Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.067200

ARTICLE

A Spectrum Allocation and Security-Sensitive Task Offloading Algorithm in MEC Using DVS

Xianwei Li[1,2], Bo Wei[3,4], Xiaoying Yang[5,6,*], Amr Tolba[7], Zijian Zeng[8] and Osama Alfarraj[7,*]

[1]School of Computer and Information Engineering, Bengbu University, Bengbu, 233000, China
[2]Anhui Engineering Research Center for Intelligent Applications and Security of Industrial Internet, Anhui University of Technology, Ma'anshan, 243032, China
[3]Faculty of Environmental, Life, Natural Science and Technology, Okayama University, Okayama, 700-8530, Japan
[4]Japan Science and Technology Agency (JST), PRESTO, Kawaguchi, 332-0012, Japan
[5]School of Information Engineering, Suzhou University, Suzhou, 237000, China
[6]State Key Laboratory of Tea Biology and Resource Utilization, Anhui Agricultural University, Hefei, 230000, China
[7]Computer Science and Engineering Department, College of Applied Studies, King Saud University, Riyadh, 11437, Saudi Arabia
[8]Institute of Computer Science and Digital Innovation, UCSI University, Kuala Lumpur, 56000, Malaysia

*Corresponding Authors: Xiaoying Yang. Email: yangxiaoying@ahszu.edu.cn; Osama Alfarraj. Email: oalfarraj@ksu.edu.sa

Received: 27 April 2025; Accepted: 22 July 2025; Published: 23 September 2025

ABSTRACT: With the advancements of the next-generation communication networking and Internet of Things (IoT) technologies, a variety of computation-intensive applications (e.g., autonomous driving and face recognition) have emerged. The execution of these IoT applications demands a lot of computing resources. Nevertheless, terminal devices (TDs) usually do not have sufficient computing resources to process these applications. Offloading IoT applications to be processed by mobile edge computing (MEC) servers with more computing resources provides a promising way to address this issue. While a significant number of works have studied task offloading, only a few of them have considered the security issue. This study investigates the problem of spectrum allocation and security-sensitive task offloading in an MEC system. Dynamic voltage scaling (DVS) technology is applied by TDs to reduce energy consumption and computing time. To guarantee data security during task offloading, we use AES cryptographic technique. The studied problem is formulated as an optimization problem and solved by our proposed efficient offloading scheme. The simulation results show that the proposed scheme can reduce system cost while guaranteeing data security.

KEYWORDS: IoT; DVS; MEC; AES

1 Introduction

In recent years, with the advancements of the next-generation communication networks and Internet of Things (IoT) technologies, the number of IoT terminals has increased at an unprecedented rate. It is expected that terminal devices (TDs) will exceed 27 billion at the end of the year 2025 [1]. This results in the emergence of a variety of computation and delay-intensive applications, such as autonomous driving and face recognition [2,3]. The execution of these applications needs a lot of computing resources and consumes a large amount of energy. Nevertheless, as TDs have limited physical space and battery power, they usually do not have sufficient resources to compute these applications. The traditional method to overcome this problem is to offload and process IoT applications in public clouds. But since there is a long distance from TDs to the location of the data centers of public cloud, long transmission latency will be caused [4]. This means that the

distant public cloud is not suitable for processing real-time IoT applications. To tackle the above issues, a new computing paradigm called mobile edge computing (MEC) has emerged [3]. By deploying edge servers at the base station (BS), the cloud services can be offered at the network edge, which is much closer to the TDs compared with the public cloud [4]. TDs can offload and execute computation-intensive and latency-aware tasks by using the edge cloud services such that both the computing time and energy consumption can be reduced.

Despite these advantages, several challenging issues should be resolved for the deployment of MEC systems [5]. Firstly, how to efficiently allocate resources is of great importance for making task offloading decisions of TDs. Secondly, security is a main concern when provisioning edge cloud service, as data attacks may lead to disruptive denial-of-service (DoS) in decentralized wireless networks.

Although extensive research work has been done to investigate resource allocation methods and task offloading in MEC systems, many of them did not consider the security issue. In addition, many prior works merely considered execution cost from the aspects of TDs, ignoring the cost of the whole MEC system. In [6], Kuang et al. studied the allocation of transmission power and task offloading in UAV-enabled MEC systems while adopting the DVS technology to minimize the total energy consumption of Ground TDs and UAVs. In [7], Liu et al. studied energy efficiency maximization in UAV-assisted NOMA-MEC networks. They also considered the allocation of transmission power and the constraint of the computing power of mobile user devices. In [8], Hossain et al. proposed a task offloading scheme based on the MRL algorithm considering transmission power control and the adoption of DVS technology. The aim is to minimize energy consumption of user equipment.

In this work, we consider spectrum resource allocation and security-sensitive task offloading to minimize the total computing time and energy consumption of the MEC system. A secure layer using AES cryptographic technique to protect the data of tasks from being attacked during the task offloading process. Moreover, as dynamic voltage scaling (DVS) technology can be used by TDs to adjust their operating frequency to reduce computing time and energy consumption [9], we adopt it when designing the task offloading scheme.

The principal contributions made in this study are listed as follows.

- We study spectrum resource allocation and security-aware full task offloading in an MEC system. The main purpose is to propose an efficient spectrum resource allocation and full task offloading scheme for optimizing the performance of the MEC system while considering the security issue. We adopt the AES cryptographic technique to protect tasks from being attacked.
- We formulate an optimization problem to minimize system costs in terms of computing time and energy consumption. Since the objective function for the formulated problem is a 0–1 linear optimization problem, which is a mixed integer nonlinear programming (MINP) problem known to be NP-hard. In order to solve it efficiently, we divide the original problem into three sub-problems. Additionally, an efficient task offloading algorithm is devised to determine task offloading decisions of the TDs.
- To show the superiority of the proposed resource allocation method and task offloading algorithm, simulation experiments are conducted. The conducted simulation results verify the performance of this algorithm.

The remaining parts of this paper are organized as follows. We review and discuss the related research in Section 2. Description of the system model and formulation of the optimization problem are presented in Section 3. In Section 4, we present a solution method to the optimization problem. In Section 5, we present simulation results to verify our solution method. Section 6 is the conclusion.

2 Related Work

Numerous studies have focused on improving the performance of MEC networks by optimizing resource allocation and proposing task offloading algorithms. The related research can be categorized according to the objectives.

