UAV-Enabled Federated Learning in Dynamic Environments: Efficiency and Security Trade-off

Xiaokun Fan, Yali Chen, Min Liu, *Senior Member, IEEE*, Sheng Sun, Zhuotao Liu, Ke Xu, *Fellow, IEEE*, and Zhongcheng Li

***Abstract*—Unmanned aerial vehicles (UAVs) can be deployed as flying base stations to provide wireless communication and machine learning (ML) training services for ground user equipments (UEs). Due to privacy concerns, many UEs are not willing to send their raw data to the UAV for model training. Fortunately, federated learning (FL) has emerged as an effective solution to privacy-preserving ML. To balance efficiency and wireless security, this paper proposes a novel secure and efficient FL framework in UAV-enabled networks. Specifically, we design a secure UE selection scheme based on the secrecy outage probability to prevent uploaded model parameters from being wiretapped by a malicious eavesdropper. Then, we formulate a joint UAV placement and resource allocation problem for minimizing training time and UE energy consumption while maximizing the number of secure UEs under the UAV's energy constraint. Considering the random movement of the eavesdropper and UEs as well as online task generation on UEs in practical application scenarios, we present the long short-term memory (LSTM)-based deep deterministic policy gradient (DDPG) algorithm (LSTM-DDPG) to facilitate real-time decision making for the formulated problem. Finally, simulation results show that the proposed LSTM-DDPG algorithm outperforms the state-of-arts in terms of efficiency and security of FL.**

***Index Terms*—Deep reinforcement learning, federated learning, physical layer security, resource allocation, unmanned aerial vehicle.**

I. Introduction

IT is expected that sixth-generation (6G) networks will support ubiquitous network connectivity and artificial intelligence (AI) services [1]. Due to the unique advantages of flexible deployment and rapid mobility, unmanned aerial vehicle (UAV) has been envisioned as an indispensable technology to provide ubiquitous communications for user equipments (UEs), in the absence of ground infrastructures or limited infrastructure coverage such as disasters, remote areas and hot spots [2]. Besides, along with the dramatic developments of AI applications such as augmented reality and face recognition, massive data generated by UEs can be used to train various machine learning (ML) models with the assistance of UAVs, where UAVs are deployed as flight BSs to flexibly collect data and perform ML model training [3]. However, this centralized training manner inevitably causes privacy disclosure, so it is unrealistic for UEs to transmit their raw data to UAVs [4].

Fortunately, federated learning (FL) has emerged as a promising distributed ML paradigm with privacy preservation [5], which enables UEs to collaboratively train ML models and only upload model parameters rather than raw data to the server. By adopting FL, UAVs can assist ground UEs in performing distributed model training without prejudice to data privacy. To be specific, UAVs work as FL servers to aggregate local models uploaded from UEs, and then broadcast the aggregated global model to all UEs for the next learning round. This process is repeated until the global model converges.

In UAV-enabled FL networks, communication capacity can be enhanced and UEs' dropout rate can be reduced during FL training, by exploiting the beneficial line-of-sight (LoS) propagation of the UAV [6]. Despite these promising benefits, there still exist some critical issues to be solved.

1) Security: Due to the broadcast nature of the UAV-UE channel, malicious eavesdroppers can easily intercept the uploaded model parameters and infer sensitive information of users (e.g., age, gender, occupation and location). The common secure schemes for achieving confidential communications in FL are the upper-layer encryption techniques such as secure multi-party computation and homomorphic encryption [10], which often inevitably add extra computational or communication overheads to the resource-limited UAV and UEs. In contrast, physical layer security (PLS) exploits wireless channel characteristics to protect UAV communications from malicious eavesdroppers, and it does not need complex encryption and decryption operations. Thus, leveraging PLS to guarantee the security of FL in UAV-enabled networks is worth studying.

2) Efficiency: The efficiency of FL is mainly reflected in both training time and energy consumption. On the one hand,

This work was supported by the National Key Research and Development Program of China under Grant 2021YFB2900102, in part by the National Natural Science Foundation of China under Grants 62202449 and 62072436, and in part by the Innovation Funding of ICT, CAS under Grant E261080. (*Corresponding author: Min Liu.*)
Xiaokun Fan and Zhongcheng Li are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: fanxiaokun@ict.ac.cn; zcli@ict.ac.cn).

Min Liu is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, also with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with Zhongguancun Laboratory, Beijing 100194, China (e-mail: liumin@ict.ac.cn).

Yali Chen and Sheng Sun are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: chenyali@ict.ac.cn; sunsheng@ict.ac.cn).

Zhuotao Liu is with the Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China (e-mail: zhuotaoliu@tsinghua.edu.cn).

Ke Xu is with the Beijing National Research Center for Information Science and Technology (BNRist) and the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and also with Zhongguancun Laboratory, Beijing 100194, China (e-mail: xuke@tsinghua.edu.cn).

the training time required to converge to a target accuracy level is one of the most important performance metrics of FL, which depends on the computation latency for model training on UEs and communication latency for transmitting model parameters between the UAV and UEs. On the other hand, the iterative training process of FL leads to considerable energy consumption of UEs (communication and computation) and UAV (propulsion), which raises great challenges to the battery constrained UAV and UEs. Thus, it is essential to reduce training time and energy consumption of FL.

To sum up, it is crucial to simultaneously optimize training time, energy consumption and security of FL in UAV-enabled networks. However, this research is very challenging and has not yet been carried out. First, it is worth noting that leveraging PLS to achieve secure FL in UAV-enabled networks is still in its infancy. In [15], secrecy rate is used to measure the security of one UE for model transmission, which is a metric in PLS, defined as the difference between the transmission rates of legitimate channels and wiretap channels. The greater the secrecy rate of one UE, the more secure it is. However, the secrecy rate cannot intuitively reflect the possibility of UEs being eavesdropped. Thus, how to accurately evaluate the security of UEs is a key issue. Second, to improve the security of FL, the UAV location can be adjusted to enhance the capacity of legitimate channels, and the UEs can also reduce their transmit power to weaken wiretap channels, but at the cost of higher communication delay. To improve the efficiency of FL, system resources (e.g., bandwidth, CPU frequency, transmit power) can be optimized to minimize training time [17], [18], energy consumption [20], [21] and trade-off between them [24], while secrecy performance may be reduced. Thus, training time, energy consumption and security of FL conflict with each other, and how to optimize UAV placement and resource allocation to strike a trade-off among these three conflicting objectives is quite difficult. Third, in practical scenarios, the random movement of eavesdroppers and UEs and online task generation on UEs pose great challenges to the optimization of UAV-enabled FL.

Driven by the above challenges, this paper proposes a secure and efficient FL framework in UAV-enabled networks to simultaneously optimize training time, energy consumption and security of FL systems, under the practical dynamic environments. The main contributions of this paper can be summarized as follows.

- We design a secure UE selection scheme based on the secrecy outage probability (SOP), where SOP is derived to obtain the probability of UEs being eavesdropped. Then, only the secure UEs whose SOP meets a target security requirement can participate in FL training because their model parameters can be securely transmitted to the UAV.
- We define a trade-off objective function called Security-Efficiency Cost (SEC) to simultaneously reduce training time and UE energy consumption, as well as increase the number of secure UEs. Then, we formulate a long-term SEC minimization problem by jointly optimizing the transmit power and CPU frequency of UEs, uplink bandwidth and UAV placement under the UAV's energy budget constraint. The problem is difficult to address due to the three conflicting objectives and environment dynamics. Thus, we reformulate the problem as a markov decision process (MDP) and exploit deep reinforcement learning (DRL) technique to find an optimal solution.
- We present the long short-term memory (LSTM)-based deep deterministic policy gradient (DDPG) algorithm (LSTM-DDPG) to make real-time decisions, where the LSTM-based actor network and critic network are designed to capture the temporal correlation of state features for improving the state representation ability.
- Simulation results demonstrate that the proposed LSTM-DDPG algorithm has good convergence performance, and outperforms the state-of-arts in training time, energy consumption and security of FL.

The remainder of this paper is organized as follows. Section II reviews some related works. Section III presents the system model and problem formulation. Section IV presents the proposed algorithm. Section V describes the simulation results. Finally, Section VI concludes the paper.

II. Related Work

Recently, many research efforts have been conducted to improve efficiency and security performance of FL. These studies can be divided into two main research directions. One mainstream direction is to design learning algorithms for FL performance boost. For example, Luo *et al.* [7] analyzed how to optimally choose the essential control variables in FL to minimize the total efficiency cost. Shen *et al.* [8], [9] proposed the novel split federated learning (SFL) scheme that integrated FL with a model split mechanism to enhance training efficiency while maintaining data privacy. Another mainstream direction is to optimize resource allocation of FL systems, which studies how the computation and communication resources can affect the training time, energy consumption and security of FL. In this paper, we focus on achieving secure and efficient FL in UAV-enabled networks from the perspective of resource allocation. To avoid being out of the scope of this work, we will present related works on resource allocation in terrestrial/aerial FL systems, and divide the related works into: (i) security performance optimization, (ii) training time minimization, (iii) energy consumption minimization, and (iv) weighted sum minimization of training time and energy consumption.

Security performance optimization. The security issue of FL has aroused extensive concern from both academia and industry. The common privacy-preserving schemes such as secure multiparty computation, homomorphic encryption and differential privacy have been widely applied on top of FL to prevent potential security and privacy threats [10]–[12]. Recently, the works in [13], [14] considered the security of FL in UAV networks and introduced the blockchain technology to ensure secure model transmissions. However, the above approaches are computationally expensive (i.e., homomorphic encryption and blockchain) or add up communication burdens (i.e., secure multiparty computation and differential privacy), which may pose challenges for the UAV and ground UEs with limited hardware resources. In this case, Yao *et al.*

[15] adopted the physical layer security (PLS) technology to achieve secure FL in fog-aided internet of drones and investigated the maximization problem of system secrecy rate by optimizing the transmit power of all drones.

Minimizing training time. Considering FL over terrestrial wireless networks, Chen *et al.* [16] optimized UE selection and wireless resource allocation to reduce the total communication delay for training. In [17], Do *et al.* studied UAV-assisted wireless powered FL networks, and applied the DDPG algorithm to optimize UAV placement and resource allocation with the goal of minimizing training time. Different from above works that considered UAVs as FL clients, Yang *et al.* [18] adopted UAVs as FL servers and proposed an asynchronous advantage actor-critic (A3C)-based algorithm to minimize the weighted sum of training time and learning accuracy by jointly optimizing device selection, UAVs placement and resource management.

