AFRL: Adaptive Federated Reinforcement Learning for Intelligent Jamming Defense in FANET

Nishat I Mowla, Nguyen H. Tran, Inshil Doh, and Kijoon Chae

Abstract: **The flying ad-hoc network (FANET) is a decentralized communication network for the unmanned aerial vehicles (UAVs). Because of the wireless nature and the unique network properties, FANET remains vulnerable to jamming attack with additional challenges. First, a decision from a centralized knowledge base is unsuitable because of the communication and power constraints in FANET. Second, the high mobility and the low density of the UAVs in FANET require constant adaptation to newly explored spatial environments containing unbalanced data; rendering a distributed jamming detection mechanism inadequate. Third, taking model-based jamming defense actions in a newly explored environment, without a precise estimation of the transitional probabilities, is challenging. Therefore, we propose an adaptive federated reinforcement learning-based jamming attack defense strategy. We developed a model-free Q-learning mechanism with an adaptive exploration-exploitation epsilon-greedy policy, directed by an on-device federated jamming detection mechanism. The simulation results revealed that the proposed adaptive federated reinforcement learning-based defense strategy outperformed the baseline methods by significantly reducing the number of en route jammer location hop counts. The results also showed that the average accuracy of the federated jamming detection mechanism, leveraged in the defense strategy, was 39.9% higher than that of the distributed mechanism verified with the standard CRAWDAD jamming attack dataset and the ns-3 simulated FANET jamming attack dataset.**

Index Terms: **Federated learning, flying ad-hoc network, jamming attack, on-device AI, reinforcement learning.**

I. INTRODUCTION

UNMANNED aerial vehicles (UAVs) are becoming increasingly popular as various challenging tasks can be accomplished by them in the three-dimensional space [1]–[4]. Because of the high degree of mobility of UAVs, there are many challenging applications such as border surveillance [1], relaying network [2], and disaster monitoring [3], where UAVs can be deployed for achieving better efficacy. A flying ad-hoc network (FANET) is a decentralized communication network formed by UAVs to mitigate the challenges faced by a fully infrastructure based UAV network [5]. Typically, in a FANET, the UAV nodes communicate among themselves over a shared wireless medium and transfer data to the base-station independently when they are in the communication range with a base station infrastructure [4]. The base station can alternatively be a multi-access edge computing (MEC) server. Because of the shared nature of the wireless medium used for communication in FANET, UAV nodes remain particularly vulnerable to wireless jamming attack, as shown in Fig. 1.

Essentially a jamming attack prevents devices from communicating by disrupting the reception of communications at the receiver using as little transmission power as possible [1], [3]. A jamming attacker model may follow a constant radio signal transmission (i.e., a constant jammer model), alternate between sleeping and jamming (i.e., a random jammer model), or transmit a radio signal as soon as it senses activity on the channel (i.e., reactive jammer model) [6]. To counter such jamming attacks, competition strategies, spectral retreat and spatial retreat based defense mechanisms are mainly considered [1], [4], [6]. In the competition strategy, the communicating nodes compete against the jammer by adaptively perceiving the threat level. The communicating nodes increase the transmission power used by the legitimate radio devices and operate at a higher signal-to-interference-plus-noise ratio (SINR). In contrast, the spectral retreat-based frequency hopping is performed by the on-demand change in the frequency to retreat from the channel in which the jammer is operating. The other option is the spatial retreat-based defense strategy which operates by evacuating from the jammer location by moving in random directions upon detecting a jammer in its range. In [7], a two-dimensional anti-jamming mobile communication scheme is proposed to enable a mobile device to retreat from a jammed frequency or area. In [8], UAVs are leveraged to relay the message of an on-board unit (OBU) to improve the network communication using reinforcement learning.

In general, the optimal defense strategy against jamming attacks in wireless networks faces various challenges. For example, any centralized defense strategy may incur significant communication cost and induce latency in the network. In fact, the traditional defense mechanisms may not scale to bigger networks and result in asynchrony issues in ad-hoc networks [6]. FANET poses some distinct properties which make selecting the optimal jamming defense strategy additionally challenging. For instance, unlike the mobile ad-hoc network (MANET) [1], vehicular ad-hoc network (VANET) [4], and some swarm UAV networks [22], [49], generally FANET has a considerably lower density of nodes [1], [2]. In particular, nodes in FANET operate in a three-dimensional space while the nodes in MANET and VANET operating in a two-dimensional space [1], [4]. Be-

Manuscript received December 20, 2019; revised March 21, 2020; approved for publication by Linglong Dai, Guest Editor, May 29, 2020.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIP) under Grant 2019R1F1A1063194. Kijoon Chae is the corresponding author.

N. Mowla and K. Chae are with the Department of Computer Science and Engineering, Ewha Womans University, email: nishat.i.mowla@gmail.com, kjchae@ewha.ac.kr.

I. Doh is with the Department of Cyber Security, Ewha Womans University, email: isdoh1@ewha.ac.kr.

N. H. Tran is with the School of Computer Science, University of Sydney, email: nguyen.tran@sydney.edu.au

K. Chae is the corresponding author.

Digital Object Identifier: 10.1109/JCN.2020.000015

1229-2370/19/$10.00 © 2020 KICS

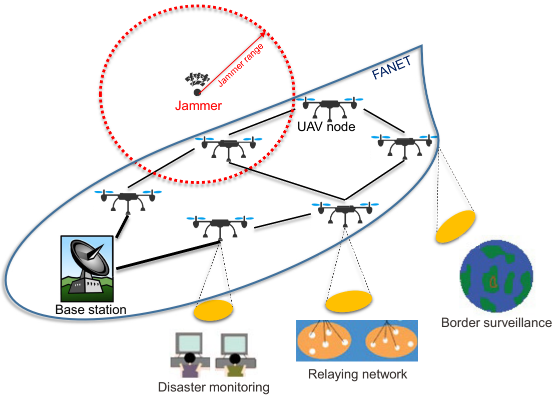


Fig. 1. Jamming Attack in FANET.

sides, the intuition of FANET is to leverage an ad-hoc network since the nodes can be far away from the conventional ground base stations. In other words, nodes in the flying ad-hoc network pose an additional challenge of asynchronous communication with any centralized controller due to several factors such as low node density, remote deployment of highly mobile UAVs, and power consumption constraints etc. [5]. As a consequence, it is not always feasible for the UAV nodes to communicate with any centralized controller. This property of the nodes in FANET, thus, makes it more vulnerable to various attacks [1], [2], [4]. Apart from that, the UAVs in FANET can provide higher coverage [32] than traditional ad-hoc networks (i.e., MANET, VANET) because of the enhanced line-of-sight (LoS) air-to-air and air-to-ground communication. Therefore, the computational and communication service coverages of the UAVs in FANET is significantly higher than that of the nodes in MANET and VANET [1], [2], [32].

In recent years, various defense mechanisms have been proposed for UAV-based networks [8]–[10]. In [8], a reinforcement learning-based approach was proposed against jamming attack by leveraging the UAV relay in VANET. In [9], a rule-based jamming attack detection mechanism was proposed for UAVs. A Bayesian game-theoretic approach was proposed for intrusion detection and ejection in UAV-aided networks [10]. However, such approaches are not suitable for jamming detection and defense in FANET as it faces the following three major challenges:

- First, the FANET nodes have power consumption constraints and because of the low density of the nodes [4], communication between the UAV nodes and the base station is also constrained. Therefore, jamming detection and defense support from any centralized knowledge base are challenging.
- Second, UAV nodes have very high mobility and rapid topology change [1], [2] which makes the FANET architecture very dynamic in nature. Hence, these nodes require constant adaptation to unbalanced sensory information collected from the newly explored spatial environments. Therefore, a traditionally distributed mechanism for jamming detection and defense may not be sufficient to address the unbalanced nature of the sensory environments.
- Third, the jamming defense strategy needs to take certain actions in the newly explored environments for which precise environmental data are initially unavailable. Therefore, any model-based jamming defense strategy will not be suitable given that the transition probabilities and the rewards are initially unknown. In this case, a model-free mechanism is essential to learn from the newly explored environments, and adapt to execute jamming preventive strategies according to the network dynamics.

A new technique in the artificial intelligence (AI) community for robust learning in a communications network is the federated learning algorithm [12], [13], which allows the efficient learning of unbalanced data besides providing communication efficiency [13], [16]. Moreover, federated learning [12] is specially designed for device-level training for the mobile devices (e.g., smartphones, UAVs, smartglasses, and smartwatches) [12]–[15]. In the case of the centralized control systems, the jamming attack detection operation is executed solely at the standalone centralized controller [12]–[14]. With the help of federated learning, a neural network model is trained locally (i.e., on-device) in any device with the help of a global model's weight updates when in range. In fact, the global weight update helps the local training models, but the final model is created locally. Thus, local decisions can be executed in the devices without the constant support of a global model as it is essential for the communication constrained UAVs in FANET. This becomes useful to take decision in a scenario where a global model is unreachable at times.

Moreover, federated learning allows low-level weight updates from the local devices to be sent and received from the global model [12], [16]. Therefore, the centralized controller is only responsible for collecting the weight updates from the local devices and performing federated averaging [12]. This property of the federated learning can help in extracting the fine-grained properties of the jamming data instances to significantly reduce the effect of the imbalance in the data faced by the UAVs in FANET. Therefore, the local devices independently perform jamming detection, and thus, making the control process decentralized [12], [13].

Finally, for any centralized control system as in the case of centralized learning-based detection [14], huge amounts of sensory data from the local devices need to be sent to the centralized controller which overlooks the critical data privacy issues of the local devices [13]. In contrast, in case of our proposal based on federated learning, only weight updates are sent to the global model and thus, effectively preserve the privacy of the local sensory data [12], [13].

This federated learning-based jamming detection can aid in developing a model-free jamming defense strategy by setting positive and negative rewards for a UAV in FANET as it explores an unbalanced data environment. In this regard, a jamming defense strategy can be tailored with a reinforcement learning approach as it is model-free [34], [37]. In particular, the reinforcement learning-based jamming defense approach can be adjusted according to the federated learning-based jamming detection result. Therefore, in this paper, we propose an adaptive federated reinforcement learning-based jamming attack detection and defense mechanism for FANET. In essence, the main contributions of this paper are as follows:

- First, we identify the key issues of applying jamming attack detection and defense strategy in a FANET architecture. In particular, we identify the key challenges of using the centralized and traditional distributed mechanisms for jamming attack detection in FANET, because of the leading causes of the communication constraints and unbalanced properties of the sensory data respectively. Furthermore, we identify the challenges of introducing a jamming defense mechanism in a newly explored environment with no initial data encountered by the UAV nodes required to adapt effectively.

