Online Edge Learning Offloading and Resource Management for UAV-Assisted MEC Secure Communications

Yu Ding, Yunqi Feng, Weidang Lu, *Senior Member, IEEE*, Shilian Zheng, *Member, IEEE*, Nan Zhao, *Senior Member, IEEE*, Limin Meng, Arumugam Nallanathan, *Fellow, IEEE* and Xiaoniu Yang

***Abstract*—The mobile and flexible unmanned aerial vehicle (UAV) with mobile edge computing (MEC) can effectively relieve the computing pressure of the massive data traffic in 5G Internet of Things. In this paper, we propose a novel online edge learning offloading (OELO) scheme for UAV-assisted MEC secure communications, which can improve the secure computation performance. Moreover, the problem of information security is further considered since the offloading information of terminal users (TUs) may be eavesdropped due to the light-of-sight characteristic of UAV transmission. In the OELO scheme, we maximize the secure computation efficiency by optimizing TUs' binary offloading decision and resource management while guaranteeing dynamic task data queue stability and minimum secure computing requirement. Since the optimization problem is fractionally structured, binary constrained and multi-variable coupled, we first utilize the Dinkelbach method to transform the fractionally structured problem into a tractable form. Then, OELO generates the offloading decision based on deep reinforcement learning (DRL) and optimizes the resource management in an iterative manner through successive convex approximation (SCA). Simulation results show that the proposed scheme achieves better computing performance and enhances the stability and security compared with benchmarks.**

***Index Terms*—UAV, MEC, secure computation efficiency, online edge learning offloading, resource management.**

I. Introduction

The developing evolution of 5G Internet of Things (5GIoT) has brought great convenience to human life and production, which also causes tremendously data traffic [1]–[3]. To quickly process the massive data of 5GIoT, mobile edge computing (MEC) is an efficient method to offload the data traffic to the edge for computation, which can effectively alleviate the pressure of data growth and improve the data computing efficiency [4]–[10]. Mao *et al.* investigated that MEC can assist terminal users (TUs) with limited resource to enable latency-critical and computation-intensive applications, and promote the achievement of 5GIoT [7]. Li *et al.* studied the online trusted collaboration offloading method for MEC systems by considering the cooperative trust risk and the variation in completion delays [8]. A model-free configuration scheme to achieve the MEC system's quality of service (QoS) was proposed by in Zhao *et al.* [9]. Guo *et al.* investigated that MEC can overcome the limitation of the multi-user system and increase the computation capability [10].

With the advantage of high mobility and flexibility, unmanned aerial vehicles (UAVs) can assist the MEC systems in 5GIoT to rapidly deploy in computation-intensive areas, and improve the quality of wireless communication links. UAV-assisted MEC system can not only extend the coverage of MEC computing service, but also save installation costs [11]–[13]. Qi *et al.* in [11] investigated the efficiency enhancement of the UAV-assisted MEC system by optimizing UAV trajectory and resource management. Han *et al.* investigated that UAV-assisted MEC can enhance the computation capability performance in IoT systems [12]. Du *et al.* investigated a resource management method by considering the UAV computing ability and the basic QoS demands of 5GIoT systems [13].

Although UAV-assisted MEC systems can provide high computation performance for multiple TUs, the offloading information of TUs may be more easily eavesdropped during the light-of-sight (LoS) transmission. Physical layer security can provide secure wireless communication through utilizing communication channels and data information transmitting modes [14]–[18]. Lu *et al.* investigated the secure computation capacity optimization in the UAV-MEC system, where one UAV assists TUs in offloading while the other UAV eavesdrops TUs offloading task data [14]. Zhao *et al.* proposed a secure optimization scheme to adjust the resource management for secure UAV-assisted MEC systems, where one UAV acts as an eavesdropper to threaten the data offloading [15]. Bai *et al.* proposed a secure computation enhancement scheme in UAV-MEC systems with passive and active eavesdroppers [16]. Lu *et al.* studied a secure computation capacity improvement scheme through considering the uncertain position of the eavesdropping UAV in NOMA-based UAV-assisted MEC systems [18].

However, most of the above works require multiple iterations to solve the formulated problem, which requires high computational complexity. Edge learning offers an alternative

Yu Ding, Yunqi Feng, Weidang Lu, Limin Meng and Xiaoniu Yang are with College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: 2112003309@zjut.edu.cn, yqfeng@zjut.edu.cn, luweid@zjut.edu.cn, mlm@zjut.edu.cn, yxn2117@126.com).

Shilian Zheng is with the Science and Technology on Communication Information Security Control Laboratory, Jiaxing 314000, China (e-mail: lianshizheng@126.com).

Nan Zhao is with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, P. R. China (e-mail: zhaonan@dlut.edu.cn).

Arumugam Nallanathan is with School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, U.K. (e-mail: a.nallanathan@qmul.ac.uk).

way to solve the problem with real-time online offloading computing, which can satisfy some online services with high real-time requirement, e.g., football live broadcast [19]–[21]. Lim *et al.* pointed out that the convergence of artificial intelligence (AI) and edge computing has facilitated the development of edge learning and the efficient deployment of AI [20]. Yang *et al.* proposed a multi-agent resource management scheme with privacy-preserving asynchronous federated learning framework for a multi-UAV-enabled network [21]. Deep reinforcement learning (DRL) is utilized in [22]–[28] to generate online offloading actions from time-varying system parameters. Jiang *et al.* investigated an effective online offloading scheme for MEC systems, which includes resource management, position optimization and user association [22]. Liu *et al.* studied the DRL-based offloading decision and resource optimization to improve the computational performance of UAV-MEC systems [23]. Chen *et al.* in [24] studied a multi-task DRL method to allocate the MEC system resource considering the task characteristics and the MEC state. Bi *et al.* investigated a joint optimizing scheme on offloading decision generation and resource management based on DRL to maximize the computation rate [25]. Min *et al.* proposed a learning-based offloading method to achieve good computation performance with low complexity for 5GIoT systems in [26]. Nguyen *et al.* proposed a parallel learning method to reduce the energy consumption and latency of multi-UAV systems [27]. Wang *et al.* proposed an online decision generation scheme with lower complexity in the randomly time-varying UAV-assisted MEC system based on DRL [28].

However, to our best knowledge, online edge learning offloading (OELO) for improving the secure computation performance in the UAV-assisted MEC system is not studied. Thus, in this paper, we propose a novel OELO scheme for the UAV-assisted MEC secure communication system. In the proposed OELO scheme, we maximize the secure computation efficiency through optimizing the binary offloading decision and resource management of TUs, including time allocation, local computation allocation and power allocation, under the conditions of guaranteeing the stability of dynamic task data queues and the minimum secure computing requirement. The key advantages of the proposed OELO are that OELO not only achieves better secure computation efficiency with lower complexity, but also maintains the stability of the task data queues with random dynamic change in the UAV-assisted MEC secure communication system. The main contributions are summarized as follows.

- We propose a UAV-assisted MEC secure communication scheme, in which the task data queues of TUs change dynamically in each time frame. The UAV Server (USV) assists TUs in computing the offloading task data, and the UAV Eavesdropper (UEV) eavesdrops on the offloading data information of TUs during the flight. In order to disturb the eavesdropping of UEV, ground jammer (GJ) generates jamming signals.
- By considering the constraints of peak transmit power of TUs, the computing capacity of USV and TUs, the minimum secure computing requirements, the causality of TUs' task data queues computation, and the stability of TUs' task data queues, we formulate an optimization problem to maximize the secure computation efficiency of the UAV-assisted MEC secure communication system by optimizing binary offloading decision and resource management for TUs, which includes time allocation, local computation allocation and power allocation.
- The optimization problem is fractionally structured, binary constrained and multivariable coupled. We first transform the fractionally structured problem as a tractable form based on Dinkelbach method. Then, the proposed OELO scheme learns offloading decision from past experience and utilizes the deep neural network (DNN) and noisy order-preserving (NOP) to generate TUs' offloading decision, and resource management is optimized in an iterative manner through SCA.

