station) via the wireless channel. For ease of exposition, the bandwidth constraint of the wireless channel is not considered in this paper and we will consider it in the next work. The mobile device has computing-intensive computation tasks that are required to be completed as soon as possible. Due to its limited battery energy and computing capabilities, the mobile device can offload some computation tasks to the MEC server, which has powerful computing capabilities. As such, the mobile device has three ways to process these computation tasks, that is, computing in the local processing unit, offloading to the MEC server through the transmission unit, and queuing in the buffer for processing in the next time slot.

Without loss of generality, we assume that the task offloading policies are made in a slotted structure and its timeline is discretized into time slots $t \in \mathcal{T} = \{1, 2, \cdots, T\}$. At each time slot t, the mobile user will newly generate N^t computation tasks to the mobile device, denoted by a set of $\mathbb{N} = \{1, 2, \cdots, N^t_{\max}\}$ ($N^t_{\max}$ is the maximum possible number of generated tasks), which depends on the user's usage pattern. And the $b^t \in \mathcal{B} = \{1, 2, \cdots, b_{\max}\}$ (with the maximum buffer size $b_{\max}$) can be denoted as the number of tasks in the buffer at time slot t.

A widely used three-parameter model [26] can be used to describe each task n, denoted by a set of $\mathcal{N}^t = \{1, 2, \cdots, (N^t$

$+ b^t)\}$. The three-parameter model consists of the input data size λ_n (bits), computation intensity δ_n (CPU cycles/bit), and maximum allowed latency τ_n (seconds). Whereupon, the computation demand for each task can be obtained by $m_n = \lambda_n * \delta_n$ (CPU cycles). In each time slot t, all the $(N^t + b^t)$ tasks (including the newly generated N^t tasks and the b^t tasks in the buffer) will be either locally executed, buffered, or remotely offloaded according to the proposed task offloading policy. More specifically, the mobile device will explore the optimal offloading policy for each task.

Based on [10], in order to minimize computing delay and energy consumption, the mobile device tends to offload all its tasks to the MEC server if the wireless channel state is good and processes all its tasks locally if the wireless channel state is bad. Under such circumstances, the user's location and usage pattern privacy are easily spied by the attacker. Hence, the proposed task offloading policy in this paper takes the privacy preservation into account. And the wireless channel power gain will not be assumed as the Markov model, which allows the proposed task offloading policy to be executed on the device level. More explicitly, the mobile device can only observe its local information (e.g., the number of tasks and the computation demand of each task) but the system-side information is not observable. Key parameter notations in this paper are listed in Table 1 for ease of reference.

3.2. Problem Formulation. We focus on privacy-aware and user-level task offloading optimization problems in this paper. In this section, we firstly formulate the task offloading decision making and then the model system cost (including processing latency and energy consumption cost) and privacy level as the performance metrics. Finally, the objective function will be presented.

3.2.1. Task Offloading Decision Making. To maintain satisfactory quality of service, an available and reliable task offloading policy should be considered. And the tasks can be dynamically offloaded to the three different positions i by the mobile device, denoted by $\mathcal{I} = \{l, s, b\}$, where l, s, and b represent the local processing unit, MEC server, and buffer, respectively. At each time slot t, the mobile device (also called the earner or operator) makes the task offloading decision for each task n. Here, we design a binary indicator $x_{n,i}^t$ to denote the dynamic task offloading decision variable; let $x_{n,i}^t = 1$ if the task $n \in \mathcal{N}^t$ is offloaded to platform $i \in \mathcal{I}$ at time slot t and $x_{n,i}^t = 0$ otherwise. Note that at a given time slot t, each task n can be offloaded to only one execution platform (l, s, or b). We have the following constraints for $x_{n,i}^t$:

$$x_{n,i}^t \in \{0, 1\}, \quad \forall t, n, i, \tag{1}$$

$$x_{n,l}^t + x_{n,s}^t + x_{n,b}^t = 1, \quad \forall n, t, \tag{2}$$

$$\sum_{n \in \left(\mathcal{N}^t \cup \mathcal{B}\right)} \sum_{i \in \mathcal{I}} x_{n,i}^t = N^t + b^t, \quad \forall t. \tag{3}$$

Equation (1) indicates that whether offloading the task n in platform i in the time slot t. Equation (2) indicates that only one of $x_{n,l}^t$, $x_{n,s}^t$, and $x_{n,b}^t$ for task n in the time slot t can be nonzero. Equation (3) indicates that the $(N^t + b^t)$ tasks will be offloaded to execution platform i in the time slot t. Based on the above definition, the system cost model (including processing latency and energy consumption) and privacy model will be further described.

Table 1: List of main parameter notations.

Notation	Definition
N^t	The number of newly generated tasks at time slot t
$n \in \mathcal{N}^t$	Tasks to be offloaded
λ_n	Input data size of task n (bits)
δ_n	Computation intensity of task n (CPU cycles/bit)
m_n	Computation demand for task n (CPU cycles)
$b^t \in \mathcal{B}$	Tasks in the buffer
$i \in \mathcal{I} = \{l, s, b\}$	Offloading execution platforms (local, MEC, and buffer)
γ_i	Available computing capability in execution platform i
$d_{n,i}^t$	Processing delay of each task
D^t	Total processing delay for processing all tasks
$e_{n,i}^t$	Energy consumption of each task
E^t	Total energy consumption
q^t	The number of tasks offloaded to the MEC server
ξ	Weighting factor of location privacy
K	The metric of the usage pattern privacy
Δ^t	The difference between N^t and q^t
P^t	Privacy metric

3.2.2. System Cost Model. Similar to [27, 28], we consider a system cost that accounts for processing latency and energy consumption cost, which are associated with the task offloading. They depend on both the tasks and the processing platforms where the tasks are computed.

(1) Processing Latency. In this paper, total processing latency consists of three parts, i.e., queuing delay in buffer, computing delay in either the local processing unit or the MEC server. For ease of exposition, we assume that the queuing delay in the buffer can be converted to computing delay and the buffer can be treated as a microprocessor, which has much lower computing capability than the local processing unit. In our system, each task $n \in (\mathcal{N}^t \cup \mathcal{B})$ will be offloaded to execution platform $i \in \mathcal{I}$ by the mobile device in time slot t. We use γ_i to denote the available computing capability (i.e., CPU cycles per second) of execution platform i for task processing at time slot t. Then, the processing delay $d_{n,i}^t$ of each task n can be expressed as follows:

$$d_{n,i}^t = \frac{m_n^t}{\gamma_i}, \tag{4}$$

where m_n^t is the computation demand of each task n at time slot t. Therefore, given the task offloading decision $x_{n,i}^t$, the total processing latency required to process $(N^t + b^t)$ tasks within time slot t can be further expressed as follows:

$$D^t = \sum_{n=0}^{N^t+b^t} \sum_{i \in \mathcal{I}} d_{n,i}^t x_{n,i}^t = \sum_{n=0}^{N^t+b^t} \sum_{i \in \mathcal{I}} \frac{m_n^t}{\gamma_i} x_{n,i}^t. \tag{5}$$

(2) Energy Consumption. Task offloading will consume the energy of the mobile device, whose battery storage capacity is rather limited. Thereby, further investigation on how to minimize the total energy consumption of the mobile device is one of the objectives of this paper. The energy consumption cost of the mobile device may include the CPU cycles, transmitting energy and electric energy. They are associated with the tasks executed in the local processing unit, offloaded to the MEC server, and queued in the buffer. To better characterize these energy consumption costs, we let the $e_{n,i}^t$ be the energy consumption in time slot t for offloading task n to execution platform i ($i \in \mathcal{I} = \{b, l, s\}$). Thus, when considering the task offloading decision $x_{n,i}^t$, the overall energy consumption at time slot t can be expressed as follows:

$$E^t = \sum_{n=0}^{N^t+b^t} \sum_{i \in \mathcal{I}} e_{n,i}^t x_{n,i}^t, \tag{6}$$

3.2.3. Privacy Model. As more and more people enjoy the benefits of MEC, the location privacy and usage pattern privacy of MEC have become a major concern. According to the simulation results and considered setting in [10], we can observe the offloading pattern that the mobile device may offload all its tasks to the MEC server when the wireless channel state is good, while it will not offload any tasks otherwise. For simplicity, it is assumed that the wireless channel states are only good and bad in this work. It can be extended to the multistate case. The wireless channel gain is highly related to the distance between the user and the MEC server. Thus, the honest-but-curious MEC server (it may be controlled by adversary) can infer not only the wireless channel state but also the distance to the mobile device based on the offloading pattern and historical statistics.