- Second, we propose a security architecture of FANET leveraging a combination of federated learning and reinforcement learning. The proposed federated learning mechanism enables on-device jamming attack detection to support a reinforcement learning-based defense strategy. Subsequently, the proposed defense strategy takes the input from the federated learning model to update a Q-table using the Bellman equation [35]. Moreover, the optimal defense paths of the UAVs are selected using an adaptive epsilon-greedy policy over the combined federated learning and reinforcement learning model. As a result, the UAVs perform efficient spatial retreat from the jamming areas, that are detected beforehand by the federated learning approach.
- Finally, we simulate the proposed architecture and apply our method to the ns-3 [47] simulated FANET dataset and the standard CRAWDAD jamming attack dataset [39]. We verify our mechanism with two different cases of a jamming attack network environment. Then, we simulate a set of jamming and non-jamming communication cells in which spatial retreat-based decisions can be made by on-device agents. The simulation results showed that the detection model achieved a higher performance gain (82.01%) in terms of the average accuracy than the distributed model (49.11%). Moreover, the jamming defense strategy supported by the detection model showed that the number of hop counts of the jammer locations covered could be significantly reduced by applying the adaptive federated reinforcement learning-based defense strategy while selecting optimal paths to the destination. We also show that an appropriate epsilon value can be selected on the basis of the federated learning accuracy gain to achieve higher reward values with fast convergence.

The rest of this paper is organized as follows. Section II overviews the related works of distributed machine learning and jamming attack detection and defense mechanisms. Section III introduces the system model of the proposed adaptive federated reinforcement learning-based jamming detection and defense strategy. Section IV presents the performance of our system model. Finally, Section V concludes the paper with some remarks and possible future directions.

II. RELATED WORK

In this section, we discuss the related works of distributed machine learning mechanisms. Then, we review the jamming attack defense mechanisms based on competition strategy, spectral retreat, and spatial retreat.

A. *Distributed Machine Learning*

Distributed machine learning is the new trend to allow a flexible learning paradigm in a communication network. In the literature, various cluster and data center-based distributed learning algorithms have been proposed [12]. In [19], a distributed training mechanism of locally trained models with an iterative averaging technique was proposed. However, most of these mechanisms fail to address the unbalanced and non-IID properties of the data. In [20] and [21], distributed learning mechanisms with a focus on communication efficiency were developed. In [24], the asynchronous distributed forms of the stochastic gradient descent (SGD) algorithm are discussed. Recently federated learning techniques were proposed to address the efficient learning of the unbalanced and non-IID properties of the data [12]. Essentially, federated learning enables on-device learning to reduce both the data privacy issues and the communication costs [13], [16].

In [17], a federated reinforcement learning approach was proposed, considering the privacy requirements of the data and the models by building a Q-network for each agent with the help of other agents, without directly transferring the data or the knowledge from one agent to another agent. In [18], a lifelong federated reinforcement learning architecture was proposed as a knowledge fusion algorithm with evolutionary transfer learning in order to improve a shared model deployed in the cloud for cloud robots' navigation aid.

B. *Competition Strategy*

The idea of the competition strategy is to compete against the jammer by adjusting of the digital networks coding and transmission power of communication in the lower layers. Stronger error-correcting code can be used to increase the likelihood of the packets successfully being decoded [6], [23]. However, this mechanism can reduce the overall throughput with a relatively low information rate. Another technique is to use increased transmission power levels (i.e., higher transmission power than the jammer), which also means expanding the radio coverage patterns for radio devices leading to interference with other devices [6]. Moreover, competition strategies become resource-intensive and may not be suitable for resource-constrained network architectures such as FANET.

C. *Spectral Retreat*

Spectral retreat or frequency hopping is the mechanism of an on-demand change in frequency [6], [22] when a jammer is detected using the same frequency for disrupting the communication. However, the mechanism becomes particularly challenging with ad-hoc networks as reliably coordinating multiple devices switching to a new network channel faces the usual challenges of distributed computing such as asynchrony, latency, and scalability issues [6], [8]. A possible alternative is the coordinated spectral retreat. Upon the detection of the absence of neighbors on the original channel, a device node probes the next channel to check whether the neighboring nodes are still nearby. If beacons are detected, the device node can inform the other device nodes by switching back channels. However, the overall coordination process can be challenging when the network scales [6].

D. Spatial Retreat

Spatial retreat-based jamming defense strategy enables retreating the device nodes from the jammer locations [6] to restore communication. Spatial retreat can also be evaded by using directional antennas [25], [26]. With directional antennas, data can be transmitted and received only in one particular direction through beamforming where the beamformer (i.e., transmitter) adjusts the phase and amplitude of the transmitter signal for its successful communication with the receiver. In particular, sectored antennas, placed at an angle forming a geometric sector shaped radiation pattern, are proved to help improve the connectivity of wireless networks [25]. Since beamforming is performed by steering an array of antennas to transmit radio signals in a specific direction, the adaptive beamforming techniques can be considered as a subset of the directional antenna-based mechanisms Therefore, the legitimate communicating nodes may bypass the adverse effect of the jammer node by the use of directional antennas and adaptive beamforming [27]–[29]. In [30], an adaptive beam nulling technique is proposed for jamming attack mitigation in multi-hop ad-hoc networks as a spatial filtering procedure. The mechanism performs periodic measurements of the radio frequency environment to detect the direction of arrival (DoA) of jamming signals and then suppresses the signals. In [31], distributed adaptive beam nulling was proposed to survive against jamming attack in the 3D UAV mesh networks by spatially filtering out signals coming from a certain direction. The potentials of optimal deployment and path planning have been discussed in [32] to address inter-cell interference and obstacle-awareness. However, to the best of our knowledge, they have not been considered as spatial retreat based jamming defense strategies for UAVs in FANET.

Spatial retreat mechanisms have been proposed in the literature by various game-theoretic analysis of an aerial jamming attack on a UAV communication network. A pursuit-evasion game between the jammers and the legitimate device nodes was proposed in [33] for optimal jamming defense strategy computation. In [9] and [10], an intrusion detection and ejection framework against lethal attacks in a UAV network was proposed using a Bayesian game-theoretic methodology. One problem with game-theoretic mechanisms is that the solutions are reactive.

Thus far, some proactive solutions have been proposed using a reinforcement learning-based methodology to evacuate from jammed regions [7], [8]. In [7], a hot-booting deep Q-network based 2-D mobile communication scheme is proposed by applying deep convolutional neural network and macro-action techniques to accelerate learning in dynamic situations by exploiting experiences in similar scenarios. In [8], UAVs are used to relay OBU messages to other road-side units (RSUs) with a better radio transmission condition if the serving RSU is heavily jammed.

While proactive solutions are promising for jamming attack defense strategy selection, most of these mechanisms are centralized, which makes them a limiting constraint for highly mobile networks such as FANET. In [11], we proposed a federated learning-based jamming detection mechanism for FANET along with a Dempter–Shafer theory-based client group prioritization technique. The goal was to federate the detection of jamming attacks over several UAV nodes in the FANET with the help of global weight updates computed from a prioritized client group, thus enabling a decentralized detection mechanism. In this paper, we extend [11] to enable an adaptive defense strategy during a jamming attack, on the basis of a collaboration between the federated learning and the reinforcement learning mechanisms.

III. Proposed Adaptive Federated Reinforcement Learning-Based Defense Strategy

The dynamics of the global model, local model, and the Q-learning model work in five detailed steps: *A.* Parameter estimation, *B.* UAV client execution and upload, *C.* MEC server model averaging and execution, *D.* UAV client download, and *E.* UAV agent Q-table update and execution as shown in Fig. 2. In addition, there are four major components that execute these steps namely, *1.* UAV network environment, *2.* sensory data, *3.* UAV client, and *4.* the MEC server as shown in Fig. 2. A detailed discussion of the major components and the steps is provided in the later sub-sections.

A. Parameter Estimation

In the proposed federated architecture for FANET, each UAV client conducts parameter estimation (as shown in Fig. 2), from the UAV network by collecting the sensory data. The parameter estimation is done by using the sensory data composed of the local networking features, such as the received signal strength indicator (RSSI), and packet delivery rate (PDR) in order to generate a set of feature vectors. Hence, this process generates a set of feature vectors x_i and its label y_i (i.e., jammer and non-jammer classes) forming a single data instance or data point. Given that D_k is the set of indexes of the data points with client index k and $p_k = |D_k|$, pairs of (x_i, y_i), $i = 1, \cdots, p_k$ are yielded for the local p_k data points as a fraction of the global p data points (lines 4–5 in Alg. 1). Thus, p_k consists of the dataset of client k whereas p consists of all the clients' dataset.

B. UAV Client Execution and Upload

In Fig. 2, the local UAV client k trains a local model by splitting the data p_k into B batches. For each local epoch, a subset b of batch B is trained via updating the local weight as follows,

$$w = w - \eta \Delta(w; b). \tag{1}$$

Here, $\Delta(w; b)$ is the change in the weight w for the mini-batch b multiplied by the learning rate η. $\eta\Delta(w; b)$ is subtracted from the w to compute the new weight (i.e., w in the left-hand side of the equation) for the local model. Accordingly, the learning rate η specifies how much of the change in the weights will be used to update the new weight. Thus, (1) signifies that the weight w is updated locally before being uploaded to an MEC server (lines 6–10 in Alg. 1).

