Federated Deep Reinforcement Learning for Efficient Jamming Attack Mitigation in O-RAN

Zakaria Abou El Houda, *Member, IEEE*, Hajar Moudoud, *Member, IEEE*, and Bouziane Brik, *Senior Member, IEEE*

***Abstract*—Open RAN (ORAN or O-RAN) revolutionizesRadio Access Networks (RAN) by offering flexibility and cost-efficiency through inter-vendor equipment interoperability.More importantly, it addresses emerging security threats, such as jamming attacks, by incorporating network softwarization and leveraging Artificial Intelligence (AI) techniques. However, AI-based systems face challenges such as limited training data, slow convergence, and vulnerability to dynamic attack patterns like Zero-day attacks. To enhance jamming attack mitigation in O-RAN, Multi-Agent Reinforcement Learning (MARL) has been introduced for improved flexibility and robustness. However, MARL requires data sharing, which consumes network bandwidth and slows down training, and the curse of dimensionality limits its benefits due to the exponential growth of the state-action space. To overcome these limitations, we provide a novel framework that combines federated learning (FL) and deep reinforcement learning (DRL) for efficient jamming attack detection in O-RAN. FL allows decentralized agents to train local models using their data sources, and the models are aggregated into a global model at a Non-real-time RAN Intelligent Controller (RIC) to guide decision-making. The federated learning process enables distributed intelligence, while deep reinforcement learning ensures adaptive and robust jamming attack detection. Our proposed framework improves security, privacy, and resilience in ORAN through collaborative FL and adaptive DRL. Extensive simulations demonstrate its superiority in detection accuracy, resource efficiency, and scalability.**

***Index Terms*—Federated learning, jamming attacks, multi-agent reinforcement learning, Open RAN, wireless sensor networks.**

I. Introduction

JAMMING attacks have witnessed a significant rise in recent years, emerging as a prominent threat to the security and reliability of O-RAN networks [1]. Wireless networks, such as the Open Radio Access Network (O-RAN), are vulnerable to jamming attacks that can severely impact their performance and compromise user experience. According to a recent report, the frequency and severity of jamming attacks have increased globally [2], affecting communication systems and disrupting critical services [3].

Attacking the open radio interface (ORI) in communication systems can indeed be complex, and adversaries may employ various techniques to disrupt or compromise the communication channel. One such method is a jamming attack, where the adversary intentionally interferes with the radio signals to disrupt communication. These attacks exploit vulnerabilities in wireless communication protocols, targeting radio signals and causing interference that hampers the transmission of data and compromises network performance. The consequences of jamming attacks can be severe, leading to service disruptions, financial losses, and potential breaches in data privacy. In light of these alarming statistics, it is imperative to develop robust countermeasures and mitigation strategies to safeguard O-RAN networks against jamming attacks, ensuring uninterrupted communication and preserving the integrity of the system.

AI algorithms can identify patterns and anomalies related to jamming attempts by analyzing large amounts of network data. Additionally, AI algorithms can optimize resource allocation and traffic management to mitigate the impact of jamming attacks and ensure efficient network operation [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. However, these AI-based systems, including those deployed in O-RAN networks, encounter challenges when dealing with jamming attacks. These challenges include limited training data, slow convergence, and vulnerability to dynamic attack patterns like zero-day attacks. To enhance the mitigation of jamming attacks in O-RAN, Multi-Agent Reinforcement Learning (MARL) has been introduced as a promising approach to improve flexibility and robustness. However, the implementation of MARL comes with certain challenges. One such challenge is the need for data sharing between agents, which can consume significant network bandwidth and result in slower training processes. Additionally, the curse of dimensionality in MARL refers to the exponential growth of the state-action space, which can limit the potential benefits of the approach. These challenges need to be addressed to fully leverage the capabilities of MARL in enhancing jamming attack mitigation in O-RAN.

To overcome these limitations, we propose a novel framework that combines FL with DRL for effective jamming attack detection in O-RAN. FL is a decentralized learning paradigm that allows multiple agents to collaboratively learn from their local

Manuscript received 30 June 2023; revised 11 November 2023 and 11 January 2024; accepted 26 January 2024. Date of publication 29 January 2024; date of current version 16 July 2024. The review of this article was coordinated by the Guest Editors of the Special Section on Open Radio Access Networks: Architecture, Challenges, Opportuities, and Use Cases in Vehicular Networks. *(Corresponding author: Zakaria Abou El Houda.)*

Zakaria Abou El Houda is with the Centre Énergie Matériaux Télécommunications, Institut National de la Recherche Scientifique (INRS-EMT), Varennes, QC J3X 1S2, Canada (e-mail: zakaria.abouelhouda@inrs.ca).

Hajar Moudoud is with the L@bISEN, ISEN Yncrea Ouest, 58000 Nevers, France (e-mail: hajar.moudoud@usherbrooke.ca).