Minimizing energy consumption. In [19], Yang *et al.* studied the energy-efficient FL over terrestrial wireless networks, where computation and communication resources were optimized to minimize the sum of all UEs' energy consumption under a latency constraint. Considering deploying UAV as a FL server, Jing *et al.* [20] used the successive convex approximation (SCA) approach to minimize UE energy consumption via jointly optimizing UAV location and resource allocation under the constraints of learning accuracy and training latency. In UAV-aided wireless powered networks, Pham *et al.* [21] designed an energy-efficient FL framework to minimize the total energy consumption of the UAV server and UEs, and proposed a joint resource allocation scheme based on an iterative algorithm.

Minimizing weighted sum of training time and energy consumption. Tran *et al.* [22] proposed first-of-its-kind "FL over wireless networks" problem design to study the trade-off between training time and UE energy consumption by using the Pareto efficiency model. In [23], Zhou *et al.* optimized bandwidth, transmission power and CPU frequency of UEs to minimize the weighted sum of training time and UE energy consumption. Different from the above works that focused on FL efficiency over terrestrial wireless networks, Tang *et al.* [24] considered the implementation of FL in UAV networks, and adopted the DDPG algorithm to optimize the bandwidth and CPU frequency of UAV clients aiming at minimizing training time and UAV energy consumption.

In summary, existing works only cover one or two of the three significant factors regarding security, training time and energy consumption, which hinders the deployment and application of FL. In addition, few studies have considered the impact of environment dynamics on the FL performance. In real scenarios, the locations of the eavesdropper and UEs as well as the training tasks on UEs are time-varying. Thus, it necessitates the design of online algorithms for making real-time decisions. The DRL has recently achieved remarkable successes in making continuous online decisions. Liu *et al.* [25] adopted the double deep Q-network (DDQN) algorithm to optimize UAV trajectory according to the locations of mobile UEs, aiming at maximizing the long-term system reward. However, DDQN can only deal with discrete action cases. Samir *et al.* [26] formulated an online optimization problem with hybrid discrete-continuous action space, and leveraged the proximal policy optimization (PPO) algorithm to find the best policy. To achieve online decision-making for a multi-objective joint optimization problem, Yu *et al.* [27] proposed an extended DDPG algorithm with multi-dimensional reward. Nevertheless, DDPG with fully-connected deep neural networks lacks the representative ability for accurate state inference. Thus, we are motivated to study secure, low-latency and energy-efficient FL in UAV-enabled networks under dynamic environments. Then, we propose the LSTM-DDPG algorithm to obtain the optimal UAV placement and resource allocation strategy in real-time, aiming at reducing training time, energy consumption while enhancing FL security.

TABLE I
SUMMARY OF KEY NOTATIONS

Notions	Description
$\mathcal{K}$, K, k	Set, number and index of UEs
$\mathcal{T}$, T, t	Set, number and index of rounds/time slots
$\mathcal{S}_t$	Set of secure UEs at time slot t
$K_{s,t}$	Number of secure UEs at time slot t
N_{local}	Number of local iterations in FL
$\tau_{k,t}^{cmp}$	Local computing time of UE k
$\tau_{k,t}^{up}$	Transmission time of local model at UE k
$\boldsymbol{q}_{u,t}$, $\boldsymbol{q}_{k,t}$, $\boldsymbol{q}_{e,t}$	Locations of UAV, UE, eavesdropper
$b_{k,t}$	Bandwidth allocated to UE k
$p_{k,t}$	Transmit power of UE k
$R_t^{k,sec}$	secrecy rate of UE k at time slot t
$P_{k,t}^{out}$	Secrecy outage probability of UE k
$D_{k,t}$	Data size of UE k at time slot t
$C_{k,t}$	Number of CPU cycles required by UE k
$f_{k,t}$	Computing capability of UE k
M	Data size of local model parameters

III. SYSTEM MODEL AND PROBLEM FORMULATION

As shown in Fig. 1, we consider a FL network with a rotary-wing UAV acting as a parameter server, K ground UEs indexed by the set $\mathcal{K} \triangleq \{k = 1, 2, \ldots, K\}$ and an eavesdropper. In particular, each UE trains a local model using its own dataset $\mathcal{D}_{k\in\mathcal{K}} = \{x_{k,n}, y_{k,n}\}_{n=1}^{D_k}$, and then sends its local model parameters to the UAV for global model aggregation. During the uplink transmission, the eavesdropper attempts to wiretap the uploaded model parameters from UEs to the UAV. The main notations in this paper are summarized in Table I.

A. FL Framework Design in UAV-Enabled Networks

In this subsection, we first introduce the basics of FL, and then present the FL framework in UAV-enabled networks.

1) FL Basics: In FL, the ML model trained on each UE is called local model, while the model generated at the UAV server by aggregating all received local models is called global model. The goal of the FL training process is to derive the global model $\boldsymbol{w}_g$ to minimize the global loss function:

$$\min_{\boldsymbol{w}_g} L(\boldsymbol{w}_g) \triangleq \sum_{k=1}^{K} \frac{D_k}{D} L_k(\boldsymbol{w}_g), \tag{1}$$

where $D_k = |\mathcal{D}_k|$ represents the number of data samples for UE k. $D = \sum_{k=1}^{K} D_k$ represents the total number of data

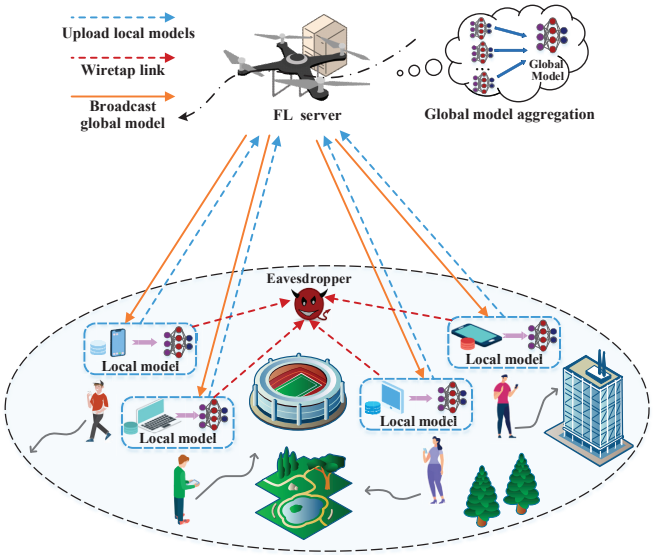


Fig. 1. Federated learning in UAV-enabled wireless networks.

samples from all UEs, and $L_k(\boldsymbol{w}_g)$ is the local loss function of UE k on its own dataset $\mathcal{D}_k$, i.e.,

$$L_k(\boldsymbol{w}_g) = \frac{1}{D_k}\sum_{n=1}^{D_k} l(\boldsymbol{w}_g, x_{k,n}, y_{k,n}), \tag{2}$$

where $l(\boldsymbol{w}_g, x_{k,n}, y_{k,n})$ is the loss function of UE k with one data sample.

2) FL Framework: The entire training process of FL is periodic with T global rounds, while each round $t \in \mathcal{T} \triangleq \{t = 1, 2, \ldots, T\}$ consists of the following three steps:

Step 1. Secure UE Selection: The UAV selects secure UEs to participate in FL training, which will be elaborated in Section III.C. The selected UEs can securely send their model parameters without being wiretapped by the eavesdropper.

Step 2. Local Model Training and Upload: Each participating UE trains a local model on its own dataset by solving the following local optimization problem:

$$\min_{\boldsymbol{h}_k^{(t)}} G_k\left(\boldsymbol{w}_g^{(t)}, \boldsymbol{h}_k^{(t)}\right) \triangleq L_k\left(\boldsymbol{w}_g^{(t)} + \boldsymbol{h}_k^{(t)}\right) - \left(\nabla L_k\left(\boldsymbol{w}_g^{(t)}\right) - \delta \nabla L\left(\boldsymbol{w}_g^{(t)}\right)\right)^T \boldsymbol{h}_k^{(t)}, \tag{3}$$

where δ is the step size, $\boldsymbol{w}_g^{(t)}$ is the global model at round t, and $\boldsymbol{h}_k^{(t)}$ is the difference between the global and local model for UE k at round t. Then, UE k uploads its updated local model (i.e., $\boldsymbol{w}_g^{(t)} + \boldsymbol{h}_k^{(t)}$) to the UAV for aggregation.

Step 3. Global Model Aggregation and Download: After collecting local models from all participating UEs, the UAV server aggregates them to produce a new version of the global model based on a certain aggregation principle. Then, the UAV will broadcast the new global model $\boldsymbol{w}_g^{(t+1)}$ to all UEs for optimizing the local models in the $(t+1)$-th round.

The above three steps are iterated until a desired accuracy is achieved. To achieve a global accuracy ϵ_0 for the global model, the solution $\boldsymbol{w}_g^{(t)}$ of problem (1) means that

$$L(\boldsymbol{w}_g^{(t)}) - L(\boldsymbol{w}_g^*) \leq \epsilon_0 \left(L(\boldsymbol{w}_g^{(0)}) - L(\boldsymbol{w}_g^*)\right), \tag{4}$$

where $\boldsymbol{w}_g^*$ is the optimal solution of problem (1). Similarly, the solution $\boldsymbol{h}_k^{(t)}$ of problem (3) with a local accuracy η means that

$$G_k\left(\boldsymbol{w}_g^{(t)}, \boldsymbol{h}_k^{(t)}\right) - G_k\left(\boldsymbol{w}_g^{(t)}, \boldsymbol{h}_k^{(t)*}\right) \leq \eta\left(G_k\left(\boldsymbol{w}_g^{(t)}, \boldsymbol{0}\right) - G_k\left(\boldsymbol{w}_g^{(t)}, \boldsymbol{h}_k^{(t)*}\right)\right), \tag{5}$$

where $\boldsymbol{h}_k^{(t)*}$ is the optimal solution of problem (3).

According to [19], assuming that $L_k(\cdot)$ is Lipschitz continuous with parameter L and strongly convex with parameter μ, the minimum number of local iterations N_{local} for each global round to reach the local target accuracy η and the minimum global learning rounds T to reach the global target accuracy ϵ_0 can be derived as follows:

$$N_{local} = \frac{2}{(2 - L\delta)\delta\mu} \log \frac{1}{\eta}, \tag{6}$$

$$T = \frac{2L^2}{\mu^2 \xi(1-\eta)} \log \frac{1}{\epsilon_0}, \tag{7}$$

where the hyper-learning parameters $\delta < \frac{2}{L}$ and $\xi \leq \frac{\mu}{L}$.

