Aerial Base Station Positioning and Power Control for Securing Communications: A Deep Q-Network Approach

Aly Sabri Abdalla*, Ali Behfarnia†, and Vuk Marojevic*
*Department of Electrical and Computer Engineering, Mississippi State University, MS 39762, USA
†Department of Engineering, University of Tennessee at Martin, TN, USA
Email: asa298@msstate.edu, a.behfarnia@tennessee.edu, vuk.marojevic@msstate.edu

***Abstract*—The unmanned aerial vehicle (UAV) is one of the technological breakthroughs that supports a variety of services, including communications. UAVs can also enhance the security of wireless networks. This paper defines the problem of eavesdropping on the link between the ground user and the UAV, which serves as an aerial base station (ABS). The reinforcement learning algorithms Q-learning and deep Q-network (DQN) are proposed for optimizing the position of the ABS and the transmission power to enhance the data rate of the ground user. This increases the secrecy capacity without the system knowing the location of the eavesdropper. Simulation results show fast convergence and the highest secrecy capacity of the proposed DQN compared to Q-learning and two baseline approaches.**

Keywords: Deep reinforcement learning, Q-learning, eavesdropping, UAV, security.

I. Introduction

Unmanned aerial vehicles (UAVs) support various applications in advanced cellular networks. A UAV can be an aerial base station (ABS), aerial relay (AR), or aerial user equipment (AUE) in the cellular network. As a network support node, it can enhance the performance of the end users. Steps to identify the challenges and solutions of emerging cellular networks serving UAVs are being taken by the 3rd Generation Partnership Project (3GPP) [1]. Major challenges are radio frequency (RF) interference and security. Security is increasingly important in modern wireless networks and needs to be ensured for establishing communications links between UAVs and terrestrial nodes [2]–[4].

Different types of attacks have been studied with the aim of strengthening the security of wireless communications links with trusted cellular network-connected UAVs. Eavesdropping is a passive attack that can compromise the confidentiality and privacy of control and user data. Early research [5], [6] considering both hovering and moving UAVs study the performance of an AR for safeguarding the ground communications links between terrestrial nodes against eavesdropping attacks. Reference [7] investigates the same problem but for multiple ground users and multiple eavesdroppers by optimizing the position of the AR and the transmission power. In [8], the UAV is deployed as an ABS to serve ground users under attack with the goal of optimizing the 3D position of the UAV for maximizing the secrecy rate. The authors of [9] maximize the secrecy rate of legitimate users while the ABS position is optimized without eavesdropper location information.

In other lines of work, learning approaches have been used for solving the above problem. Reference [10] develops model-free reinforcement learning (RL) algorithms to maximize the system secrecy rate by transmitting artificial noise with optimized beamforming from the AR. A multi-agent deep RL has been proposed in [11] to maximize the secrecy capacity by jointly optimizing the trajectory, transmit power, and jamming power for both relay and friendly jamming UAVs protecting against eavesdroppers with known locations. Nevertheless, limited study items utilized RL in secrecy rate analysis, such as [10], [11]. These approaches assume knowledge of the eavesdroppers and their locations. Also, learning methods are only recently being explored for physical layer security. To the best of our knowledge, none of recent studies have explored the potential of RL solutions without assuming the availability of imperfect or perfect location information of the eavesdropper.

In this paper, we propose a RL algorithm for an ABS to assist with the uplink transmission of ground users that are subject to eavesdropping. Our solution is based on a deep Q-network (DQN) that aims to maximize the secrecy capacity by optimizing the legitimate capacity of the ground user. This is achieved by finding the position of the ABS and the transmission power that maximizes the data and secrecy rates without the eavesdropper's location information.

The rest of the paper is organized as follows. Section II provides the system model and problem formulation. Section III introduces the DQN as our learning-based physical layer security solution to eavesdropping. Section IV presents the numerical analysis and Section V derives the conclusions.