Some studies merely put focus on maximizing energy efficiency. For example, Wang et al. studied communication and computation resources optimization in blockchain based MEC networks in [10]. They considered both the computation of MEC tasks and blockchain federated learning tasks intending to reduce the energy consumption of edge servers. In [11], Yang et al. investigated energy efficiency in reconfigurable intelligent surface (RIS)-assisted networks. They analyzed both single user equipment (UE) case and multiple UEs case. In [12], the authors investigated energy efficiency maximization and task offloading in Internet of Vehicles (IoV) networks enabled by MEC. They proposed an algorithm maximizing energy efficiency based on the multi-agent DRL. In the work [13], a heuristic algorithm was proposed for energy-efficient computation offloading. The authors considered the allocation of wireless and computation resources. Due to the complexity of the primal optimization problem, it was divided into several sub-problems and solved separately. In [14], the authors also studied allocation of wireless and computation resources for computation offloading in MEC systems. They aimed to minimize the average power consumption of TDs under the completion deadline constraints. In [15], Bi et al. studied partial computation offloading targeting energy minimization in hybrid cloud-edge computing systems. They established a cost minimization problem and addressed it by proposing an LSAG algorithm. In [16], Panda et al. proposed an offloading scheme by applying the DVS and DRL to cut down the energy consumption of TDs under execution time constraints. But this work did not consider security issues and resource allocation.

Some other studies focused on reducing the execution time of TDs. The authors studied task execution time minimization by optimizing the computation resources allocation in multiple unmanned aerial vehicle (UAV) assisted MEC systems [17]. They considered two offloading models and developed alternating optimization algorithms to handle the formulated problems. Wang et al. studied the subtask offloading problem in MEC systems with multiple edge servers [18]. They formulated a latency minimization problem and proposed a sub-task offloading strategy with the aim of reducing system latency while considering energy utilization. Chen et al. theoretically analyzed the experienced latency of TDs caused by transmission and computation processes [19]. Their study shows that the offloaded decisions, bandwidth, and CPU resources can affect the processing latency of TDs. Hossain and Ansari studied computing and communication resources optimization, and applied the NOMA and network slicing (NS) techniques to reduce the computing latency of system tasks [20]. They proposed a binary scheme for task offloading and formulated an optimization problem to minimize total computing latency of tasks.

In addition to the above two categories, some works pay attention to the balance of execution time and energy consumption of TDs. For example, Pervez et al. studied transmission power and CPU resource allocation to minimize energy and latency of the multi-UAV assisted MEC systems [21]. They proposed an alternating iterative approach to solve the formulated optimization problem. In the reference [22], Younis investigated task offloading and approximate computing in MEC networks. Their objective is to balance energy consumption and latency. The formulated problem is decomposed into three main subproblems and solved efficiently. In [23], the authors built three models and the computation offloading problem of minimizing latency and energy consumption is a multiobjective optimization problem and solved by utilizing the NSGA-II. Xu et al. explored the problem of joint minimization of energy and latency costs by optimizing the computing resources of edge servers in the IRS-enabled multi-cell MEC systems [24]. The problem was divided into two subproblems namely the MEC subproblem and the IRS subproblem, and solved by an efficient algorithm. In [25], Mustafa et al. studied optimal task offloading actions and resource allocation

in vehicular edge computing networks. Due to the challenging issues caused by vehicles and tasks, they proposed a three-layered architecture and presented a two-level algorithm to get optimal task offloading actions and resource allocation. But a security issue was not considered in this work.

Some studies of task offloading and resource allocation considered security issue. Samy et al. studied energy and time overhead minimization of each TD in blockchain-based MEC systems [5]. They considered a multiuser multitask scenario and proposed a framework to secure task offloading by applying deep reinforcement learning. Huo et al. studied latency minimization in FDMA-based MEC systems while considering physical layer security [26]. They proposed a joint power and computing resources allocation for task offloading policy to achieve latency minimization. In [27], Wu et al. also considered physical layer security of computation offloading in NOMA-assisted MEC systems where an eavesdropping attack exists. They tried to minimize total energy consumption of IoT users by optimizing the transmission power of TDs. In order to secure the physical layer security of partial computation offloading in NOMA-assisted MEC system, Zheng et al. adopted Wyner's secrecy encoding scheme [28]. Their main objective is to minimize energy consumption of each TD by optimally allocating the transmit power and bandwidth. Nevertheless, these previous works either ignored the security issue or only considered the energy or time cost of TDs without considering them jointly.

To overcome the limitations of existing works, we investigate security-sensitive task offloading and spectrum allocation in an MEC system. Inspired by [29,30], a secured layer using AES cryptographic technique is introduced to protect the data of the offloaded tasks. In addition, DVS technology is adopted to reduce the computing time and energy consumption of TDs. Our main goal is to minimize the whole MEC system costs including both computing time and energy cost.

3 System Model and Problem Formulation

Assume that there is an MEC system consisting of N TDs and one edge cloud server. The system model is described in Fig. 1. In the system model, the cloud server is placed near the base station (BS). All TDs are associated with the edge server via the 5G cellular networks. Each TD has a task to be executed either on its device or on the edge cloud server. The task of each TD can be part of various applications, such as face recognition [29]. For TD k, we use D_k to represent the data size of its task and C_k to denote computing intensity. Each TD can either compute its task itself or use the computing resources of the edge cloud server. We let a_k denote the computing decision of TD k. If the task of TD k is computed in the cloud server, $a_k = 1$. Otherwise, if it is computed in the TD, $a_k = 0$. In other words, we consider the full task offloading scenario.

Remark 1: *It should be noted that there are two types of tasks offloading scenarios, namely, full task offloading scenarios and partial task offloading scenarios. IoT applications, such as file compression applications, can be partitioned into two parts. One part can be processed on TD while the other part can be offloaded and processed on the edge cloud server. In this paper, for simplicity, we only considered the full task offloading scenario. Hybrid task offloading scenarios can be left as a future research study. It should also be noted that there may exist multiple servers in the edge cloud. In this case, load balancing strategies should be adopted to handle server load, which is out of the scope of this study.*

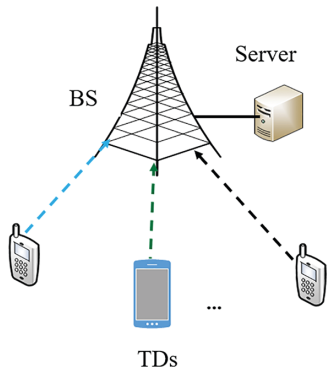

Figure 1: A MEC system

3.1 Local Computing Model

In local computing model, tasks of TDs are processed by using their computing resources. For TD k, $k \in \{1, 2, ..., N\}$, the computing time and the energy consumption caused by processing its task are respectively denoted as

$$T_k^l = \frac{D_k C_k}{f_k^l} \tag{1}$$

$$E_k^l = P_k \frac{D_k C_k}{f_k^l} \tag{2}$$

where f_k^l is the computing capacity of TD k, and P_k denotes its consumed power per CPU cycle.

The power consumption of TD k is

$$P_k = \rho_k (f_k^l)^2 \tag{3}$$

where ρ_k is a constant value related to TD k's chip architecture. The value of ρ_k can be set as 10^{-26} [24].