B. Mobility Model of UEs, Eavesdropper and UAV

Without loss of generality, we consider a three-dimensional Cartesian coordinate system. The UAV flies at a fixed altitude H and its horizonal location is denoted by $\boldsymbol{q}_{u,t} = (x_{u,t}, y_{u,t})$. Moreover, for practical reasons, the locations of the eavesdropper and UEs are considered to be time-varying. The locations of the eavesdropper and UEs are assumed to be known by the UAV.[1] The locations of the k-th UE and the eavesdropper at time slot t are denoted by $\boldsymbol{q}_{k,t} = (x_{k,t}, y_{k,t})$ and $\boldsymbol{q}_{e,t} = (x_{e,t}, y_{e,t})$, respectively.

1) Mobility Model of UEs and Eavesdropper: We assume that all UEs and the eavesdropper, denoted by the set $\mathcal{I} \triangleq \{i = 1, \ldots, K, e\}$, follow the Gauss-Markov mobility model [34]. Specifically, the velocity $v_{i,t}$ and the direction $\theta_{i,t}$ of UE/eavesdropper i in the t-th time slot are derived as

$$v_{i,t} = \eta_{1,i} v_{i,t-1} + (1 - \eta_{1,i})\bar{v}_i + \sqrt{1 - \eta_{1,i}^2}\Omega_{1,i}, \tag{8}$$

$$\theta_{i,t} = \eta_{2,i} \theta_{i,t-1} + (1 - \eta_{2,i})\bar{\theta}_i + \sqrt{1 - \eta_{2,i}^2}\Omega_{2,i}, \tag{9}$$

where $0 \leq \eta_{1,i}, \eta_{2,i} \leq 1$ represent the memory level. $\bar{v}_i$ and $\bar{\theta}_i$ represent the average velocity and direction, respectively. $\Omega_{1,i}$ and $\Omega_{2,i}$ are random variables following two independent Gaussian distributions, which reflect the randomness of the eavesdropper and UE movement.

[1]We assume that the eavesdropper acts as a passive receiver who always stays silent. The passive eavesdropper can be detected due to its inevitable power leakage of the local oscillator [29]. Moreover, it has been demonstrated that the UAV can use on-board optical cameras or synthetic aperture radars to obtain the location information of the eavesdropper and UEs [30]–[33].

2) UAV Mobility Model: The flight control of the UAV is described by flight speed $v_t \in [0, V_{max}]$ and direction $\theta_t \in [0, 2\pi]$. Moreover, the UAV can only move within the served rectangle-shaped area, whose side lengths are denoted as X_{max} and Y_{max}. While flying, the propulsion power consumption can be calculated as follows [35]:

$$P_t^{fly}(v_t) = P_0\left(1+\frac{3v_t^2}{U_{tip}^2}\right) + P_1\left(\sqrt{1+\frac{v_t^4}{4v_0^4}}-\frac{v_t^2}{2v_0^2}\right)^{\frac{1}{2}} + \\ + \frac{1}{2}d_0\rho s_0 A v_t^3, \tag{10}$$

where P_0 and P_1 represent the blade profile power and derived power of the UAV in the hovering state, respectively. U_{tip} is the tip speed of rotor blade, and v_0 is the mean rotor induced velocity under the hover condition. d_0, ρ, s_0 and A represent the fuselage drag ratio, air density, rotor solidity and rotor disc area, respectively. Accordingly, the hovering power consumption is $P^{hover} = P_0 + P_1$ by setting $v_t = 0$.

C. Secure UE Selection

In this subsection, we first present the channel models of UAV-UE and UE-eavesdropper links, and then elaborate the SOP-based secure UE selection scheme.

1) Channel Model: For the legitimate channel from the k-th UE to the UAV, we adopt the practical probabilistic LoS channel model [36]. The LoS and NLoS path loss between the UAV and UE k at time slot t can be given by

$$L_{k,t} = \begin{cases} \alpha_0 (d_t^{k,u})^{-\beta_0}, & \text{if LoS link,} \\ \mu^{\mathrm{NLoS}} \alpha_0 (d_t^{k,u})^{-\beta_0}, & \text{if NLoS link,} \end{cases} \tag{11}$$

where α_0 represents channel power gain at the reference distance of 1 m, and β_0 represents the path loss exponent. μ^{NLoS} is the additional attenuation coefficient of NLoS links. In the communication model, the LoS probability between the k-th UE and the UAV at time slot t can be expressed as

$$P_{k,t}^{\mathrm{LoS}} = \frac{1}{1 + a\exp\big(-b(\theta_{k,t}-a)\big)}, \tag{12}$$

where a and b are constant values that depend on the communication environment. $\theta_{k,t} = \frac{180}{\pi}\sin^{-1}\left(\frac{H}{d_t^{k,u}}\right)$ is the elevation angle from the UAV to UE k in degree, and $d_t^{k,u} = \sqrt{(x_{u,t}-x_{k,t})^2 + (y_{u,t}-y_{k,t})^2 + H^2}$ is the distance between the UAV and UE k. Accordingly, the probability of non-LoS (NLoS) can be given by $P_{k,t}^{\mathrm{NLoS}} = 1 - P_{k,t}^{\mathrm{LoS}}$. Thus, the channel power gain between the UAV and UE k at time slot t can be given by

$$h_t^{k,u} = \left(P_{k,t}^{\mathrm{LoS}} + \mu^{\mathrm{NLoS}} P_{k,t}^{\mathrm{NLoS}}\right)\alpha_0 (d_t^{k,u})^{-\beta_0}. \tag{13}$$

We assume the frequency domain multiple access protocol is applied for uplink channels. The transmission rate of the k-th UE at time slot t can be expressed as

$$R_t^{k,u} = b_{k,t}\log_2\left(1+\frac{p_{k,t}h_t^{k,u}}{\sigma_u^2}\right), \tag{14}$$

where $b_{k,t}$ is the bandwidth allocated to UE k at time slot t, and $\sum_{k=1}^{K} b_{k,t} \le B$. B is the system bandwidth. $p_{k,t}$ is the transmit power of UE k at time slot t, and $\sigma_u^2 = b_{k,t}N_u$, where N_u is the noise power spectrum density at the UAV.

For the ground wiretap channel from the k-th UE to the eavesdropper, both large-scale path loss and small-scale fading are considered. Then, the channel gain between the k-th UE and the eavesdropper at time slot t is given by

$$h_t^{k,e} = \alpha_1 (d_t^{k,e})^{-\beta_1}\tilde{h}_t^{k,e}, \tag{15}$$

where α_1 is the channel power gain at the reference distance of 1 meter, and $d_t^{k,e} = \|\boldsymbol{q}_{k,t} - \boldsymbol{q}_{e,t}\|$ is the distance between the eavesdropper and UE k at time slot t. β_1 is the path loss exponent of the terrestrial link. $\tilde{h}_t^{k,e}$ is the Rayleigh fading following exponential distribution with unit mean.

Accordingly, the data leaking rate from the k-th UE to the eavesdropper at time slot t is given by

$$R_t^{k,e} = b_{k,t}\log_2\left(1+\frac{p_{k,t}h_t^{k,e}}{\sigma_e^2}\right), \tag{16}$$

where $\sigma_e^2 = b_{k,t}N_e$ is the noise power at the eavesdropper, and N_e is the noise power spectrum density at the eavesdropper.

2) Secure UE Selection Based on SOP Metric: In PLS, secrecy rate is used to measure the security of wireless communication. The secrecy rate of UE k can be calculated as the non-negative difference between the transmission rates of the legitimate channel and wiretap channel [37], i.e.,

$$R_t^{k,sec} = \left[R_t^{k,u} - R_t^{k,e}\right]^+, \tag{17}$$

where $[x]^+ = \max(x, 0)$. The secrecy outage probability (SOP) is defined as the probability that the instantaneous secrecy rate $R_t^{k,sec}$ falls below a secrecy rate threshold R_{th} [38]. Accordingly, the SOP can be expressed as

$$\mathbb{P}_{k,t}^{out} = \Pr\left(b_{k,t}\log_2\left(\frac{1+p_{k,t}h_t^{k,u}/\sigma_u^2}{1+p_{k,t}h_t^{k,e}/\sigma_e^2}\right) < R_{th}\right). \tag{18}$$

Then, we define the received SNRs for the legitimate and wiretap channels as $\gamma_t^{k,u} = p_{k,t}h_t^{k,u}/\sigma_u^2$ and $\gamma_t^{k,e} = p_{k,t}h_t^{k,e}/\sigma_e^2$, respectively.[2] Therein, $\gamma_t^{k,e}$ follows the exponential distribution with mean $\lambda_t^{k,e} = p_{k,t}\alpha_1 (d_t^{k,e})^{-\beta_1}/\sigma_e^2$.

[2]For the legitimate UAV-UE channel, it is assumed that the perfect channel state information (CSI) can be obtained by using channel estimation techniques [39]. For the wiretap channel, we assume the availability of statistical CSI, due to the inadvertent signal leakage from the eavesdropper [40]. Moreover, SOP only requires to know the statistical information about the wiretap channel (i.e., the statistical mean and variance), which relaxes the assumption of knowing the eavesdropper's perfect CSI [41].

Therefore, the SOP is given as follows:

$$\begin{aligned}\mathbb{P}_{k,t}^{out} &= \Pr\left(b_{k,t}\log_2\left(\frac{1+\gamma_t^{k,u}}{1+\gamma_t^{k,e}}\right) < R_{th}\right)\\ &= \Pr\left(\frac{1+\gamma_t^{k,u}}{1+\gamma_t^{k,e}} < 2^{R_{th}/b_{k,t}}\right)\\ &= \Pr\left(\gamma_t^{k,e} > \frac{1+\gamma_t^{k,u}}{2^{R_{th}/b_{k,t}}} - 1\right)\\ &= \int_{\gamma_t^{k,e}=\beta}^{\infty} f(\gamma_t^{k,e})\mathrm{d}\gamma_t^{k,e}\\ &= \int_{\gamma_t^{k,e}=\beta}^{\infty} \frac{1}{\lambda_t^{k,e}}\exp\left(\frac{-\gamma_t^{k,e}}{\lambda_t^{k,e}}\right)\mathrm{d}\gamma_t^{k,e}\\ &= \exp\left(\frac{-\beta}{\lambda_t^{k,e}}\right),\end{aligned} \tag{19}$$

where $\beta = \frac{1+\gamma_t^{k,u}}{2^{R_{th}/b_{k,t}}} - 1$.

Accordingly, we define the UEs satisfying the constraint $\mathbb{P}_{k,t}^{out} \leq \epsilon_e$ as secure UEs, where ϵ_e is the maximum allowable value of SOP. The secure UEs $\mathcal{S}_t \subseteq \mathcal{K}$ will be selected to participate in FL training, which is given by

$$\mathcal{S}_t = \{k \in \mathcal{K} | \mathbb{P}_{k,t}^{out} \leq \epsilon_e\}. \tag{20}$$

D. Computation and Transmission Model

During FL training, the latency and energy consumption of each UE in a global round mainly comprise two parts, i.e., local model computing and uploading.