Accordingly, when the mobile device communicates with multiple MEC servers, its location information may be jointly inferred by these MEC servers. Besides, if a mobile device always maintains a good channel state (e.g., its office near the base station), it will always offload all its tasks to the MEC server. The total number of tasks is highly related to the user's usage pattern (i.e., user's app running if a certain pattern exists in the number of tasks generated by the app), which may be very important for the privacy-sensitive users. Hence, the MEC server may be able to infer the personal information of the user through monitoring the total number of offloading tasks and analyzing the historical statistics.

Hence, from the privacy perspective, we propose a metric to jointly quantify the location and usage pattern privacy and strike a balance between the privacy-preserving level and system cost. Firstly, the total number of tasks offloaded to the MEC server at the end of time slot t is defined as q^t and we have

$$q^t = \sum_{n=0}^{N^t+b^t} x_{n,s}^t, \quad \forall t. \tag{7}$$

Then, the privacy metric of P^t can be obtained by

$$P^t = \mathbb{I}\left(\Delta^t = 0\right) \cdot K + \mathbb{I}\left(\Delta^t \neq 0\right) \cdot \left[\mathbb{I}\left(q^t = 0\right) \cdot \xi + \mathbb{I}\left(q^t \neq 0\right) \cdot \frac{\widehat{\xi}}{\Delta^t}\right], \tag{8}$$

where the $\mathbb{I}$ represents the indicator function that equals 1 if the statement is true and 0 otherwise; Δ^t indicates the difference between N^t and q^t, and it has $\Delta^t = |N^t - q^t|$; ξ and $\widehat{\xi}$ are the weighting factors reflecting the importance of the location privacy over the usage pattern privacy in different situations; $K \in [1, N^t]$ denotes the metric of usage pattern privacy, which is the number of dummy tasks. The dummy tasks may sacrifice some system performance but will increase the privacy level, and the proposed algorithm will balance them.

The first term of equation (8) represents that if the mobile device offloaded all its tasks to the MEC server ($\Delta^t = 0$), in order to protect the usage pattern privacy, it will continue to offload K dummy tasks to the MEC server to confuse the attacker. As such, the attacker cannot pinpoint the number of tasks actually generated by the user. According to the second term of equation (8), there are two situations correspond to the $\Delta^t \neq 0$. In the first situation, $q^t = 0$ denotes that the tasks either queued in the buffer or processed locally otherwise. In order to protect the location privacy, the mobile device needs to offload ξ tasks (which is queuing in the buffer, $0 \leq \xi \leq b^t$) to the MEC server for preventing the attacker from inferring the wireless channel status. In the second situation of $\Delta^t \neq 0$, some tasks are offloaded to the MEC server ($q^t \neq 0$) and the privacy level can be achieved by $\widehat{\xi}/\Delta^t$. It denotes the importance of the location privacy over the usage pattern privacy $\widehat{\xi}$, which will increase as Δ^t decreases.

3.2.4. Objective Function. In order to achieve a desirable trade-off between the system cost (i.e., computing delay and energy consumption) and the user's privacy level, we design different weights ω_{delay}^t, ω_{energy}^t, and $\omega_{\text{privacy}}^t$ to indicate the different preference device. These weights also can convert the privacy level and system cost into the same dimension. Thus, the objective of this paper is to achieve robust minimization of a weighted sum of the privacy level and system cost for the mobile device. Based on [29], given a finite time horizon T, the problem can be formulated as

$$\begin{aligned} &\min \sum_{t=0}^{T} \omega_{\text{delay}}^t D^t + \omega_{\text{energy}}^t E^t + \omega_{\text{privacy}}^t \frac{1}{P^t}, \\ &\text{s.t.} (1) - (3), \end{aligned} \tag{9}$$

From the mobile device perspective, it is difficult for them to explore the system-wide information (e.g., the wireless channel states and resource availability) in advance and it may need extremely expensive energy cost. Therefore, devising a device-level adaptive privacy-preserving task offloading policy is highly desirable, in which the future system-level information will not be needed.

4. Algorithm Design

In this section, we focus on the privacy preserving task offloading problem in the MEC-enabled network and propose a device-level privacy-aware online learning scheme to minimize the objective in equation (9) for the mobile device without knowing the system-side information.

Firstly, we transform the information-constrained multi-objective optimization problem to a contextual multi-armed bandit (MAB) problem [30] with a semiparametric reward model. Then, we propose a privacy-aware online task offloading (PAOTO) algorithm which can accommodate the network dynamics at the device level and learn the optimal offloading policy for the mobile device while maintaining the user's privacy.

4.1. Problem Transformation. In this work, we focus on the device-level and privacy-aware task offloading problem, which is a typical sequential decision problem. For decoupling the time dependency, we formulate this problem as a contextual MAB problem with a relaxed, semiparametric reward model in [30]. It is an extended version of the conventional contextual MAB that has a linear reward model [31]. Both versions can utilize the contextual feature vector to indicate the use-side information for overcoming the challenges of lacking future system information. However, why we use the contextual MAB with a relaxed, semiparametric reward model is that the privacy metric in our model is difficult to formulate as a linear reward model. The semiparametric reward model can provide a more relaxed reward model, and this proof can be found in the literature [30].

Accordingly, in order to learn the network dynamics and take the privacy protection into consideration, the problem in this paper can be transformed as a semiparametric contextual MAB problem [13], which can address the tradeoffs inherent in the sequential decision problem and has a relaxed, semiparametric reward model. This model can be described as

$$\mathbb{E}\left(\sum_{n\in\mathcal{N}^t} r_{n,i}(t) \mid \mathcal{F}^{t-1}\right) = \nu(t) + \sum_{n\in\mathcal{N}^t} b(t)^\top \bar{\mu}_{n,i}(t), \tag{10}$$

where $r_{n,i}(t)$ is the received cost of offloading task n to execution platform i; $\nu(t)$ is a nonparametric component; $b(t)$ is a current contextual feature vector; $\bar{\mu}_{n,i}(t)$ is a fixed but unknown underlying expectation of the feature vector $\mu_{n,i}(t)$; the $\mathcal{F}^{t-1}$ is the union of historical information and $b(t)$. Furthermore, it has assumptions about the upper bound of some parameters, which is $\|b(t)\|_2 \le 1$, $\|\bar{\mu}_{n,i}(t)\|_2 \le 1$, $\|\nu(t)\|_2 \le 1$, and $\|\cdot\|_2$ denotes the L_2 norm.