Therefore, the computing costs of TD k in local computing model can be denoted as

$$C_k^l = \alpha_k^t T_k^l + \alpha_k^e E_{k,l} = \alpha_k^t \frac{D_k C_k}{f_k^l} + \alpha_k^e \rho_k (f_k^l)^2 D_k C_k \tag{4}$$

where α_k^t and $\alpha_k^e \in [0, 1]$ are weighting values, and $\alpha_k^t + \alpha_k^e = 1$. If $\alpha_k^t > \alpha_k^e$, TD k puts more focus on computing time. If $\alpha_k^t < \alpha_k^e$, TD k focuses more on the energy cost.

3.2 Cloud Server Computing Model

In this model, TDs will offload and process their tasks on the edge cloud server. For TD k, its transmission rate in uplink channel is

$$r_k = B_k \log_2 \left(1 + \frac{h_k P_k}{\sigma_0^2}\right) \tag{5}$$

where B_k is the uplink channel bandwidth, h_k and P_k are the channel gain and the transmit power, respectively; σ_0^2 denotes the white Gaussian noise power during transmission.

Then, the time caused by transmission and energy consumption can be represented as, respectively,

$$T_k^t = \frac{D_k}{r_k} = \frac{D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} \tag{6}$$

$$E_k^t = P_k T_k^t = \frac{P_k D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} \tag{7}$$

When TD k's task is processed on edge cloud server, the incurred computing time is

$$T_k^c = \frac{D_k C_k}{F_k} \tag{8}$$

where F_k is the cloud computing resources of edge cloud server allocated to TD k.

Hence, the total computing time on edge cloud server processing model can be denoted as

$$T_k^e = T_k^t + T_k^c = \frac{D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} + \frac{D_k C_k}{F_k} \tag{9}$$

Hence, the processing cost for the MEC server model is denoted as

$$c_k^e = \alpha_k^t T_k^e + \alpha_k^e E_k^t = \alpha_k^t \left(\frac{D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} + \frac{D_k C_k}{F_k} \right) + \alpha_k^e \frac{P_k D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} \tag{10}$$

For TD k, its computing costs under unsecured model can be denoted as

$$\begin{aligned} C_k^u = {} & (1 - a_k) C_k^l + a_k C_k^e = (1 - a_k) \left(\alpha_k^t \frac{L_k C_k}{f_k^l} + \alpha_{k,e} \rho_k (f_k^l)^3 L_k C_k \right) \\ & + a_k \left[\alpha_k^t \left(\frac{L_k}{B_k \log_2 \left(1 + \frac{h_k p_k}{\sigma_0^2}\right)} + \frac{C_k L_k}{F_k} \right) + \alpha_k^e \frac{p_k L_k}{B_k \log_2(1 + \frac{h_k p_k}{\sigma_0^2})} \right] \end{aligned} \tag{11}$$

3.3 Security Model

If TDs offload their tasks to an edge cloud server, their tasks may be attacked by different types of threats. In order to protect the data, a security-layer AES cryptographic technique is introduced [29,30]. For each TD, its task will be encrypted before offloading and decrypted before transmitting to the edge cloud server. Consequently, the incurred time and energy costs for TD k under the security model can be respectively denoted as

$$T_k^s = T_k^e + T_k^d = \frac{C_k^e}{f_k^l} + \frac{C_k^d}{F_k} \tag{12}$$

$$E_k^s = \rho_k (f_k^l)^2 C_k^e \tag{13}$$

where C_k^e and C_k^d represent the number of computing resources to encrypt and decrypt TD k's data on TD and edge cloud server.

Based on the above two models, for TD k, the total time and energy cost can be respectively expressed as

$$T_k^o = T_k^e + T_k^s = \frac{C_k^e}{f_k^l} + \frac{C_k^d}{F_k} + \frac{D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} + \frac{D_k C_k}{F_k} \tag{14}$$

$$E_k^o = E_k^s + E_k^t = \rho_k (f_k^l)^2 C_k^e + \frac{P_k D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} \tag{15}$$

Therefore, the total cost of TD k under secured model can be denoted as

$$C_k^o = \alpha_k^t T_k^o + \alpha_k^e E_k^o = \alpha_k^t \left(\frac{C_k^e}{f_k^l} + \frac{C_k^d}{F_k} + \frac{D_k}{B_k \log_2 \left(1 + \frac{h_k P_k}{\sigma_0^2}\right)} + \frac{D_k C_k}{F_k} \right) + \alpha_k^e \left(\rho_k (f_k^l)^2 C_k^e + \frac{P_k D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})} \right) \tag{16}$$

Remark 2: *It should be noted that other cryptographic methods* (e.g., *RSA, ECC) can also be adopted for data security. However, for resource-constrained edge devices, computational efficiency is crucial. Compared with RSA and ECC, AES encryption technique is much faster for encryption and decryption, and consumes much less energy. Therefore, AES is the only viable choice for encrypting large offloaded datasets efficiently [29,30]. A comparison of these cryptographic methods is left as future work.*

3.4 Problem Formulation

We try to minimize whole system costs by allocating bandwidth resource optimally and adopting DVS technology. For TD k, its task can either be computed locally or offloaded to edge cloud server. Therefore, its computing cost can be denoted as

$$C_k = (1 - a_k) C_k^l + a_k C_k^o \tag{17}$$

Based on system model, the formulated optimization problem can be expressed as

Problem 1:

$$\min_{B_k, f_k^l, a_k} \sum_{k=1}^{N} C_k$$

s.t.

$$0 < f_k^l \leq f_k^m$$

$$\sum_{k=1}^{N} B_k \leq B$$

$$a_k \in \{0, 1\}$$

where constraint one is the CPU frequency constraint, constraint two enforces that bandwidth allocated to TDs should not exceed the total communication resources, and constraint three is offloading strategy.

The third constraint of Problem 1 is a binary integer value, but the bandwidth allocation and the CPU frequency constraints are both continuous. This means that the feasible set of Problem 1 is not convex. Therefore, we have the conclusion that the above problem is a MINLP problem. This kind of problem is NP-hard, and its solution is difficult to obtain. Traditional algorithms, such as the Branch-and-Bound and the bat algorithm can solve the MINLP problem. But a weak point of these algorithms is that their time complexity is high [31].