1) Local Computing Model: We denote the training task generated by UE k at time slot t as $(D_{k,t}, C_{k,t})$, where $D_{k,t}$ is the number of task data samples and $C_{k,t}$ (cycles/sample) is the number of CPU cycles required by UE k to process one sample data. Let $f_{k,t}$ denote the computing capability of UE k at time slot t. Note that the number of local training iterations is N_{local}, so the local computing time of UE k at time slot t can be given by

$$\tau_{k,t}^{cmp} = N_{local}\frac{C_{k,t}D_{k,t}}{f_{k,t}}. \tag{21}$$

Accordingly, the energy consumption of UE k for local computing can be written as

$$E_{k,t}^{cmp} = N_{local}\zeta_k C_{k,t} D_{k,t} f_{k,t}^2, \tag{22}$$

where ζ_k is the coefficient of UE k's chip.

2) Transmission Model: The data size of the transmitted model parameters for each UE is the same, denoted as M. For the k-th UE, the transmission time for uploading its local model to the UAV at time slot t is given by

$$\tau_{k,t}^{up} = \frac{M}{R_t^{k,u}}. \tag{23}$$

Accordingly, the communication energy consumed by UE k for model uploading is given by

$$E_{k,t}^{up} = \tau_{k,t}^{up} p_{k,t}. \tag{24}$$

E. Problem Formulation

The global model aggregation latency and broadcast latency are very small and can be neglected [23]. Thus, one-round training time for UE k includes the local computing time and model uploading time, which can be expressed as

$$\tau_{k,t} = \tau_{k,t}^{cmp} + \tau_{k,t}^{up}. \tag{25}$$

Note that synchronous FL is considered in this paper, which is still currently the most commonly used FL approach because of good convergence properties [42]. Thus, the training time of one learning round is determined by the slowest UE, i.e.,

$$\tau_t = \max_{k\in\mathcal{S}_t} \tau_{k,t}. \tag{26}$$

In addition, the total energy consumption of all UEs in the t-th round is given by

$$E_t = \sum_{k\in\mathcal{S}_t} E_{k,t}^{cmp} + E_{k,t}^{up}. \tag{27}$$

Given the above system model, we aim to minimize training time and UE energy consumption while maximizing the number of secure UEs by joint optimizing UAV placement, and communication as well as computation resource allocation under the UAV's energy budget constraint. Thus, we define a system Security-Efficiency Cost (SEC):

$$\kappa_t = \frac{\lambda\tau_t + (1-\lambda)E_t}{K_{s,t}}, \tag{28}$$

where $K_{s,t} = |\mathcal{S}_t|$ is the number of secure UEs at round t, and $\lambda \in [0,1]$ is a constant weight parameter that is used to balance the one-round training time τ_t and UE energy consumption E_t. Accordingly, we consider a long-term SEC minimization problem, which can be formulated as follows:

$$\mathbf{P1}: \min_{\{v_t,\theta_t,\boldsymbol{b}_t,\boldsymbol{f}_t,\boldsymbol{p}_t\}} \sum_{t=1}^{T}\kappa_t \tag{29a}$$

$$\text{s.t. } v_t \in [0, V_{max}],\ \forall t, \tag{29b}$$

$$\theta_t \in [0, 2\pi],\ \forall t, \tag{29c}$$

$$0 \leq x_{u,t} \leq X_{max},\ 0 \leq y_{u,t} \leq Y_{max},\ \forall t, \tag{29d}$$

$$\sum_{k=1}^{K} b_{k,t} \leq B,\ b_{k,t} > 0,\ \forall t, \tag{29e}$$

$$f_k^{min} \leq f_{k,t} \leq f_k^{max},\ \forall k, \forall t, \tag{29f}$$

$$p_k^{min} \leq p_{k,t} \leq p_k^{max},\ \forall k, \forall t, \tag{29g}$$

$$\sum_{t=1}^{T} E_{u,t} \leq E_{max}, \tag{29h}$$

where $\boldsymbol{b}_t = [b_{1,t}, \cdots, b_{K,t}]$, $\boldsymbol{f}_t = [f_{1,t}, \cdots, f_{K,t}]$ and $\boldsymbol{p}_t = [p_{1,t}, \cdots, p_{K,t}]$. (29b)-(29d) limit the UAV's movement. (29e)-(29g) are the bandwidth resource constraint, computing resource and transmit power constraints for UE k, respectively. (29h) indicates that the long-term energy consumption of the UAV should not exceed the energy budget E_{max}, where $E_{u,t} = P_t^{fly}(v_t) + P^{hover}\tau_t$ is the UAV's energy consumption for flying and hovering in the t-th round. The non-convex problem **P1** cannot be solved in an offline manner since it requires real-time decision-making based on the current

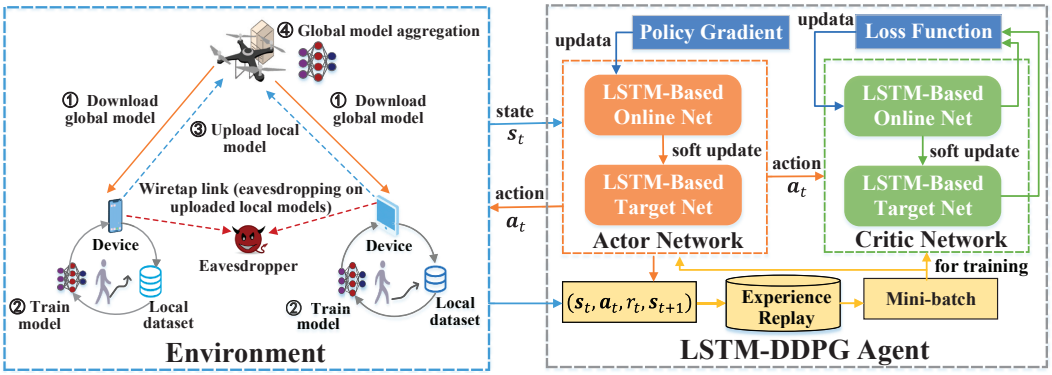


Fig. 2. Overall framework of LSTM-DDPG.

environment information. Thus, we reformulate the problem as an MDP and design an online DRL algorithm.

IV. Proposed Solution

In this section, we first give the definitions of state, action and reward of MDP. Then, we introduce the proposed DRL algorithm named LSTM-DDPG for solving problem **P1**.

A. State, Action, and Reward

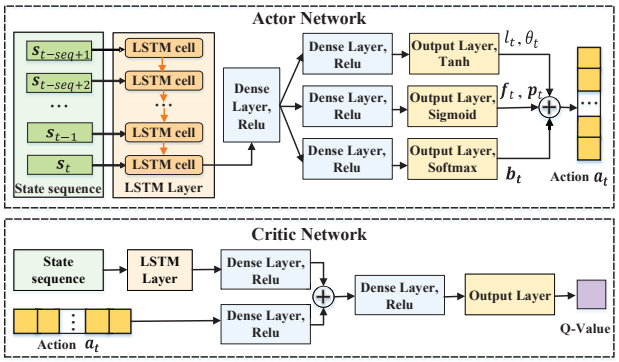
- **State**: The state of the environment at time slot t is defined as $\boldsymbol{s}_t = \{\boldsymbol{q}_{k,t}, \boldsymbol{q}_{e,t}, \boldsymbol{q}_{u,t}, D_{k,t}, C_{k,t}, \tilde{E}_{u,t}, N_{f,t}\}_{k\in\mathcal{K}}$, where $\boldsymbol{q}_{k,t}$, $\boldsymbol{q}_{e,t}$ and $\boldsymbol{q}_{u,t}$ are the locations of UEs, the eavesdropper and UAV, respectively. $(D_{k,t}, C_{k,t})$ is the training task of UE k. $\tilde{E}_{u,t} = E_{max} - \sum_{t=1}^{T} E_{u,t}$ is the current remaining energy of the UAV, and $N_{f,t}$ is the cumulative number of times that the UAV flies out the target area by the time t.
- **Action**: The action is defined as $\boldsymbol{a}_t = \{v_t, \theta_t, \boldsymbol{b}_t, \boldsymbol{p}_t, \boldsymbol{f}_t\}$, where v_t and θ_t denote the flying speed and direction of the UAV, respectively. $\boldsymbol{b}_t$, $\boldsymbol{p}_t$ and $\boldsymbol{f}_t$ denote the bandwidth, transmit power and CPU frequency of UEs, respectively.
- **Reward**: According to the objective function in (29a), the immediate reward can be defined as

$$r_t = -\tilde{\kappa}_t + \Delta_1^{penlty} + \Delta_2^{penlty}, \tag{30}$$

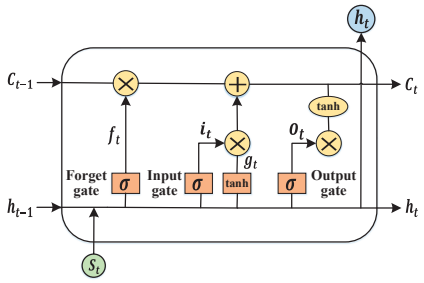
where $\tilde{\kappa}_t = \frac{\lambda\tau_t + (1-\lambda)E_t}{K_{s,t}/K}$, i.e., the enhancement of FL security is measured by the proportion of secure UEs $K_{s,t}/K$. Δ_1^{penlty} and Δ_2^{penlty} are the negative penalty constants when the UAV flies out of the target area or runs out of the energy budget, respectively. That is, (29d) or (29h) is not satisfied.

B. Overall Framework

The overall framework of our proposed LSTM-DDPG algorithm is presented in Fig. 2. LSTM-DDPG consists of a LSTM-based actor network and a LSTM-based critic network. Both the LSTM-based actor and critic networks are further composed of one online network and one target network. By applying the policy gradient method, the actor network can generate a deterministic action according to states observed from the environment. The critic network interacts with the actor network and learns the Q-function by minimizing the loss function to accurately evaluate the action derived from the actor. The target actor and critic networks are respectively copies of the online actor and critic networks, which are used to improve the training efficiency and stability. Moreover, the experience replay buffer stores the historical transition tuples. Then a mini-batch of transitions are randomly sampled from replay memory to train the neural networks, which can effectively decrease data correlation. The details of LSTM-based actor-critic networks are described as follows.



(a) LSTM-based actor-critic networks.



(b) The structure of LSTM cell.

Fig. 3. The structure of LSTM-based actor-critic networks.