When the computation tasks arrive, the task offloading decision can be executed for each task by the mobile device. Nonetheless, only the device-side status information can observable, which can be described as a contextual feature vector $b(t) = [\omega^t_{\text{delay}}\mathbb{M}^t, \omega^t_{\text{energy}}\mathbb{S}^t] \in \mathbb{R}^{(2(N^t_{\max}+b_{\max}))}$ for arriving tasks. More specifically, $\mathbb{M}^t \in \mathbb{R}^{(N^t_{\max}+b_{\max})}$ denotes the computation demand vector of tasks. The first $N^t + b^t$ values of $\mathbb{M}^t$ are corresponding computation demand m^t_n of each task n, and the remaining values are 0; $\mathbb{S}^t \in \mathbb{R}^{(N^t_{\max}+b_{\max})}$ is a transition vector, which denotes the number of tasks in time slot t. The first $N^t + b^t$ values of $\mathbb{S}^t$ are 1, and the remainder are 0. According to the system cost (including processing latency and energy consumption cost) defined in Section 3, we transform them as a feature vector $\mu_{n,i}(t) = [(1/\gamma^t_i), e^t_{n,i}] \in \mathbb{R}^{(2(N^t_{\max}+b_{\max}))}$ to better learn the network uncertainty and resource availability, which is related to $\bar{\mu}_{n,i}(t)$. Besides, the privacy level in this task offloading policy will be formulated as the aggregated nonparametric component $\nu(t) = \omega^t_{\text{privacy}}(1/P^t)$based on the reward model of contextual MAB in equation (10). The reason of this is that it cannot be directly formulated as a linear component like other metrics (such as computing delay and energy consumption). We assume that it can be calculated when all task decisions are completed at the end of t.

Hereinafter, we define $\mathcal{H}^{t-1} = \{\mathcal{A}(\tau), r_{\mathcal{A}(\tau)}(\tau), b(\tau)\}$, $\tau = \{1, 2, \cdots, t-1\}$ as the historical observations until $t-1$, where $\mathcal{A}(\tau)$ represents the set of actions for all tasks at time slot τ and $r_{\mathcal{A}(\tau)}(\tau)$ denotes the total received cost at time slot τ. And the $\mathcal{F}^{t-1} = \{\mathcal{H}^{t-1}, b(t)\}$ can be denoted as the union of historical information $\mathcal{H}^{t-1}$ and the current contextual feature vector $b(t)$. Given that $\mathcal{F}^{t-1}$, we assume that the expectation of the total received cost $r_{\mathcal{A}(t)}(t)$ can be decomposed into a time-invariant linear component $\sum_{n=0}^{N^t+b^t} b(t)^\top \bar{\mu}_{n,\mathcal{A}(t)}(t)$ (associated with processing delay and energy consumption cost) and a nonparametric component $\nu(t)$ (associated with the privacy-preserving level). Therefore, according to equation (10), we have

$$\begin{aligned}\mathbb{E}\left(r_{\mathcal{A}(t)}(t) \mid \mathcal{F}^{t-1}\right) &= \mathbb{E}\left(\sum_{n=0}^{N^t+b^t} r_{n,i}(t) \mid \mathcal{F}^{t-1}\right) \\ &= \nu(t) + \sum_{n=0}^{N^t+b^t} b(t)^\top \bar{\mu}_{n,i}(t) \\ &= \omega^t_{\text{privacy}} \frac{1}{P^t} + \sum_{n=0}^{N^t+b^t} b(t)^\top \bar{\mu}_{n,i}(t).\end{aligned} \tag{11}$$

The task offloading scheme needs to select an execution platform i (or called an arm at MAB) for every task n at time slot t. Specifically, we let $a_n(t) \in \mathcal{A}(t)$ denote the choice for every task and let the optimal action to be $a^*_n(t) \in \mathcal{A}^*(t)$ based on equation (10). Additionally, it must be noted that the nonparametric component $\nu(t)$ in equation (11) depends on time and historical information, but not on the current action [30]. Hence, the optimal received cost $r_{n,i}(t)$ of each

task can be obtained by minimum $b(t)^\top \bar{\mu}_{n,i}(t)$ and we can achieve the optimal offloading decision of each task by $a_n^*(t) = \text{argmin}_i b(t)^\top \bar{\mu}_{n,i}(t)$. Indeed, the privacy level $\nu(t)$ will have an impact on the aggregated received cost of all tasks at the end of time slot t and this aggregated received cost will be used to update the contextual feature vector for the next interval t.

Beyond that, the regret at time slot t is defined as the difference between the average cost of the optimal choices and the universal choices for all tasks and it does not depend on $\nu(t)$ either. Hence, the regret can be expressed as

$$\text{Regret}(t) = b(t)^\top \bar{\mu}_{\mathscr{A}^*(t)} - b(t)^\top \bar{\mu}_{\mathscr{A}(t)}. \tag{12}$$

Moreover, given a finite time horizon T, the total regret can be described as

$$R(t) = \sum_{t=1}^{T} \text{regret}(t) = \sum_{t=1}^{T} b(t)^\top \bar{\mu}_{\mathscr{A}^*(t)} - b(t)^\top \bar{\mu}_{\mathscr{A}(t)}. \tag{13}$$

This regret is used to evaluate the effectiveness of task offloading decision making based on the online learning of system-level information.

4.2. Privacy-Aware Online Task Offloading Algorithm. In order to minimize the total system cost (e.g., delay and energy consumption cost) and protect user's privacy without exploring any system-level information, a novel PAOTO algorithm is proposed in this work. In particular, this algorithm keeps the framework of the Thompson sampling (TS) with a semiparameter reward model [32]. Its key idea is to estimate and learn the device's performance by selecting different actions over time based on the contextual information. At the same time, the privacy metric can be abstracted into the semiparametric reward model.

In the proposed PAOTO algorithm, the mobile device will learn the network information while executing the task offloading policy. As time goes by, the mobile device learns abundant information and it can estimate how to offload these $(N^t + b^t)$ tasks for achieving the optimal system cost and privacy-preserving level. According to the aforementioned MAB transformation of our problem, it is known that the optimal offloading decision and the received cost $r_{n,i}(t)$ of each task mainly depend on the current contextual feature vector $b(t)$ and the fixed but unknown feature vector $\bar{\mu}_{n,i}(t)$. Through the previous trial and error, the underlying relationship between the feature vectors and received cost will be learned by the mobile device. The $\nu(t)$ is related to the privacy metric P^t, and it can be achieved after all tasks are offloaded at the end of time slot t. Hence, we let $\widehat{\mu}_{n,i}(t)$ denote the estimate of the feature vector $\mu_{n,i}(t)$ and $B_i(t)$ represent the cumulative contextual vector. The estimate of feature vector $\widehat{\mu}_{n,i}(t)$ and the cumulative contextual vector $B_i(t)$ can be denoted as

$$\begin{aligned}
\widehat{\mu}_{n,i}(t) &= (I_d + \Sigma\Lambda_t + \Sigma_t)^{-1} \sum_{\tau=1}^{t-1} 2X_\tau r_{\mathscr{A}(\tau)}(\tau) = B_i(t)^{-1} \sum_{\tau=1}^{t-1} 2X_\tau r_{\mathscr{A}(\tau)}(\tau), \\
B_i(t) &= I_d + \widehat{\Sigma}_t + \Sigma_t = I_d + \sum_{\tau=1}^{t-1} X_\tau X_\tau^\top + \sum_{\tau=1}^{t-1} \mathbb{E}\left(X_\tau X_\tau^\top \mid \mathscr{F}_{\tau-1}\right),
\end{aligned} \tag{14}$$

where $X_\tau = b(\tau) - \mathbb{E}(b(\tau) \mid \mathscr{F}_{\tau-1})$; I is a d dimensional identity matrix, where $d = 2(N_{\max}^t + b_{\max})$.