II. System Model and Problem Formulation

The scenario studied in this paper is illustrated in Fig. 1. The ground user communicates with the ABS in the presence of a terrestrial eavesdropper. The user and eavesdropper are independent of each other. The ABS is positioned to provide a secure and reliable uplink (UL) for the user under attack. It achieves this by leveraging its 3D mobility and strong line of sight (LoS) channel that allows low power transmission. Without loss of generality, we model and analyze the UL

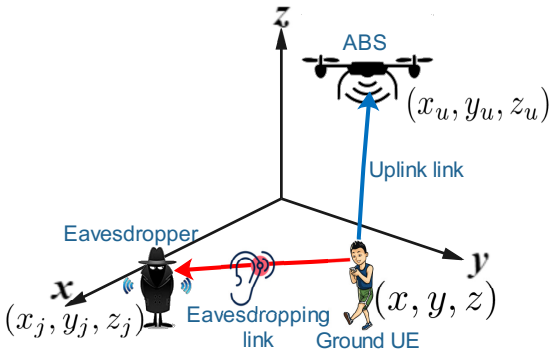

Fig. 1: The simulation scenario.

channel here. The same principles can be applied for the downlink (DL). In the rest of this paper, we use the UAV and ABS interchangeably.

We define the location of the ABS, the eavesdropper, and the user in the 3D Cartesian coordinate system as (x_u, y_u, z_u), (x_j, y_j, z_j), and (x, y, z), respectively. For practicality, time slots are used to capture the different radio frames and statistical channel conditions, as well as the momentarily static position of the nodes. Therefore, the ABS coordinates in time slot t are expressed as $(x_u[t], y_u[t], z_u[t])$.

A. Communications Channel

The received signal at the ABS when the ground user equipment (UE) transmits the signal s with power P is as follows

$$r = \sqrt{P\hbar}s + n, \tag{1}$$

where $\hbar$ is the air-to-ground (A2G) channel gain between the UE and the ABS, and n denotes the additive white Gaussian noise (AWGN) of zero mean and σ^2 variance. Based on the measurements presented in [12], the LoS model is a good approximation for the A2G channel in rural areas. The channel gain has a path loss exponent of two and can be written as

$$\begin{aligned}\hbar[t] &= \zeta_0 d_u^{-2}[t] \\ &= \frac{\zeta_0}{(x[t]-x_u[t])^2 + (y[t]-y_u[t])^2 + (z[t]-z_u[t])^2},\end{aligned} \tag{2}$$

where ζ_0 is the channel gain at the reference point d_0 = 1 m. Parameter ζ_0 is the same for the A2G and the ground-to-ground (G2G) channel because of the same the antenna gains and carrier frequency in both cases. Parameter $d_u[t]$ denotes the distance between the ground user and the ABS.

The capacity of the channel between the UE and the ABS in time slot t is calculated as follows:

$$C_u[t] = \log_2\left(1 + \frac{P[t]\hbar[t]}{\sigma^2}\right) = \log_2\left(1 + \frac{\zeta_0 P[t]}{d_u^2[t]\sigma^2}\right), \tag{3}$$

where ζ_0/σ^2 is the signal-to-noise ratio (SNR) at the reference point.

B. Eavesdropping Channel

Often, UEs have fewer antennas and transmit in broadcast mode without beamforming. The eavesdropper is intercepting the transmitted signal from the ground UE and the received signal can be formulated as

$$v = \sqrt{P\emptyset}s + w, \tag{4}$$

where w denotes the zero-mean AWGN with a variance of σ^2 at the eavesdropper. As a result of the G2G communications link between the UE and the eavesdroppers, the exponent of the path loss for this link is assumed to be four. Within the time slot t, the G2G channel gain represented by $\emptyset$ between the UE and eavesdropper can be modeled as

$$\begin{aligned}\emptyset &= \zeta_0 d_j^{-4}[t] \\ &= \frac{\zeta_0}{\left((x[t]-x_j[t])^2 + (y[t]-y_j[t])^2 + (z[t]-z_j[t])^2\right)^2},\end{aligned} \tag{5}$$

where $d_j[t]$ corresponds to the distance between the UE and the eavesdropper in the tth time slot. The corresponding capacity of the wiretap channel is

$$C_j[t] = \log_2\left(1 + \frac{P[t]\emptyset[t]}{\sigma^2}\right) = \log_2\left(1 + \frac{\zeta_0 P[t]}{d_j^4[t]\sigma^2}\right). \tag{6}$$