C. MEC Server Model Averaging and Execution

A global model at the MEC server of the security architecture (as shown in Fig. 2) initializes the global weight. For each m round for client k, w^k_{m+1} is updated by the $ClientUAVExecution(k, w)$ (lines 24–30 in Alg. 1). Local

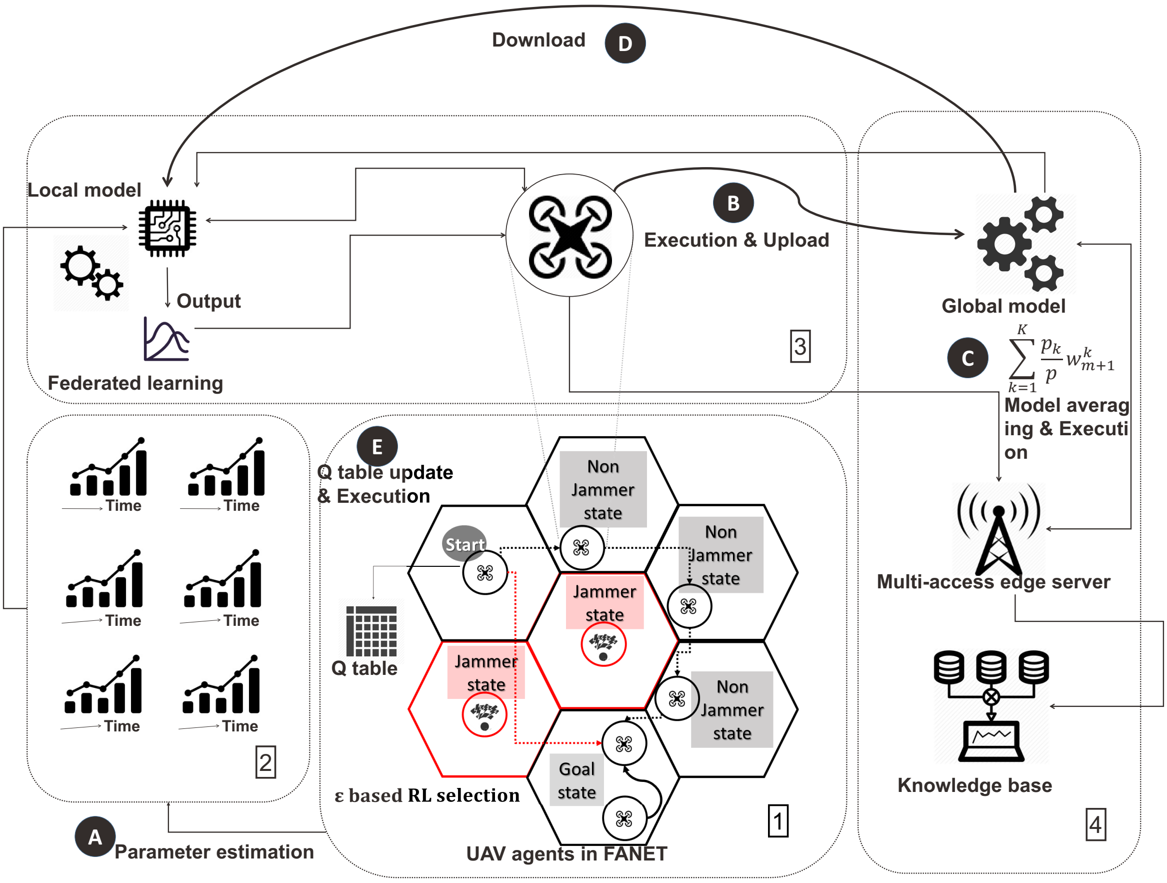


Fig. 2. Proposed adaptive federated reinforcement learning-based jamming defense strategy.

client updates as w_{m+1}^{k} is received at the MEC server and used to improve the global model leveraging the federated averaging algorithm by computing a weighted average of the local updates received from the local UAV clients as follows,

$$w_{m+1} = \sum_{k=1}^{K} \frac{p_k}{p} w_{m+1}^{k}. \tag{2}$$

In (2), for a total of K clients, w_{m+1} is the global weight after m rounds over all the p data points (line 28 in Alg. 1). The loss function of client k on the local dataset p_k is defined as follows,

$$L_k(w) = \frac{1}{p_k} \sum_{i \in p_k} f_i(w), \tag{3}$$

where,

$$f_i(w) = f(x_i, y_i; w). \tag{4}$$

$f_i(w)$ is a function for x_i which is the ith feature paired with the label y_i and the global weight w. Based on the above formulation, the global loss function minimization problem is derived as follows,

$$\min_{w \in R^d} L(w) = \sum_{k=1}^{K} \frac{p_k}{p} L_k(w). \tag{5}$$

D. UAV Client Download

The MEC server returns the global weight w (lines 29–30 in Alg. 1). This global weight w can now be downloaded by the UAV clients to be used in their local training (as shown in Fig. 2). This enables a globally verified update to aid the local training by incorporating global knowledge, while the individual local clients also learn iteratively. This helps in generating a model that is suited to their local surroundings as well. The model generated by a local UAV client then senses the environment as it moves into newly explored spatial environments to detect the presence of a jammer.

After the jamming detection, the UAV client needs to apply a jamming defense strategy to restore stable communication. As discussed before, there are three major defense strategies, namely, the competition strategy, spectral retreat, and spatial retreat [6]. Because of power consumption constraint, the applicability of competition strategy is challenging for the UAVs. Moreover, an increased power level means a larger radio coverage pattern for radio devices that simultaneously increases the likelihood of collisions and unintentional interference with the legitimate radio devices [1], [4]. In contrast, the spectral retreat based strategies are challenging with ad-hoc networks as the reliable coordination between multiple devices that switch to new channels faces asynchrony and scalability issues [6]. Moreover, the spectral retreat based defense strategies can incur significant latency due to the network scalability that leads to an unstable phase consisting of old and new channels [4], [6]. The third major strategy of spatial retreat is suitable for mobile networks and is particularly practical for highly mobile UAVs. Nevertheless, the spectral retreat strategy still faces partition issues [6] for nodes that always need to be connected. As the UAVs in FANET maintain minimal communication and are aided by the on-device jamming detection local learning model, the partition issue does not apply to the FANET scenario.

Hence, we propose a spatial retreat based jamming defense strategy that leverages the federated learning-based on-device jamming detection to retreat from the jammer locations. Moreover, the UAVs applying the jamming defense strategy need to take certain spatial retreat actions in the newly explored environments for which precise environmental data are initially unavailable. Hence, any model-based jamming defense strategy will not be suitable.

In this regard, we propose a jamming defense strategy based on a model-free reinforcement learning model [34], [37] particularly designed for the UAVs in FANET. As the transition probabilities and the rewards are initially unknown by the UAVs in FANET, we applied a Q-learning algorithm that learnt the newly explored environments and chooses spatial retreat-based routes adaptively. Meanwhile, the federated learning-based jamming detection approach enables adopting an epsilon-greedy policy of the reinforcement learning model for selecting the jamming defense strategies. Thus, the federated learning-based jamming detection model and the reinforcement learning-based jamming defense strategy maintain a mutualistic relation. In other words, the reinforcement learning-based model acts as a model-free defense strategy of the UAVs which adapts to the newly explored environments (lines 15–17 in Alg. 1). Moreover, for evolving the defense strategies with newly available information, the federated learning-based model acts as an as evolutionary detection model (line 18–22 in Alg. 1).

E. UAV Agent Q-table Update and Execution

Q-learning is an off-policy, model-free reinforcement learning algorithm that seeks the best action to take given the current state [36]–[38]. It enables the adaptation of the Q-value iteration algorithm in a situation where the transition probabilities and the rewards are initially unknown [35]. The main benefit of applying a Q-learning function (as shown in Fig. 2) is that it allows learning from actions that are outside the current policy (e.g., allowing random actions). Moreover, the precise estimation or the use of an exact environmental model is not needed, thus making it model-free [34], [37]. Thus, Q-learning learns a policy that maximizes the total reward. In the case of our adaptive federated reinforcement learning model as shown in Fig. 2, we used the Q-learning model to learn from the newly explored environment, based on the trial-and-error experience enhanced by the federated learning-based jamming detection accuracy for setting up the optimal policy. In this regard, a UAV client received a negative reward if it moved closer to a jammer location detected by the federated learning-based on-device jamming detection model. These rewards were then incorporated in the Q-table used by the UAV client where the actions were the physical relocation from one location (i.e., current state) to another location (i.e., next state) in the aerial space. Then, an epsilon-greedy policy was used to balance between the exploration and the exploitation opportunities of the reinforcement learning model on the basis of the achieved federated learning accuracy.

At each UAV agent, a Q-table was initialized to 0 values for each state-action pair $Q(S_t, A_t)$ in the local UAV agent, where $S_t = (s_t, s_{t+1}, \cdots)$ was a sequence of states from t to ∞ and $A_t = (a_t, a_{t+1}, \cdots)$ was a sequence of actions from t to ∞. Here, a state-action pair corresponded to one state at a specific location of the UAV in the aerial environment and a possible action to physically move from the current state (i.e., current location) to the next state (i.e., next location) until the UAV reaches the goal state (i.e., destination) from the source. Hence, the set of states S_t represented the position of the UAV nodes in the communication cells. Consequently, the set of actions A_t represented the movement from one communication cell to another cell (line 12 in Alg. 1).

For the UAV agent's decision making, we saved the average accuracy of the model by using the local weight w (line 13 in Alg. 1). The updated Q-table enabled taking a defense strategy by selecting a state-action pair that maximized the Q-value. The Bellman equation to update the Q-value could be derived as follows [35],

$$\begin{aligned} Q(S_t, A_t) \leftarrow & Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) \\ & - Q(S_t, A_t)] \end{aligned} \tag{6}$$

In (6), the learning rate, α is initialized to represent how quickly or slowly the model will learn. A discount factor, γ is set to indicate the immediate or delayed response. More specifically, a discount factor γ close to 0 indicates a delayed reward and a value close to 1 indicates an immediate reward. Meanwhile, $R_{t+1} \in \{1, -1\}$ indicates the reward at time $t + 1$ calculated from the federated learning-based detection model with local weight w, for a change in the discrete state-action pairs (lines 14–15 in Alg. 1). In this case, a negative reward (i.e., -1) indicates that a jammer is detected whereas a positive reward (i.e., 1) indicates a non-jamming environment is detected. Moreover, (6) returns the Q-value for an action a taken in the current state S_t that eventually maximizes the Q-value, $Q(S_{t+1}, a)$ of the future state at time $t + 1$. Therefore, the difference between this maximum Q-value, $Q(S_{t+1}, a)$ and the current Q-value, $Q(S_t, A_t)$ at $t + 1$, was taken to update $Q(S_t, A_t)$ (line 16 in Alg. 1). The Q-value was updated for n number of iterations as follows,

$$Q_n(S_t, A_t) \rightarrow Q^*(S_t, A_t), n \rightarrow \infty, \tag{7}$$

where, $Q^*(S_t, A_t)$ is the converged Q-value [36] after n number of iterations at time t (line 17 in Alg. 1). To choose an adaptive federated reinforcement learning model, the local detector's jamming detection accuracy was compared to a certain threshold value (δ) (line 18 in Alg. 1). If the accuracy of the local model was higher than δ, an epsilon-greedy policy value ϵ (initialized to 0.5), was reduced by a random measure ($rand(0, 1)$) derived by a random function generating a value between 0 and 1. Then, the derived model was used to select a feasible spatial retreat defense route to the destination. Otherwise, the ϵ value was increased by the random measure, and the derived model was used to select a feasible spatial retreat defense route to the destination (lines 19–22 in Alg. 1). The epsilon value (ϵ), ranged between 0 and 1, was used to balance between the exploration and exploitation opportunities of the reinforcement learning model. A value of ϵ closer to 1 indicated that more exploration would be