4 Solution Method

In this section, we propose an efficient solution method to Problem 1 formulated in the previous section. We observe from the structure of Problem 1 that it is a convex optimization problem on condition that each TD's offloading strategy is given. Thereby, problem Problem 1 can be divided into three subproblems:

1. Local computing model subproblem.
2. Secured model subproblem.
3. Offloading strategy subproblem.

4.1 Local Computing Subproblem

As long as the offloading strategies of DTs are determined, the local computing subproblem can be formulated as

Problem 2:

$$\min_{B_k} \sum_{k=1}^{N} G_k(f_k^l)$$

s.t.

$$0 < f_k^l \leq f_k^m$$

where $G_k(f_k^l)$ is

$$G_k\left(f_k^l\right) = \alpha_k^t \frac{C_k^e}{f_k^l} + \alpha_k^e \rho_k \left(f_k^l\right)^2 C_k^e \tag{18}$$

The second-order derivative of Eq. (18) is

$$\frac{\partial^2 G_k}{\partial \left(f_k^l\right)^2} = \alpha_k^t \frac{2C_k^e}{\left(f_k^l\right)^3} + 2\alpha_k^e \rho_k C_k^e > 0 \tag{19}$$

which means that $G_k^{''}$ is positive. Hence, G_k is a convex function. By setting the first-order derivative of Eq. (18) to zero,

$$\frac{\partial G_k}{\partial f_k^l} = -\alpha_{k,t} \frac{C_k^e}{\left(f_k^l\right)^2} + 2\alpha_k^e \rho_k f_k^l C_k^e = 0 \tag{20}$$

we have

$$f_k^{l*} = \sqrt[3]{\frac{\alpha_k^t}{2\alpha_k^e \rho_k}} \tag{21}$$

It is obvious that $G_k(f_k^l)$ monotonously first decreases in $(0, f_k^{l*}]$ and then increases in $(f_k^{l*}, f_k^m]$ when $f_k^{l*} < f_k^m$, and monotonously decreases when $f_k^{l*} \geq f_k^m$.

Therefore, the optimal solution to Problem 2 is

$$G_k(f_k^l) = \begin{cases} G_k\left(f_k^{l*}\right), if & f_k^{l*} < f_k^m \\ G_k\left(f_k^m\right), if & f_k^{l*} \geq f_k^m \end{cases} \tag{22}$$

Remark 3: *From Eq. (21), it can be observed that the optimal voltage-frequency adjustment for each TD is associated with weighting values α_k^t and α_k^e. Therefore, each TD can adjust its CPU frequency according to task offloading decision-making process.*

4.2 Security Model Subproblem

After the offloading strategies of the TDs are determined, the security model computing subproblem is formulated as

Problem 3:

$$\min_{f_k^l, B_k} \sum_{k=1}^{N} H_k(f_k^l, B_k)$$

s.t.

$$0 < f_k^l \leq f_k^m$$

$$\sum_{k=1}^{N} B_k \leq B$$

where $H_k(f_k^l, B_k)$ is

$$H_k(f_k^l, B_k) = \alpha_k^t \left(\frac{C_k^e}{f_k^l} + \frac{C_k^d}{F_k} + \frac{D_k}{B_k \log_2\left(1 + \frac{h_k P_k}{\sigma_0^2}\right)} + \frac{D_k C_k}{F_k} \right) + \alpha_k^e (\rho_k (f_k^l)^2 C_k^e + \frac{P_k D_k}{B_k \log_2(1 + \frac{h_k P_k}{\sigma_0^2})}) \tag{23}$$

Problem 3 can be further divided into the following two subproblems:

Problem 3.1:

$$\min_{f_k^l} \sum_{k=1}^{N} H_{1k}(f_k^l)$$

s.t.

$$0 < f_k^l \leq f_k^m$$

where $H1_k(f_k^l)$ is

$$H1_k(f_k^l) = \alpha_k^t \frac{C_k^e}{f_k^l} + \alpha_k^e \rho_k (f_k^l)^2 C_k^e \tag{24}$$

As Problem 3.1 has the same structure as Problem 2, its optimal solution can be denoted as

$$H1_k(f_k^l) = \begin{cases} H1_k\left(f_k^{l*}\right), if & f_k^{l*} < f_k^m \\ H1_k\left(f_k^m\right), if & f_k^{l*} \geq f_k^m \end{cases} \tag{25}$$

Problem 3.2:

$$\min_{f_k^l} \sum_{k=1}^{N} H_{1k}(f_k^l)$$

s.t.

$$\sum_{k=1}^{N} B_k \leq B$$

where $H2_k(B_k)$ is

$$H2_k(B_k) = \frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{B_k \log_2\left(1 + \frac{h_k P_k}{\sigma_0^2}\right)} \tag{26}$$

The second-order derivative of Eq. (26) is

$$\frac{\partial^2 H2_k}{\partial B_k^2} = \frac{2(\alpha_k^t + \alpha_k^e P_k) D_k}{B_k^3 \log_2\left(1 + \frac{h_k P_k}{\sigma_0^2}\right)} > 0 \tag{27}$$

From Eq. (27), it can be concluded that the objective function for Problem 3.2 is convex, and this problem is a convex optimization problem. The Lagrangian function of Problem 3.2 is formulated as

$$L\left(B_k\right) = \frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{B_k \log_2\left(1 + \frac{h_k P_k}{\sigma_0^2}\right)} + \lambda\left(\sum_{k=1}^{N} B_k - B\right) \tag{28}$$

where λ denotes the non-positive Lagrange multiplier.

The dual problem of 3.2 is

$$\psi\left(\lambda\right) = \max_{\lambda \geq 0} \min_{B_k > 0} L\left(B_k, \lambda\right) = \frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{B_k \log_2\left(1 + \frac{h_k P_k}{\sigma_0^2}\right)} + \lambda\left(\sum_{k=1}^{N} B_k - B\right) \tag{29}$$

According to Karush-Kuhn-Tucker (KKT) conditions [32],

$$\frac{\partial L}{\partial B_k} = -\frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{B_k^2 \log_2\left(1 + \frac{h_k P_k}{\sigma_0^2}\right)} + \lambda = 0 \tag{30}$$

$$\lambda\left(\sum_{k=1}^{N} B_k - B\right) = 0 \tag{31}$$

From Eq. (29), it is obvious that $\lambda > 0$. Hence, from Eq. (31), we have

$$\sum_{k=1}^{N} B_k - B = 0 \tag{32}$$

From Eq. (30), we get

$$B_k^* = \sqrt{\frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{\lambda \log_2(1 + \frac{h_k p_k}{\sigma_0^2})}} \tag{33}$$

Combining Eqs. (32) and (33), the value of λ can be obtained and denoted as

$$\lambda = \left(\frac{\sum_{k=1}^{N} \sqrt{\frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{\log_2(1 + \frac{h_k p_k}{\sigma_0^2})}}}{B} \right)^2 \tag{34}$$

Substituting Eq. (34) into Eq. (33), the optimal spectrum resource allocated to TD k can be denoted as

$$B_k^* = \frac{\sqrt{\frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{\log_2(1 + \frac{h_k p_k}{\sigma_0^2})}}}{\sum_{k=1}^{N} \sqrt{\frac{(\alpha_k^t + \alpha_k^e P_k) D_k}{\log_2(1 + \frac{h_k p_k}{\sigma_0^2})}}} B \tag{35}$$

The pseudocodes for obtaining the optimal solutions to B_k and λ are summarized in Algorithm 1.