C. Secrecy Capacity Metric

The security of the system is evaluated using the secrecy capacity metric which is commonly employed in the literature for analyzing eavesdropping security problems. The secrecy capacity is the rate at which the malicious node cannot decode any data when the legitimate channel capacity is higher than the wiretap channel capacity [13]. The secrecy capacity over T time slots is then obtained as

$$C_{sec} = \frac{1}{T}\sum_{t=1}^{T}\Big(C_u[t] - C_j[t]\Big)^+, \tag{7}$$

where $[\omega]^+ \triangleq max(\omega, 0)$.

D. Problem Formulation

The optimization problem is defined according to the following assumptions: First, for the sake of simplicity and without loss of generality, we assume that an ABS flies at a constant height that facilitates a LoS link between the ABS and the ground UE. In general, the lower the UAV height can be, the higher the resulting legitimate channel and secrecy capacity in the considered context. Second, we assume that the location of the passive eavesdropper is unknown. This scenario is of interest in practice where it is difficult to detect the presence and location of eavesdroppers because of their passive nature. Thus, we consider the capacity of the legitimate user in the above problem formulation. The eavesdropper location is used only for the calculation of the resulting secrecy rate to evaluate the performance of the proposed solution.

The objective of this paper is thus to maximize the legitimate capacity C_u by selecting the position for the ABS and controlling transmit power for the UE.

The optimization of the UE capacity will result in an improved secrecy capacity of the system. However, the UL capacity of the ground user relies on having a short distance to

the ABS with a strong LoS link. Therefore, the ABS position constraint is formulated as follows:

$$(x_u[t], y_u[t]) \leq (L_x, L_y), \forall t, \tag{8}$$

where (L_x, L_y) represent the maximum 2D coordinates of the UAV ground location projection. We introduce the peak transmission power P_{max} and the UL transmission power constraint for the legitimate UE per time slot as

$$0 \leq P[t] \leq P_{max}, \forall t. \tag{9}$$

The optimization problem of the UE capacity is then given as

$$(P1): \max_{x_u, y_u, P} \frac{1}{T} \sum_{t=1}^{T} \left[\log_2 \left(1 + \frac{\zeta_0 P[t]}{d_u^2[t]\sigma^2} \right) \right], \quad s.t.\ (8), (9). \tag{10}$$

where x_u and y_u are the UAV positioning parameters and P is the uplink transmission power controlled by the ABS.

III. Proposed Solution

The optimization problem (10) is challenging because it needs a joint UAV positioning and UE transmission power adjustment in the presence of an eavesdropper. Since the objective function is non-convex with respect to parameters x_u, y_u, and P, and the constraints, the problem becomes NP-hard [7] [11] [14]. Alternatively, the ABS position and the UE transmission power will be selected through a transition process based on the current system state. Since the next state of the system is independent from the previous state and action, the process can be modeled as a Markov decision process (MDP). This allows applying a RL algorithm to a UAV agent without requiring the knowledge of the system model. In this regard, instead of solving the problem using conventional optimization algorithms, we apply a RL method that can solve the problem in an efficient and accurate way, and thereby improve the secrecy rate in the network.

In what follows, we first describe the MDP model by defining the settings that include states, actions, and the reward for the UAV agent. Then, we introduce the Q-learning method based on the defined MDP settings to solve the problem. In order to avoid intractably high dimensionality for the high state-action space, we propose the DQN method in which a deep neural network (DNN) is employed to estimate the action value function for the UAV agent.

A. MDP Settings

The MDP for the UAV agent is composed of the state space $\mathcal{S}$, the action space $\mathcal{A}$, the reward space $\mathcal{R}$, and the transition probability space $\mathcal{T}$, i.e., $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T})$. At time slot t, the agent observes the state $s_t \in \mathcal{S}$, and based on its policy, it takes an action $a_t \in \mathcal{A}$. Depending on the distribution of the transition probability $\mathcal{T}(s_{t+1}|s_t, a_t)$, the agent will be transferred to the new state s_{t+1}. Since the transition probability is highly dependent on a specific environment and is difficult to obtain, we choose the Q-learning method as a model-free algorithm to directly find the best policy for each action in each state. This means that we do not need to know $\mathcal{T}$, but we need to carefully define states, actions, and the reward of the agent as follows.