Algorithm 1: Adaptive federated reinforcement learning-based jamming defense trategy in FANET

```
ClientUAVExecution($k, w$):
Initialize $\epsilon$
Initialize detection threshold $\delta$
Pre-process $p_k$
Extract $p_k$ feature set $(x_i, y_i)$
$b \leftarrow$ split data $p_k$ into batches of size $B$
for each local epoch e from 1 to E do
    for $b \in \beta$ do
        $w \leftarrow w - \eta\Delta(w; b)$
return $w$ to server
UAVAgentExecution:
$Q(S_t, A_t) \leftarrow 0$
Accuracy = Detection with $w$
for each change in state and action pair do
    $R_{t+1} \leftarrow \{1, -1\}$
    $Q(S_t, A_t) \leftarrow$
      $Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$
    $Q^*(S_t, A_t) \leftarrow Q_n(S_t, A_t), n \rightarrow \infty$
    if $Accuracy > \delta$ then
        $\epsilon = 0.5 - rand(0, 1)$
        Select state-action pair based on model with $\epsilon$
    $\epsilon = 0.5 + rand(0, 1)$
    Select state-action pair based on model with $\epsilon$
MECExecution:
Initialize $w_0$
for each round m=1,2,$\cdots$ do
    for each client $k$ do
        $w^k_{m+1} \leftarrow$ ClientUAVExecution($k, w$)
        $w_{m+1} \leftarrow \sum_{k=1}^{K} \frac{p_k}{p} w^k_{m+1}$
        $w \leftarrow w_{m+1}$
return $w$
```

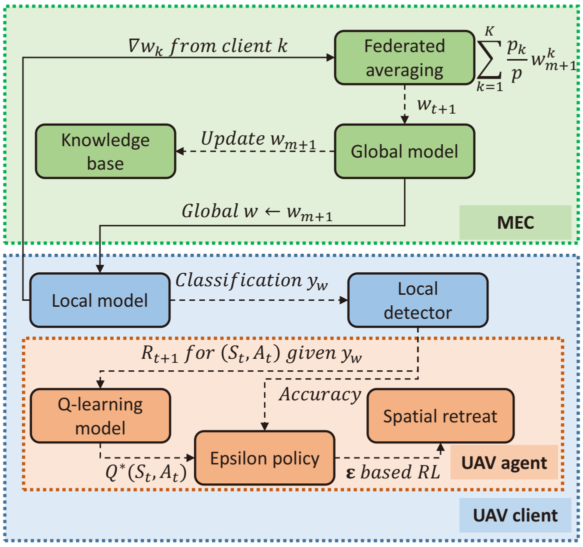


Fig. 3. Proposed workflow of the adaptive federated reinforcement learning.

allowed and the random next actions would be taken. In contrast, a value of ϵ closer to 0 indicated that more exploitation would be allowed and the next actions would be taken on the basis of the best Q-value. For the proposed federated reinforcement learning-based jamming defense, if the federated average accuracy was lower than a threshold δ, then we increased the epsilon value to allow more random next actions (i.e., more exploration). However, if the federated average accuracy was higher than that of the threshold δ, then we decreased the epsilon value as described above to bias more on the Q-table updates and take the next decision on the basis of the best Q-value (i.e., more exploitation). Thus, an adaptive exploration-exploitation based reinforcement learning model was selected.

The defense strategy for FANET relies on two major components, namely the MEC server and the UAV client. The UAV agent is the defense strategy module of the UAV client to execute certain spatial retreat actions. Therefore, the client is considered as the agent when it performs the reinforcement learning-based spatial retreat. Fig. 3 shows the detailed workflow of the proposed adaptive federated reinforcement learning mechanism. Essentially, a global model runs in the MEC server which is updated by the federated averaging algorithm. The consequent global weight updates, w_{m+1} after each round m, are stored in a knowledge base of the MEC server for record-keeping (as shown in Fig. 2). The global model sends the updated global weight w_{m+1} after m rounds, as the current global weight w to the local model, when the UAV client is within the communication range. A local model runs in the UAV client to support local jamming attack detection by an on-device local detector. The local model also periodically sends an updated local weight of ∇w^k to the global model. The local detection derived from the local model is used to set the rewards, R_{t+1} at time $t + 1$ for each change in state s_{t+1} with action a_{t+1}. The generated reward matrix R is then used to update the Q-table of the UAV agent. An updated Q-table trained by the Q-learning is then sent to the epsilon value-based policy maker. The epsilon value-based policy maker checks the detection accuracy received from the local detector based on which the epsilon value is adjusted as discussed before. Hence, a reinforcement learning model with the adjusted epsilon value is selected to perform spatial retreat by adaptive exploration and exploitation. Thus, the spatial retreat approach enables choosing alternative paths to the destination by spatially retreating from the jammed spaces. In the next section, we extensively evaluate the proposed mechanism and verify the benefits of the proposed federated reinforcement learning-based defense strategy.

IV. PERFORMANCE EVALUATION

The experiment was divided into two parts. The first part was the performance evaluation of the federated learning-based jamming detection model. The second part was the performance evaluation of the spatial retreat-based defense strategy leveraging Q-learning and the supporting federated learning-based jamming detection mechanism.

A. Experimental Settings

A 64-bit, Intel i7-47900 CPU @ 3.60-GHz processor and 8.00-GB RAM simulation environment was used to train the detection model. We simulated a FANET topology in the ns-3 [47] with three-dimensional (3-D) mobility model in the ad-hoc setting, communicating over the WiFi physical standard 802.11n [42]–[44]. We have considered all the three main levels of jamming, i.e., constant jammer, random jammer and reactive jammer [6], [25], [48] where one-third of the total jamming data instances belong to each of these three jamming levels, respectively. Besides, we have also collected and trained our model on data instances with different levels of jammer power ranging between -90.00 dBm to -100.97 dBm in our ns-3 based experiment. A jammer node was introduced with constant, random, and reactive radio frequency (RF) jamming signals that interfered in the communication between three UAV nodes and one server node over the 3-D UAV ad-hoc Gauss-Markov mobility model [45], [46]. We extracted 3000 instances with six features each consisting of signal to noise ratio (SNR), noise, received signal strength indicator (RSSI), throughput, data rate, and modulation and coding scheme (MCS) value. Apart from that, for further verification of our proposal, we validated the proposed mechanism on the standard public dataset of the CRAWDAD jamming attack in a wireless vehicular ad-hoc network [39]. The dataset provides the RSSI and PDR features of jammed and non-jammed wireless vehicular ad-hoc networks. We pre-processed the dataset to extract 3000 instances with 100 features each. The features consisted of 50 RSSI readings and 50 PDR readings taken over a period of time. The communication technology that we considered for the dataset is wireless and therefore, is also applicable for FANET. Additionally, the features used were RSSI and PDR, as in the case of FANET for interpreting jamming scenarios. The dataset contains traces of 802.11p packets, collected in a rural area located on the periphery of Aachen (Germany) in 2012, with the presence of a radio frequency jamming signal with constant, random, and reactive jamming patterns. We have used one-third of the total jamming instances from constant, random, and reactive jamming instances respectively. The jamming power of the dataset ranged from -10 dBm to -100 dBm. However, it is noteworthy to mention that the dataset was pre-processed to create a pathologically unbalanced dataset [12] by generating an unbalanced proportion of the classes to particularly address the unbalanced sensory environment of the UAVs in FANET.

We formulated a binary class problem [16] for both the datasets, where the two classes were labeled as jammer and non-jammer classes with 50% training and 50% testing dataset respectively. We pre-processed the dataset to have one set of 20% jammer and 80% non-jammer instances and another set of 80% jammer and 20% non-jammer instances. The accuracy from these two sets was then averaged to yield the average accuracy of the pathologically transformed unbalanced dataset.

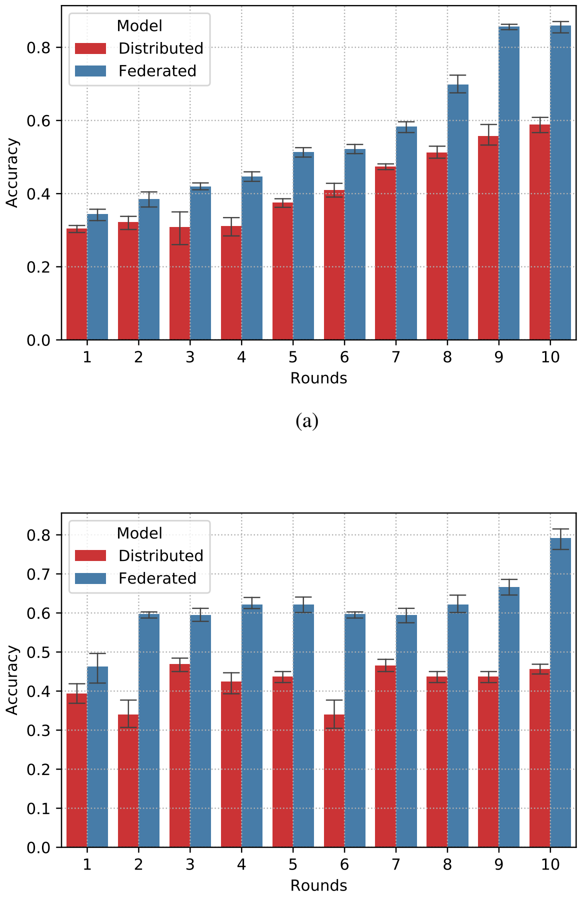
Moreover, a local Q-learning model is initialized with a Q-table with 0 values. For the Q-learning model, a learning rate (α) and discount factor (γ) of 0.9 and 0.8 were initialized respectively. For the evaluation of the federated reinforcement learning model, the epsilon value was initially set to 0 to indicate a full exploitation of the Q-table values for the defense strategy selection. For the evaluation of the adaptive federated reinforcement learning model, the threshold (δ) of the federated learning accuracy was set to 0.5 tested with varying epsilon value accordingly.

B. Simulation Result of Federated Learning Model for Jamming Attack Detection

For the federated learning-based jamming attack detection, a three-layered neural network model was generated where the first layer was the flattened layer that converted the input features to a vector. Next, there were two fully connected dense layers with the rectified linear unit (ReLU) and softmax functions respectively. Adam optimizer was used with a learning rate of 0.001. The model and the dataset were distributed and trained over six local client instances and one global instance for the federated learning of the jamming attack detection.