C. LSTM-Based Actor-Critic Networks

In the traditional DDPG algorithm, the actor-critic architecture consisting of an actor network and a critic network is adopted, where both the actor and critic networks employ fully-connected deep neural networks (DNNs) to extract state

and action features. However, the fully-connected DNNs fail to capture the temporal pattern of environment dynamics (e.g., user mobility and time-varying UE tasks), resulting in inaccurate state inference. In order to exploit the temporal pattern of states and continuously adapt to environment dynamics, we design the LSTM-based state characterization layer in actor-critic networks. As a modified type of recurrent neural network, LSTM introduces a memory cell to capture long-term dependencies from input sequence data. Therefore, it has been successfully used to deal with various sequential tasks like time series prediction and translation.

The structure of LSTM-based actor-critic networks is illustrated in Fig. 3(a). We first construct a state sequence $\mathbb{S}_t = [\boldsymbol{s}_{t-seq+1}, \ldots, \boldsymbol{s}_{t-1}, \boldsymbol{s}_t]$, which includes the state of the current time slot t and the historical states of the previous $seq-1$ time slots. Then, the states of seq time steps are fed into LSTM cell one by one (one at each time step). As shown in Fig. 3(b), the LSTM cell contains the forget gate f_t, input gate i_t and output gate o_t, which can control the extent of memorizing new information and forgetting historical information. Formally, the relationships between different parts of the LSTM cell can be expressed as follows:

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{C}_t), \tag{31a}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{s}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_o), \tag{31b}$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t \circ \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \circ \tanh(\boldsymbol{W}_c \cdot [\boldsymbol{s}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_c), \tag{31c}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W_f} \cdot [\boldsymbol{s}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_f), \tag{31d}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i [\boldsymbol{s}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_i), \tag{31e}$$

where $\boldsymbol{W}_o, \boldsymbol{W}_c, \boldsymbol{W}_f, \boldsymbol{W}_i, \boldsymbol{b}_o, \boldsymbol{b}_c, \boldsymbol{b}_f$ and $\boldsymbol{b}_i$ are the parameters of the neural networks. $\sigma(\cdot)$ and $\tanh(\cdot)$ are sigmoid and tanh activation functions, respectively. Notation $\circ$ denotes the Hadamard product. Then, the hidden state $\boldsymbol{h}_t$ at the last time step as the output of the LSTM layer, is fed into the fully-connected neural networks (dense layers) to further extract state features for achieving a proper fitting effect. Afterwards, the output layers with different activation functions are employed in the actor network to generate the corresponding policies. Finally, we concatenate all the results of the output layers with the concatenation operator "$\oplus$" to obtain the action $\boldsymbol{a}_t$. The critic network adopts a similar structure to the actor, but the output of the dense layer that takes the state sequence as input is concatenate with the output of another dense layer that takes the action generated by the actor network as input. The concatenated result is mapped into a Q-value via a dense layer and an output layer, which is used to evaluate the effectiveness of action generated by the actor network.

D. Training Process

As illustrated in Algorithm 1, the proposed LSTM-DDPG algorithm operates as follows.

At the beginning of training, we initialize the LSTM-based online networks in the actor $\mu(\cdot)$ and the critic $Q(\cdot)$ with random weights $\boldsymbol{\theta}^{\mu}$ and $\boldsymbol{\theta}^{Q}$, respectively (Line 1). The parameters of the target actor network $\mu'(\cdot)$ and critic target network $Q'(\cdot)$ are copied from the online networks, i.e., $\boldsymbol{\theta}^{\mu'} \leftarrow \boldsymbol{\theta}^{\mu}$ and $\boldsymbol{\theta}^{Q'} \leftarrow \boldsymbol{\theta}^{Q}$ (Line 2). Meanwhile, we initialize

Algorithm 1 Training Process of LSTM-DDPG Algorithm

1: Initialize the LSTM-based actor online net $\boldsymbol{\theta}^{\mu}$ and LSTM-based critic online net $\boldsymbol{\theta}^{Q}$;
2: Initialize the LSTM-based actor target net $\boldsymbol{\theta}^{\mu'}$ and LSTM-based critic target net $\boldsymbol{\theta}^{Q'}$;
3: Initialize the experience replay buffer $\mathcal{B}$;
4: **for** each episode **do**
5: Initialize environment and observe initial state $\boldsymbol{s}_0$;
6: **for** time slot $t = 1, 2, \cdots, T$ **do**
7: Observe state $\boldsymbol{s}_t$ and construct the state sequence $\mathbb{S}_t = [\boldsymbol{s}_{t-seq+1}, \ldots, \boldsymbol{s}_{t-1}, \boldsymbol{s}_t]$ as input of the LSTM-based actor and critic networks;
8: Select action $\boldsymbol{a}_t$ according to the actor network and carry out it;
9: Obtain immediate reward r_t, observe new state $\boldsymbol{s}_{t+1}$, and construct the new state sequence $\mathbb{S}_{t+1} = [\boldsymbol{s}_{t-seq+2}, \ldots, \boldsymbol{s}_t, \boldsymbol{s}_{t+1}]$;
10: Store transition $(\mathbb{S}_t, \boldsymbol{a}_t, r_t, \mathbb{S}_{t+1})$ into $\mathcal{B}$;
11: **if** update **then**
12: Randomly sample a mini-batch of transitions from the replay buffer $\mathcal{B}$;
13: Calculate the target Q-value by (32);
14: Update the critic network by minimizing the critic loss defined in (33);
15: Update the actor network by using the policy gradient defined in (34);
16: Update the target networks according to (35);
17: **end if**
18: **end for**
19: **end for**

the experience replay buffer $\mathcal{B} = \emptyset$ (Line 3). In the exploration phase (Lines 4-10), the UAV agent first receives the initial state $\boldsymbol{s}_0$ at the beginning of each episode. In each time slot t, the agent observes state $\boldsymbol{s}_t$ and constructs the state sequence $\mathbb{S}_t = [\boldsymbol{s}_{t-seq+1}, \ldots, \boldsymbol{s}_t]$. Then, the state sequence is input into the actor network to generate action $\boldsymbol{a}_t$. After executing the action $\boldsymbol{a}_t$, the corresponding reward r_t can be obtained according to the reward function (30), and the next state $\boldsymbol{s}_{t+1}$ as well as the new state sequence $\mathbb{S}_{t+1} = [\boldsymbol{s}_{t-seq+2}, \ldots, \boldsymbol{s}_t, \boldsymbol{s}_{t+1}]$ are updated. Next, the transition $(\mathbb{S}_t, \boldsymbol{a}_t, r_t, \mathbb{S}_{t+1})$ will be stored in the experience replay buffer $\mathcal{B}$. In the update period (Lines 11-16), a mini-batch of transitions are randomly sampled from $\mathcal{B}$ to train DDPG networks. According to the sampled data, the online critic network is used to calculate the estimation Q-value with the inputs of $\mathbb{S}_t$ and $\boldsymbol{a}_t$, i.e., $Q_{eva} = Q(\mathbb{S}_t, \boldsymbol{a}_t; \boldsymbol{\theta}^{Q})$, while the target Q-value is calculated by

$$Q_{tar} = r_t + \gamma Q' \left(\mathbb{S}_{t+1}, \boldsymbol{a}'_{t+1}; \boldsymbol{\theta}^{Q'} \right), \tag{32}$$

where γ is the discount factor for balancing the future reward and the immediate reward, and $\boldsymbol{a}'_{t+1}$ is generated by the target actor network. Note that $\boldsymbol{a}'_{t+1}$ is only used to update the network and will not be executed. Then, we use the gradient descent method to minimize the loss function of the critic network, which is defined as follows:

$$L(\boldsymbol{\theta}^{Q}) = \mathbb{E}\left[\left(Q_{eva} - Q_{tar} \right)^2 \right]. \tag{33}$$

Similarly, the policy gradient for updating the actor network can be expressed as

$$\nabla_{\boldsymbol{\theta}^{\mu}} J = \mathbb{E}_{\mathbb{S}_t \sim \varrho} \Big[\nabla_{\boldsymbol{a}} Q\left(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta}^{Q}\right) |_{\boldsymbol{s}=\mathbb{S}_t, \boldsymbol{a}=\mu(\mathbb{S}_t;\boldsymbol{\theta}^{\mu})} \nabla_{\boldsymbol{\theta}^{\mu}} \mu\left(\boldsymbol{s}; \boldsymbol{\theta}^{\mu}\right) |_{\boldsymbol{s}=\mathbb{S}_t} \Big], \tag{34}$$

where ϱ denotes the distribution of $\mathbb{S}_t$. Finally, the target critic and actor networks are updated by a soft update method:

$$\begin{cases} \boldsymbol{\theta}^{\mu'} \leftarrow (1-\tau)\boldsymbol{\theta}^{\mu'} + \tau\boldsymbol{\theta}^{\mu}, \\ \boldsymbol{\theta}^{Q'} \leftarrow (1-\tau)\boldsymbol{\theta}^{Q'} + \tau\boldsymbol{\theta}^{Q}, \end{cases} \tag{35}$$

where $\tau \in [0, 1]$ is the step size of soft update.

V. Simulation Results

In this section, extensive experiments are conducted to evaluate the effectiveness of our proposed LSTM-DDPG algorithm in improving FL performance. Simulation setup is first illustrated, followed by results and discussions.

A. Simulation Setup

We consider that the UAV serves ground mobile UEs in a 200 m $\times$ 200 m area. The eavesdropper and UEs follow the Gauss-Markov mobility model with the average speed $\bar{v}_i$ = 1 m/s, direction $\bar{\theta}_i = [0, 2\pi)$ and memory level $\eta_{1,i} = \eta_{2,i} =$ 0.4. The number of training data samples for each UE (i.e., $D_{k,t}$) is uniformly distributed in [400, 600], and the number of CPU cycles required for computing one sample data (i.e., $C_{k,t}$) is uniformly distributed in [1, 3]$\times 10^4$. The UAV flies at the fixed height H = 80 m with maximum flight speed v_{max} = 25 m/s. Moreover, we assume that the loss function $L_k(\cdot)$ is L-Lipschitz and μ-strongly convex, and further set the parameters $L = 4$, $\mu = 2$, $\delta = \frac{1}{4}$, $\xi = \frac{1}{4}$, $\eta = 0.5$ and $\epsilon_0 = 10^{-3}$, respectively.