State: The set of states is defined as

$$\mathcal{S} = \{s_1, s_2, ..., s_t, .., s_T\}, \tag{11}$$

where T is the total number of time slots. Each state s_t at a time slot t has three elements that are defined as

$$s_t = \{\Delta x, \Delta y, \Delta z\}, \tag{12}$$

where $\Delta x, \Delta y$, and Δz represent the distance difference between the UAV and UE along the x, y, and z axes, respectively. It is worth noting that the value of each state affects the channel gain and, hence, the SNR.

Action: The states are transited according to the defined actions. A set of actions is defined as

$$\mathcal{A} = \{a_1, a_2, ..., a_t, .., a_T\}, \tag{13}$$

where each action at time t consists of two parts related to the UAV movement and one part related to the transmit power adjustment. That is,

$$a_t = \{\delta_x, \delta_y, \delta_p\}, \tag{14}$$

where δ_x and δ_y represent the movement in the x and y directions and δ_p denotes the change in power. The altitude of the UAV is assumed to be constant.

As a sample configuration, the movement in x and y, i.e., δ_x and δ_y, can be assumed to change by $+1$ unit or -1 unit, and the power level, i.e., δ_p, can be assumed to change by p_1, 0, or $-p_1$, where p_1 is an arbitrary number. Hence, here we consider 4 possible directional movements and 3 power level changes, resulting in 12 possible actions for the ABS, which controls the UE transmission power, in any state.

Reward: After taking an action a_t in a state s_t at time slot t, the UAV agent will receive a reward $R_t(s_t, a_t)$. The UAV should get more rewards for the actions that may lead it higher secrecy rates. In this respect, we define the reward function of the system based on the instantaneous SNR between the UAV and the UE as

$$R_t(s_t, a_t) = \frac{\zeta_0 P[t]}{d_u^2[t]\sigma^2}. \tag{15}$$

B. Q-Learning Method

The UAV agent can apply the Q-learning method to find the best policy for the state-action relationship. Q-learning is a classical table-based RL algorithm in which the state-action pair has the value of $Q(s, a)$. The rows and columns of the Q-table consist of the environmental states and possible actions of the UAV agent, respectively. For example, for the aforementioned sample of states and actions, the table has 3 rows and 12 columns. The Q-values in the table are initially filled by random numbers. Then, the Bellman equation is used to obtain the optimal state-action pairs in the table [15]:

$$Q^*(s, a) = E_{s'} \left[R(s, a) + \gamma \times \max_{a \in \mathcal{A}} Q(s', a') \right], \tag{16}$$

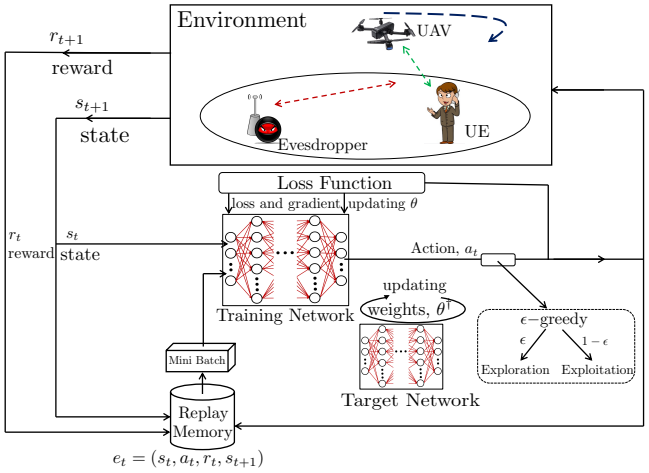

Fig. 2: Block diagram of the proposed DQN architecture.

where the s' and a' symbolize the next state and action. The parameter $\gamma \in (0, 1)$ denotes the discount factor that affects the importance of the future reward. The Bellman equation is used through an iterative process to update the Q-values, where a learning rate parameter α is applied to determine how quickly an agent leaves the previous Q-value for the new Q-value in the table. That is, $Q^{\text{new}}(s, a) = (1 - \alpha)\ Q_{i-1}(s, a) + \alpha\ Q_i(s, a)$, where subscript i indicates the i-th iteration in the process. Finally, in each step, the Q-value is updated while applying the $\epsilon-$greedy algorithm. This algorithm is used to balance between the exploration and the exploitation of the environment [15].