For ease of exposition, the $\mathbb{E}(b(\tau) \mid \mathscr{F}_{\tau-1})$ in X_τ can be denoted as $\bar{b}(\tau)$ and it can be calculated as

$$\bar{b}(\tau) = \mathbb{E}\left(\sum_{n=1}^{N^t+b^t} \sum_{i=1}^{3} I(a_n(\tau) = i) b(\tau) \mid \mathscr{F}_{\tau-1}\right) = \sum_{n=1}^{N^t+b^t} \sum_{i=1}^{3} \pi_{n,i}(\tau) b(\tau), \tag{15}$$

where $\pi_{n,i}(\tau) = \mathbb{P}(a_n(\tau) = i \mid \mathscr{F}_{\tau-1})$ is the probability of offloading task n to the ith execution platform at time τ.

Besides, we can calculate the covariance $\mathbb{E}(X_\tau X_\tau^\top \mid \mathscr{F}_{\tau-1})$ as follows:

$$\mathbb{E}\left(X_\tau X_\tau^\top \mid \mathscr{F}_{\tau-1}\right) = \sum_{n=1}^{N^t+b^t} \sum_{i=1}^{3} \pi_{n,i}(\tau) \left(b(\tau) - \bar{b}(\tau)\right) \left(b(\tau) - \bar{b}(\tau)\right)^\top. \tag{16}$$

Accordingly, the mobile device can continuously explore and then gather the relationship between the feature vector of each task and the system cost of the chosen execution platform. Then, it also measures the corresponding privacy level to estimate which execution platform is likely to give the minimum system cost while maintaining a good privacy level.

In this paper, the TS-based online learning algorithm will be applied to learn the underlying relation between the feature vector and received cost. Hence, we should construct a distributional likelihood function to sample the estimated cost. Firstly, the standard deviation of the estimated cost $\widehat{r}_{n,i}(t) = b(t)^\top \widehat{\mu}_{n,i}(t)$ can be defined as $\widehat{s}_{n,i}(t) = \sqrt{b(t)^\top B_i(t)^{-1} b(t)}$ and the standard deviation of the sampling cost $\tilde{r}_{n,i}(t) = b(t)^\top \tilde{\mu}_{n,i}(t)$ can be denoted as $\tilde{s}_{n,i}(t) = \nu\sqrt{b(t)^\top B_i(t)^{-1} b(t)} = \nu \widehat{s}_{n,i}(t)$, where $\nu = (2R + 6)\sqrt{6d \log(T/\delta)}$ is a control parameter.

According to Bayes' theorem $P(B_i \mid A) \propto P(B_i) P(A \mid B_i)$, we have:

$$\Pr\left(b(t)^\top \tilde{\mu}_{n,i}(t) \middle| \tilde{r}_{n,i}(t)\right) \propto \Pr\left(\tilde{r}_{n,i}(t) \middle| b(t)^\top \tilde{\mu}_{n,i}(t)\right) \Pr\left(b(t)^\top \tilde{\mu}_{n,i}(t)\right). \tag{17}$$

Based on the TS algorithm in [31], if the prior for received cost $r_{n,i}(t) = b(t)^\top \bar{\mu}_{n,i}(t)$ at time slot t is given by $\mathcal{N}(b(t)^\top \widehat{\mu}_{n,i}(t), \nu^2 b(t)^\top B_i(t)^{-1} b(t))$, it is easy to compute the posterior distribution at time slot $t+1$, i.e., $\mathcal{N}(b(t+1)^\top \widehat{\mu}_{n,i}(t+1), \nu^2 b(t+1)^\top B_i(t+1)^{-1} b(t+1))$ (details of this

Input: N^t, $N^t_{\max}$, b^t, $b_{\max}$, m^t_n, $i \in \mathcal{I}$;
Output: total received cost $r_{\mathcal{A}(t)}(t)$.
1: **Initialization:** Initialize the cumulative contextual vector $B_i = I_d$, where I_d is a $2(N^t_{\max} + b_{\max})$ dimensional identity matrix, the cumulative contextual system cost $y_i = 0_d$, and a control parameter $v = (2R + 6)\sqrt{6d \log(T/\delta)}$, $\delta \in (0, 1)$.
2: **End initialization**
3: **for** $t = 1, 2, \cdots, T$ **do**
4: **for** $n = 1, 2, \cdots, N^t + b^t$ **do**
5: **for** $i = 1, 2, 3$ **do**
6: **Compute** the estimated feature vector $\widehat{\mu}_{n,i}(t) = B_i^{-1} y_i$.
7: **Sample** the cost $\tilde{r}_{n,i}(t)$ independently for each task n and each execution platform i from the Gaussian distribution $\mathcal{N}(b(t)^\top \widehat{\mu}_{n,i}(t), v^2 b(t)^\top B_i(t)^{-1} b(t))$.
8: **Compute** the probability of offloading task n to the i-th execution platform $\pi_{n,i}(t) = \mathbb{P}(a_n(t) = i \mid \mathcal{F}_{t-1})$.
9: **End for**
10: **Select** the offloading execution platform $a_n(t) = \arg\ \min_i \tilde{r}_{n,i}(t)$.
11: **Record** the selection $a_n(t)$ into offloading decisions vector $\mathcal{A}(t)$.
12: **End for**
13: **Compute** the aggregated received cost $r_{\mathcal{A}(t)}(t)$ via $\mathcal{A}(t)$, Equation (5), (6), (8) and (9).
14: **Update** B and y:
15: $B_{\mathcal{A}(t)} \longleftarrow B_{\mathcal{A}(t)} + (b(t) - \bar{b}(t))(b(t) - \bar{b}(t))^\top + \sum_{n=1}^{N^t+b^t} \sum_{i=1}^{i=3} \pi_{n,i}(t)(b(t) - \bar{b}(t))(b(t) - \bar{b}(t))^\top$
16: $y_{\mathcal{A}(t)} \longleftarrow y_{\mathcal{A}(t)} + 2(b(t) - \bar{b}(t)) r_{\mathcal{A}(t)}(t)$.
17: **End for**

Algorithm 1: Privacy-aware online task offloading (PAOTO) algorithm.

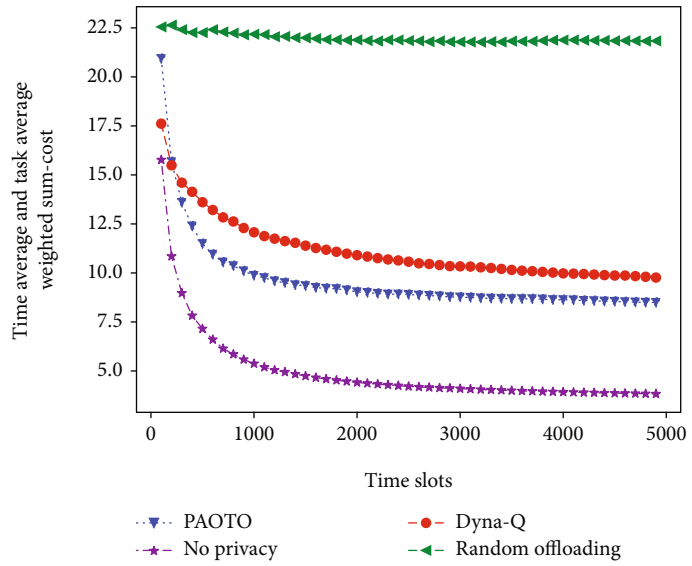

Figure 2: The simulation result comparison of time average and task average weighted sum-cost for the PAOTO algorithm, random offloading algorithm, Dyna-Q algorithm, and no privacy scenario.

computation can be seen in Appendix A.1 of [31]). Hence, at every time step t, we can use this Gaussian likelihood function $\mathcal{N}(b(t)^\top \widehat{\mu}_{n,i}(t), v^2 b(t)^\top B_i(t)^{-1} b(t))$ to sample the cost $\tilde{r}_{n,i}(t)$ for offloaded task n at execution platform i in our algorithm. Then, the sampling cost $\tilde{r}_{n,i}(t)$ will be used to estimate the performance of offloaded task n at execution platform i and finally the execution platform that has minimum $\tilde{r}_{n,i}(t)$.