The global model was updated by the federated averaging technique to derive a weighted average of the received local weights after several rounds of communication. Hence, we conducted a comparison analysis between the federated learning model and the base-line traditional distributed learning model [13] over 10 rounds of communication. The main difference between the base-line distributed learning model and the federated learning model exists in the assumptions made regarding the local dataset's properties [12], [13]. In particular, the goal of the distributed learning model is to parallelize computing power while the goal of the federated learning model initially aims at training on heterogeneous datasets. Both the distributed learning model and the federated learning model train a model on multiple servers. Nevertheless, a general assumption in the distributed learning model is that the local datasets are identically distributed, and the classes are roughly the same size [12], [13]. In contrast, in the federated learning model the datasets can be heterogeneous, and the classes are allowed to be unbalanced. Therefore, in the case of distributed learning model, a central server usually averages the updates of the average gradients collected from the local clients which could then alternatively download an updated model from the central server. However, in the federated learning model, weight updates are sent to the MEC server other than more generalized average gradients to allow better local learning from a much lower level of the neural networks [13]. Fig. 4 shows the comparison between the distributed learning model and the proposed federated learning model for the jamming detection over 10 rounds of communication after 25 epochs each averaged over 10 samples. Here, Fig. 4(a) shows the performance of the ns-3 [47]-simulated FANET dataset and Fig. 4(b) shows the performance of the CRAWDAD ad-hoc network dataset.

In the ns-3-simulated FANET dataset, as shown in Fig. 4a, the distributed learning model and the federated learning model exhibited an average accuracy of 30.40% and 34.35% respectively after round 1. However, after round 10, the distributed learning model's average accuracy increased to 58.91%, whereas the federated learning model's average accuracy increased to 85.98%. In the case of the CRAWDAD ad-hoc network dataset, as shown in Fig. 4(b), the distributed learning model and the federated learning model exhibited an average accuracy of 39.38% and 46.28% respectively after round 1. However, after round 10,



(a)

(b)

Fig. 4. Comparison between the distributed model and the proposed federated model for jamming detection in terms of average accuracy: (a) Ns–3 simulated FANET unbalanced dataset and (b) CRAWDAD ad-hoc network unbalanced dataset.

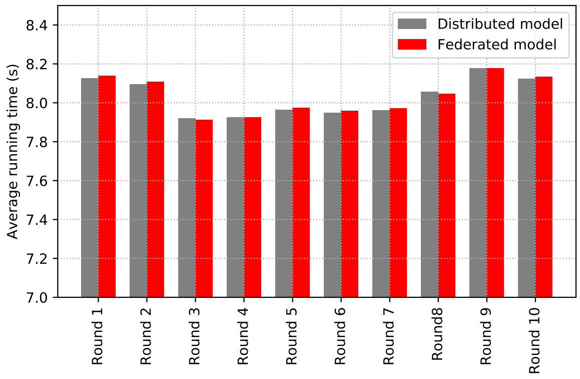
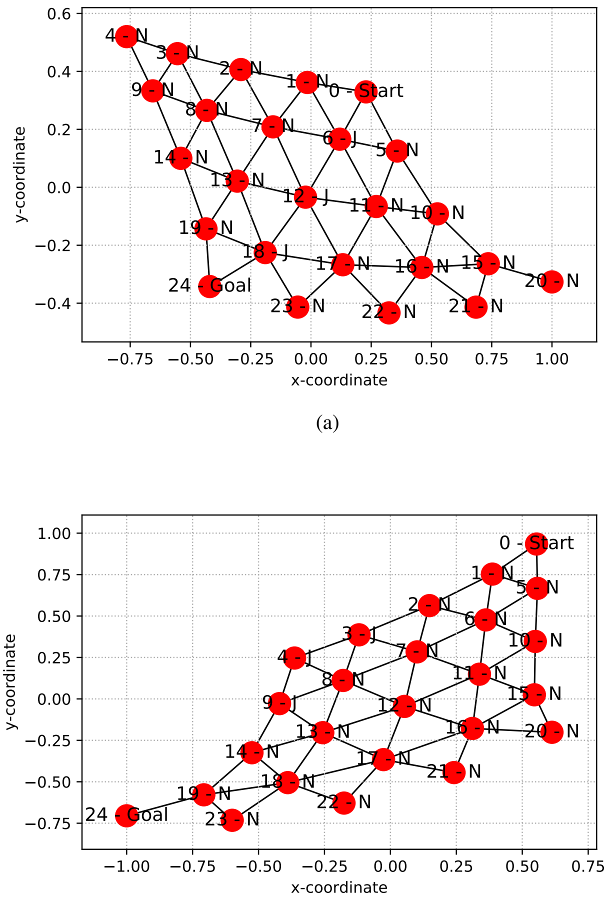


Fig. 5. Comparison of the average local running time between the distributed learning model and federated learning model.

the distributed learning model (45.63%) was significantly outperformed by the federated learning model (79.16%). As can be seen, over the several rounds of communication, the performance of the distributed learning model remained lower than that of the proposed federated learning model. This was because the distributed model applied a generalized global average gradient that could not adjust to the unbalanced data provided in the local client. In contrast, the proposed federated learning model enabled the recognition of each instance on the basis of the fine-grained individual weight updates other than a more generalized average gradient. This ensured that the underlying weights performed better as they could learn from a considerably lower level of the neural network model. As a result, the overall performance gain became significant with an increase in the number of communication rounds. Moreover, in the publicly available dataset, as shown in Fig. 4(b), the performance of the distributed learning model remained quite unstable as it encountered real-world features in an unbalanced data environment. Fig. 5 shows the comparison of the average running time of the local distributed learning model and the federated learning model for the local jamming attack detection over the 10 rounds of communication. The average running time for the distributed learning model and the federated learning model are 8.02 s and 8.03 s over the 10 rounds of communication respectively. While the jamming attack detection performance of the proposed federated learning model (with average accuracy of 82.01%) is much better than that of the distributed learning model (with average accuracy of 49.11%), the average running time for both the learning techniques are almost similar. This clearly shows that the federated learning-based model can achieve much better performance while incurring no additional running time. Note that, the federated learning model is specially designed for communication networks that have a diverse range of communication constraints. In other words, one of the main advantages of the federated learning model for FANET is that minimal wireless communication can be maintained between the MEC server and the other communicating nodes. Since continuous connectivity is not required for the federated learning, there are also no synchronization issues that reduce the overall complexity and control signaling between the MEC server and the UAV nodes in FANET. Moreover, since the learning is federated between the UAV nodes and the MEC server, the system can be considered not to be fully centralized. Besides, it is to be noted that the MEC server only receives the weight updates from the peripheral nodes other than raw data which reduces the burden on the wireless communication channel bandwidth and complex modulation by another factor.

C. Simulation Result of Spatial Retreat-based Defense Strategy

To evaluate the spatial retreat-based defense strategy, we considered a topology of 25 discrete communication cells (as depicted in Fig. 6) where a UAV traveled from a starting point (i.e., source) to an endpoint goal (i.e., destination) laid down in a 3-D space. In Fig. 6, the N tagged cells are the cells where there is a non-jammer present and the J tagged cells are the cells where there is a jammer present. The x-axis and the y-axis represent the spatial coordinates of the communication cells. The number of jammers is increased from a single jammer to ten jammers located around the 25 communication cells. The starting point is initialized at communication cell 0, and the goal is initialized at communication cell 24. For the experiment, we considered the following two use-cases:



(a)

(b)

Fig. 6. Experimental scenario of network topology consisting of 25 communication cells: (a) Case 1 topology and (b) Case 2 topology.

- **Case 1:** Shorter distance between the source and destination as shown in Fig. 6(a). In Case 1, the UAV requires 5 hop counts from the source to the destination. Three jammers are placed in the UAVs path at cell 6, cell 12, and cell 18.
- **Case 2:** Longer distance between the source and destination as shown in Fig. 6(b). In Case 2, the UAV requires 9 hop counts from the source to the destination. Three jammers are placed in the UAVs path at cell 3, cell 4, and cell 9.

The adaptivity of the proposed method in reacting to jamming attacks can be visualized by Figs. 7 and 8. Figs. 7 and 8 show the evaluation of the Q-learning cumulative score (i.e., reward) of the UAV agent after a certain number of iterations for Case 1 and Case 2, respectively. By definition, a reward is a numeric feed-back that evaluates the performance of certain action [17], [34]. As mentioned before, in our scenario, the UAV agent receives a negative reward (i.e., -1) if it detects a jammer and a positive reward ($+1$) if a non-jamming environment is detected. Fig. 7 shows the performance for Case 1 over 3000 iterations for the proposed federated reinforcement learning-based defense strategy (federated RL) with an increasing number of en-route jammers from the minimum of one jammer to the maximum of ten jammer locations. The performance is compared with that of the random policy. The random policy represents the Q-learning strategy to find the best route from the source to the destination [40]. The proposed adaptive federated reinforcement learning is built on top of the random policy algorithm by supporting it with the knowledge of the jammer detection and the epsilon greedy policy. Figs. 7 and 8 show a comparison between the random policy and the proposed federated reinforcement learning mechanism without the epsilon-greedy policy (i.e., ϵ set to 0). This means that full exploitation of the federated RL mechanism is performed and an evaluation of the impact of the federated learning-based jammer detection knowledge for a UAV agent under attack with increasing number of jammers can be clearly drawn. As shown in Fig. 7, note that the convergence is achieved considerably faster when more knowledge about the jammer locations is provided to the reinforcement learning model by the federated detection in Federated RL. For example, the Q-learning score for the random policy converged after 2000 iterations. In contrast, the Q-learning score converged after 1000 iterations for the proposed federated RL models that were trained with jammer detection knowledge of 4 to 10 jammer locations. The federated RL model with 1 jammer location detected, 2 jammer locations detected and 3 jammer locations detected converge at 2000 iterations, 1500 iterations, and 1500 iterations, respectively.

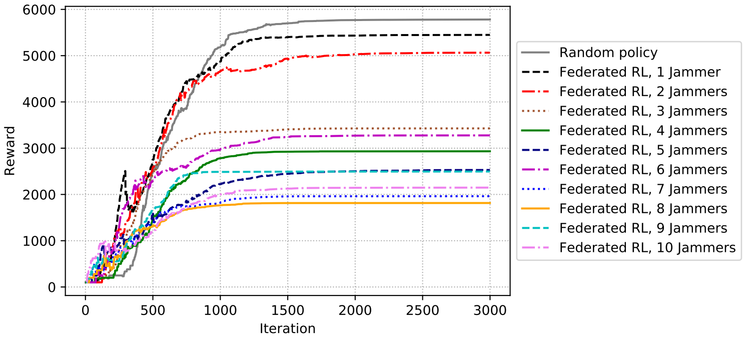


Fig. 7. Comparison of random policy and federated RL with increasing jammer locations in Case 1.