Although Q-learning provides a general framework for RL, it requires to store the Q-values for each state-action pair in the table. The number of state-action pairs in the table grows quickly with the number of states and actions. As the Q-table becomes large, the process becomes more time-consuming and impractical. Therefore, we consider the DQN method where a DNN is used to estimate the $Q(s, a)$ values, as opposed to Q-learning, where the Q-table is used to estimate the state-action values. This has advantages in terms of handling a large number of states and actions and the associated processing time.

C. Deep Q-Network Method

The DQN, initially proposed by Google Deep Mind [16], integrates the RL and deep learning methods. This technique uses the power of nonlinear functions, specifically DNNs, in order to approximate the Q-values and handle highly dimensional state-action problems.

Figure 2 shows the block diagram of the proposed DQN method. There are two DNNs of the same structure: a training network and a target network. The training network outputs the Q-values associated with the actions of the UAV in each state. The target network supervises the training network by providing the target Q-values obtained from the Bellman equation. The target values are compared with the outputs of the training network to minimize the loss function described below. Also, the target network prevents the learning model from suffering from the noise in the environment. The inputs to the networks are the states of the UAV agent in the environment, see (11)-(12). The outputs of the network are the Q-values corresponding to the actions of the UAV agent, i.e., $Q(s, a; \theta)$, where θ denotes the weights of the DNNs.

As the UAV takes an action, the system generates a record of experience. At time step t, the experience contains the current state s_t, the action a_t, the reward r_t, and the next state s_{t+1}, formed as a tuple $e_t = (s_t, a_t, r_t, s_{t+1})$. Each such experience is stored in a replay memory with the capacity of N, such that $\mathcal{M} = \{e_1, ..., e_t, ..., e_N\}$. The memory is a queue-like buffer that stores the latest N experience vectors. We use a mini-batch sample from the replay memory to feed the input of the training network as shown in Fig. 2. The main reason for using the mini-batch samples from the reply memory is to break possible correlations between sequential states of the environment, and thereby facilitate generalization.

In order to minimize the error prediction of the DNNs, a loss function is used that is defined as

$$L(\theta) = \mathbb{E}\Bigg[\Bigg(\Big[r_t + \gamma \times \max_{a \in \mathcal{A}}\ Q(s_{t+1}, a_{t+1}; \theta^{\dagger})\Big] - \Big[Q(s_t, a_t; \theta)\Big]\Bigg)^2\Bigg], \quad (17)$$

where the Q-value of the first term is obtained from the target network and the Q-value of the second term from the training network. Parameters $\theta^{\dagger}$ and θ denote the weights of the target network and training network, respectively. The $\theta^{\dagger}$ coefficients are updated every few time slots in order to ensure the stability of the target values and, hence, facilitate stable learning.

The UAV applies a gradient descent algorithm,

$$\nabla_{\theta}\, L(\theta) = -\mathbb{E}\Bigg[2\ \nabla_{\theta} Q(s_t, a_t; \theta)\Bigg(r_t + \gamma \times \max_{a \in \mathcal{A}}\ Q(s_{t+1}, a_{t+1}; \theta^{\dagger}) - Q(s_t, a_t; \theta)\Bigg)\Bigg], \quad (18)$$

to update θ an $\theta^{\dagger}$ as the weights of the DNNs with the aim of minimizing the prediction error.