Algorithm 1: The optimal solutions to B_k and Lagrange Multiplier λ

Input:

1: The total spectrum resources B and the Lagrange multiplier λ.
2: While the end condition is not satisfied
3: Get the partial derivative of L concerning B_k according to Eq. (30);
4: Based on Eqs. (30) and (31) to know that $\lambda > 0$;
5: From Eq. (31) to get Eq. (32);
6: Solving Eq. (30) to get the expression of B_k, which is a function of λ;
7: Based on Eqs. (32) and (33) to get the value of λ, which is denoted in Eq. (34);
8: According to Eqs. (32) and (33) to get the optimal bandwidth allocation B_k^*.

4.3 Offloading Strategy Algorithm

An algorithm for the offloading strategy of each TD is presented, as shown in Algorithm 2. This algorithm provides an optimal offloading strategy for each DT. In this algorithm, each TD initially sets their offloading strategy as $a_k = 0$, which means the local computing model. Then, each TD calculates the optimal computing capacity f_k^{l*} by solving Problem 2, and sends the data size of its task D_k, computing intensity C_k, and transmission power p_k to edge cloud server. The edge cloud server calculates the transmission data rate based on Eq. (5), and determines their offloading strategies by making comparisons between the computing costs of the local computing model and that of the secured model,

$$a_k = \begin{cases} 1, & C_k^l \geq C_k^o \\ 0, & C_k^l < C_k^o \end{cases} \tag{36}$$

When the computing costs for all TDs are obtained, the computing cost of TD k can be denoted as

$$C_k = (1 - a_k)C_k^l + a_k C_k^o \tag{37}$$

Then, the overall computing costs of whole MEC system can be minimized.

Algorithm 2: The proposed security-aware task offloading strategy

Input:

1: Each TD initially sets its offloading strategy as $a_k = 0$, the data size of its task D_k, computing intensity C_k, and transmission power p_k.

Output:

2: The allocated bandwidth, the CPU frequency, the offloading strategy, and computing costs;
3: Solving subproblem Problem 2;
4: Obtaining the computing costs of the local computing model based on Eq. (4);
5: Solving the subproblem Problem 3;
6: Obtaining the optimal spectrum of each TD based on Eq. (35);
7: Obtaining the computing costs of the secured model based on Eq. (37);
8: **if** $C_k^l \geq C_k^0$ **then**
9: $a_k = 1$;
10: **else**
11: $a_k = 0$;
12: **end if**

5 Simulation Results

The efficacy of the proposed secured model and task offloading algorithm is validated through simulation results. Besides, the performance is compared against the following benchmark models:

Local Computing (LC) Model: In the LC Model, all TDs compute their tasks in themselves, that is $a_k = 0$.

Cloud Server Computing (CSP) Model: In the CSP Model, all TDs compute their tasks on edge cloud server, that is $a_k = 1$.

Unsecured Model: In the Unsecured Model, all TDs make offloading decisions according to the current system environment and do not adopt the security layer.

Secured Model: In the Secured Model, all TDs make offloading decisions according to the current system environment, and adopt the security layer to protect the data security of tasks.

5.1 Simulation Parameters

Assume that there is an MEC system that includes $N = 20$ TDs and one edge cloud server. The number of computing resources allocated to TD k by the edge cloud server F_k is 1 GHz. Each TD has a task to be computed using its computing resources or the cloud server's. The task data size ranges from 0.2 to 1 Mbits [29,32]. The computing intensity C_k and the total bandwidth B are 1000 cycles one bit and 5 MHz, respectively. The transmission power p_k and Gaussian noise σ_0^2 are respectively set as 0.2 W and 10^{-7} W. The number of computing resources to encrypt and decrypt the TD k's data C_k^e and C_k^d are both set as 100 megacycles. Besides, the channel gain is set as 10^{-6}, and the default values of α_k^t and α_k^e are set as 0.5. The set of values for the parameters is referring to [29,33–35] and listed in Table 1.

Table 1: Parameter setting

Parameters	Values
TD's number N	30
Bandwidth B	5 MHz
Task data size D_k	[0.1, 1] Mbits
Computing intensity C_k	1000 Cycles/bit
Transmission power p_k	0.1 W
Channel gain h_k	10^{-6}
Gaussian noise σ_0^2	10^{-7} W
Computing capacity f_k^l	[0.1, 1] GHz
Allocated computing resources to TD k F_k	1 GHz
The number of computing resources to encrypt TD k's data C_k^e	0.1 GHz
The number of computing resources to decrypt TD k's data C_k^d	0.1 GHz
Weighting value α_k^t	0.5
Weighting value α_k^e	0.5

We set up a MEC system environment by using the simulation tool MATLAB. The simulation steps are shown as follows:

Step 1: From solving Problem 2 to get the optimal solution of local computing resource f_k^{l*}, from which we can get the optimal CPU frequency for each TD.

Step 2: From solving Problem 3.1, we have the optimal CPU frequency for each TD in Secured Model.

Step 3: From solving Problem 3.2, we have optimal spectrum resource allocation for each TD.

Step 4: From the Eqs. (4), (11), and (16), we can have total costs under the local computing model, unsecured model, and secured model, respectively.

Step 5: From Algorithm 1, we get task offloading decision for each TD.

Step 6: System costs can be obtained based on Eq. (37).

5.2 Simulation Results

Firstly, we analyze the offloading strategies of TD users vs. the number of TDs under the secured and unsecured models. The number of spectrum resources varies from 1 to 5 WHz. Fig. 2 plots the impact of the number of spectrum resources on the offloading percentage. The offloading percentage is the ratio of TD users who offload their tasks to the total number of TD users. As depicted in Fig. 2, the offloading percentage of TD users is very small when the number of spectrum resources is less than 1 WHz. However, the offloading percentage increases rapidly as the number of spectrum resources increases. A clear observation can be made that the offloading percentage is larger under the Unsecured Model compared with the Secured Model.

We next analyze the impact of the number of spectrum resources on task energy consumption, computing time, and the total costs of TDs. We show the results in Figs. 3–5. In Fig. 3, we show the impact of the number of spectrum resources on task energy consumption under the (a) Unsecured Model and (b) Secured Model, respectively. As the number of spectrum resources has no impact on the task computing in the Local Model, the energy consumption remains constant. By comparing the energy consumption among the CSP Model, Unsecured Model, and Secured Model, it can be obviously observed that TDs

consume more energy in the Secured Model than the other two models. Besides, the CSP Model has the least energy consumption.