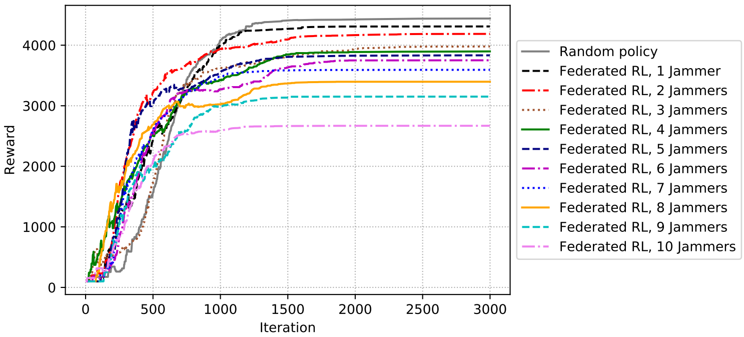


Fig. 8. Comparison of random policy and federated RL with increasing jammer locations in Case 2.

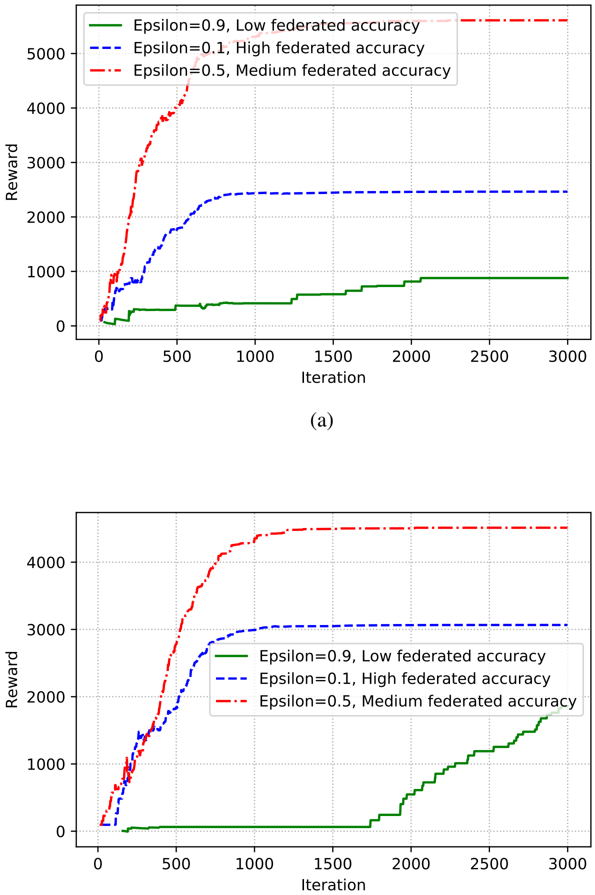
Fig. 8 shows the Q-learning cumulative score comparison over 3000 iterations for Case 2. Similar to that in Case 1, the proposed federated RL model with an increased number of jammer locations detected, converged considerably faster than that of the random policy which converged after 1800 iterations. The Q-learning score converged after 1500 iterations for the 3 jammer locations detected to 10 jammer locations detected cases under the proposed federated RL model. Meanwhile, the federated RL model with 1 jammer location detected, and 2 jammer locations detected converged at 1700 iterations and 1650 iterations, respectively. The phenomena occurred as the proposed federated RL was provided with additional information about the surrounding environment by the federated jamming detection mechanism that enabled the Federated RL to narrow its decision paths rapidly. Moreover, in the case of a shorter path as in Case 1, the random policy took a longer time to converge, as it experienced more path options than the longer path during its training, as in Case 2.

As shown in Figs. 7 8, as we increase the knowledge of the jammer detection to the federated reinforcement learning model, the average cumulative reward decreases as it does not explore all the different paths to the destination. Therefore, we apply an adaptive epsilon-greedy policy to adjust between the exploration and exploitation opportunity of the agent. In order to perform the adaptive federated reinforcement learning, an adaptive exploration-exploitation based reinforcement learning model is developed where the epsilon value was adjusted according to the average accuracy of the federated averaging model.

Fig. 9 shows the performance of the three average accuracy states, low federated accuracy, medium federated accuracy and high federated accuracy with the corresponding epsilon values of 0.9, 0.5, and 0.1 for both Case 1 and Case 2, respectively given all the ten jammer locations are present. Hence, if the model had low federated accuracy, we allowed the model to take more random decisions to reach at convergence. In contrast, if the model had high federated accuracy, we allowed the model to take fewer random decisions and rely more on the Q-table values to take the next action. As shown in both Case 1 and Case 2, an epsilon value of 0.1 converged faster than that for an epsilon value of 0.5 and 0.9. In Case 1, the model with an epsilon value of 0.1 converged after 1400 iterations, whereas the model with an epsilon value of 0.5 converged after 1800 iterations, and the model with an epsilon value of 0.9 almost never converged as it continued to take more random actions. Similarly, in Case 2, the model with an epsilon value of 0.1 converged after 1500 iterations, whereas the model with an epsilon value of 0.5 converged after 2000 iterations and the model with an epsilon value of 0.9 hardly converged by relying mostly on random actions than the best Q-value-based actions. However, note that the model with an epsilon value of 0.5 reached a higher reward value in both Case 1 (5400) and Case 2 (4600). Therefore, a good trade-off is to take an epsilon value close to 0.5, as higher rewards could be achieved with fast convergence. Therefore, for the proposed adaptive federated reinforcement learning-based jamming defense strategy, if the federated accuracy of the jamming detection was lower than a certain threshold value δ (i.e., less than 0.5), we selected an epsilon value that was slightly higher than 0.5. Conversely, if the federated accuracy of the jamming detection was higher than the threshold value δ (i.e., more than 0.5), we selected an epsilon value that was slightly lower than 0.5 to enable a higher reward score with fast convergence while exploiting the Q-values.

The threshold of the detection accuracy was selected arbitrarily to be 0.5 as a rule of thumb. The intuition behind selecting such a threshold is backed-up by a several reasons. First, as shown in Fig. 9, there are various disadvantages to using a high or a low federated accuracy as a threshold since it has severe consequences on the epsilon value. If the detection accuracy threshold is set to lower federated accuracy, the epsilon value would need to be increased which will reduce the overall exploitation and heavy exploration will not allow the average reward to converge thoroughly. In contrast, if the detection accuracy threshold is set to high federated accuracy, the epsilon value will tend to be very low. As a result, the proposed approach will converge early without enough exploration and that will lead to lower average reward value. Therefore, we select a mid-range federated accuracy value (i.e., 0.5) to ensure sufficient exploration and exploitation opportunities. As shown by the medium federated accuracy (i.e., red dotted line) in the figure, a medium federated accuracy with an epsilon value of 0.5 converges well and also achieves a high average reward score. Second, the goal of the proposed mechanism is to work well even when the detection accuracy is low. Therefore, given the jamming detection doesn't achieve high accuracy, as it is highly likely in a real-world scenario with an unbalanced data environment, we do not want the model to continuously perform random exploration due to higher threshold for the detection accuracy. In fact, high random exploration is unsuitable for the proposed scenario since it will not be an efficient approach for learning the jamming Q-table as the algorithmic convergence will be hard to attain. Besides, the average reward will be low due to the lower exploitation of the Q-table caused by heavy exploration. In such a scenario, a threshold accuracy value of around 0.5 allows to balance between the exploration and exploitation resulting in stable convergence and high average reward value. This is also confirmed from our experiment shown in Fig. 9 to be more suitable for our scenario.

Next, we perform the test phase of the trained Adaptive federated RL model by evaluating the spatial retreat paths selected with an increasing number of jammer locations. We compared the performance of our trained adaptive federated RL model with the best-selected route by the random policy, a centralized Dijkstra's algorithm [41] without any knowledge of the jammer locations and a centralized Dijkstra's algorithm with a centralized global knowledge of all the jammer locations as our baseline methods. The Dijkstra's algorithm (or Dijkstra's shortest path first algorithm) finds the shortest paths between the nodes in a graph based on the edge costs. For the Dijkstra's algorithm without knowledge of jammer locations, we set the edge weights to 1 to indicate that all the edges are considered equal to one another. For the Dijkstra's algorithm with the global knowledge of the jammer locations, we set the edge weights between a non-jammer location and a jammer location to 100 while the other edge weights remain 1. The epsilon value of the adaptive federated RL is set to 0.4 considering that the federated detection of the jamming attack is acceptable. Fig. 10 shows the different routes taken by the different models from the source to the destination in a 3-D space in Case 1. The random policy and Dijkstra's algorithm (without the knowledge of jammer locations)



(a)

(b)

Fig. 9. Comparison of cumulative reward for different epsilon value and level of federated detection accuracy: (a) Cumulative reward in Case 1 and (b) cumulative reward in Case 2.

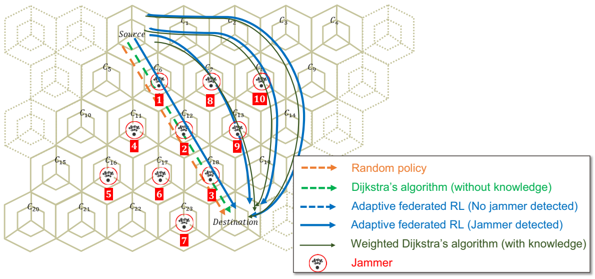

Source
Destination
Random policy
Dijkstra's algorithm (without knowledge)
Adaptive federated RL (No jammer detected)
Adaptive federated RL (Jammer detected)
Weighted Dijkstra's algorithm (with knowledge)
Jammer
1
2
3
4
5
6
7
8
9
10

Fig. 10. Routes taken by the random policy, Dijkstra's algorithm (without knowledge), adaptive federated RL (with no jammer detected), adaptive federated RL (with jammer detected), and weighted Dijkstra's algorithm (wiht knowledge) in Case 1.