Finally, we apply the $\epsilon-$greedy algorithm to select an action while balancing the exploration and the exploitation of the UAV in the environment (Fig. 2). In this algorithm, the UAV explores the environment with the probability of ϵ by choosing a random action. More precisely, the UAV exploits the environment with the probability of $1 - \epsilon$ by choosing the actions that maximize the Q-value function, i.e., $a^* = \text{argmax}_{a \in \mathcal{A}}\ Q(s, a; \theta)$. A high value of ϵ is initially set in the model for the UAV to spend more time for the exploration. As the agent obtains more knowledge about the environment, the ϵ value is gradually decreased to leverage the experience and choose the best actions for the UAV, rather than continuing with the exploration.

The details of the DQN-based algorithm used by the UAV agent for optimizing the UE SNR and calculating the secrecy

rates is presented in Algorithm 1. The brief description of the pseudocode is as follows: The parameters of the algorithm are initialized in lines 1 to 4. Line 5 starts the first loop for K episodes. The environment is reset in line 6 to initialize the starting state. The second loop begins at line 7, representing T time slots for the UAV to adjust its trajectory and power. Line 8 denotes the state in each time slot, and lines 9 to 13 apply the $\epsilon-$greedy algorithm to balance the exploration versus exploitation. The UAV takes an action in line 14. Then it receives the reward and goes to the new state, as denoted in line 15. The replay memory collects the new experience in line 16. Using the mini-batch in line 17, the training DNN is trained in line 18. The weights of the training DNN are updated in line 19 using the gradient descent algorithm on the loss function of (18). Line 20 updates the weights of the target DNN every B time slots. Once the algorithm runs out of time slots in each episode, it updates the value of ϵ (line 21). The rewards for each episode are stored for each episode according to line 22.

Algorithm 1 DQN for secrecy rate optimization.

1 **Initialize** ϵ_{start}, ϵ_{end}, decay
2 **Initialize** T time slots, K episodes
3 **Initialize** *replay memory* $\mathcal{M}$ *to capacity N*
4 **Initialize** θ, $\theta^{\dagger}$, γ, α, B
5 **for** *episode = 1, 2, ..., K* **do**
6 *Reset Environment*
7 **for** *t = 1, 2, ..., T* **do**
8 $s_t = (\Delta x_t, \Delta y_t, \Delta z_t)$
9 **if** $\epsilon > random(0, 1)$ **then**
10 *Select random* $a_t \in \mathcal{A}$
11 **else**
12 $a_t = \text{argmax}_{a \in \mathcal{A}} \ Q(s_t, a; \theta)$
13 **end**
14 *UAV takes an action,* a_t
15 *Obtain* $\mathbf{s}_{t+1}$, r_t
16 **Replay Memory:** $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathbf{s}_t, a_t, r_t, s_{t+1}\}$
17 **Minibatch from** $\mathcal{M}$**:** $\mathbf{e}_i = (s_i, a_i, r_i, s_{i+1})$
18 **Train the DNN training network**
19 **Update** θ **in training network via eq. (18)**
20 **Update** $\theta^{\dagger}$ **in target network every B steps**
21 $\epsilon \leftarrow update\epsilon$
22 *Store reward for each episode*
end
Result: Optimal Secrecy Rate in eq. (7)

IV. Numerical Analysis and Discussion

We numerically analyze the performance of the proposed DQN-based UAV positioning and power control scheme in optimizing the UL SNR of the ground user and its effect on the secrecy capacity in the presence of an eavesdropping attack.

The simulation scenario consists of a single antenna ground UE, an ABS mounted on the UAV, and a single antenna malicious node that is performing a passive eavesdropping attack on the UL transmission (Fig. 1). The UE and the eavesdropper are randomly distributed in a 2D area that has a L_x length and a L_y width. The ABS is launched at a random location with a fixed altitude and is equipped with an omnidirectional antenna to enable communications with UEs.

Table I provides the simulation parameters and the hyper-parameters for the proposed DQN solution. The training and target DNN networks consist of 4 layers where each DNN contains two fully connected hidden layers, one with 24 and the other with 32 neurons. Each DNN has 3 neurons at the input layer and 12 neurons at the output layer, corresponding to the number of states and possible actions defined in (12) and (14), respectively. The simulator is implemented in Python, using PyTorch to train the DQN.