For the LSTM-DDPG algorithm, the actor network consists of one LSTM layer with 800 neurons, one shared dense layer with 512 neurons, three-branch dense layers with 256 neurons and three-branch output layers. Each of the three-branch output layers is also the dense layer, whose structure is determined by its corresponding action dimension. The critic network consists of one LSTM layer with 800 neurons, two-branch dense layers with 256 neurons, one shared dense layer with 256 neurons and one output layer with 1 neuron. In actor and critic networks, the length of the state sequence input to the LSTM layer is 3. We adopt the Adam optimizer to train the actor and critic with the same learning rate 0.001. Moreover, we set the discounted factor to 0.9, the batch size to 32, the size of replay buffer to 10000, and the soft update parameter to 0.001. Other simulation parameters are listed in Table II.

To evaluate the performance of our proposed LSTM-DDPG algorithm, we compare it with the following approaches:

- PPO [26]: It is a highly stable state-of-the-art model-free DRL algorithm for obtaining the online control policy for UAV locations and resource allocation.
- DDPG [27]: It learns the online control policy of the UAV and shows excellent performance in the multi-objective optimization problem.

TABLE II
Simulation Parameters

Parameter	Symbol	Value
Number of UEs	K	10
Model size	M	500 Kb
CPU frequencies of UEs	$f_{k,t}$	[0.1, 2.0] GHz
Effective capacitance coefficient	ζ_k	10^{-28}
Environmental parameters	a, b	11.95, 0.136
Reference channel gain	α_0, α_1	-40 dB, -60 dB
Path loss exponent	β_0, β_1	2.3, 3
Communication bandwidth	B	3 MHz
Noise power spectral density	N_u, N_e	-174 dBm/Hz
Transmit power of UEs	$p_{k,t}$	[5, 100] mW
Weight parameter	λ	0.6
Air density	ρ	1.225 kg/m^3
Tip speed	U_{tip}	120 m/s
Blade profile power	P_0	79.86 W
Derived power	P_1	88.63 W
Body resistance ratio	d_0	0.6
Robustness of the rotor	s	0.05
The area of the rotor disk	A	0.503 m^2
Mean rotor induced velocity	v_0	4.03 m/s

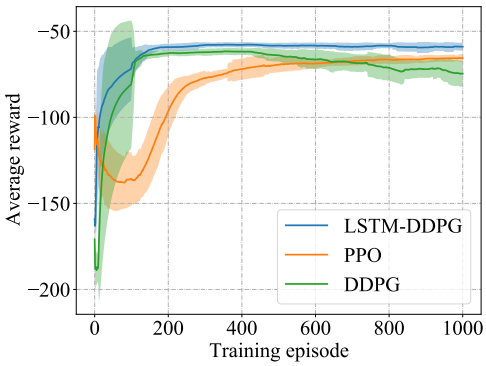


Fig. 4. Training convergence comparison of different DRL algorithms.

- Uniform-FA: The location of the UAV and the transmit power of UEs are fixed, while the bandwidth resources are equally allocated to each UE [17]. The CPU-frequency allocation decisions of UEs are learned by using the proposed LSTM-DDPG algorithm.
- Uniform-PC: The UAV's location and the CPU frequency of UEs are fixed, while the bandwidth resources are equally allocated to each UEs and only the transmit power of UEs is optimized, which is inspired by [15]. Similarly, we use the proposed LSTM-DDPG algorithm to learn power control decisions.

B. Results and Discussions

*1) **Training Convergence**:* In Fig. 4, we show the convergence performance of different DRL algorithms over training episodes. Specifically, the average rewards are represented by solid curves and the standard deviations are represented by shaded areas. All average reward values are obtained by applying the moving average method with a window of 100 episodes. It can be seen that LSTM-DDPG attains not only the higher average reward but also the lower variance than the PPO and DDPG algorithms, which indicates that our proposed LSTM-DDPG algorithm outperforms PPO and DDPG on both FL system performance and learning stability. Moreover, we

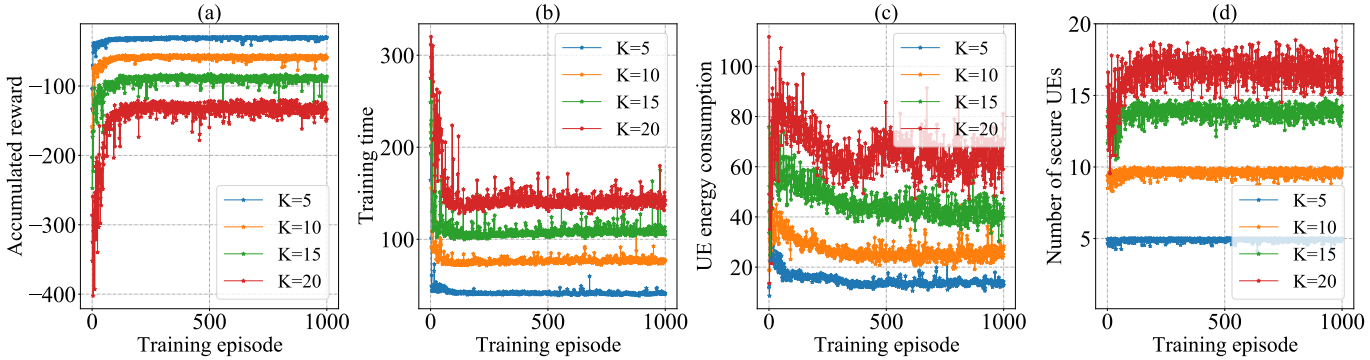


Fig. 5. Training curves: (a) Accumulated reward, (b) training time, (c) UE energy consumption and (d) number of secure UEs versus episode with different numbers of UEs.

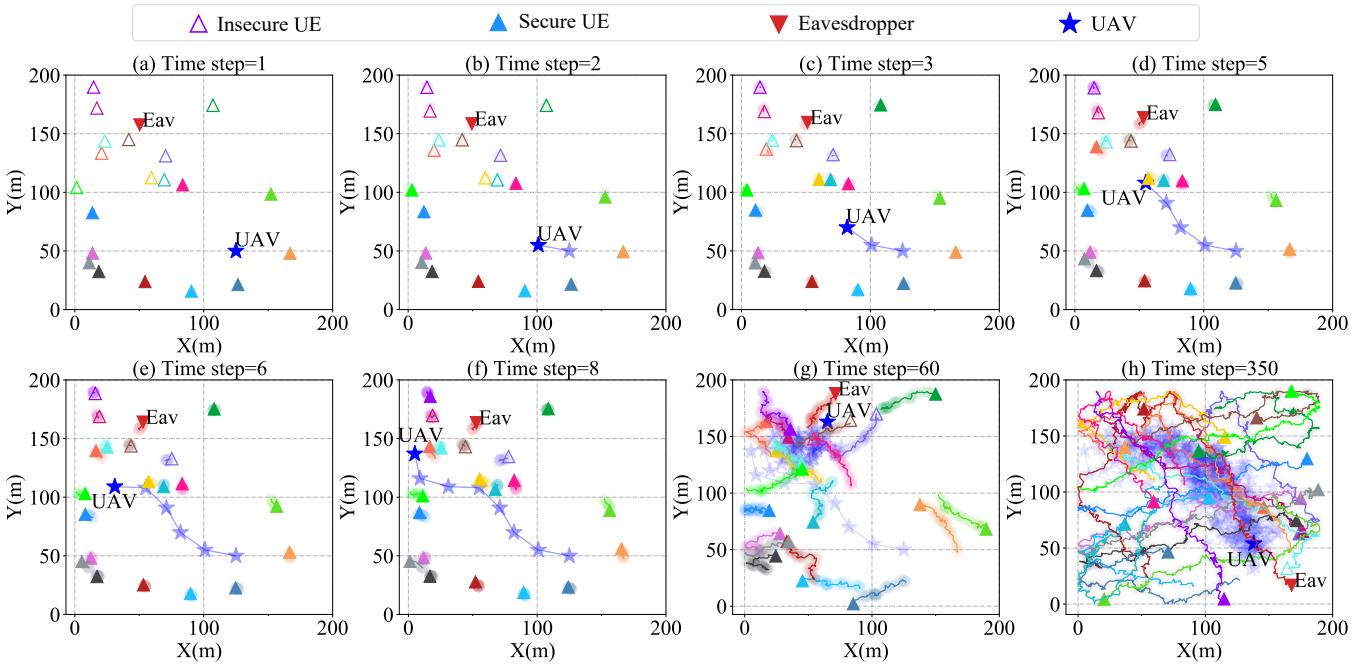


Fig. 6. UAV locations and secure UE selection obtained by LSTM-DDPG for a scenario of a moving eavesdropper and 20 moving UEs.

can observe that the training stability of PPO is significantly better than that of DDPG. However, the average reward and convergence speed of DDPG are slightly better than those of PPO at first about 400 episodes, but DDPG undergoes a performance deterioration in later training. In view of this, we choose DDPG as the starting point of our design, and then leverage the LSTM networks to exploit the potential of DDPG. Overall, the results confirm the effectiveness of our designed LSTM-DDPG algorithm, which can better learn environment dynamics to train DDPG more stably and accurately.

Fig. 5 illustrates the convergence performance of the proposed LSTM-DDPG algorithm with different numbers of UEs. Therein, the changes of accumulated reward, FL training time, UE energy consumption and the number of secure UEs over time during training are shown in Fig. 5(a), Fig. 5(b), Fig. 5(c) and Fig. 5(d), respectively. First, we can observe from Fig. 5(a) that all the learning curves fluctuate violently at the beginning because the policy learned by the agent is incorrect, which leads to a large loss in actor-critic networks. Afterwards, the accumulated rewards of all the learning curves gradually grow as training progresses and then converge steadily at a high level. In addition, it can be seen from Fig. 5(b), Fig. 5(c) and Fig. 5(d) that our algorithm can explore and learn a better policy over time, which can simultaneously reduce the training time and UE energy consumption as well as enhance FL security. Finally, the LSTM-DDPG algorithm can converge quickly and stably under different numbers of UEs, which demonstrates that our algorithm has good convergence performance.