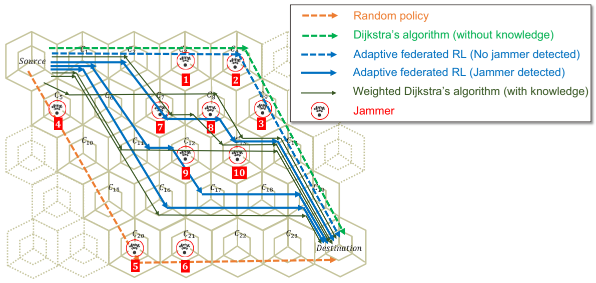

Source
Destination
Random policy
Dijkstra's algorithm (without knowledge)
Adaptive federated RL (No jammer detected)
Adaptive federated RL (Jammer detected)
Weighted Dijkstra's algorithm (with knowledge)
Jammer
1
2
3
4
5
6
7
8
9
10

Fig. 11. Routes taken by the random policy, Dijkstra's algorithm (without knowledge), adaptive federated RL (with no jammer detected), adaptive federated RL (with jammer detected), and weighted Dijkstra's algorithm (with knowledge) in Case 2.

chose the best route from the source to the destination through the communication cells C6, C12, and C18. The jammer locations were placed in 10 communication cells incrementally as shown by the numbers in red boxes. When jammer locations were placed at C6, C12, and C18, the proposed adaptive federated RL and the weighted Dijkstra's algorithm chose the following route: Source to C1, C7, C13, and C19 to the destination. When jammer locations were places at C11, C16, C17, and C23, it did not affect the path of the adaptive federated RL model and the weighted Dijkstra's algorithm as the jammer did not fall in its selected path. When jammer locations were placed at C7 and C13, adaptive federated RL and the weighted Dijkstra's algorithm alternatively took the following route: Source to C1, C2, C8, C14, and C19 to the destination. Upon detecting the jammer location at C8, the adaptive federated RL and the weighted Dijkstra's algorithm chose the following route: Source to C1, C2, C3, C9, C14, and C19 to the destination. Fig. 11 shows the different routes taken by the three models from the source to the destination in a three-dimensional space in Case 2. The random policy selected the best route from the source to the destination through the following communication cells: Source to C5, C10, C15, C20, C21, C22, and C23 to the destination. The Dijkstra's algorithm (without knowledge of the jammer locations) chose the best route from the source to the destination through the following communication cells: Source to C1, C2, C3, C4, C9, C14, and C19 to the destination. Technically, both the routes were equivalent in terms of distance and resulted because of the random selection by the random policy and the Dijkstra's algorithm (without knowledge of the jammer locations). When no jammer locations were detected, the adaptive federated RL and the weighted Dijkstra's algorithm also chose one of these two paths, i.e., source to C1, C2, C3, C4, C9, C14, and C19 to the destination. When jammer locations were detected at C3, C4, and C9, the adaptive federated RL and the weighted Dijkstra's algorithm chose the following route: Source to C1, C2, C7, C8, C13, C14, and C19 to the destination. When jammer locations were detected at C7 and C8, the adaptive federated RL model and the weighted Dijkstra's algorithm chose the following route: Source to C1, C6, C11, C12, C17, C18, and C19 to the destination. When jammer locations were detected at C12, and C13, the adaptive federated RL model and the weighted Dijkstra's algorithm alternatively chose the following route: Source to C1, C6, C11, C16, C17, C18, and C19 to the destination to avoid the jammer locations. It is noteworthy to mention that, in a jamming scenario the adaptive federated RL can operate even if the centralized global system cannot be reached. However, the weighted Dijkstra's algorithm would require to communication with the centralized system to get the route decision. In addition, for the weighted Dijkstra's algorithm a complete global knowledge of the jammer locations in the network is required in the centralized system to compute the best route prior to the flight of the UAV. Moreover, a centralized jamming detection may not be accurate from the unbalanced data in the UAV spatial environment. From the above description, it can be seen that the adaptive federated RL performs equally as well to the baseline weighted Dijkstra's algorithm, which is supported by the global knowledge, while on-device decisions can be performed by the UAVs in the unbalanced data environment.

Table 1. Performance comparison of the adaptive federated RL with random policy, Dijkstra's algorithm (without knowledge), and weighted Dijkstra's algorithm (with knowledge).

Algorithm	**Case 1**	**Case 2**
Success rate (%)		
Random policy	50.909	81.818
Dijkstra's algorithm (w/o knowledge)	50.909	72.727
Weighted Dijkstra's algorithm (with knowledge)	**100**	**100**
Adaptive federated RL	**100**	**100**
Average number of hop count		
Random policy	**5**	**9**
Dijkstra's algorithm (w/o knowledge)	**5**	**9**
Weighted Dijkstra's algorithm (with knowledge)	6.273	**9**
Adaptive federated RL	6.273	**9**
Average number of iterations to reach convergence		
Random policy	2000	1800
Adaptive federated RL	**1800**	**1700**
Average cumulative reward		
Random policy	**5820**	**4710**
Adaptive federated RL	5626	4657

Table 1 summarizes the overall performance of the proposed adaptive federated RL model compared with the random policy, Dijkstra's algorithm (without knowledge) and the weighted Dijkstra's algorithm (with knowledge of the jammer locations) for a node under attack with up to a total ten en-route jammer locations. The success rate was calculated by the total number of hop counts without a jammer location present over the total number of hop counts averaged over 11. In other words, there were 10 episodes where a new jammer is introduced up to a total of 10 jammers and 1 episode with no jammer present. The average hop count is the total number of hops taken from the source to the destination averaged over the 11 episodes. The average number of iterations to reach convergence indicates the average number of times the algorithm's parameters are updated and the average cumulative reward is the total Q-learning score received by the model averaged over all the episodes.

In Case 1, as the number of jammer locations are increased from 0 to 1, 2, and 3, the number of jammer hops covered by random policy and Dijkstra's algorithm (without knowledge) was also increased by 1, 2, and 3 hop counts respectively, as the jammers fell in their selected route. However, the number of jammer locations covered by the adaptive federated RL model and weighted Dijkstra's algorithm remained 0, as they chose a route with no jammer locations present. When the total number of jammer locations was increased from 4 to 10, the total number of jammer locations covered by the random policy and the Dijkstra's algorithm (without knowledge) remained constant at three, as the hop counts for the random policy and Dijkstra's algorithm (without knowledge) were the same five communication cells. However, the total number of hop counts from the source to the destination for the adaptive federated RL model and the weighted Dijkstra's algorithm increased from 5 to 7 and 7 to 8 as the total number of jammer locations was increased from 7 to 8 and 9 to 10, respectively.

The proposed adaptive federated RL and the weighted Dijkstra's algorithm was aided with the detection of the jammer locations in its route (i.e., the optimal route to the destination) Therefore, the two algorithms select an alternative route other than the same optimal route selected by the random policy to reach its destination. However, it is noteworthy to mention that the alternate route for the UAVs is the feasible and near-optimal route to reach its destination. Thus, the success rate of the adaptive federated RL and the weighted Dijkstra's algorithm increased to 100% and 100%, respectively. In contrast, the success rate of the random policy and the Dijkstra's algorithm reduced to 50.91% and 81.82%. However, the average number of hop counts for the random policy and the Dijkstra's algorithm (without knowledge) in Case 1 is 5 and 5, respectively. Nevertheless, because of the jammer locations, the proposed adaptive federated RL and the weighted Dijkstra's algorithm takes an alternate longer path for which their average hop count were 6.273 and 6.273, respectively. However, in terms of the average number of iterations required for convergence, the adaptive federated RL (1800) is lower than the random policy (2000). Moreover, the average cumulative reward of the adaptive federated RL (5626) is also close to the random policy (5820) as it adjusts between the exploitation and exploration opportunity of the model. Therefore, in the Case 1 scenario, it can be observed that there is a 3.33% decrease in the average cumulative reward while a convergence is reached 200 iterations earlier by our proposed adaptive federated RL model in comparison to the random policy.

In Case 2, as the number of jammer locations are increased from 0 to 3, the number of jammer hops for the Dijkstra's algorithm (without knowledge) increased from 0 to 3 after which it remained constant at 3, as jammer locations were not in its selected route anymore. However, after the total number of jammer locations was increased from 3 to 6, the number of jammer hops covered by the random policy also increased from 0 to 3 as the jammer locations fell into its selected path. After six jammer locations, the number of jammer hops for random policy and Dijkstra's algorithm (without knowledge) steadily remained at 3, as the jammer locations were placed in the other cells. In contrast, the number of jammer locations covered by the adaptive federated RL model and the weighted Dijkstra's algorithm remained 0 for all the 10 jammer locations detected. Thus, the success rate of the random policy and the Dijkstra's algorithm reduced to 81.82% and 72.73%. In contrast, the success rate of the adaptive federated RL and the weighted Dijkstra's algorithm increased to 100% and 100% respectively. However, the average number of hop counts for the random policy, Dijkstra's algorithm (without knowledge), and adaptive federated RL as the alternate routes taken by the adaptive federated RL also consisted of 9 hop counts. In fact, the weighted Dijkstra's algorithm also consisted of 9 hop counts. As we increase the distance from the source to the destination in Case 2, the total number of hop counts for the proposed mechanism and the random policy become the same (i.e., 9 hop counts) for the increasing number of jammer locations. This is because there are more available alternate routes and all of these alternate routes have the same number of hop counts due to the larger distance from the source to the destination. However, in terms of the average number of iterations required for convergence, the adaptive federated RL (1700) is lower than the random policy (1800). Moreover, the averageMoreover, the average cumulative reward of the adaptive federated RL (4657) is significantly close to the random policy (4710) as it exploits the adaptive epsilon-greedy policy. As a result, in Case 2, there is a 1.12% decrease in the average cumulative reward while a convergence is reached 100 iterations earlier by our proposed adaptive federated RL model in comparison to the random policy.

V. CONCLUSION

In this paper, we proposed an adaptive federated reinforcement learning-based jamming defense strategy in FANET consisting of UAV nodes. Then, an epsilon-greedy policy-based Q-learning spatial retreat jamming defense strategy was proposed on the basis of a federated learning-based jamming detection mechanism. We showed that the proposed adaptive federated reinforcement learning-based approach enabled performing better spatial retreat defense strategies. For doing so, the proposed mechanism leverages an efficient federated jamming detection mechanism to locate and retreat from the jammers in a newly explored environment. The supporting federated detection mechanism provided environment-specific knowledge about the jammer locations to the Q-learning module to converge its Q-learning score faster and adapt the exploration-exploitation property of the model. In the future, we will consider a global model for the Q-learning architecture to further federate the defense strategy.

REFERENCES

[1] I. Bekmezci, O. K. Sahingoz, and S. Temel, “Flying ad-hoc networks (FANETs): A survey,” *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, May 2013.

[2] A. Guillen-Perez, and M. Cano, “Flying ad hoc Networks: A new domain for network communications,” *Sensors*, vol. 18, no. 10, p. 3571, Oct. 2018.

[3] I. Bekmezci, E. Senturk, and T. Turker, “Security issues in flying ad-hoc networks (FANETS),” *J. Aeronautics Space Technologies*, vol. 9, no. 2, pp. 13–21, July 2016.

[4] O. Sahingoz, “Networking models in flying ad-hoc networks (FANETs): Concepts and challenges,” *J. Intelligent Robotic Systems*, vol. 74, no. 1–2, pp. 513–527, Oct. 2014.

[5] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun, “FANET: Communication, mobility models and security issues,” *Comput. Netw.*, vol. 163, p. 106877, Nov. 2019.

[6] W. Xu Wenyuan, K. Ma, W. Trappe, and Y. Zhang, “Jamming sensor networks: Attack and defense strategies,” *IEEE Netw.*, vol. 20, no. 3, pp. 41–46, June 2006.

[7] L. Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and H. Vincent Poor, “Two-dimensional antijamming mobile communication based on reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9499–9512, July 2018.