Bouziane Brik is with the Computer Science Department, College of Computing and Informatics, Sharjah University, Sharjah J3X 1S2, UAE (e-mail: bbrik@sharjah.ac.ae).

Digital Object Identifier 10.1109/TVT.2024.3359998

data without directly sharing it. In FL, each agent trains its local model using its local data and periodically communicates only the model updates with a centralized server. This approach reduces the amount of data exchanged between agents, mitigating the bandwidth and privacy concerns associated with transmitting large amounts of raw data. In our proposed framework, multiple agents are deployed within the near real-time RIC module of O-RAN, where they generate local learning models using FL. These models capture the knowledge and insights gained from their respective local data sources. The agents then communicate the model updates to a centralized Non-real-time RIC, where the local models are aggregated to create a global model that represents the collective intelligence of the agent network. The global model, derived from the FL process, is utilized within the DRL framework to facilitate effective jamming attack detection. The agents interact with the environment, taking actions based on the global model's guidance, and receive rewards and system state information. The DRL algorithm, powered by the deep neural network model, enables the agents to learn and update their policies, considering the aggregated knowledge from the federated learning process [17].

By combining FL with DRL, our approach addresses the challenges of data sharing and the curse of dimensionality in MARL. It enables effective jamming attack detection in O-RAN by utilizing the benefits of distributed learning, localized decision-making, and the expressive power of deep neural networks. Furthermore, the integration of FL and DRL ensures efficient utilization of network resources and promotes scalability in large-scale wireless networks. The combination of FL and DRL allows agents to benefit from the distributed intelligence and localized decision-making capabilities of FL while leveraging the expressive power and generalization abilities of DRL models. We evaluate our approach through extensive simulations, comparing it with existing methods. The results demonstrate the superiority of our approach in terms of detection accuracy, resource efficiency, and scalability. The federated learning process enables distributed intelligence and localized decision-making, while the deep reinforcement learning component ensures adaptive and robust detection of jamming attacks. Extensive simulations show that our proposed framework outperforms recent AI-based models in terms of accuracy and F1 score while significantly reducing training delay and achieving fast convergence.

The structure of this paper is organised as follows. In Section II, we overview some of the prominent works in the field of jamming attack detection. Then, Section III provides our system model. In Section IV, we evaluate the performance of our proposed framework in terms of efficiency and compare it to recent work. Section V concludes the paper.

II. Related Work

O-RAN seeks to enhance network performance by employing a virtualized network architecture and open standardized interfaces for deployment. This novel and distinctive network environment, however, has given rise to new vulnerabilities that the current security frameworks are ill-equipped to handle. Traditional cellular network security methods, such as authentication protocols, are based on the belief in robust trust connections. This assumption, however, is at odds with the changing architecture and deployment paradigm beyond 5 G networks.

A handful of research studies have begun to explore the security challenges associated with O-RAN deployment and suggest solutions to address them [18]. Habler et al. [19] performed a systematic Adversarial Machine Learning (AML) threat assessment for O-RAN, examining ML use cases and deployment scenarios, and establishing a threat model that identifies potential adversaries, their capabilities, and objectives. Similarly, Mohammadi et al. [20] presented a comprehensive examination of the SVM-based intrusion detection and feature selection methods. Nair et al. [21] presented a novel unsupervised ML algorithm designed specifically for rapid RF fingerprinting of LoRa-modulated chirps in emerging networks. The objective is to establish a robust method for authenticating IoT sensors in the context of new network deployments.

To enhance the performance of Intrusion Detection Systems (IDS), researchers have explored the application of RL approaches. RL methods are well-suited for the ORAN environment, as they enable systems to adapt their behavior based on continuous feedback to maximize rewards. Several studies have utilized DRL algorithms for intrusion detection in both real-world and simulated environments [22]. Iannucci et al. [23] proposed an innovative approach called Intrusion Response DRL, which employs a stateful Markov Decision Process to handle large-scale systems and sophisticated attack scenarios. Alazizadeh et al. [24] investigated the use of Deep Q-Learning (DQL), a combination of RL and a deep neural network, for network intrusion detection. Their approach leverages automated trial and error and continual learning to effectively detect various types of intrusions, enhancing overall detection capabilities. However, while RL techniques have demonstrated effectiveness in detecting intrusions in IoT networks, none of the existing works have considered the impact of low device and agent reliability on intrusion detection accuracy. The presence of malicious software within a network can potentially overwhelm or compromise its functionality, leading to decreased reliability and accuracy of intrusion detection systems.