Hence, guided by the problem transformation and key vectors mentioned above, we introduce the PAOTO algorithm in Algorithm 1.

Algorithm 1 gives the details of exploring the optimal solution that can make an adaptive task offloading decision and preserve the privacy of users. It estimates the offloading cost $r_{n,i}(t)$ of each task based on context information $b(t)$ and performance feature vector $\mu_{n,i}(t)$ and selects the best offloading action based on the minimum received cost $r_{n,i}(t)$. At the same time, it calculates the offloading probability $\pi_{n,i}(t)$ to fit the MAB problem with a semiparametric reward model for privacy preservation. At the end, it utilizes total received cost $r_{\mathcal{A}(t)}(t)$ to update cumulative

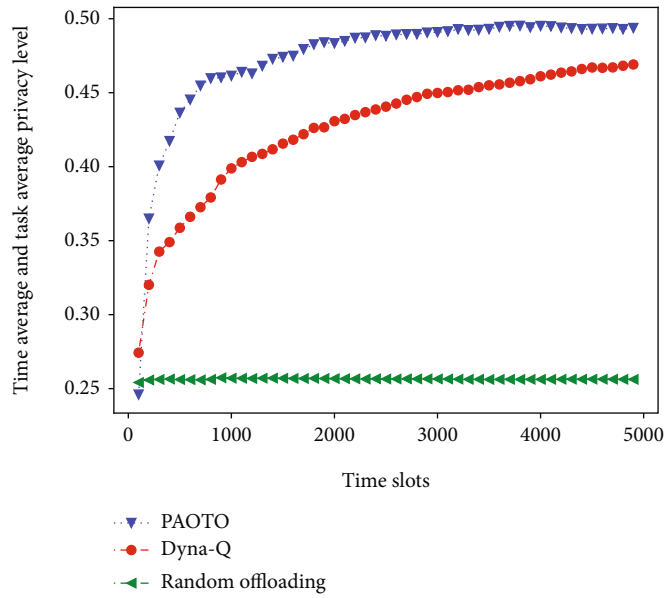

Figure 3: The simulation result comparison of the time average and task average privacy level for the PAOTO algorithm, random offloading algorithm, and Dyna-Q algorithm.

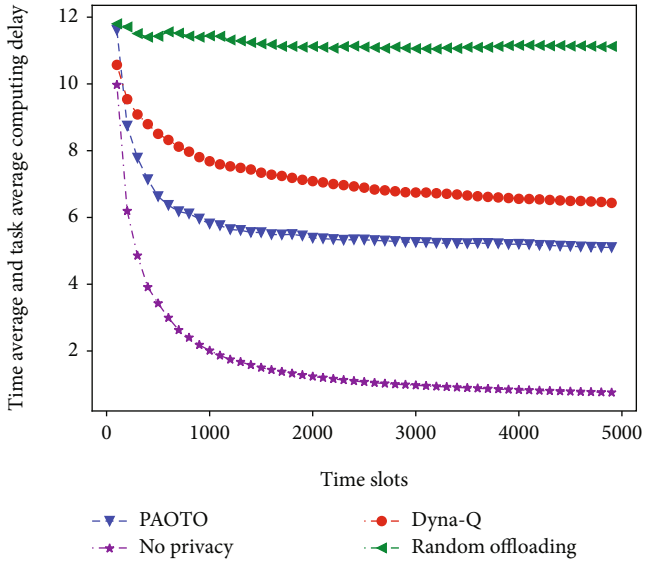

Figure 4: The simulation result comparison of the time average and task average computing delay for the PAOTO algorithm, random offloading algorithm, Dyna-Q algorithm, and no privacy scenario.

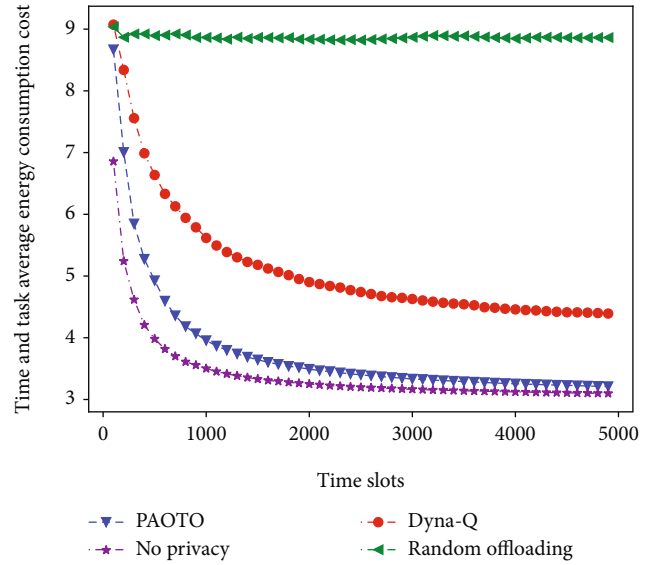

Figure 5: The simulation result comparison of the time average and task average energy consumption cost for the PAOTO algorithm, random offloading algorithm, Dyna-Q algorithm, and no privacy scenario.

contextual vector B and cumulative contextual system cost y corresponding to the decisions vector $\mathcal{A}(t)$ of all tasks at every time slot t.

5. Simulation Results

In this section, extensive simulations are conducted to evaluate the performance of the proposed PAOTO algorithm under different scenarios. We build our simulations in Python 3.6. The implementations are conducted on a Lenovo desktop PC equipped with Intel(R) core (TM) i7-4500U CPU @1.80 GHz processor and 12.0 GB (11.7 GB available) RAM. The simulation settings, algorithm benchmarks, and performance evaluation are elaborated below.