[8] L. Xiao *et al.*, “UAV relay in VANETs against smart jamming with reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, Jan. 2018.

[9] H. Sedjelmaci, S. M. Senouci, and N. Ansari, “A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks,” *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1594–1606, Mar. 2018.

[10] H. Sedjelmaci, S. M. Senouci, and N. Ansari, “Intrusion detection and ejection framework against lethal attacks in UAV-aided networks: A bayesian game-theoretic methodology,” *IEEE Trans. Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1143–1153, May 2017.

[11] N. Mowla, I. Doh, and K. Chae, “Federated learning-based cognitive detection of jamming attack in flying ad-hoc network,” *IEEE Access*, vol. 8, no. 1, pp. 4338–4350, Dec. 2019.

[12] H. McMahan, E. Moore, D. Ramage, and S. Hampson, “Communication-efficient learning of deep networks from decentralized data,” *arXiv preprint*, arXiv:1602.05629, 2016.

[13] J. Konecny *et al.*, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint* arXiv:1610.05492, 2016.

[14] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon et al, “Towards federated learning at scale: System design,” *arXiv preprint* arXiv:1902.01046, 2019.

[15] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *Proc. IEEE ICC*, 2019, pp. 1–7.

[16] N. Mowla, I. Doh, and K. Chae, “On-device AI-based cognitive detection of bio-modality spoofing in medical cyber physical system,” *IEEE Access*, vol. 7, no. 1, pp. 2126–2137, Dec. 2018.

[17] H. H. Zhuo, W. Feng, Q. Xu, Q. Yang, and Y. Lin, “Federated reinforcement learning,” *arXiv preprint*, arXiv, 1901.08277, 2019.

[18] B. Liu, L. Wang, M. Liu, and C. Xu, “Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems,” *arXiv preprint* arXiv:1901.06455, 2019.

[19] S. Zhang, A. E. Choromanska, and Y. LeCun, “Deep learning with elastic averaging SGD,” in *Proc. NIPS*, 2015, pp. 685–693.

[20] Y. Zhang, J. Duchi, M. I. Jordan, and M. J. Wainwright, “Information-theoretic lower bounds for distributed statistical estimation with communication constraints,” in *Proc. NIPS*, 2013, pp. 2328–2336.

[21] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, C. Hong, “Federated Learning over wireless networks: Optimization model design and analysis,” in *Proc. IEEE INFOCOM*, 2019, pp. 1387–1395.

[22] P. Cevik, I. Kocaman, A. S. Akgul, and B. Akca, “The small and silent force multiplier: a swarm UAV-electronic attack,” *J. Intelligent Robotic Systems*, vol. 70, no. 1–4, pp. 595–608, Apr. 2013.

[23] Y. Shi *et al.*, “Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies,” in *Proc. IEEE ICC Workshops*, 2018, pp. 1–6.

[24] J. Dean *et al.*, “Large scale distributed deep networks,” in *Proc. NIPS*, 2012, pp. 1223–1231.

[25] S. Vadlamani, B. Eksioglu, H. Medal, and A. Nandi, “Jamming attacks on wireless networks: A taxonomic survey,” *International J. Production Economics*, vol. 172, pp. 76–94, Feb. 2016.

[26] G. Noubir, “On connectivity in ad hoc networks under jamming using directional antennas and mobility,” in *Proc. IFIP WWIC*, Springer, Berlin, Heidelberg, 2019 pp. 186–200.

[27] J. Seongah, O. Simeone, A. Haimovich, and J. Kang, "Beamforming design for joint localization and data transmission in distributed antenna system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 1, pp. 62–76, Jan. 2014.

[28] A. Mukherjee and A. Swindlehurst, "Robust beamforming for security in MIMO wiretap channels with imperfect CSI", *IEEE Trans. Signal Processing*, vol. 59, no. 1, pp. 351–361, Jan. 2011.

[29] F. Zhu, "Joint information- and jamming-beamforming for physical layer security with full duplex base station", *IEEE Trans. Signal Processing*, vol. 62, no. 24, pp. 6391–6401, Dec. 2014.

[30] S. Bhunia, V. Behzadan, P. Alexandre Regis, and S. Sengupta, “Adaptive beam nulling in multi hop ad hoc networks against a jammer in motion,” *Computer Netw.*, vol. 109, pp. 50–66, Nov. 2016.

[31] S. Bhunia, V. Behzadan, P. Alexandre Regis, and S. Sengupta, “Distributed adaptive beam nulling to survive against jamming in 3d uav mesh networks,” *Computer Netw.*, vol. 137, pp. 83–97, June 2018.

[32] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, “A tutorial on UAVs for wireless networks: Applications, challenges, and open problems,” *IEEE Commun. Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, thirdquarter 2019.

[33] S. Bhattacharya, and Tamer Basar, “Game-theoretic analysis of an aerial jamming attack on a UAV communication network,” in *Proc. ACC*, 2010, pp. 818–823.

[34] J. Glascher, N. Daw, P. Dayan, and J. P. O’Doherty, “States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning,” *Neuron*, vol. 66, no. 4, pp. 585–595, May 2010.

[35] A. Geron, “Hands-on machine learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems,” *O’Reilly Media, Inc.*, 2017.

[36] Q. Wei, F. L. Lewis, Q. Sun, P. Yan, and R. Song, “Discrete-time deterministic *Q*-learning: A novel convergence analysis,” *IEEE Trans. cybernetics*, vol. 47, no. 5, pp. 1224–1237, May 2016.

[37] B. Luo, D. Liu, T. Huang, and D. Wang, “Model-free optimal tracking control via critic-only Q-learning,” *IEEE Trans. neural networks and learning systems*, vol. 27, no. 10, pp. 2134–2144, Oct. 2016.

[38] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, “Adaptive optimal control of unknown constrained-input systems using policy iteration and neural network,” *IEEE Trans. Neural Netw. learning Systems*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.

[39] O. Punal, C. Pereira, A. Aguiar, and J. Gross, CRAWDAD dataset uportorwthaachen/vanetjamming2014 (v. 2014-05-12), downloaded from https://crawdad.org/uportorwthaachen/vanetjamming2014/20140512, https://doi.org/10.15783/C7Q306, May 2014.

[40] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman, “Deep successor reinforcement learning,” *arXiv preprint arXiv:1606.02396*, 2016.

[41] S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, “Applying Dijkstra algorithm for solving neutrosophic shortest path problem,” in *Proc. ICAMechS*, IEEE, 2016, pp. 412–416.

[42] M. Asadpour, D. Giustiniano, K. A. Hummel, and S. Heimlicher. “Characterizing 802.11 n aerial communication,” in *Proc. ACM MobiHoc workshop*, 2013, pp. 7–12.

[43] S. Rosati, K. Kruzelecki, L. Traynard, and B. R. Mobile. "Speed-aware routing for UAV ad-hoc networks,” in *Proc. IEEE Globecom Workshops*, 2013, pp. 1367–1373.

[44] N. Goddemeier, S. Rohde, and C. Wietfeld, “Experimental validation of RSS driven UAV mobility behaviors in IEEE 802.11s networks,” in *Proc. IEEE Globecom Workshops*, 2012, pp. 1550–1555.

[45] D. Broyles, A. Jabbar and J.P.G. Sterbenz, “Design and Analysis of a 3-D Gauss-Markov mobility model for highly-dynamic air bourne network,” in *Proc. ITC*, 2010, pp. 25–28.

[46] J. P. Rohrer *et al.*, “AeroRP Performance in Highly Dynamic Airborne networks using 3D Gauss Markov Mobility model,” in *Proc. MILCOM*, 2011, pp. 834–841.

[47] G.F Riley, and T. R. Henderso,. “The ns-3 network simulator,” *Modeling and tools for network simulation*, pp. 15-34. Springer, Berlin, Heidelberg, 2010.

[48] ns-3, “Wireless jamming model”. [Online] Available at https://www.nsnam.org

[49] A. Tahir, J. Boling, M. H. Haghbayan. H. T.Toivonen, J. Plosila “Swarms of Unmanned Aerial Vehicles — A Survey,” *J. Industrial Information Integration*, vol. 16, Dec. 2019.

Nishat I Mowla (S’18) received her B.S in Computer Science from Asian University for Women, Chittagong, Bangladesh, in 2013, M.S. degree in Computer Science and Engineering from Ewha Womans University, Seoul, Korea in 2016. She worked at Asian University for Women, Chittagong, Bangladesh as Senior Teaching Fellow. She is currently a Ph.D. student at Ewha Womans University, Seoul, Korea. Her research interests include next generation network security, IoT network security, machine intelligence, and network traffic analysis. Ms. Mowla was awarded the best paper award at the Qualcomm paper awards 2017, Ewha Womans University, Seoul, Korea paper competition. She is a student member of IEEE.

Nguyen H. Tran (S’10–M’11–SM’18) received his B.S. from Hochiminh City University of Technology and Ph.D. from Kyung Hee University, in Electrical and Computer Engineering, in 2005 and 2011, respectively. Since 2018, he has been with the School of Computer Science, The University of Sydney, where he is currently Senior Lecturer. He was Assistant Professor with Department of Computer Science and Engineering, Kyung Hee University, Korea from 2012 to 2017. His research interest is to apply the analytical techniques of optimization, game theory, and stochastic modeling to cutting-edge applications such as cloud and mobile edge computing, data centers, heterogeneous wireless networks, and big data for networks. He received the best KHU thesis award in engineering in 2011 and best paper award at IEEE ICC 2016. He has been Editor of IEEE Trans. Green Communications and Networking since 2016, and served as Editor of the 2017 Newsletter of Technical Committee on Cognitive Networks on Internet of Things.

Inshil Doh received her B.S. and M.S. in Computer Science and Engineering at Ewha Womans University, Korea, in 1993 and 1995, respectively. She received her Ph.D. degree in Computer Science and Engineering from Ewha Womans University in 2007. From 1995-1998, Prof. Doh worked at Samsung SDS of Korea. She was a Research Professor at Ewha Womans University and at Sungkyunkwan University. She is currently an Associate Professor of the Department of Cyber Security at Ewha Womans University, Seoul, Korea. Her research interests include wired/wireless network security, sensor network security, and IoT network security.

Kijoon Chae received his B.S. in Mathematics from Yonsei University in 1982, an M.S. in Computer Science from Syracuse University in 1984. He received his Ph.D. in Electrical and Computer Engineering from North Carolina State University in 1990. He is currently a Professor at the Department of Computer Science and Engineering at Ewha Womans University, Seoul, Korea. His research interests include blockchain, security of FANET, sensor network, smart grid, CDN, SDN and IoT, network protocol design, and performance evaluation.