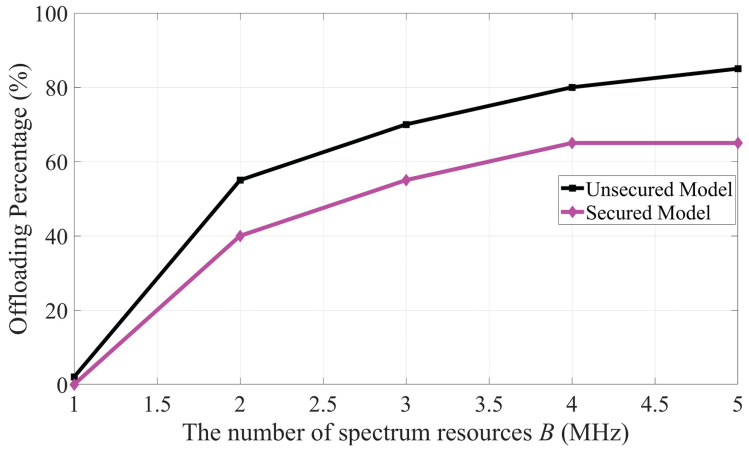

Figure 2: The impact of the number of spectrum resources on offloading percentage

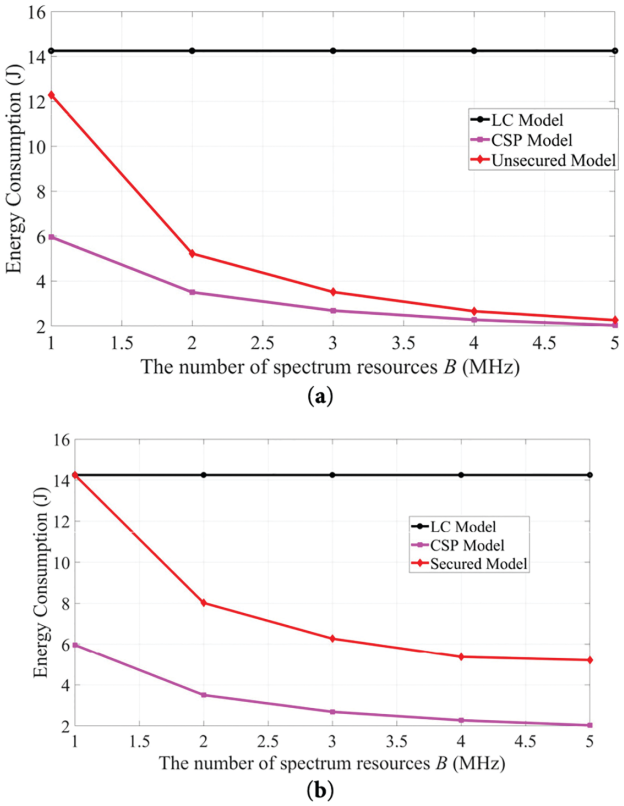

Figure 3: The impact of the number of spectrum resources on energy consumption. (**a**) Unsecured model; (**b**) Secured model

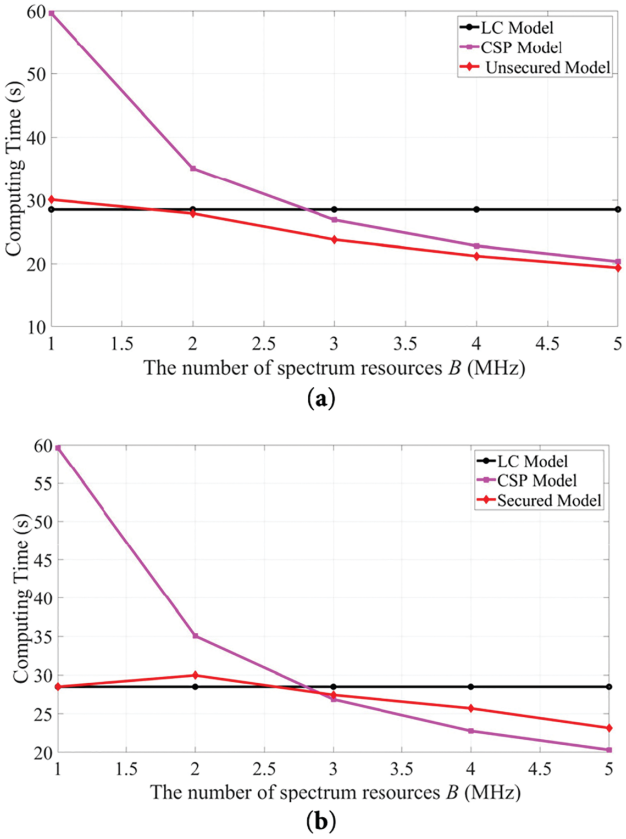

Figure 4: The impact of the number of spectrum resources on computing time. (**a**) Unsecured model; (**b**) Secured model

In Fig. 4, we show the impact of the number of spectrum resources on task computing time under the (a) Unsecured Model and (b) Secured Model, respectively. As TDs do not need spectrum resources in the Local Model, the computing time in this environment remains unchanged. By comparing the computing time among the CSP Model, Unsecured Model, and Secured Model, we can have the following observations. First, when the number of spectrum resources is less than 2 MHz, TDs will experience more computing time in the CSP Model. Second, with the number of spectrum resources increasing, computing TDs' tasks will need more time in the Secured Model than the other two models. Additionally, it can be observed that the TDs in the Local Model have the largest computing time when the number of spectrum resources increases. This indicates that TDs are inclined to offload their tasks to the edge cloud under the condition that the number of spectrum resources is satisfied.

In Fig. 5, the impact of the number of spectrum resources on overall computing costs concerning computing time and energy consumption under the scenario of the Secured Model and Unsecured Model is illustrated and analyzed. For the Local Model, as the spectrum resources have no impact on the computing cost, the trend of computing cost remains constant under this model. For the CSP Model, it can be found that the overall computing cost has an evident reduction when the number of spectrum resources increases. When the spectrum resources increase, there will be a decrease in the transmission time between the TDs and the edge cloud server, which ultimately influences the overall computing cost. For the Unsecured Model, it can be seen that the overall computing cost decreases as the number of spectrum resources increases. But the computing cost is low when $B = 1$ MHz compared with $B = 2$ MHz or $B = 3$ MHz. The reason for this case is that the scarcity of communication resources makes the TD users compute their tasks locally. The answer

to this reason can also be obtained from Fig. 2. As far as the Secured Model is concerned, the computing cost is larger compared with the CSP Model and the Unsecured Model. This is because computing the tasks of TDs needs more computing time and consumes more energy, which leads to an increase in the overall computing cost. Looking at Fig. 5, we can also observe that computing cost tends to decrease. An illustration for this reason is that the increase in communication resources can lead to a reduction of computing time and energy consumption. Another trend that can be observed is that the computing cost under the Local Model is much higher than the other three computing models when the number of spectrum resources increases. From Fig. 5, it can be obviously observed that adopting the AES technique will put extra energy consumption, computing time, and computing cost on TDs. Therefore, TDs have to make choices among the Local Model, Unsecured Model, and Secured Model.