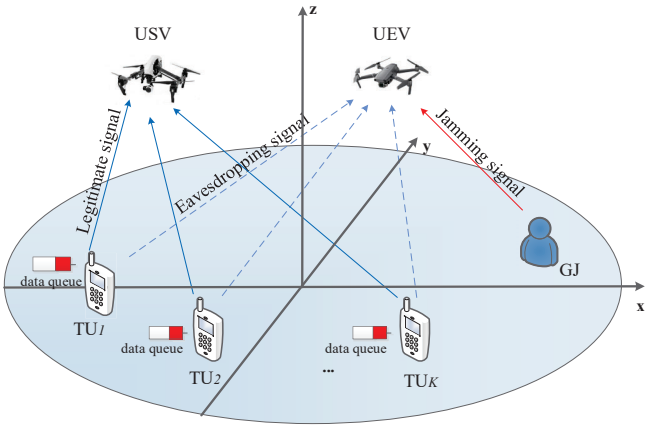

Fig. 1. The UAV-assisted MEC secure communication system model.

The rest of the paper is summarized as follows. Section II introduces the UAV-assisted MEC secure communication system model. Section III formulates the secure computation efficiency maximization problem. Section IV studies the optimization for maximizing secure computation efficiency. Section V presents simulation results. Simulation results are presented in Section V. Section VI presents the conclusion of this paper.

II. System Model

Fig. 1 shows the considered UAV-assisted MEC secure communication system, where K TUs have randomly dynamic task data queues arriving in each frame. Mobile USV assists TUs in task information computation. Mobile UEV eavesdrops on the task information sent by TUs to USV. In order to disturb UEV eavesdropping behavior, GJ broadcasts artificial interference signal. Since both GJ and USV belong to the legitimate network, USV knows GJ's jamming signal in advance, USV can subtract the artificial interference signal from its received signals to ensure that it will not be affected by the interference signal [14]. However, UEV does not know the presence of GJ. It will regard the signals it received as the offloading signals sent by TUs. Thus, the signals broadcasted by GJ will cause the interference to UEV.

TABLE I
NOTATIONS

Notation	Description
σ_t	The frame size
$q_s[n]$	Location of USV
$q_e[n]$	Location of UEV
H_s	Flying Altitudes of USV
H_e	Flying Altitudes of UEV
K	Number of TUs
w_j	Location of GJ
w_k	Location of TU_k
$A_k[n]$	Arrival task data queue of TU_k
G	Rayleigh factor
β_0	Path loss at reference distance
P_j	GJ transmit power
$P_{\max}$	Peak transmit power of TUs
B	Bandwidth of transmit links
$p_k[n]$	Transmit power of TU_k
$\tau_k[n]$	Time allocating factor of TU_k
σ_s^2, σ_e^2	Noise power received at USV and UEV
$\gamma_k[n]$	Offloading decision of TU_k
c_k	CPU cycles required for TU_k to compute one bit task data
$l_{loc,k}[n]$	Local computing bits of TU_k
$F_k^{\max}$	TU_k maximum CPU computation frequency
c_s	CPU cycles required for USV to compute one bit task data
$F_s^{\max}$	USV maximum CPU computation frequency
v_u	Communication overhead
Q_m	Each TU minimum secure computing requirement
k_k	TU_k CPU capacity coefficient
$S_{\max}$	Maximum capacity of TUs' data queue storage

Define the horizonal coordinates of USV and UEV with continuous time t as $q_s(t)$ and $q_e(t)$. Assume that USV and UEV perform their respective tasks by flying at the fixed altitudes H_s and H_e. For the sake of discussion, we discretize continuous time, i.e., one time frame is σ_t. The horizonal coordinates of USV and UEV are $q_s[n] = (x_s[n], y_s[n])^T$ and $q_e[n] = (x_e[n], y_e[n])^T$ in the frame $n, n \in \{1, 2, ..., \infty\}$, respectively. The horizonal coordinates of GJ and $\mathrm{TU}_k, k \in \{1, 2, ..., K\}$ are defined as $w_j = (x_j, y_j)$ and $w_k = (x_k, y_k)$, respectively.

Note that TU_k has a new arrival task data in every frame. Define $A_k[n]$ as the arrival task data queue that reaches the original data queue of TU_k in frame n. It is assumed that the arrival task data queue follows a bounded second-order distribution, i.e., $E\left[(A_k[n])^2\right] = \psi_k < \infty, k = 1, ..., K$. The value of ψ_k is deterministic, which can be estimated from past observations. We assume that both of USV and UEV already know the location of all TUs and the channel status information of all channels in advance by means of synthetic aperture radar, etc. The detailed notations in this paper are shown in Table I.

In the frame $n, n \in \{1, 2, ..., \infty\}$, the distance between $\mathrm{TU}_k, k \in \{1, 2, ..., K\}$ and USV, TU_k and UEV, GJ and UEV are expressed as

$$d_{k,s}[n] = \sqrt{||w_k - q_s[n]||^2 + H_s^2}, \tag{1}$$

$$d_{k,e}[n] = \sqrt{||w_k - q_e[n]||^2 + H_e^2}, \tag{2}$$

$$d_{j,e}[n] = \sqrt{||w_j - q_e[n]||^2 + H_e^2}. \tag{3}$$

We consider the channel coefficient follows Rician fading channel model [11], the channel coefficient between TU_k and USV in frame n is given by

$$\begin{aligned} h_{k,s}[n] &= \sqrt{\frac{\beta_0}{(G+1)d_{k,s}^2[n]}}\left(\sqrt{G}e^{j\frac{2\pi}{\lambda}d_{k,s}[n]} + f_{k,s}[n]\right) \\ &= \sqrt{\frac{G}{G+1}}\sqrt{\frac{\beta_0}{||w_k - q_s[n]||^2 + H_s^2}}e^{j\theta_{k,s}[n]} \\ &+ \sqrt{\frac{1}{G+1}}\sqrt{\frac{\beta_0}{||w_k - q_s[n]||^2 + H_s^2}}f_{k,s}[n], \end{aligned} \tag{4}$$

where G represents the Rayleigh factor, β_0 represents the path loss at the reference distance of $1m$, $\theta_{k,s}[n] = \frac{2\pi}{\lambda}d_{k,s}[n]$ represents the phase of the LoS channel, λ represents the wavelength and $f_{k,s}[n] \sim CN(0, 1)$.

Similarly, the channel coefficients between TU_k and UEV, GJ and UEV are $h_{k,e}[n]$ and $h_{j,e}[n]$, which are expressed as

$$\begin{aligned} h_{k,e}[n] &= \sqrt{\frac{G}{G+1}}\sqrt{\frac{\beta_0}{||w_k - q_e[n]||^2 + H_e^2}}e^{j\theta_{k,e}[n]} \\ &+ \sqrt{\frac{1}{G+1}}\sqrt{\frac{\beta_0}{||w_k - q_e[n]||^2 + H_e^2}}f_{k,e}[n], \end{aligned} \tag{5}$$

$$\begin{aligned} h_{j,e}[n] &= \sqrt{\frac{G}{G+1}}\sqrt{\frac{\beta_0}{||w_j - q_e[n]||^2 + H_e^2}}e^{j\theta_{j,e}[n]} \\ &+ \sqrt{\frac{1}{G+1}}\sqrt{\frac{\beta_0}{||w_j - q_e[n]||^2 + H_e^2}}f_{j,e}[n]. \end{aligned} \tag{6}$$

In the frame n, the transmit power of TU_k, $p_k[n]$, cannot exceed to the peak transmit power $P_{\max}$,

$$0 \le p_k[n] \le P_{\max}. \tag{7}$$

III. PROBLEM FORMULATION

In the UAV-assisted MEC secure communication system, TUs offload task information to USV with time division multiple access (TDMA) mode, where one frame is divided into K sub-frames. Define $\tau_k[n]$ as the time allocating factor, the time allocated to TU_k is $\tau_k[n]\sigma_t$, which satisfies

$$0 \le \tau_k[n] \le 1, \tag{8a}$$

$$\sum_{k=1}^{K} \tau_k[n] \le 1. \tag{8b}$$

A. Communication Model

As mentioned above, USV can separate the interference signal broadcasted by GJ from the received signals since both GJ and USV belong to the legitimate network. However, UEV will regard the signals it received as useful signals since UEV does not know the presence of GJ. Thus, signal-to-interference-and-noise-ratio (SINR) of USV and UEV in frame n are written as

$$r_{k,s}[n] = \frac{p_k[n]|h_{k,s}[n]|^2}{\delta_s^2}, \forall k, n, \tag{9}$$

$$r_{k,e}[n] = \frac{p_k[n]|h_{k,e}[n]|^2}{P_j|h_{j,e}[n]|^2 + \delta_e^2}, \forall k, n, \tag{10}$$

where δ_s^2 and δ_e^2 are the Gaussian noise at UEV and USV, respectively. P_j represents GJ's transmit power.