*2) **UAV Placement and Secure UE Selection**:* Fig. 6 shows the UAV trajectory and secure UE selection polices learned by LSTM-DDPG, under time-varying task arrivals of UEs as well as random movement of the eavesdropper and UEs. Therein, the blue pentagrams and transparent pentagrams respectively represent the current locations and previous locations of the UAV, the triangles and transparent circles respectively represent the current locations and previous locations of the eavesdropper and UEs, the solid triangles represent secure

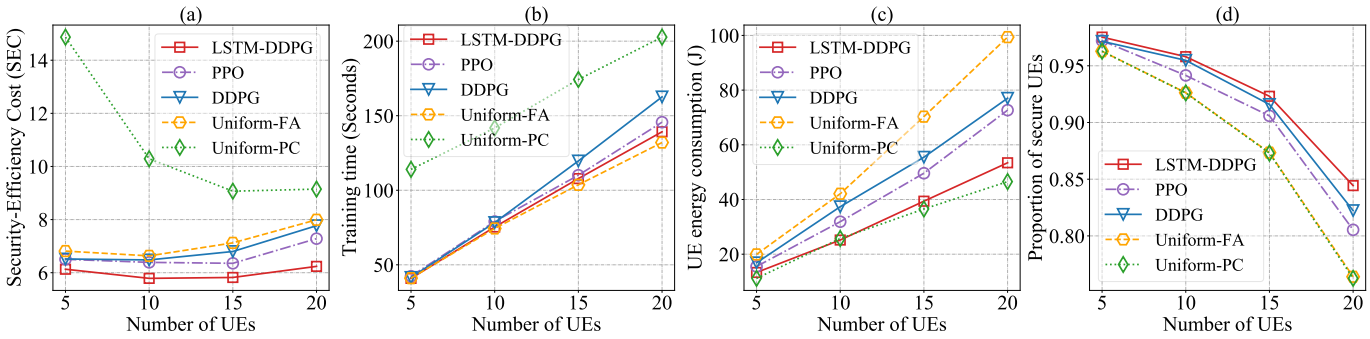


Fig. 7. (a) Security-Efficiency Cost (SEC), (b) FL training time, (c) UE energy consumption and (d) proportion of secure UEs versus number of UEs.

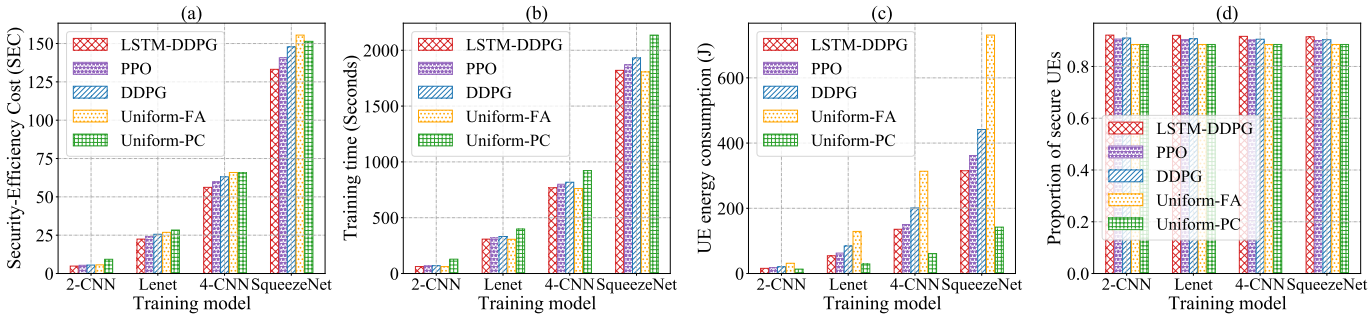


Fig. 8. (a) Security-Efficiency Cost (SEC), (b) FL training time, (c) UE energy consumption and (d) proportion of secure UEs versus training model.

UEs that will be selected to participate in FL training, and the hollow triangles represent insecure UEs that are not selected to participate in FL training. In Fig. 6(a), we can see that only 50% of secure UEs can participate in FL training at the beginning. The UEs who are close to the eavesdropper (abbreviated as Eav) and far away from the UAV, are insecure and vulnerable to eavesdropping threats. As shown in Fig. 6(b)-(f), in order to enhance FL security, the UAV gradually moves to the locations near the insecure UEs to improve their secrecy capacities. It can be observed from Fig. 6(f) that 17 secure UEs can participate in FL training in the eighth time slot, that is, the proportion of secure UEs increases from 50% to 85%. Furthermore, we can analyze the location distribution and movement trend of the UAV in a longer time period from Fig. 6(g) and Fig. 6(h). First of all, we can find that when there are some insecure UEs who are vulnerable to eavesdropping risks, the UAV tends to chase the moving eavesdropper and then move back and forth in its nearby area. The UAV can also fly closer to the insecure UEs to improve their secrecy rate, thus increasing the number of secure UEs. Accordingly, the FL security performance is enhanced. Secondly, in order to reduce the training time and UE energy consumption, the UAV tends to fly to locations with high UE distribution density or fly to locations which most users will move to in the near future. These observations indicate that the proposed LSTM-DDPG algorithm can well adapt to the changing new environments, and make wise decisions with foresight to simultaneously improve the efficiency and security of the FL system.

*3) **Performance Comparison Versus Number of UEs**:* Fig. 7 shows the performance comparison for different algorithms with various numbers of UEs. First of all, we can find that Uniform-FA achieves the minimization of training time and the maximization of UE energy consumption, while Uniform-PC achieves the minimization of UE energy consumption and the maximization of training time. The reasons are as follows: (1) The Uniform-FA algorithm only optimizes computing resources related to computation latency and computation energy consumption. Because the uploaded model size is small, the training time mainly depends on computation latency rather than communication delay. In order to reduce the training time, the CPU frequency of all UEs increases, so UE energy consumption increases. Since the training time accounts for a larger weight, Uniform-FA will mainly optimize the training time to maximize the system reward. (2) The Uniform-PC algorithm only optimizes transmit power of all UEs related to communication energy consumption and system security performance. Since the transmit power optimization has a greater impact on UE energy consumption than that on security performance, the optimal strategy of Uniform-PC is to control the transmit power of all UEs to be very low. Thus, UE energy consumption is greatly reduced, training time is significantly increased, and security performance is not optimized. Thus, we can approximately regard the training time obtained by Uniform-FA as the optimal value and UE energy consumption obtained by Uniform-PC as the optimal value, to evaluate the performance of the proposed algorithm in achieving multi-objective optimization.

From Fig. 7(a), we can observe that LSTM-DDPG consistently outperforms all baseline approaches in terms of Security-Efficiency Cost (SEC). On average, for SEC, LSTM-DDPG significantly improves 9.5%, 12.7%, 15.8% and 42.5% over PPO, DDPG, Uniform-FA and Uniform-PC, respectively.

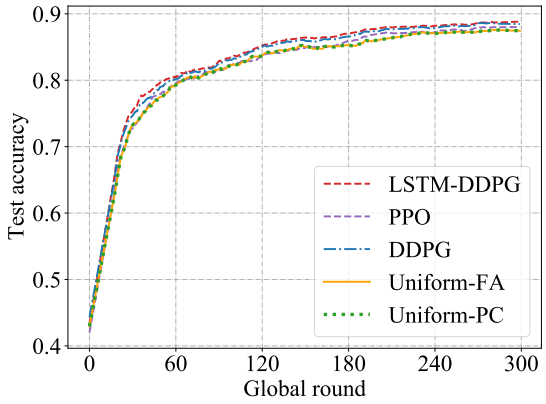


Fig. 9. Accuracy comparison versus number of global rounds.

We can observe from Fig. 7(b) that the FL training time increases with the increase of UEs. It is easy to understand that since the UAV server needs to wait for all selected UEs to complete and upload their local model updates at each round, the limited bandwidth resource and the increasing number of UEs will exacerbate the communication bottleneck of FL, resulting in consuming longer time for global model aggregation. As shown in Fig. 7(c), there is no doubt that the total energy consumption of all UEs increases with the increase of UEs. In Fig. 7(d), as the number of UEs increases, the bandwidth allocated to each UE decreases, so the SOP increases and the number of secure UEs decreases. We can also observe that the proportion of secure UEs in the Uniform-FA and Uniform-PC schemes is the same, because neither of them optimizes security performance. In addition, it can be seen that the training time of LSTM-DDPG is approximately equal to or very close to that of Uniform-FA, and UE energy consumption of LSTM-DDPG is approximately equal to or very close to that of Uniform-PC. On average, the LSTM-DDPG algorithm can save 4.0%, 7.8% of training time and 20.6%, 28.4% of UE energy consumption compared with the PPO and DDPG algorithms, respectively. It can also be observed from Fig. 7(d) that LSTM-DDPG achieves superior security performance. For 20 UEs, LSTM-DDPG improves the proportion of secure UEs by 4.8%, 2.7%, 10.5% and 10.5% compared with PPO, DDPG, Uniform-FA and Uniform-PC, respectively. In summary, the above results show the necessity of jointly optimizing UAV placement, bandwidth allocation, CPU frequency and transmit power of UEs for reducing the FL training time and UE energy consumption and enhancing FL security. In addition, our proposed LSTM-DDPG algorithm can make better decisions for UAV placement and resource management, thus achieving excellent performance in FL efficiency and security.

*4) **Performance Comparison Versus Training Model**:* Fig. 8 presents the Security-Efficiency Cost (SEC), training time, UE energy consumption and proportion of secure UEs achieved by different algorithms under different training models. In this simulation, the two-layer Convolutional Neural Network (2-CNN), Lenet [43], four-layer CNN (4-CNN) and SqueezeNet [44] models are respectively trained with FashionMNIST dataset. The classic FedAvg is used as the FL algorithm, and the number of global learning rounds is set to 300. The number of UEs is set to 10, and only secure UEs are selected to participate in FL training. From Fig. 8(a), it can be seen that LSTM-DDPG consistently outperforms all baseline approaches in terms of the key metric, SEC (the lower the better). On average, for SEC, LSTM-DDPG significantly improves 6.8%, 11.2%, 14.9% and 23.7% over PPO, DDPG, Uniform-FA and Uniform-PC, respectively. From Fig. 8(b) and Fig. 8(c), we can see that the training time and UE energy consumption successively increase on the 2-CNN, LeNet, 4-CNN and SqueezeNet models. This is because the model sizes of 2-CNN, LeNet, 4-CNN and SqueezeNet are in increasing order, which are 0.083, 0.462, 1.165 and 2.776 MB, respectively. When the model size becomes larger, each participating UE needs to spend more time transmitting its local model or use a higher transmit power, resulting in longer training time and more energy consumption of UEs. In addition, it can be observed that the proposed LSTM-DDPG algorithm outperforms other baseline approaches in terms of efficiency and security performance. On average, the LSTM-DDPG algorithm can reduce 4.8%, 7.7% of training time and 11.7%, 30.0% of UE energy consumption compared with the PPO and DDPG algorithms, respectively. It can also be seen from Fig. 8(d) that the proportion of secure UEs for the LSTM-DDPG algorithm is the highest. This is mainly because LSTM-DDPG can capture temporal correlation of state sequences to better explore actions with high rewards. In contrast, DDPG with the fully-connected DNNs fails to accurately learn the environment dynamics due to lacking the state representative ability. Since PPO is an on-policy DRL algorithm without a replay buffer to integrate previous experience, it is always difficult to converge at an optimal point.