TABLE I: Simulation Parameters

Parameter	Value	Parameter	Value
(L_x, L_y)	$(10\,m, 10\,m)$	$Discount\ factor, \gamma$	0.9
$\#\ of\ episodes,\ K$	9×10^4	$Replay\ memory,\ N$	500
$\#\ of\ time\ slots,\ T$	200	$Batch\ size$	50
$Learning\ rate, \alpha$	10^{-6}	$Update\ of\ target\ network, B$	10

For the performance evaluation of the proposed DQN algorithm, we compare the resulting secrecy capacity of the DQN against the values resulting from employing Q-learning, a greedy policy, and a random state selection scheme. The random scheme selects the next position and power action at each time slot randomly, whereas the greedy policy assigns the next action based on the highest Q-value.

Figure 3(a) shows our accumulative reward function over the total number of episodes that is defined as the legitimate UL SNR of the ground UE for the proposed DQN and for the other techniques. The illustration of rewards informs that the proposed solution has the fastest convergence rate to the highest SNR. The SNR performance of the Q-learning is similar, but takes longer to converge because of the nature of the problem that has a large number of states and actions.

Since the optimization problem in (10) relies on the location of the ABS and the transmission power, it is critical to study the convergence of these two parameters. Figures 3(b) and 3(c) illustrate the convergence of the UAV position and the transmission power over the number of episodes. Comparing the converged position of the ABS and the transmission power for the DQN and Q-learning, we observe that both techniques reach the same ABS position; however, there is a slight increase of the converged transmission power of the DQN over the converged Q-learning transmission power. This difference is the main explanation behind the difference in the optimized legitimate SNR levels presented in Fig. 3(a). The low SNR of the random approach results from low the transmission power value of Fig. 3(c). Note that the ABS position of the random scheme converges to the same value as the greedy algorithm.

Figure 4 shows the secrecy capacity of the DQN compared to the other techniques over the number of episodes. The secrecy capacity is calculated using the optimized ABS position and transmission power level after finalizing the learning. The proposed DQN algorithm improves the secrecy capacity by as much as 40%, 10%, and 5% when compared to the random, greedy, and the Q-learning solutions, respectively. The DQN reaches a relatively stable secrecy capacity value already after 2×10^4 episodes. On the other hand, the convergence of Q-learning occurs after 5×10^4 episodes. This proves the

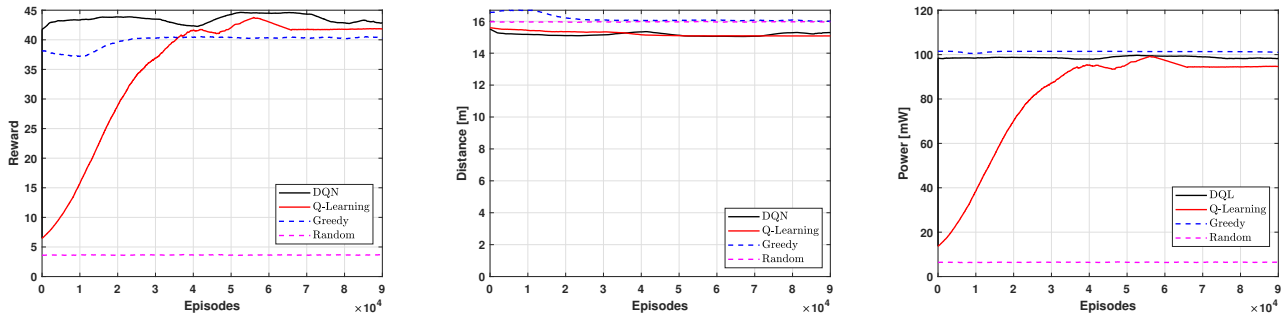

Fig. 3: The reward (a), distance (b), and power (c) convergence of the DQN, Q-learning and two baseline techniques.

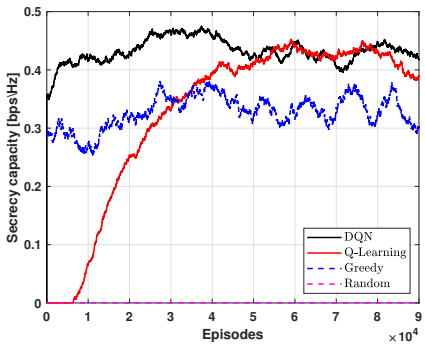

Fig. 4: Secrecy capacity achieved by the DQN, Q-learning and baseline techniques.

superiority of the DQN regarding the speed of convergence.