Thus, the task data offloading rate from TU_k to USV in frame n is given by

$$R_{k,s}[n] = \tau_k[n]\log_2\left(1 + r_{k,s}[n]\right). \tag{11}$$

The task data eavesdropping rate from TU_k to UEV is given by

$$R_{k,e}[n] = \tau_k[n]\log_2\left(1 + r_{k,e}[n]\right). \tag{12}$$

Thus, the achievable secure computation rate from TU_k to USV is written as

$$R_{k,\mathrm{sec}}[n] = [R_{k,s}[n] - R_{k,e}[n]]^+, \tag{13}$$

where $[x]^+$ means $\max(x, 0)$.

B. Computation Model

In this paper, TUs adopt a binary offloading strategy. Define $\gamma_k[n]$ as the offloading decision of TU_k, which should satisfy the following constraint

$$\gamma_k[n] \in \{0, 1\}, \tag{14}$$

where $\gamma_k[n] = 0$ represents that TU_k chooses to execute the computing task data locally, and $\gamma_k[n] = 1$ represents that TU_k chooses to offload the task data to USV for computation.

Denote c_k as the CPU cycles required for TU_k to compute one bit task data, $l_{loc,k}[n]$ as the local computing bits of TU_k in frame n, and $F_k^{\max}$ as maximum CPU computation frequency of TU_k. The computing bits of TU_k locally cannot exceed its own maximum computing capacity, which should satisfy

$$c_k l_{loc,k}[n] \le F_k^{\max}\sigma_t, \forall k, n. \tag{15}$$

Furthermore, the energy consumption when TU_k computes locally in frame n is given by

$$E_{loc,k}[n] = \frac{\left(l_{loc,k}[n]\right)^3 {c_k}^3 k_k}{\sigma_t^2}, \forall k, n, \tag{16}$$

where k_k represents the effective capacitance coefficient of TU_k.

Denote c_s as the CPU cycles required for USV to compute one bit task data, $F_s^{\max}$ as maximum CPU computation frequency of USV. Similarly, the computing bits of USV cannot exceed USV's maximum computing capacity [18], [25], which should satisfy

$$c_s \frac{B\sigma_t R_{k,\mathrm{sec}}[n]}{v_u} \le F_s^{\max}\tau_k[n]\sigma_t, \forall k, n, \tag{17}$$

where v_u represents the communication overhead, and B is the bandwidth.

Furthermore, the energy consumption by TU_k in the transmission of offloading information in frame n is expressed as

$$E_{off,k}[n] = p_k[n]\tau_k[n]\sigma_t. \tag{18}$$

Therefore, the amount of secure computation bits and energy consumption of TU_k in frame n are written as

$$L_k[n] = \left(1 - \gamma_k[n]\right) l_{loc,k}[n] + \gamma_k[n]\frac{B\sigma_t R_{k,\mathrm{sec}}[n]}{v_u}, \tag{19}$$

$$E_k[n] = \left(1 - \gamma_k[n]\right) E_{loc,k}[n] + \gamma_k[n] E_{off,k}[n]. \tag{20}$$

Denote Q_m as minimum secure computing requirement of TUs in each frame. In order to ensure that TUs have a basic secure computing requirement, we have

$$L_k[n] \ge Q_m, \forall k, n. \tag{21}$$

Define $Q_k[n]$ as the task data queue of TU_k at the beginning of frame n. Thus, the dynamic queue of TU_k is given by

$$Q_k[n+1] = Q_k[n] - L_k[n] + A_k[n], \forall k, n. \tag{22}$$

According to the causality of the task data queue computation, we have

$$L_k[n] \le Q_k[n], \forall k, n. \tag{23}$$

To guarantee the stability of TUs' task data queues, the task data queues of TUs cannot be allowed to accumulate for a long time, which should satisfy

$$Q_k[n] \le S_{\max}, \forall k, n, \tag{24}$$

where $S_{\max}$ represents the maximum value of data queue storage in TU_k.

Thus, the long-term secure computation efficiency of the UAV-assisted MEC secure communication system is obtained as

$$U_{eff} = \sum_{k=1}^{K}\sum_{n=1}^{\infty}\left(\frac{L_k[n]}{E_k[n]}\right). \tag{25}$$

C. Problem Formulation

To maximize the long-term secure computation efficiency of the considered UAV-assisted MEC secure communication system, offloading decision $\gamma_k[n]$, time allocating factor $\tau_k[n]$, local computing bits $l_{loc,k}[n]$ and transmit power $p_k[n]$ are optimized. The original problem is formulated as

$$\begin{aligned}&(\mathrm{P1}): \max_{\{\gamma_k[n],\tau_k[n],p_k[n],l_{loc,k}[n]\}} U_{eff} \\ &s.t. (7), (8), (14), (15), (17), (21), (23), (24).\end{aligned} \tag{26}$$

The arrival task data queue $A_k[n]$ in frame n is random and unknown. It is difficult to satisfy the long-term computation constraints when offloading decision is optimized in each frame without knowing the future realizations of task data queue. Thus, we convert the optimization objective from long-term secure computation efficiency to secure computation efficiency $\eta[n] = \sum_{k=1}^{K}\left(\frac{L_k[n]}{E_k[n]}\right)$ in each frame, so as to approximately achieve the objective of maximizing the UAV-assisted MEC secure communication system's long-term secure computation efficiency [15].

Our optimization objective is to maximize the secure computation efficiency maximization of the UAV-assisted MEC secure communication system in each frame under the condition of ensuring the stability of dynamic task data queues

and the minimum secure computing requirement of TUs, the original problem (P1) is transformed as

$$(\mathrm{P2}): \max_{\{\gamma_k[n],\tau_k[n],l_{loc,k}[n],p_k[n]\}} \eta[n] \quad (27)$$
$$s.t. (7), (8), (14), (15), (17), (21), (23), (24).$$

IV. Optimization for Maximizing Secure Computation Efficiency

In this section, the secure computation efficiency optimization problem of the UAV-assisted MEC secure communication system will be solved by the proposed OELO. The problem (P2) is fractionally structured, binary constrained and multivariable coupled, which is intractable. To overcome this challenge, we first transform the fractionally structured problem as a tractable form based on Dinkelbach method [29]. Then, based on OELO, we decompose the transformed problem (P2) into two sub-problems, namely offloading decision generation and resource management. Specifically, OELO learns offloading decision from past experience and utilizes the DNN to generate relaxed offloading actions. Candidate decisions are generated based on the relaxed offloading actions through NOP. Resource management is optimized with given the candidate decisions in an iterative manner through SCA.

A. Dinkelbach-based Problem Reconstruction and Solution

Based on Dinkelbach method, the fractional problem can be transformed as a parametric programming form. Denote the maximum secure computation efficiency as $\eta^*[n]$ in frame n. We can obtain $\eta^*[n] = \max_{\{\gamma_k[n],\tau_k[n],l_{loc,k}[n],p_k[n]\}} \sum_{k=1}^{K} \left(\frac{L_k[n]}{E_k[n]}\right)$.

Lemma 1: The maximum secure computation efficiency of the UAV-assisted MEC secure communication system is obtained if and only if the following formula is established:

$$\max_{\{\gamma_k[n],\tau_k[n],l_{loc,k}[n],p_k[n]\}} \sum_{k=1}^{K} \left(L_k[n] - \eta^*[n]E_k[n]\right) = 0. \quad (28)$$

Proof: Please refer to [29]. ■

Problem (P2) is difficult to solve because the objective function of (P2) is fractional. In order to solve (P2) more flexibly, based on Lemma 1, parameter $\alpha[n]$ is introduced to transform the problem (P2) into (P3)

$$(\mathrm{P3}): \max_{\{\gamma_k[n],\tau_k[n],l_{loc,k}[n],p_k[n]\}} \sum_{k=1}^{K} \left(L_k[n] - \alpha[n]E_k[n]\right) \quad (29)$$
$$s.t. (7), (8), (14), (15), (17), (21), (23), (24).$$

Since objective function is non-convex, constraints (17), (21), (23) and (24) are also non-convex, and constraint (15) is binary and incoherent, (P3) is difficult to solve directly.