C. Accuracy Comparison Versus Global Round

In this subsection, we compare the test accuracy of FL versus the number of global rounds under different approaches. In the simulation, we employ FedAvg model aggregation mechanism and a 2-CNN model is trained with FashionMNIST. The number of UEs is set to 20. As shown in Fig. 9, in the LSTM-DDPG, PPO and DDPG algorithms, FL requires about 200, 270 and 230 global rounds to achieve an accuracy of 88%, respectively. In the Uniform-FA and Uniform-PC algorithms, FL requires about 250 global rounds to achieve an accuracy of 87%. To sum up, LSTM-DDPG achieves the best FL performance with the highest accuracy and the fastest convergence, and DDPG's FL performance is comparable to LSTM-DDPG. The reason is that the number of secure UEs of the LSTM-DDPG and DDPG algorithms is more than that of PPO, Uniform-FA and Uniform-PC. In other words, LSTM-DDPG and DDPG can select more UEs to participate in the FL model aggregation, thus achieving higher accuracy and faster convergence speed.

VI. Conclusion

In this paper, we investigate the problem of designing a secure and efficient FL framework in UAV-enabled networks. Specifically, we propose a secure UE selection scheme based on SOP to prevent the uploaded model parameters from

being wiretapped by a malicious eavesdropper. Then, we formulate a joint UAV placement and resource allocation problem to reduce training time and UE energy consumption while enhancing FL security, subject to the UAV's energy budget constraint. Considering the random movement of the eavesdropper and UEs as well as online task generation on UEs, we present a LSTM-based DDPG algorithm to solve this problem, which makes full use of historical state information and replaces the fully-connected neural networks with LSTM, so as to capture the complex temporal correlation of environment states. Simulation results show that the proposed algorithm achieves superior performance in terms of training time, energy consumption and security of FL.

REFERENCES

[1] K. B. Letaief *et al.*, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[2] N. Cheng *et al.*, "AI for UAV-assisted IoT applications: A comprehensive review," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14438–14461, Aug. 2023.

[3] C. Dong, Y. Shen, Y. Qu, K. Wang, J. Zheng, Q. Wu, and F. Wu, "UAVs as an intelligent service: Boosting edge intelligence for air-ground integrated networks," *IEEE Netw.*, vol. 35, no. 4, pp. 167–175, Jul./Aug. 2021.

[4] S. A. Khowaja *et al.*, "Towards energy-efficient distributed federated learning for 6G networks," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 34–40, Dec. 2021.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[6] W. Y. B. Lim *et al.*, "UAV-assisted communication efficient federated learning in the era of the artificial intelligence of things," *IEEE Netw.*, vol. 35, no. 5, pp. 188–195, Sep./Oct. 2021.

[7] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2021, pp. 1–10.

[8] J. Shen *et al.*, "RingSFL: An adaptive split federated learning towards taming client heterogeneity," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2023.3309633.

[9] J. Shen *et al.*, "Effectively heterogeneous federated learning: A pairing and split learning based approach," 2023, *arXiv:2308.13849*.

[10] Z. Liu, J. Guo, K. Lam, and J. Zhao, "Efficient dropout-resilient aggregation for privacy-preserving machine learning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1839–1854, 2023.

[11] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 493–506.

[12] K. Wei *et al.*, "Low-latency federated learning over wireless channels with differential privacy," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 290–307, Jan. 2022.

[13] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1055–1069, Apr.–Jun. 2021.

[14] C. Feng *et al.*, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3582–3592, May. 2022.

[15] J. Yao and N. Ansari, "Secure federated learning by power control for internet of drones," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1021–1031, Dec. 2021.

[16] M. Chen, H. V. Poor, W. Saad and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.

[17] Q. V. Do, Q. V. Pham and W. J. Hwang, "Deep reinforcement learning for energy-efficient federated learning in UAV-enabled wireless powered networks," *IEEE Commun. Lett.*, vol. 26, no. 1, pp. 99–103, Jan. 2022.

[18] 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.

[19] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[20] Y. Jing, Y. Qu, C. Dong, Y. Shen, Z. Wei and S. Wang, "Joint UAV location and resource allocation for air-ground integrated federated learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

[21] Q.-V. Pham, M. Le, T. Huynh-The, Z. Han, and W.-J. Hwang, "Energy-efficient federated learning over UAV-enabled wireless powered communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4977–4990, May. 2022.

[22] N. H. Tran, W. Bao, A. Zomaya, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. Int. Conf. Comput. Commun. (INFOCOM)*, Paris, 2019, pp. 1387–1395.

[23] X. Zhou, J. Zhao, H. Han and C. Guet, "Joint optimization of energy consumption and completion time in federated learning," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2022, pp. 1005–1017.

[24] S. Tang, W. Zhou, L. Chen, L. Lai, J. Xia, and L. Fan, "Battery-constrained federated edge learning in UAV-enabled IoT for B5G/6G networks," *Phys. Commun.*, vol. 47, Aug. 2021, Art. no. 101381.

[25] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, May. 2020.

[26] M. Samir, C. Assi, S. Sharafeddine and A. Ghrayeb, "Online altitude control and scheduling policy for minimizing AoI in UAV-assisted IoT wireless networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2493–2505, Jul. 2022.

[27] Y. Yu, J. Tang, J. Huang, X. Zhang, D. K. C. So and K.-K. Wong, "Multi-objective optimization for UAV-assisted wireless powered IoT networks based on extended DDPG algorithm", *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6361–6374, Sep. 2021.

[28] Z. Xiong *et al.*, "UAV-assisted wireless energy and data transfer with deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 85–99, Mar. 2021.

[29] Y. Zhang, Z. Mou, F. Gao, J. Jiang, and Z. Han, "UAV-enabled secure communications by multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11599–11611, Oct. 2020.

[30] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing UAV communications via joint trajectory and power control," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1376–1389, Feb. 2019.

[31] S. Li, B. Duo, M. D. Renzo, M. Tao, and X. Yuan, "Robust secure UAV communications with the aid of reconfigurable intelligent surfaces," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6402–6417, Oct. 2021.

[32] Y. Tao, X. Wang, B. Li, and C. Zhao, "Pilot spoofing attack detection and localization with mobile eavesdropper," *IEEE Trans. Mobile Comput.*, vol. 22, no. 3, pp. 1688–1701, Mar. 2023.

[33] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.

[34] S. Batabyal and P. Bhaumik, "Mobility models, traces and impact of mobility on opportunistic routing algorithms: A survey," *IEEE Commun. Surveys Tut.*, vol. 17, no. 3, pp. 1679–1707, Sep. 2015.

[35] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[36] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, 2014.

[37] N. Yang, L. Wang, G. Geraci, M. Elkashlan, J. Yuan, and M. D. Renzo, "Safeguarding 5G wireless communication networks using physical layer security," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 20–27, Apr. 2015.

[38] Y. Liu *et al.*, "Enhancing the physical layer security of non-orthogonal multiple access in large-scale networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1656–1672, Mar. 2017.

[39] B. Zheng, and R. Zhang, "Intelligent reflecting surface enhanced OFDM: channel estimation and reflection optimization," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 518–522, Apr. 2020.

[40] X. Yu, D. Xu, Y. Sun, D. W. K. Ng, and R. Schober, "Robust and secure wireless communications via intelligent reflecting surfaces," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2637–2652, Nov. 2020.

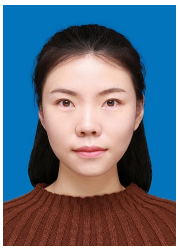
[41] M. S. Allahham, A. A. Abdellatif, A. Mohamed, A. Erbad, E. Yaacoub, and M. Guizani, "I-SEE: Intelligent, secure, and energy-efficient techniques for medical data transmission using deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6454–6468, Apr. 2021.

[42] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

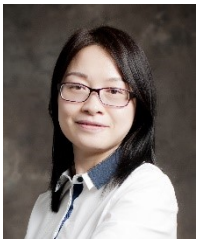
[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[44] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and >0.5MB model size," 2016, *arXiv:1602.07360*.

Xiaokun Fan received the B.S. degree in network engineering from the Hebei University of Technology, Hebei, China, in 2019. She is currently working toward the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her current research interests include unmanned aerial vehicles, edge intelligence, and federated learning.



Yali Chen received the B.S. degree in communication engineering from the Taiyuan University of Science and Technology, Taiyuan, China, in 2016, and the Ph.D. degree in communication and information system from Beijing Jiaotong University, Beijing, China, in 2022. She is currently an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her research interests include unmanned aerial vehicles, optimization under uncertainty, and mobile edge computing.



Min Liu (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Xi'an Jiaotong University, China, in 1999 and 2002, respectively. She got her Ph.D. degree in computer science from the Graduate University of the Chinese Academy of Sciences in 2008. She is currently a professor at the Networking Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences. Her current research interests include mobile computing and edge intelligence.

Sheng Sun is currently an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences. She received her B.S. degree from Beihang University, and her Ph.D. from the Institute of Computing Technology, Chinese Academy of Sciences. Dr. Sun has led or executed 5 major funded research projects and published over 20 technical papers in journals and conferences related to computer network and distributed systems, including IEEE Transactions on Parallel and Distributed Systems (TPDS), IEEE Transactions on Mobile Computing (TMC), and IEEE International Conference on Computer Communications (INFOCOM). Her research interests include federated learning, edge intelligence, and privacy computing.

Zhuotao Liu received the BS degree in electrical engineering from Shanghai Jiao Tong University in 2012 and the Ph.D. degree in computer engineer from the University of Illinois at Urbana-Champaign in 2017. He is currently an assistant professor with the Institute for Network Sciences and Cyberspace, Tsinghua University. He was a technical lead with Google, managing Google's private Wide Area Network that hyper-connects Google's massive-scale Datacenters across the globe. His research interests include systems and networking, with special interest in blockchain infrastructure, next-generation network architecture, privacy-preserving computation, systems security, and datacenter networking.

Ke Xu (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He serves as a Full Professor at Tsinghua University. He has published more than 200 technical papers and holds 11 U.S. patents in the research areas of next-generation internet, blockchain systems, the Internet of Things, and network security. He is a member of ACM. He served as the Steering Committee Chair for IEEE/ACM IWQoS. He has guest-edited several special issues in IEEE and Springer journals. He is an Editor of the IEEE Internet of Things Journal.

Zhongcheng Li received the B.S. degree in computer science from Peking University in 1983, and the M.S. and Ph.D. degrees from Institute of Computing Technology, Chinese Academy of Sciences in 1986 and 1991, respectively. From 1996 to 1997, he was a visiting professor at University of California at Berkeley. He is a professor at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer networks and communications.