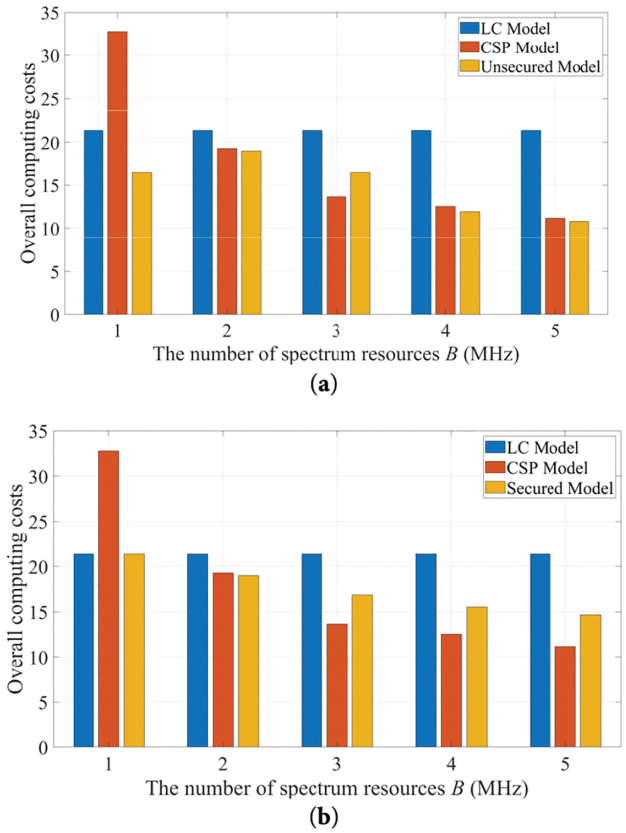

Figure 5: The impact of the number of spectrum resources on overall computing costs. (**a**) Unsecured model; (**b**) Secured model

6 Conclusions

In this work, we have investigated the optimization of spectrum allocation and task offloading in an MEC system. The DVS technology is applied by TDs to reduce the energy consumption and computing time. To guarantee data security of tasks during offloading, the AES cryptographic technique is used. We formulate an optimization problem, whose objective function is not convex. It is solved by the proposed algorithm. The simulation results are presented to show their efficacy. Compared to benchmark methods, it is verified that the overall computing cost can be reduced while guaranteeing data security.

As Deep Neural Networks (DNNs) can resolve the challenges of task offloading when applied to the traditional numerical optimization methods [36], DNNs can be applied in computation offloading in future work. In the future, we will also consider spectrum allocation and task offloading in 6G wireless networks, where machine learning tools, such as DRL, will be adopted to optimize spectrum resources, and RIS will be applied to improve the network performance.

Acknowledgement: The authors would like to thank the reviewers and assistant editor for their beneficial comments and suggestions.

Funding Statement: This work is supported in part by Key Scientific Research Projects of Colleges and Universities in Anhui Province (2022AH051921), Science Research Project of Bengbu University (2024YYX47pj, 2024YYX48pj), Anhui Province Excellent Research and Innovation Team in Intelligent Manufacturing and Information Technology (2023AH052938), Big Data and Machine Learning Research Team (BBXYKYTDxj05), Funding Project for the Cultivation of Outstanding Talents in Colleges and Universities (gxyqZD2021135), the Key Scientific Research Projects of Anhui Provincial Department of Education (2022AH051376), Start Up Funds for Scientific Research of High-Level Talents of Bengbu University (BBXY2020KYQD02), Scientific Research and Development Fund of Suzhou University (2021fzjj29), Research on Grain Logistics Data Processing and Safety Issues (ALAQ202401017), and the Open Fund of State Key Laboratory of Tea Plant Biology and Utilization (SKLTOF20220131). This work was funded by the Ongoing Research Funding Program (ORF-2025-102), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and design: Xianwei Li, Bo Wei, Xiaoying Yang, Amr Tolba, Osama Alfarraj; data collection: Xianwei Li, Xiaoying Yang, Zijian Zeng; analysis and interpretation of results: Xianwei Li, Xiaoying Yang, Zijian Zeng, Osama Alfarraj; draft manuscript preparation: Xianwei Li, Bo Wei, Amr Tolba, Osama Alfarraj. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Readers can access the data used in the study from the corresponding authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Shi T, Cai Z, Li J, Gao H, Qiu T, Qu W. An efficient processing scheme for concurrent applications in the IoT edge. IEEE Trans Mob Comput. 2024;23(1):135–49. doi:10.1109/TMC.2022.3219983.
2. Vaezi M, Azari A, Khosravirad SR, Shirvanimoghaddam M, Azari MM, Chasaki D, et al. Cellular, wide-area, and non-terrestrial IoT: a survey on 5G advances and the road toward 6G. IEEE Commun Surv Tutor. 2022;24(2):1117–74. doi:10.1109/comst.2022.3151028.
3. Satyanarayanan M. The emergence of edge computing. Computer. 2017;50(1):30–9. doi:10.1109/MC.2017.9.
4. Fan Q, Bai J, Zhang H, Yi Y, Liu L. Delay-aware resource allocation in fog-assisted IoT networks through reinforcement learning. IEEE Internet Things J. 2022;9(7):5189–99. doi:10.1109/JIOT.2021.3111079.
5. Samy A, Elgendy IA, Yu H, Zhang W, Zhang H. Secure task offloading in blockchain-enabled mobile edge computing with deep reinforcement learning. IEEE Trans Netw Serv Manag. 2022;19(4):4872–87. doi:10.1109/TNSM.2022.3190493.
6. Kuang Z, Pan Y, Yang F, Zhang Y. Joint task offloading scheduling and resource allocation in air-ground cooperation UAV-enabled mobile edge computing. IEEE Trans Veh Technol. 2024;73(4):5796–807. doi:10.1109/TVT.2023.3334143.
7. Liu Z, Qi J, Shen Y, Ma K, Guan X. Maximizing energy efficiency in UAV-assisted NOMA-MEC networks. IEEE Internet Things J. 2023;10(24):22208–22. doi:10.1109/JIOT.2023.3303491.
8. Hossain MA, Liu W, Ansari N. Computation-efficient offloading and power control for MEC in IoT networks by meta-reinforcement learning. IEEE Internet Things J. 2024;11(9):16722–30. doi:10.1109/JIOT.2024.3355023.