Therefore, we introduce the auxiliary variables $u, u_{1,k}[n], u_{2,k}[n]$ to simplify (P3), which is transformed as

$$(\mathrm{P4}): \max_{\{\gamma_k[n],\tau_k[n],l_{loc,k}[n],p_k[n]\}} u \quad (30a)$$
$$s.t. (7), (8), (14), (15),$$
$$u \le \sum_{k=1}^{K} \left(\left(1-\gamma_k[n]\right) l_{loc,k}[n] + \gamma_k[n] l_{off,k}[n]\right) \quad (30b)$$
$$- \alpha[n] \sum_{k=1}^{K} \left(E_k[n]\right), \forall k, n,$$
$$u_{1,k}[n] \le \log_2\left(1 + \frac{p_k[n]|h_{k,s}[n]|^2}{\sigma_s^2}\right), \forall k, n, \quad (30c)$$
$$u_{2,k}[n] \ge \log_2\left(1 + \frac{p_k[n]|h_{k,e}[n]|^2}{P_j|h_{j,e}[n]|^2 + \sigma_e^2}\right), \forall k, n, \quad (30d)$$
$$c_s l_{off,k}[n] \le F_s^{\max} \tau_k[n] \sigma_t, \forall k, n, \quad (30e)$$
$$\left(1-\gamma_k[n]\right) l_{loc,k}[n] + \gamma_k[n] l_{off,k}[n] \ge Q_m, \forall k, n, \quad (30f)$$
$$\left(1-\gamma_k[n]\right) l_{loc,k}[n] + \gamma_k[n] l_{off,k}[n] \le Q_k[n], \forall k, n, \quad (30g)$$
$$Q_k[n] - \left(\left(1-\gamma_k[n]\right) l_{loc,k}[n] + \gamma_k[n] l_{off,k}[n]\right) + A_k[n]$$
$$\le S_{\max}, \forall k, n, \quad (30h)$$

where $l_{off,k}[n] = \frac{\tau_k[n]\sigma_t B(u_{1,k}[n]-u_{2,k}[n])}{v_u}$. Variable u is introduced to represent the lower bound value of the objective $\sum_{k=1}^{K} \left(L_k[n] - \alpha[n]E_k[n]\right)$. Then, the optimization target is represented by u, which is expressed by (30a) and (30b). Variable $\tau_k[n]u_{1,k}[n]$ is introduced to represent the lower bound value of $R_{k,s}[n]$, which is expressed by (30c). Variable $\tau_k[n]u_{2,k}[n]$ is introduced to represent the upper bound value of $R_{k,e}[n]$, which is expressed by (30d). Then, the achievable secure computation rate $R_{k,\mathrm{sec}}[n]$ is expressed by $\tau_k[n](u_{1,k}[n]-u_{2,k}[n])$. Therefore, constraint (17), (21), (23) and (24) are rewritten as (30e), (30f), (30g) and (30h), respectively.

Note that problem (P4) is still non-convex. (P4) is firstly divided into two sub-problems to obtain the optimizing solution. A novel OELO scheme is proposed to solve (P4). According to the communication links of the UAV-assisted MEC secure communication system and the task data queues of TUs, M_n candidate offloading decision $\gamma_k^m[n], m \in \{1, 2, ..., M_n\}$ are generated by the DNN and NOP. Then, the corresponding resource management is optimized through SCA and block coordinate descent with given $\{\gamma_k^m[n]\}$, which includes time allocating factor $\{\tau_k[n]\}$, local computing bits $\{l_{loc,k}[n]\}$ and transmit power $\{p_k[n]\}$. Finally, we choose the corresponding offloading decision and resource management of the maximum target value as the joint optimization action in the frame n.

Denote $\xi[n] = \left\{h[n], \{Q_k[n]\}_{k=1}^{K}\right\}$ as the input parameter of OELO, which is consisted of the time-varying channel coefficient $h[n] = \left\{\{h_{k,s}[n], h_{k,e}[n]\}_{k=1}^{K}, h_{j,e}[n]\right\}$ and task data queue $\{Q_k[n]\}_{k=1}^{K}$. Denote $G\left(\gamma[n], \xi[n]\right)$ as the target value obtained by optimizing resource management with given offloading decision $\gamma[n]$ and the input parameter $\xi[n]$. The key point to solve problem (P4) is to find the optimized offloading

decision,

$$(\mathrm{P5}):(\gamma^*[n])=\underset{\{\gamma[n]\in\{0,1\}^K\}}{\arg\max}\ G\left(\gamma[n],\xi[n]\right), \tag{31}$$

where $\gamma[n]=\{\gamma_1[n],\gamma_2[n],...,\gamma_K[n]\}$. Moreover, it is worth noting that we need to compute 2^K candidate actions if we want to obtain the optimal offloading decision $(\gamma^*[n])$, which causes high complexity and seriously affects online offloading. Thus, we propose the OELO scheme to generate candidate offloading decision and optimize resource management with low complexity.

B. Algorithm Description

The proposed OELO algorithm is shown in Fig. 2. We observe that the relaxed offloading action $\hat{\gamma}[n]=\{\hat{\gamma}_k[n]\}_{k=1}^K$ can be generated by the DNN. We quantize $\hat{\gamma}[n]$ into M_n binary actions through NOP. Then, an efficient method is proposed to obtain the optimizing resource management $y[n]=\{\tau_k[n],l_{loc,k}[n],p_k[n]\}_{k=1}^K$.

OELO is consisted by four modules. Actor module takes $\xi[n]$ as its input and outputs M_n candidate offloading actions $\gamma[n]=\{\gamma_k[n]\}_{k=1}^K$. Critic module selects the best offloading decision $\gamma^*[n]$ through substituting the candidate actions into the objective function. Policy update module updates the DNN's training set over frame. To improve the secure computation efficiency of the system, queueing module updates the task data queue state $\{Q_k[n]\}_{k=1}^K$ of the system after preforming offloading actions and resource management. The details of the four modules are as follows.

Actor module: Actor module includes the DNN and the quantizer for candidate actions. We aim to design an offloading decision function $\pi_{\theta[n]}\left(\xi[n]\right)$ to generate the optimized offloading action $\gamma[n]\in\{0,1\}^K$ quickly. The initialized parameters of the DNN $\theta[1]$ is randomly obtained, which follows standard normal distribution. In frame n, the relaxed offloading action function is denoted as

$$\pi_{\theta[n]}:\xi[n]\rightarrow\hat{\gamma}[n]. \tag{32}$$

The sigmoid function is utilized as the output layer of the DNN. Thus, the output, the relaxed offloading action, is continuous and follows $[0,1]^K$. We adopt NOP to quantify offloading actions, which can generate $M_n, M_n\leq 2N$ candidate decisions. Specifically, the first candidate decision $\gamma_1[n]=[\gamma_{1,1}[n],\gamma_{1,2}[n],...,\gamma_{1,K}[n]]$ is given by

$$\gamma_{1,k}=\begin{cases}1,\ \hat{\gamma}_k[n]>0.5\\ 0,\ \hat{\gamma}_k[n]\leq 0.5\end{cases} \tag{33}$$

To generate the next $\frac{M_n}{2}-1$ decisions, we first sort the distance between $\hat{\gamma}_k[n]$ and 0.5, i.e., $|\hat{\gamma}_{(1)}[n]-0.5|\leq|\hat{\gamma}_{(2)}[n]-0.5|\leq...\leq|\hat{\gamma}_{(k)}[n]-0.5|\leq...\leq|\hat{\gamma}_{(K)}[n]-0.5|$. The m-th candidate decision is given by

$$\gamma_{m,k}=\begin{cases}1,\ [\hat{\gamma}_k[n]>\hat{\gamma}_{m-1}[n]],\\ 1,\ \hat{\gamma}_k[n]=\hat{\gamma}_{m-1}[n]\cap\hat{\gamma}_{m-1}[n]\leq 0.5,\\ 0,\ \hat{\gamma}_k[n]=\hat{\gamma}_{m-1}[n]\cap\hat{\gamma}_{m-1}[n]>0.5,\\ 0,\ \hat{\gamma}_k[n]<\hat{\gamma}_{m-1}[n],\end{cases} \tag{34}$$

Next, $\frac{M_n}{2}$ candidate decisions are derived from the original relaxed actions and Gaussian noise $\delta^{'}$. The relaxed action with noise $\breve{\gamma}[n]$ is denoted as $sigmoid\left(\breve{\gamma}[n]+\delta^{'}\right)$. Then, the last $\frac{M_n}{2}$ candidate decisions can be obtained by applying (33) and (34) with replacing $\hat{\gamma}[n]$ as $\breve{\gamma}[n]$.