5.1. Simulation Settings. In our simulation environment, we consider a MEC system, in which the access point is deployed with the MEC server. The N^t computing tasks are randomly generated by the mobile device at every time slot t, where the maximum of N^t is in the range from 10 to 60. The maximum buffer capacity $b_{\max}$ of the mobile device can be set to 10. The length of each time slot is 1 s. The total computation capacity

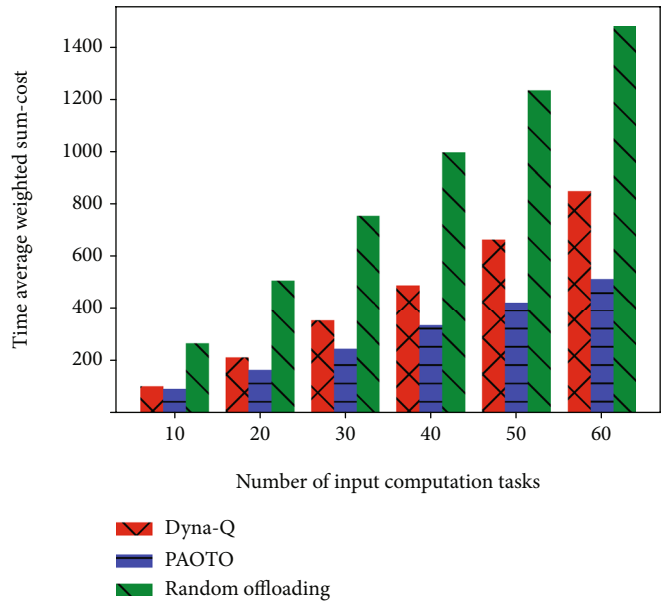

FIGURE 6: The simulation result comparison of time average weighted sum-cost for the PAOTO algorithm, random offloading algorithm, and Dyna-Q algorithm under different numbers of input computation tasks.

for MEC server γ_s is uniformly distributed in (10, 15) GHz. In order to accommodate the dynamics, we assume that the computation capacity of the mobile device γ_l is determined randomly from 1 to 3 GHz and the converted computation capacity of buffer γ_b is uniformly distributed in (0.1, 0.15) GHz. Based on [33], the data size of each task λ_n is distributed in (300 K, 800 K) bits and the computation intensity δ_n is taken randomly within (250, 1000) CPU cycles/bit. Thus, we can get the required CPU cycles of a computing task m_n by $m_n = \lambda_n * \delta_n$ (CPU cycles). Besides, the energy consumption for transmitting one task to the MEC server is uniformly distributed in (0.1, 0.5) J and the mobile device consumes 0.8 J to 3 J to locally compute one task and 0.5 J to 1 J to buffer one task. The weights of processing delay, energy consumption, and privacy metric, which are ω^t_{delay}, ω^t_{energy} and $\omega^t_{\text{privacy}}$,respectively, can be dynamically set by the users according to the user's preferences and the running application demands. In our simulation, we set them to 1, 1, and 10, respectively. The control parameter v of the PAOTO algorithm is usually set to 1.

5.2. Benchmarks. The simulations are carried out based on the above setting. In order to better manifest the advantages and effectiveness of the proposed algorithm, two typical benchmarks are implemented for comparison with the PAOTO algorithm, which are presented as follows:

(1) *Random offloading algorithm*: the random offloading algorithm is chosen as one of the baselines, which will arrange the offloading in a random way. This is the method for the resource-constrained mobile device to decentralize computing tasks. However, it does consider the privacy preservation and performance optimization. The purpose of this benchmark is to evaluate the necessity of the proposed algorithm

(2) *Dyna-Q algorithm*: we implemented the Dyna-Q algorithm as one of the benchmarks in our simulations. The implementation details may be slightly different from that of [15], but the main framework is the same. The Dyna-Q in [15] is a reinforcement learning- (RL-) based privacy-aware offloading scheme. It is an improvement of the Q-learning method, combining the model-independent and model-dependent methods. But, it requires more system-level information (e.g., assumption of the Markov model) than the proposed algorithm. As the most state-of-the-art and relevant scheme to our works, the implementation of Dyna-Q can bring more reliable performance guarantees for our algorithm evaluation

(3) *No privacy scenario*: the scenario that does not consider privacy protection is also used as one of our baselines. According to [10], when mobile devices do not consider privacy protection but focuses solely on optimizing delay and energy consumption, the optimal latency and energy consumption performance can be obtained. Comparing with a scenario that does not consider privacy protection, it can reflect that the proposed algorithm will compromise the system performance in order to protect user's privacy

As such, our algorithms are comparing the performance with these two benchmarks for analysis and these values match those used in previous works.

5.3. Numerical Results. In this section, the numerical results are presented to evaluate the effectiveness of the proposed algorithm. The weighted sum-cost, privacy level, computing delay, and energy consumption cost of the PAOTO algorithm in a period of time are compared with the two

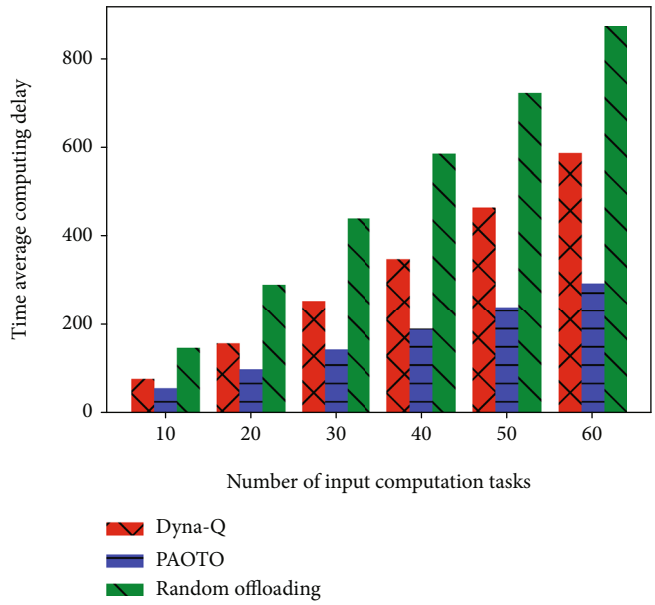

Figure 7: The simulation result comparison of time average computing delay for the PAOTO algorithm, random offloading algorithm, and Dyna-Q algorithm under different numbers of input computation tasks.

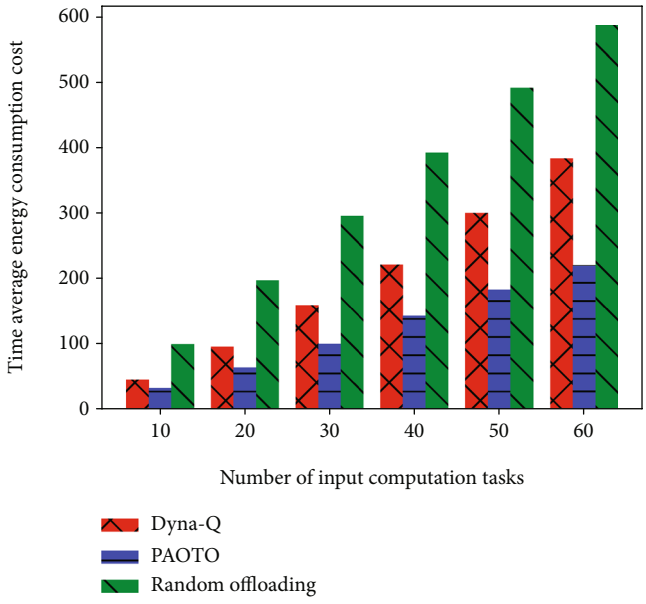

Figure 8: The simulation result comparison of time average energy consumption for the PAOTO algorithm, random offloading algorithm, and Dyna-Q algorithm under different numbers of input computation tasks.

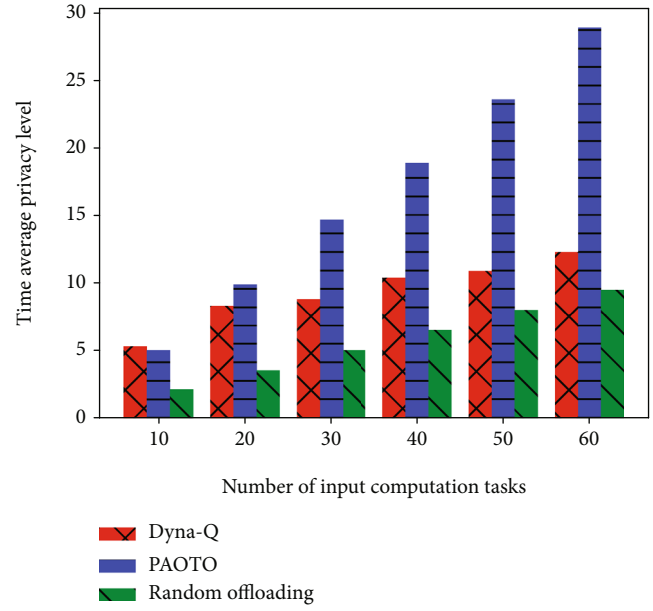

Figure 9: The simulation result comparison of the time average privacy level for the PAOTO algorithm, random offloading algorithm, and Dyna-Q algorithm under different numbers of input computation tasks.

benchmarks to evaluate the performance of the proposed algorithm.