9. Li S, Sun W, Sun Y, Huo Y. Energy-efficient task offloading using dynamic voltage scaling in mobile edge computing. IEEE Trans Netw Sci Eng. 2021;8(1):588–98. doi:10.1109/TNSE.2020.3046014.
10. Wang Z, Hu Q, Xiong Z, Liu Y, Niyato D. Resource optimization for blockchain-based federated learning in mobile edge computing. IEEE Internet Things J. 2024;11(9):15166–78. doi:10.1109/JIOT.2023.3347524.
11. Yang Y, Hu Y, Gursoy MC. Energy efficiency of RIS-assisted NOMA-based MEC networks in the finite blocklength regime. IEEE Trans Commun. 2024;72(4):2275–91. doi:10.1109/TCOMM.2023.3334811.
12. Ernest TZH, Madhukumar AS. Computation offloading in MEC-enabled IoV networks: average energy efficiency analysis and learning-based maximization. IEEE Trans Mob Comput. 2024;23(5):6074–87. doi:10.1109/tmc.2023.3315275.
13. Mei J, Tong Z, Li K, Zhang L, Li K. Energy-efficient heuristic computation offloading with delay constraints in mobile edge computing. IEEE Trans Serv Comput. 2023;16(6):4404–17. doi:10.1109/TSC.2023.3324604.
14. Chen H, Todd TD, Zhao D, Karakostas G. Wireless and service allocation for mobile computation offloading with task deadlines. IEEE Trans Mob Comput. 2023;23(5):5054–68. doi:10.1109/TMC.2023.3301577.
15. Bi J, Wang Z, Yuan H, Zhang J, Zhou M. Cost-minimized computation offloading and user association in hybrid cloud and edge computing. IEEE Internet Things J. 2024;11(9):16672–83. doi:10.1109/JIOT.2024.3354348.
16. Panda SK, Lin M, Zhou T. Energy-efficient computation offloading with DVFS using deep reinforcement learning for time-critical IoT applications in edge computing. IEEE Internet Things J. 2023;10(8):6611–21. doi:10.1109/JIOT.2022.3153399.
17. Xu Y, Zhang T, Loo J, Yang D, Xiao L. Completion time minimization for UAV-assisted mobile-edge computing systems. IEEE Trans Veh Technol. 2021;70(11):12253–9. doi:10.1109/TVT.2021.3112853.
18. Wang H, Li W, Sun J, Zhao L, Wang X, Lv H, et al. Low-complexity and efficient dependent subtask offloading strategy in IoT integrated with multi-access edge computing. IEEE Trans Netw Serv Manag. 2024;21(1):621–36. doi:10.1109/TNSM.2023.3295653.
19. Chen CL, Brinton CG, Aggarwal V. Latency minimization for mobile edge computing networks. IEEE Trans Mob Comput. 2023;22(4):2233–47. doi:10.1109/TMC.2021.3117511.
20. Hossain MA, Ansari N. Energy aware latency minimization for network slicing enabled edge computing. IEEE Trans Green Commun Netw. 2021;5(4):2150–9. doi:10.1109/TGCN.2021.3083153.
21. Pervez F, Sultana A, Yang C, Zhao L. Energy and latency efficient joint communication and computation optimization in a multi-UAV-assisted MEC network. IEEE Trans Wirel Commun. 2023;23(3):1728–41. doi:10.1109/TWC.2023.3291692.
22. Younis A, Maheshwari S, Pompili D. Energy-latency computation offloading and approximate computing in mobile-edge computing networks. IEEE Trans Netw Serv Manag. 2024;21(3):3401–15. doi:10.1109/TNSM.2024.3360850.
23. Zhao T, Liu Y, Shou G, Yao X. Joint optimization of latency and energy consumption for mobile edge computing based proximity detection in road networks. China Commun. 2022;19(4):274–90. doi:10.23919/jcc.2022.04.020.
24. Xu W, Yu J, Wu Y, Tsang DHK. Energy-latency aware intelligent reflecting surface aided multi-cell mobile edge computing. IEEE Trans Green Commun Netw. 2023;8(1):362–74. doi:10.1109/TGCN.2023.3330247.
25. Mustafa E, Shuja J, Rehman F, Riaz A, Maray M, Bilal M, et al. Deep neural networks meet computation offloading in mobile edge networks: applications, taxonomy, and open issues. J Netw Comput Appl. 2024;226(6):103886. doi:10.1016/j.jnca.2024.103886.
26. Huo Y, Liu Q, Gao Q, Wu Y, Jing T. Joint task offloading and resource allocation for secure OFDMA-based mobile edge computing systems. Ad Hoc Netw. 2024;153(1):103342. doi:10.1016/j.adhoc.2023.103342.
27. Wu Y, Ji G, Wang T, Qian L, Lin B, Shen X. Non-orthogonal multiple access assisted secure computation offloading via cooperative jamming. IEEE Trans Veh Technol. 2022;71(7):7751–68. doi:10.1109/TVT.2022.3167861.
28. Zheng TX, Chen X, Wen Y, Zhang N, Ng DWK, Al-Dhahir N. Secure offloading in NOMA-enabled multi-access edge computing networks. IEEE Trans Commun. 2024;72(4):2152–65. doi:10.1109/tcomm.2023.3342242.
29. Elgendy IA, Zhang W, Tian YC, Li K. Resource allocation and computation offloading with data security for mobile edge computing. Future Gener Comput Syst. 2019;100(7):531–41. doi:10.1016/j.future.2019.05.037.

30. Zhang H, Wang J, Zhang H, Bu C. Security computing resource allocation based on deep reinforcement learning in serverless multi-cloud edge computing. Future Gener Comput Syst. 2024;151(3):152–61. doi:10.1016/j.future.2023.09.016.
31. Lyu X, Tian H, Ni W, Zhang Y, Zhang P, Liu RP. Energy-efficient admission of delay-sensitive tasks for mobile edge computing. IEEE Trans Commun. 2018;66(6):2603–16. doi:10.1109/TCOMM.2018.2799937.
32. Boyd S, Vandenberghe L. Convex optimization. Cambridge, UK: Computational Cambridge University Press; 2004.
33. Chen X, Jiao L, Li W, Fu X. Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans Netw. 2016;24(5):2795–808. doi:10.1109/TNET.2015.2487344.
34. Lu H, He X, Zhang D. Security-aware task offloading using deep reinforcement learning in mobile edge computing systems. Electronics. 2024;13(15):2933. doi:10.3390/electronics13152933.
35. Yang Y, Yu H, Zhao Y, Chen M, Du J, Ren Y. A dynamic-pricing-based offloading and resource allocation scheme with data security for vehicle platoon. IEEE Internet Things J. 2024;12(6):7149–63. doi:10.1109/JIOT.2024.3492694.
36. Mustafa E, Shuja J, Rehman F, Namoun A. Deep reinforcement learning and SQP-driven task offloading decisions in vehicular edge computing networks. Comput Netw. 2025;262:1–13. doi:10.1016/j.comnet.2025.111180.