Critic module: The best action $\gamma^*[n]$ is selected by

$$\gamma^*[n]=\arg\max_{\gamma[n]\in\left\{\{\gamma_m[n]\}_{m=1}^M\right\}}G(\gamma[n],\xi[n]). \tag{35}$$

With given candidate offloading decision, the resource management problem is solved in Section IV.C.

Policy update module: The proposed OELO utilizes $(\xi[n],\gamma^*[n])$ as the labeled samples of to update the DNN's offloading policy. Specifically, an empty replay memory with limited capacity q is maintained. First, the labeled samples are collected when the amount of the labeled samples is less than $\frac{q}{2}$. Then, the DNN is trained when the amount of the labeled samples is more than $\frac{q}{2}$. To avoid over-fitting, the DNN is periodically trained with ω_T frames. In frame n, if $\mathrm{mod}\ (n,\omega_T)=0$, we randomly select a batch of training samples from the replay memory. The DNN's network parameter $\theta[n]$ can be updated through using the Adam optimizer, the loss function is expressed as

$$\begin{aligned}LS\left(\theta[n]\right)=\frac{-1}{|S[n]|}\sum\nolimits_{i\in S[n]}\Big(&\left(\gamma[i]\right)^{\mathrm{T}}\log f_{\theta[i]}\left(\xi[i]\right)\\&+\left(1-\gamma[i]\right)^{\mathrm{T}}\log\left(1-f_{\theta[i]}\left(\xi[i]\right)\right)\Big),\end{aligned} \tag{36}$$

where $|S[n]|$ represents the size of the labeled samples batch.

Queueing module: After solving the resource management problem, the optimizing offloading decision and resource management are obtained. We execute the joint action to obtain the queueing module,

$$\xi[n+1]=\{h[n+1],Q_k[n+1]\}_{k=1}^K. \tag{37}$$

The details of OELO optimization algorithm for solving (P1) are summarized in algorithm 1.

C. Resource Management

In the proposed OELO, the resource management problem is solved with given the offloading decision $\gamma_k[n]$. Problem (P4) is transformed as

$$\begin{aligned}&(\mathrm{P6}):\max_{\{\tau_k[n],l_{loc,k}[n],p_k[n],u_{1,k}[n],u_{2,k}[n]\}}u\\&s.t.(7),(8),(15),(30b)-(30h).\end{aligned} \tag{38}$$

Problem (P6) is non-convex because of the non-convexity of the constraints (30b) and (30d). Specifically, the right side of constraint (30b) is related to $L_k[n]$. $L_k[n]$ is composed of multiple optimization variables, e.g., $\gamma_k[n]$, $\tau_k[n]$ and $p_k[n]$, which makes (30b) non-convex. The right side of the constraint (30d) is concave, which makes it non-convex. We approximately solve it by applying SCA, in which time allocation $\tau_k[n]$, local computation allocation $l_{loc,k}[n]$ and power allocation $p_k[n]$ can be obtained by considering the others as given [30].

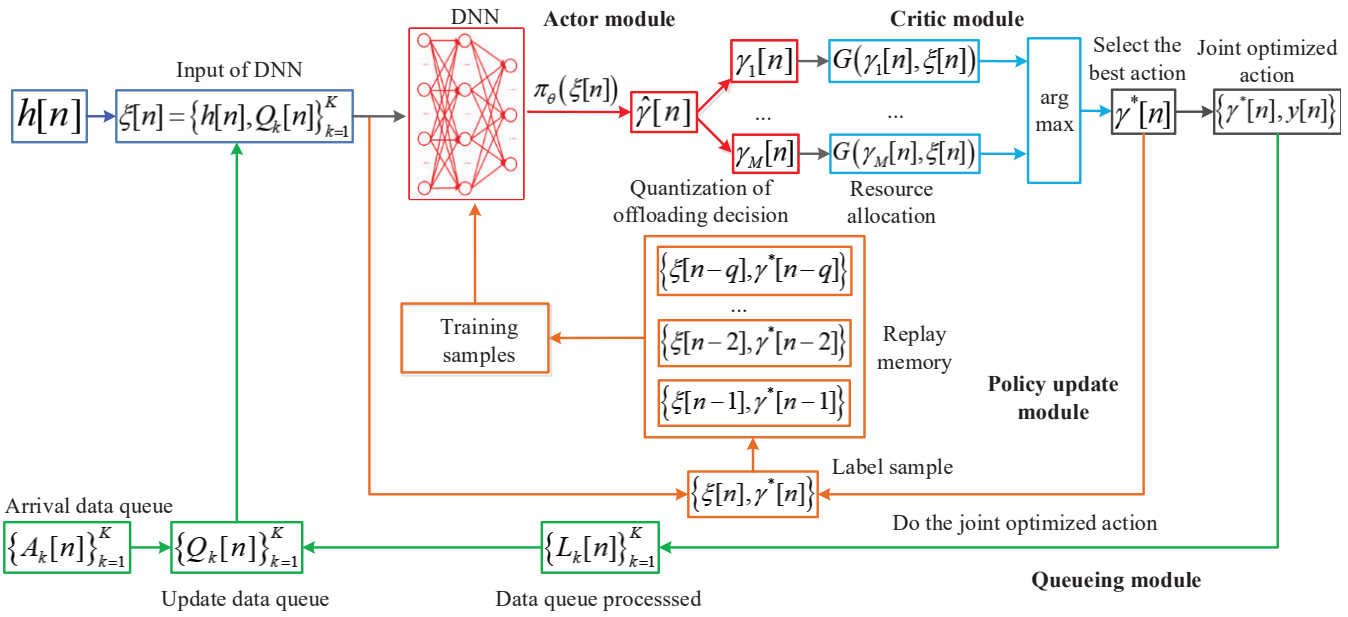

Fig. 2. The proposed OELO algorithm.

Algorithm 1: The OELO optimization algorithm

Input: Feasible points $\left\{\tau_k^r[n], l_{loc,k}^r[n], p_k^r[n]\right\}_{k=1}^K$, total frames N, the channel coefficient and $\{h[n]\}$ the amount of candidate actions M_n, training interval ω_T.

Output: The joint optimizing action $(\gamma[n], y[n])$.

Initialization: The initial task data of TU_k $\{Q_k[0]\}_{k=1}^K$, random parameters θ_1 of the DNN and the empty replay memory;

for $n = 1, 2, ..., N$ **do**

  Observe the input $\xi[n] = \{h[n], Q_k[n]\}_{k=1}^K$;

  Generate the relaxed offloading decision function $\hat{\gamma}[n] = \pi(\xi[n])$ with the DNN;

  Quantize $\hat{\gamma}[n]$ into M_n binary decision by using NOP;

  Optimize resource management to compute $G(\gamma_m[n], \xi[n])$;

  Select the best solution of the resource management $\gamma^*[n] = \arg\max_{\gamma_m[n]} G(\gamma_m[n], \xi[n])$ and execute the joint optimizing action $(\gamma^*[n], y[n])$;

  Update the replay memory through adding the joint optimizing action $(\gamma^*[n], y[n])$;

  if $\mod(n, \omega_T) = 0$ **then**

    Randomly select training samples from the replay memory to train DNN and update θ_n through using the Adam algorithm;

  end

  $n = n + 1$;

  Update $\{Q_k[n]\}_{k=1}^K$ according to the joint action $(\gamma^*[n-1], y[n-1])$ and arrival task data queue $\{A_k[n-1]\}_{k=1}^K$.

end

1) Time allocation: With given the local computing bits and the transmit power of TUs, the time allocation problem is formulated as

$$(\mathrm{P6.1}): \max_{\{\tau_k[n], u_{1,k}[n], u_{2,k}[n]\}} u \quad (39)$$
$$s.t. (8), (30\mathrm{b}) - (30\mathrm{h}).$$

Note that the constraints (8), (30b)-(30h) in the (P6.1) are linear, (P6.1) is convex. We can use standard optimization techniques, e.g., CVX to solve it [31].

2) Local computation allocation: With given the time allocating factor and transmit power of TUs, the local computation problem is formulated as

$$(\mathrm{P6.2}): \max_{\{l_{loc,k}[n], u_{1,k}[n], u_{2,k}[n]\}} u \quad (40)$$
$$s.t. (15), (30\mathrm{b}) - (30\mathrm{h}).$$

(P6.2) is convex due to the constraints (15), (30b)-(30h) in (P6.2) are convex. We can use CVX to solve it.