5.3.1. The First Set of Simulations. In the first set of simulations, we randomly generate some tasks for the mobile device at each time slot t, which are the same for the three algorithms. The number of newly generated tasks are taken randomly within [5, 30]. The task offloading policy is executed for each task per round. Since the number of tasks in each round is dynamic, the simulation results are averaged for each task. The results of average weighted sum-cost, privacy level, computing delay, and energy consumption cost are reported as the following and the results are plotted at every 100 time slots.

As shown in Figure 2, we trace the average weighted sum-cost of the PAOTO algorithm, random offloading algorithm, and Dyna-Q algorithm at each time slot t. It can be seen that the PAOTO algorithm can obtain lower system cost for each

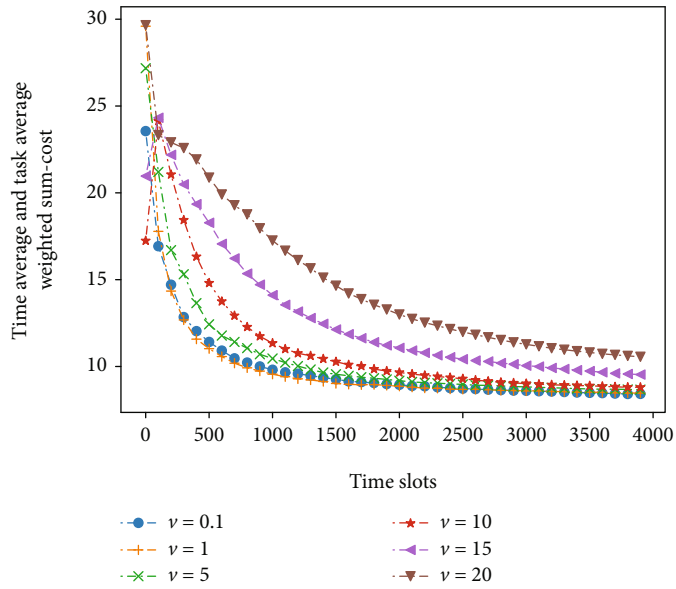

Figure 10: Performance comparison with different values of v.

task with about 23.0% reduction comparing to the Dyna-Q algorithm and about 50.1% to the random offloading at the 1000th time slot. However, compared to another scenario that the privacy is not considered, the proposed PAOTO algorithm has a higher system cost. This shows that the PAOTA algorithm has compromised the cost in order to protect privacy, which obtains the suboptimal solution.

According to Figure 3, we can see that the proposed algorithm achieves better performance of the privacy-preserving level comparing to the random offloading and Dyna-Q algorithm. The proposed algorithm improves 4.8% and 19.1% of the privacy level compared with the Dyna-Q scheme and random offloading algorithm, respectively, at the 2000th time slot. The performance comparison of computing delay and energy consumption cost also verifies the improvement of the proposed algorithm, which are shown in Figures 4 and 5, respectively. For instance, compared to Dyna-Q, the computing delay and the energy consumption cost of the PAOTO algorithm decrease by 22.5% and 25.4%, respectively, at the 1000th time slot. It is a pity that compared to the scenarios that privacy is not considered (i.e., the optimal solution), the latency and energy consumption performance of the proposed algorithm is slightly worse. Because it sacrifices some performance in order to preserve privacy. Besides, the simulation results of the random offloading scheme are very poor, which further proves that it is of great significance to study the task offloading and preserve the privacy of the users in the MEC system. Given these facts in the first set of simulations, it can be observed that the PAOTO algorithm outperforms the other two benchmarks and it obtains suboptimal task offloading performance while protecting user's privacy.

5.3.2. The Second Set of Simulations. In the second set of simulations, we investigate the performance of the proposed algorithm with different maximum number of input computation tasks $N^t_{\max}$, which ranged from 10 to 60. The data size of each task is uniformly distributed in (300, 800 K) bits. These simulation results in the second set are averaged over the first 2000 time slots. As shown in Figures 6–8, the PAOTO algorithm can get a lower average weighted sum-cost, computing delay, and the energy consumption cost than the other two benchmarks. And the improvements of these performances (histogram difference) increase as the $N^t_{\max}$ increases from 10 to 60. For instance, when the number of computing tasks is 20, compared with Dyna-Q, the average sum-cost, computing delay, and the energy consumption cost of the PAOTO algorithm increase by 28.1%, 44.7%, and 28.6%, respectively. Whereas, when the number of computing tasks is 60, they are 29.3%, 51.6%, and 37.2%, respectively. The reason is that as the total number of tasks increases, the Dyna-Q algorithm requires more time to learn, and the random offloading does not have any performance optimization effects, but the proposed algorithm has stable processing efficiency to obtain a lower cost.

Additionally, as shown in Figure 9, with the increment of the number of tasks, the privacy level of the proposed algorithm will increase significantly but the privacy level of the Dyna-Q algorithm will decrease slightly. It is also because the processing efficiency of the Dyna-Q algorithm will decrease as the number of tasks increases. Besides, the privacy level of the random offloading algorithm is not affected by the number of tasks. Hence, the simulation results of the second set validate that the PAOTO algorithm has superior and stable system performance and privacy-preserving level for increasing computing-intensive tasks.

From the two set of simulations mentioned above, it can be seen that the PAOTO algorithm meets the objective of this paper that receives the close-to-optimal delay and energy consumption performance for MD while protecting the user's privacy. And it has a significant performance improvement comparing to the other two benchmarks.

5.3.3. The Third Set of Simulations. In order to analyze the effect of different key parameters (i.e., δ and K) on the

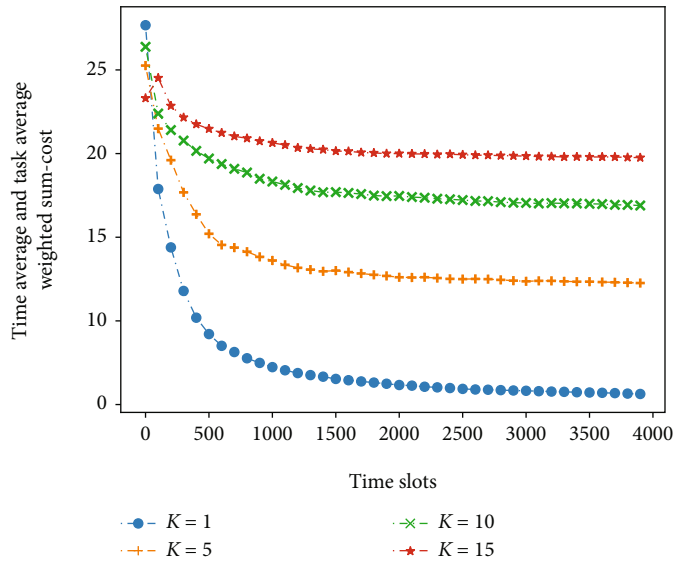

Figure 11: Performance comparison with different values of K.