V. Conclusions

This paper has presented a novel positioning and UL transmission power control approach for ABSs that serve a ground user. We consider the legitimate user capacity as the metric to optimize in eavesdropping scenarios without knowledge of the eavesdropper location. This contributes to improving the secrecy capacity of the ground user under attack. We have provided detailed information about the designed DQN and the Q-learning algorithms. The obtained results show that the highest capacity and secrecy capacity are achieved with the proposed DQN when compared with Q-learning and two baseline techniques.

In future research, we will extend the simulation environment to cover multiple legitimate and malicious nodes, analyze the performance of using a multi-antenna systems, and consider both the uplink and the downlink. Moreover, the presented technique can be implemented into a testbed, such as AERPAW [17], which provides robust drones with modular software radio hardware and software, collocated computers, and an experimental license for RF radiation, enabling A2G wireless experiments. An AERPAW experiment can deploy ground users and UAVs, where the UAV can implement an ABS that uses the proposed method to position itself for serving a legitimate user and control the transmission power in such a way to maximize the user rate in the presence of an eavesdropper.

Acknowledgement

The work of A. S. Abdalla and V. Marojevic was supported in part by the NSF awards CNS-1939334 and CNS-2120442.

References

[1] A. S. Abdalla and V. Marojevic, "Communications standards for unmanned aircraft systems: The 3GPP perspective and research drivers," *IEEE Commun. Standards Mag.*, vol. 5, no. 1, pp. 70–77, 2021.

[2] A. S. Abdalla, K. Powell, V. Marojevic, and G. Geraci, "UAV-assisted attack prevention, detection, and recovery of 5G networks," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 40–47, 2020.

[3] R. Amer, W. Saad, and N. Marchetti, "Mobility in the sky: Performance and mobility analysis for cellular-connected UAVs," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3229–3246, 2020.

[4] B. Shang, V. Marojevic, Y. Yi, A. S. Abdalla, and L. Liu, "Spectrum sharing for UAV communications: Spatial spectrum sensing and open issues," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 104–112, 2020.

[5] A. S. Abdalla *et al.*, "Performance evaluation of aerial relaying systems for improving secrecy in cellular networks," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1–5.

[6] A. S. Abdalla, B. Shang, V. Marojevic, and L. Liu, "Securing mobile IoT with unmanned aerial systems," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–6.

[7] A. S. Abdalla and V. Marojevic, "Securing mobile multiuser transmissions with UAVs in the presence of multiple eavesdroppers," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 11 011–11 016, 2021.

[8] D. Wang, B. Bai, G. Zhang, and Z. Han, "Optimal placement of low-altitude aerial base station for securing communications," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 869–872, 2019.

[9] H. Kang *et al.*, "Secrecy-aware altitude optimization for quasi-static UAV base station without eavesdropper location information," *IEEE Communications Letters*, vol. 23, no. 5, pp. 851–854, 2019.

[10] M. T. Mamaghani and Y. Hong, "Intelligent trajectory design for secure full- duplex MIMO-UAV relaying against active eavesdroppers: A model-free reinforcement learning approach," *IEEE Access*, vol. 9, pp. 4447–4465, 2021.

[11] Y. Zhang *et al.*, "UAV-Enabled Secure Communications by Multi-Agent Deep Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11 599–11 611, 2020.

[12] X. Lin *et al.*, "The sky is not the limit: LTE for unmanned aerial vehicles," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 204–210, 2018.

[13] 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, 2015.

[14] J. Li *et al.*, "Deep reinforcement learning based computation offloading and resource allocation for mec," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[15] K. Arulkumaran *et al.*, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[16] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[17] V. Marojevic, I. Guvenc, R. Dutta, and M. Sichitiu, "Aerial experimentation and research platform for mobile communications and computing," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.