3) Power allocation: With given the time allocating factor and local computing bits of TUs, the transmit power problem is formulated as

$$(\mathrm{P6.3}): \max_{\{p_k[n], u_{1,k}[n], u_{2,k}[n]\}} u \quad (41)$$
$$s.t. (7), (30\mathrm{b}) - (30\mathrm{h}).$$

Due to the constraints of (30c) and (30d) in (P6.3) are non-convex, (P6.3) is difficult to solve directly. With SCA method, we approximate (P6.3) into a convex form in each iteration. Then, the solution of (P6.3) can be obtained by updating it iteratively.

Denote $\{p_k^r[n]\}$ as TU_k's transmit power after the r-th iteration. Convex function can be a first-order Taylor expansion of the global lower bound. Therefore, (30c) is approximated

into

$$u_{1,k}[n] \le \log_2\left(1+\frac{p_k^r[n]|h_{k,s}[n]|^2}{\sigma_s^2}\right) + \frac{1}{In2}\frac{|h_{k,s}[n]|^2}{\sigma_s^2}\frac{(p_k[n]-p_k^r[n])}{\left(1+\frac{p_k^r[n]|h_{k,s}[n]|^2}{\sigma_s^2}\right)}, \forall k,n \ . \quad (42)$$

Similarly, for (30d), we have

$$u_{2,k}[n] \ge \log_2\left(1+\frac{p_k^r[n]|h_{k,e}[n]|^2}{P_j|h_{j,e}[n]|^2+\sigma_e^2}\right) + \frac{1}{In2}\frac{|h_{k,e}[n]|^2}{P_j|h_{j,e}[n]|^2+\sigma_e^2}\frac{(p_k[n]-p_k^r[n])}{\left(1+\frac{p_k^r[n]|h_{k,e}[n]|^2}{P_j|h_{j,e}[n]|^2+\sigma_e^2}\right)}, \forall k,n \ . \quad (43)$$

Then, problem (P6.3) ia rewritten as

$$(\text{P6.3.1}): \max_{\{p_k[n],u_{1,k}[n],u_{2,k}[n]\}} u \quad (44)$$
$$s.t. (7), (30\text{b}), (30\text{e})-(30\text{h}), (42), (43).$$

Since the constraints (7), (30b), (30e)-(30h), (42) and (43) in (P6.3.1) are convex, (P6.3.1) can be solved by CVX.

In conclusion, we obtain the optimizing solution of the resource management problem (P6) by alternately solving (P6.1), (P6.2), and (P6.3.1). The solution of (P6) is summarized in the algorithm 2.

Algorithm 2: The resource management optimization algorithm

1 **Initialize:** Give feasible points $\gamma_k^r[n], \tau_k^r[n], l_{loc,k}^r[n], p_k^r[n]$, set initial iteration $r=0$ and maximum number of iterations r_{max}, set accuracy $\varepsilon > 0$;
2 **Repeat** Solve problem (P5.1) and obtain the solution $\tau_k[n]$ with given $\left\{\gamma_k^r[n], l_{loc,k}^r[n], p_k^r[n]\right\}$;
3 Solve problem (P5.2) and obtain the solution $l_{loc,k}[n]$ with given $\{\gamma_k^r[n], \tau_k^r[n], p_k^r[n]\}$;
4 Solve problem (P5.3.1) and obtain the solution $p_k[n]$ with given $\left\{\gamma_k^r[n], \tau_k^r[n], l_{loc,k}^r[n]\right\}$;
5 Update $\tau_k^r[n] \leftarrow \tau_k[n], l_{loc,k}^r[n] \leftarrow l_{loc,k}[n], p_k^r[n] \leftarrow p_k[n]$;
6 Update $r \leftarrow r+1$;
7 **Until** The algorithm converges to ε or r is equal to r_{max};
8 **Output:** The optimizing actions $\{\tau_k[n], l_{loc,k}[n], p_k[n], u, u_{1,k}[n], u_{2,k}[n]\}$.

D. Complexity of OELO

We first analyze the complexity of algorithm 1. The OELO optimization algorithm mainly consists of offloading decision generation and offloading decision update. Offloading decision generation is executed in each frame. However, offloading decision update is not always executed and it can be executed in parallel with offloading decision generation. Thus, we mainly analyze the complexity of offloading decision generation in each frame. It is concentrated in the resource management of algorithm 2. Since M_n candidate actions are generated, algorithm 1 needs to perform the resource management of algorithm 2 M_n times per frame.

Then, we analyze the complexity of algorithm 2, which is related to $3K$ optimization variables. Thus, the complexity of the resource management optimization algorithm is approximately $I_1 O\left((3K)^{3.5}\log_2\left(\frac{1}{\varepsilon}\right)\right)$, where I_1 represents the number of the iterations.

In conclusion, the complexity of OELO is approximately $M_n I_1 O\left((3K)^{3.5}\log_2\left(\frac{1}{\varepsilon}\right)\right)$.

V. Simulation Results

In this section, the simulation results are presented to evaluate the performance of OELO. In the OELO scheme, 5 TUs are randomly placed in $400\text{m}\times400\text{m}$ area. USV and UEV fly in a circle centered at $(0,0)$ with the radius of $\rho_s = 50\text{m}$ and $\rho_e = 60\text{m}$, respectively. Other parameters of the UAV-assisted MEC secure communication system are shown in Table II.

TABLE II
The UAV-assisted MEC Secure Communication System Parameters Setting

Parameters	Values
The frame size	$\sigma_t = 0.5\text{s}$
Flying altitudes of USV and UEV	$H_s = H_e = 100\text{m}$
Path loss at the reference distance	$\beta_0 = -60\text{dB}$
Bandwidth of transmit links	$B = 1\text{MHz}$
Noise power received at USV and UEV	$\sigma_s^2 = \sigma_e^2 = -110\text{dBm}$
GJ transmit power	$P_j = 20\text{dBm}$
Peak power of TUs	$P_{\max} = 20\text{dBm}$
TU_k maximum CPU computation frequency	$F_k^{\max} = 1\text{GHz}$
USV maximum CPU computation frequency	$F_s^{\max} = 10\text{GHz}$
Required CPU cycles of TU_k and USV for computing one bit	$c_k = c_s = 1.0\times10^3\text{cycles/bit}$
CPU capacity coefficient of TUs	$\kappa_k = 10^{-27}$
Maximum capacity of TUs' data queue storage	$S_{\max} = 4\text{Gbits}$

The DNN model in OELO is a four-layer fully connected multilayer perceptron, which consists of one input layer, two hidden layers with 256 and 128 neurons respectively, and one output layer. The learning rate is set to 0.01, the training interval ω_T is set to 10 and the memory size is set to 1000.

Fig. 3 shows the secure computation efficiency variation versus time frames with different ρ_s when the initial task data queue of TU_k is 0bits and the arrival task data queue of TU_k $A_k[n]$ is 2Mbits in frame n. We can find out that the secure computation efficiency of the UAV-assisted MEC secure communication system increases and gradually converges with the increase of the time frames. It is because that the DNN training gradually adapts to the system environment. However, it will fluctuate because the channel coefficients between USV and TUs changes in each frame. In addition, the secure computation efficiency will vary with the difference of USV's flight radius ρ_s. Since the distance between USV and TUs are

different, the channel coefficient of the links between USV and TUs are different, which affects the communication quality between USV and TUs.

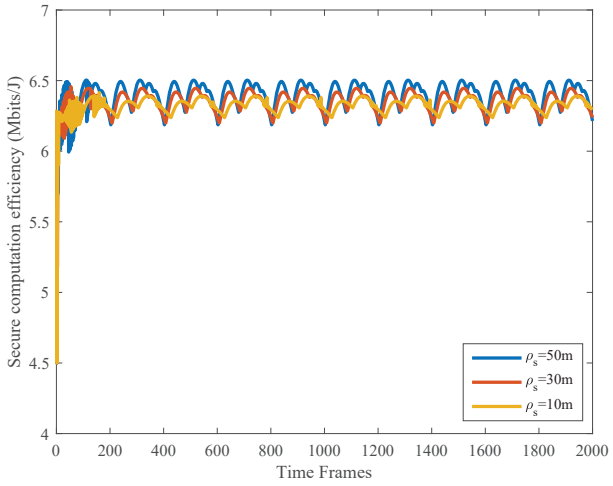

Fig. 3. Secure computation efficiency variation versus time frames with different ρ_s.