PAOTO algorithm, the weighted sum-cost is plotted under different values of v and the number of dummy tasks K. First, the parameter v in the PAOTO algorithm is associated with the standard deviation of the sampling, where $v = (2R + 6)\sqrt{6d \log(T/\delta)}, \delta \in (0, 1)$. Thus, we set the values of v as 0.1, 1, 5, 10, 15, and 20. As shown in Figure 10, we can observe that the values of v and the average weighted sum-cost of the PAOTO algorithm are positively correlated when $v \geq 1$, such as the curves $v = 1$, $v = 5$, and $v = 10$. As the value of v is larger, the convergence of the PAOTO algorithm becomes worse. However, when $v < 1$, the average cost of the PAOTO algorithm will increase with the decrease of v, such as curves $v = 1$ and $v = 0.1$. The reason is the cost tradeoff in the theoretical bound, and there are different effects before and after reaching the bound.

Second, the different number of dummy tasks K that is related to privacy metric P^t is simulated for weighted sum-cost of the proposed algorithm. As shown in Figure 11, K can be set as 1, 5, 10, and 15. Then, we can observe that as K increases, the weighted sum-cost will also increase. However, as K is larger, the increment of the weighted sum-cost will decrease. According to equations (8) and (9) in Section 3, K is directly proportional to the privacy metric P^t and the privacy metric P^t is inversely proportional to the weighted sum-cost. When the number of dummy tasks K increases, the system cost will increase at the beginning. Nevertheless, taking dummy tasks into the privacy metric can restrict the increment of the weighted sum-cost. That is, there is a tradeoff between the system cost and the privacy metric.

6. Conclusions

In this paper, we investigated joint task offloading and privacy preservation for the small-size and low-power mobile devices without any system-level network information in the MEC system. The objective is to minimize a weighted sum of the computing delay, energy consumption cost, and reciprocal of the privacy metric. In particular, the joint optimization problem has been formulated as a contextual MAB problem with a semiparametric reward model to accommodate network dynamics, in which the privacy metric is taken into account. Subsequently, a privacy-aware online task offloading (PAOTO) algorithm is proposed to explore the balance between the optimal system cost and the privacy level. The simulation results show that the proposed algorithm can provide near-optimal solutions in a short computing time. In the future, we will extend our work to the scenarios that have multiple MEC servers with distinct computing capability and take the bandwidth constraint into account.

Data Availability

The (DATA TYPE) data used to support the findings of this study are included within the article.

Disclosure

Additionally, a preliminary version of this work was accepted by WASA 2020. However, we have extended the conference paper significantly and the difference between the journal version and the conference version is above 50%.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant no. XDC02040300.

References

[1] J. Liu and Q. Zhang, "Code-partitioning offloading schemes in mobile edge computing for augmented reality," *IEEE Access*, vol. 7, pp. 11222–11236, 2019.

[2] R. Gu, L. Yu, and J. Zhang, "Mefill: a multi-edged framework for intelligent and low latency mobile IOT services," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Seoul, Korea, 2020.

[3] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-aware offloading in mobile edge computing," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, 2017.

[4] T. Yang, F. Wolff, and C. Papachristou, "Connected car networking," in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, pp. 60–64, Dayton, OH, USA, 2018.

[5] CISCO, *Cisco Annual Internet Report (2018–2023) White Paper*, 2020, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[6] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[8] Q. Tang, R. Xie, T. Huang, and Y. Liu, "Jointly caching and computation resource allocation for mobile edge networks," *IET Networks*, vol. 8, no. 5, pp. 329–338, 2019.

[9] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.

[10] X. He, R. Liu, and H. Dai, "Privacy-aware offloading in mobile-edge computing.," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, 2017.

[11] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, 2020.

[12] N. Eshraghi and B. Liang, "Joint offloading decision and resource allocation with uncertain task computing requirement," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1414–1422, Paris, France, 2019.

[13] H. Kim, H. Kim, and J. Chang, "A privacy-preserving kNN classification algorithm using Yao's garbled circuit on cloud computing," in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pp. 766–769, Honolulu, CA, USA, 2017.

[14] W. Juang and Y. Shue, "A secure and privacy protection digital goods trading scheme in cloud computing," in *2010 International Computer Symposium (ICS2010)*, pp. 288–293, Tainan, Taiwan, 2010.

[15] M. Min and X. Wan, "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *Internet of Things Journal*, vol. 6, no. 3, pp. 4307–4316, 2018.

[16] C. Wang, P. Liu, T. Zhang, and J. Sun, "The adaptive vortex search algorithm of optimal path planning for forest fire rescue UAV," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 400–403, Chongqing, China, 2018.

[17] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, "Federated edge-assisted mobile clouds for service provisioning in heterogeneous IoT environments," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 591–596, Milan, Italy, 2015.

[18] T. Ouyang, X. Chen, L. Zeng, and Z. Zhou, "Cost-aware edge resource probing for infrastructure-free edge computing: from optimal stopping to layered learning*2019 IEEE Real-Time Systems Symposium (RTSS)*, pp. 380–391, Hong Kong, China, 2019.

[19] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2018, pp. 207–215, Honolulu, HI, USA, 2018.

[20] F. Wei, S. Chen, and W. Zou, "A greedy algorithm for task offloading in mobile edge computing system," *China Communications*, vol. 15, no. 11, pp. 149–157, 2018.

[21] B. Dab, N. Aitsaadi, and R. Langar, "Q-Learning algorithm for joint computation offloading and resource allocation in edge cloud," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 45–52, Arlington, VA, USA, 2019.

[22] G. Li and J. Cai, "An online incentive mechanism for collaborative task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 624–636, 2020.

[23] X. He, R. Jin, and H. Dai, "Deep PDS-learning for privacy-aware offloading in MEC-enabled IoT," *Internet of Things Journal*, vol. 6, no. 3, pp. 4547–4555, 2019.

[24] H. Zhang and K. Zeng, "Pairwise Markov chain: a task scheduling strategy for privacy-preserving SIFT on edge," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1432–1440, Paris, France, 2019.

[25] P. Zhou, W. Chen, S. Ji, and H. Jiang, "Privacy-preserving online task allocation in edge-computing-enabled massive crowdsensing," *Internet of Things Journal*, vol. 6, no. 5, pp. 7773–7787, 2019.

[26] X. Yu, M. Guan, M. Liao, and X. Fan, "Pre-migration of vehicle to network services based on priority in mobile edge computing," *IEEE Access*, vol. 7, pp. 3722–3730, 2019.

[27] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation efficiency in a wireless-powered mobile edge computing network with NOMA," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, Shanghai, China, 2019.

[28] W. Zhang, Z. Zhang, S. Zeadally, and H. Chao, "Efficient task scheduling with stochastic delay cost in mobile edge computing," *IEEE Communications Letters*, vol. 23, no. 1, pp. 4–7, 2019.

[29] T. Ouyang, R. Li, X. Chen, Z. Zhou, and X. Tang, "Adaptive user-managed service placement for mobile edge computing: an online learning approach," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1468–1476, Paris, France, 2019.

[30] K. Gi-Soo and P. Myunghee Cho, *Contextual Multi-Armed Bandit Algorithm for Semiparametric Reward Model*, International Conference on Machine Learning, 2019.

[31] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, pp. 127–135, Atlanta, USA, 2013.

[32] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen et al., "A tutorial on Thompson sampling," *Foundations and Trends in Machine Learning*, vol. 11, no. 1, pp. 1–96, 2018.

[33] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, 2015.