To analyze the secure computation efficiency performance of the UAV-assisted MEC secure communication system, we define the average secure computation efficiency of the system as $\frac{1}{N}\sum_{n=1}^{N}\left(\sum_{k=1}^{K}(L_k[n]) - \alpha[n]\sum_{k=1}^{K}(E_k[n])\right)$. Fig. 4 shows the average secure computation efficiency of the UAV-assisted MEC secure communication system versus TU_k's arrival task data queue $A_k[n]$ with different $P_{\max}$ when the initial task data queue of TU_k is set to $10\mathrm{Mbits}$. It is observed from Fig. 4 that the average secure computation efficiency increases firstly with the larger of $A_k[n]$. Note that TUs have more task data that can be calculated with the increase of $A_k[n]$, and the computing resources of both local computation and secure offloading computation can be fully utilized at this time. Then, the average secure computation efficiency tends to be stable when $A_k[n]$ continues to increase. Because the computing power of the UAV-assisted MEC secure communication system is limited. In addition, we can see that the average secure computation efficiency of the UAV-assisted MEC secure communication system increases with the larger of $P_{\max}$. Because more energy can be obtained by TUs to offload the task information to USV for computing.

In order to fully reflect the role of UAV-assisted MEC to help TUs to compute the offloading task data, the data queue of every TU over time frames is shown in Fig. 5. We set the initial task data queue of TUs is $4\mathrm{Gbits}$ and the arrival task data queue is $0.7\mathrm{Mbits}$. It can be observed from Fig. 5 that the data queue of each TU continues to decline even if there are new queues arriving in each time frame, which shows that UAV-assisted MEC plays an important role in assisting computing.

Furthermore, we set the initial task data queue to $10\mathrm{Mbits}$ and arrival task data queue to $1.0\mathrm{Mbits}$ in Fig. 6. From Fig. 6, it is observed that most data queues of TUs in the UAV-assisted

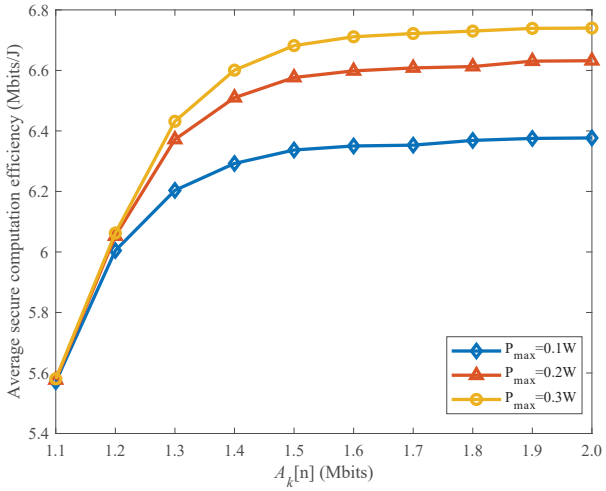

Fig. 4. Average secure computation efficiency variation versus $A_k[n]$ with different P_{max}.

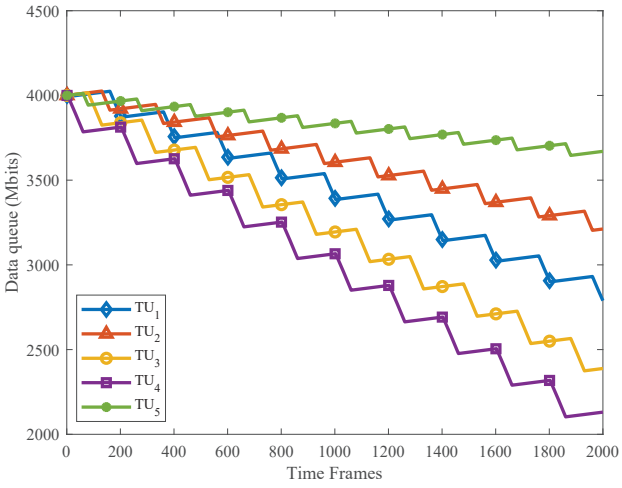

Fig. 5. Data queue variation of TUs versus time frames when the initial task data queue of TUs is $4\mathrm{Gbits}$.

MEC secure communication system are computed over time, which shows that the proposed OELO can quickly help the secure computation UAV-assisted MEC system to compute the task data.

Fig. 7 shows the average secure computation efficiency variation of the UAV-assisted MEC secure communication system versus $P_{\max}$ with different arrival task data queue $A_k[n]$ when the initial task data queue of each TU is $0\mathrm{bits}$. From Fig. 7, it is observed that the average secure computation efficiency increases with the increase of the peak transmit power of TUs $P_{\max}$, because TUs can obtain more energy to transmit task offloading information when $P_{\max}$ is larger.

Fig. 8 shows the average secure computation efficiency versus P_{max} with different CPU cycles required for TU_k to compute one bit task data c_k. We can observe that the average secure computation efficiency of the UAV-assisted MEC secure communication system decreases with larger c_k. This is because when the required CPU cycles of TU_k

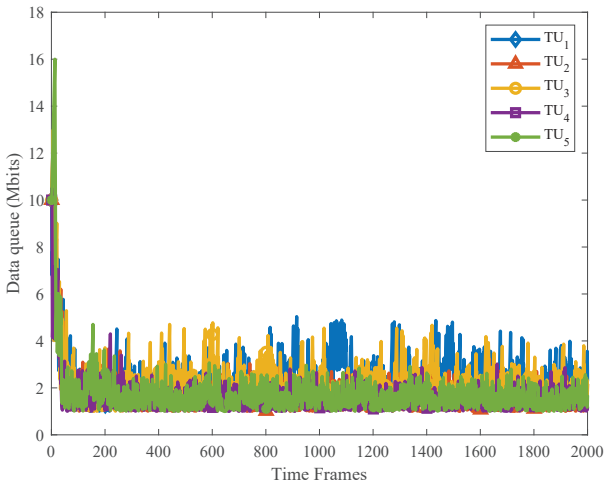

Fig. 6. Data queue variation of TUs versus time frames when the initial task data queue of TUs is 10Mbits.

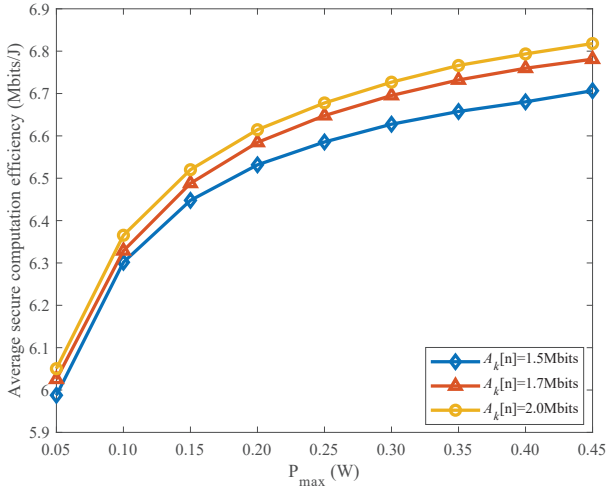

Fig. 7. Average secure computation efficiency versus $P_{\max}$ with different $A_k[n]$.

for computing one bit is larger, its computation speed will decrease. When TU_k selects local computation, the local computing bits will be reduced. Therefore, the UAV-assisted MEC system's average secure computation efficiency decreases.

To demonstrate the effectiveness of the proposed OELO, we compare the average secure computation efficiency performance of the UAV-assisted MEC secure communication system with three benchmarks in Fig. 9.

Scheme 1: The average secure computation efficiency is maximized by optimizing offloading decision, local computation and transmit power based on OELO while the time allocating factor of TUs is equally allocated.

Scheme 2: The average secure computation efficiency is maximized by optimizing offloading decision, time allocating factor and local computation based on OELO while the transmit power is fixed with equal value.

Scheme 3: The offloading decision is optimized in an iterative manner, where the candidate offloading decisions are

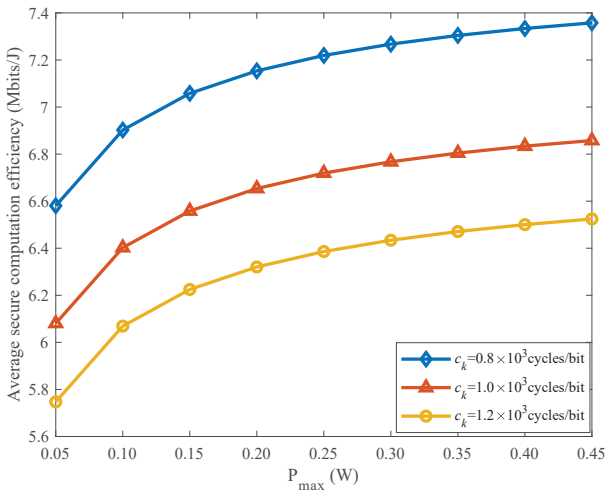

Fig. 8. Average secure computation efficiency versus P_{max} with different c_k.

obtained by performing the modulo-two addition operation on the initial offloading decision of TU_k [6].

We can see from Fig. 9 that the average secure computation efficiency performance of the OELO scheme is much better than that of scheme 1 and scheme 2 versus $A_k[n]$. This is because the resource management of the OELO includes the time allocation, local computation allocation and power allocation. Since scheme 3 traverses many situations in generating candidate offloading decisions, it can bring good performance. However, scheme 3 brings high complexity, which is not suitable for online offloading scenarios. As can be seen from Fig. 9, OELO achieves nearly as good average secure computation efficiency performance as scheme 3 with lower complexity, which is benefit of the high efficiency of edge learning in the OELO.

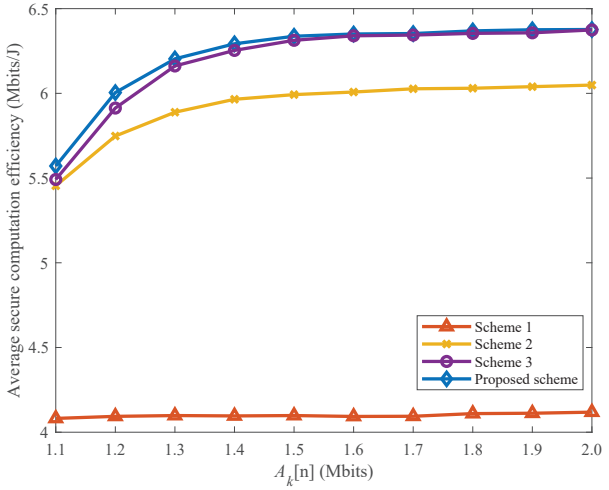

Fig. 9. Average secure computation efficiency comparison with different schemes versus $A_k[n]$.

VI. Conclusion

In this paper, we considered a UAV-assisted MEC secure communication system, where the offloading information of TUs will be eavesdropped by UEV when USV assists in the computation of TUs with dynamic arrival task data queue. To maximize the secure computation efficiency, TUs' binary offloading decision and resource management was jointly optimized while guaranteeing dynamic task data queues stability and minimum secure computing requirement. A novel OELO scheme was proposed to solve the optimization problem. Firstly, the fractionally structured problem is transformed as a tractable form based on Dinkelbach method. Then, OELO generated offloading decision based on DRL and optimized resource management in an iterative manner through SCA. Simulation results showed that the proposed scheme achieves better computing performance and enhances the stability and security of the system compared with benchmarks. In the future work, we will extend to our research into using DRL to generate resource management to reduce the complexity and optimizing the trajectory of USV to improve the secure computation performance.

References

[1] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16-32, Jan. 2020.

[2] Z. Peng, X. Chen, W. Xu, C. Pan, L. -C. Wang and L. Hanzo, "Analysis and optimization of massive access to the IoT relying on multi-pair two-way massive MIMO relay systems," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4585-4598, Jul. 2021.

[3] D. W. K. Ng, M. Breiling, C. Rohde, F. Burkhardt and R. Schober, "Energy-efficient 5G outdoor-to-indoor communication: SUDAS over licensed and unlicensed spectrum," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3170-3186, May 2016.

[4] S. Zhu, W. Xu, L. Fan, K. Wang and G. K. Karagiannidis, "A novel cross entropy approach for offloading learning in mobile edge computing," *IEEE Wireless Commun. Letters*, vol. 9, no. 3, pp. 402-405, Mar. 2020.

[5] Z. Ding, D. W. K. Ng, R. Schober and H. V. Poor, "Delay minimization for NOMA-MEC offloading," *IEEE Signal Process. Letters*, vol. 25, no. 12, pp. 1875-1879, Dec. 2018.

[6] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177-4190, Jun. 2018.

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

[8] Y. Li, X. Wang, X. Gan, H. Jin, L. Fu and X. Wang, "Learning-aided computation offloading for trusted collaborative mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 12, pp. 2833-2849, Dec. 2020.

[9] P. Zhao, H. Tian, K. -C. Chen, S. Fan and G. Nie, "Context-aware TDD configuration and resource allocation for mobile edge computing," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1118-1131, Feb. 2020.

[10] Y. Guo, R. Zhao, S. Lai, L. Fan, X. Lei and G. K. Karagiannidis, "Distributed machine learning for multiuser mobile edge computing systems," *IEEE J. Sel. Topics Signal Process.*, doi: 10.1109/JSTSP.2022.

[11] X. Qi, M. Yuan, Q. Zhang and Z. Yang, "Joint power-trajectory-scheduling optimization in a mobile UAV-enabled network via alternating iteration," *China Commun.*, vol. 19, no. 1, pp. 136-152, Jan. 2022.

[12] R. Han, Y. Wen, L. Bai, J. Liu and J. Choi, "Rate splitting on mobile edge computing for UAV-aided IoT systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1193-1203, Dec. 2020.

[13] Y. Du, K. Wang, K. Yang and G. Zhang, "Energy-efficient resource allocation in UAV based MEC system for IoT devices," *2018 IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1-6.

[14] W. Lu, Y. Ding, Y. Gao, S. Hu, Y. Wu, N. Zhao and Y. Gong, "Resource and trajectory optimization for secure communications in dual-UAV-MEC systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2704-2713, Apr. 2022.

[15] R. Zhao, J. Xia, Z. Zhao, S. Lai, L. Fan and D. Li, "Green MEC Networks Design Under UAV Attack: A Deep Reinforcement Learning Approach," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1248-1258, Sept. 2021.

[16] T. Bai, J. Wang, Y. Ren and L. Hanzo, "Energy-efficient computation offloading for secure UAV-edge-computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6074-6087, Jun. 2019.

[17] J. Xu, W. Xu, D. W. K. Ng and A. L. Swindlehurst, "Secure communication for spatially sparse millimeter-wave massive MIMO channels via hybrid precoding," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 887-901, Feb. 2020.

[18] W. Lu, Y. Ding, Y Gao, Y. Chen, N. Zhao, Z. Ding and A. Nallanathan, "Secure NOMA-based UAV-MEC network towards a flying eavesdropper," *IEEE Trans. Commun.*, doi: 10.1109/TCOMM.2022.

[19] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, M. Debbah, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," arXiv preprint arXiv:2206.00422, 2022.

[20] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536-550, 1 Mar. 2022.

[21] H. Yang, J. Zhao, Z. Xiong, K. -Y. Lam, S. Sun and L. Xiao, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. Sel. Areas. Commun.*, vol. 39, no. 10, pp. 3144-3159, Oct. 2021.

[22] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu and K. Yang, "Deep-learning-based joint resource scheduling algorithms for hybrid MEC networks," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6252-6265, Jul. 2020.

[23] Y. Liu, S. Xie and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12229-12239, Oct. 2020.

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

[25] S. Bi, L. Huang, H. Wang and Y. -J. A. Zhang, "Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7519-7537, Nov. 2021.

[26] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930-1941, Feb. 2019.

[27] K. K. Nguyen, S. Khosravirad, D. B. Da Costa, L. D. Nguyen and T. Q. Duong, "Reconfigurable intelligent surface-assisted multi-UAV networks: Efficient resource allocation with deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, doi: 10.1109/JSTSP.2021.

[28] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing," *IEEE Trans. Mobile Comput.*, doi: 10.1109/TMC.2021.

[29] W. Dinkelbach, "On nonlinear fractional programming," *Management Science*, vol. 13, pp. 492-498, Mar. 1967. Available: http://www.jstor.org/stable/2627691.

[30] B. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, 1978.

[31] S. Boyd and L. V andenberghe, Convex optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004.