FL has emerged as a distributed ML paradigm, specifically designed to address privacy concerns associated with large-scale interactions such as O-RAN. It has become a crucial component in securing open networks, offering numerous benefits. Extensive research has been conducted on various aspects of FL, including its effectiveness [25], privacy preservation [26], and security enhancements [27]. In [28], Lu et al. proposed a secure data sharding framework where the blockchain technology is integrated with the FL framework. In this work, the authors described the specific mechanisms employed to ensure privacy, data integrity, and security in the context of IoV data sharing. The provided experimental evaluations have demonstrated the effectiveness and efficiency of their proposed approach based on metrics such as communication overhead, training accuracy, and the resilience of the system against attacks. In [29], Yi et al. proposed a novel methodology that utilizes FL to predict traffic flow while preserving the privacy of individual data sources. The

proposed FL enables model training to be performed locally on individual devices without sharing sensitive data. By aggregating the locally trained models, a global traffic flow prediction model is generated. In [30], Yin et al. proposed a traffic flow prediction model, called FedGRU (Federated Gated Recurrent Unit) based on the principles of FL. FedGRU is a solution to enable secure collaboration and knowledge sharing among IoT devices while protecting the privacy of individual data sources. FedGRU utilized FL to train deep learning models collaboratively without sharing raw data. The mechanism incorporates privacy-preserving techniques and secure aggregation methods to ensure data security during the model training process. In [31], Guowen et al. proposed a framework called Verifynet, which aims to enhance the security and veracity of FL. Verifynet employs cryptographic techniques and secure multi-party computation protocols to protect the privacy of participants' data during the model training process. It also provides mechanisms for verifying the correctness and integrity of the contributed model updates from different participants. The authors present an in-depth analysis of the Verifynet framework, explaining its technical components and the cryptographic techniques utilized. It may discuss the experimental evaluation of Verifynet's security guarantees and the veracity of the properties using real-world datasets or simulations.

While integrating DRL enhances the effectiveness of conventional AI-driven mitigation strategies against jamming attacks, applying DRL directly to a large-scale network poses challenges. The need for an extensive policy space to govern all nodes complicates the DRL agent's ability to reach a stable state and attain optimal performance. Additionally, the substantial data (i.e., state-action space) exchanged among agents can strain network bandwidth, potentially impeding the training process. By combining FL with DRL, our approach addresses the challenges of data sharing and the curse of dimensionality in MARL. It enables effective jamming attack detection in O-RAN by utilizing the benefits of distributed learning, localized decision-making, and the expressive power of deep neural networks. Furthermore, our proposed framework ensures efficient utilization of network resources and promotes scalability in large-scale wireless networks.

III. FRL-Enabled Jamming Attack Mitigation in O-RAN

This section starts by presenting the identified problem, followed by a comprehensive exploration of our proposed solution and its associated mathematical formulation.

A. System Architecture

We formalize the problem of jamming attack mitigation as a Federated Multi-Agent Reinforcement Learning (FMARL) problem. In FMARL, we consider multiple agents that actively cooperate and learn to make effective decisions in an environment affected by jamming attacks. The environment in this case represents the Radio Access network, including components such as O-RU, O-DU, and O-CU. The agents interact with this environment through the E2 interface, which allows them to execute a set of actions and receive feedback in the form of rewards and system state information.

The system architecture, as illustrated in Fig. 1, shows the hierarchical wireless topology network we consider. Within this architecture, the near real-time RIC module of O-RAN plays a crucial role. It hosts intelligent agents that are responsible for generating local learning models. These agents operate in a federated manner, allowing for distributed intelligence and localized decision-making. Once the local models are created, they are transmitted to the centralized Non-real-time RIC, where they are aggregated to create a global model. This collaborative approach ensures that the knowledge and insights gathered by the intelligent agents are effectively utilized for system-wide decision-making.

However, the sharing of data, particularly in terms of the state-action space, between agents in MARL setup can present challenges such as consuming network bandwidth and potentially slowing down the training process. Additionally, the curse of dimensionality in MARL, characterized by the exponential growth of the discrete state-action space, can limit the potential benefits of traditional RL approaches. To address these issues and ensure effective jamming attack detection in O-RAN, we propose a novel approach that combines FL with DRL. In our system, intelligent agents are deployed within the near real-time RIC module of O-RAN, where they generate local learning models using FL. These models capture the knowledge and insights gained from their respective local data sources. The agents then communicate the model updates to a centralized Non-real-time RIC, where the local models are aggregated to create a global model that represents the collective intelligence of the agent network.

The global model, derived from the federated learning process, is utilized within the DRL framework to facilitate effective jamming attack detection. The agents interact with the environment, taking actions based on the global model's guidance, and receive rewards and system state information. The DRL algorithm, powered by the deep neural network model, enables the agents to learn and update their policies, considering the aggregated knowledge from the federated learning process. To detect jamming attacks, the intelligent agents periodically execute actions via the E2 interface. These actions are aimed at identifying and mitigating the effects of jamming in the network (see the following section). For instance, an agent may adjust the resource allocation and scheduling policy within the O-DU's MAC layer to isolate and minimize the impact of jammers on the overall system performance. The rewards obtained by the agents are determined based on the consequences of their actions, taking into account metrics such as the quality of experience for the users. Simultaneously, the new state of the system, including information about the total number of allocated resource blocks and user density, is conveyed to the agents through the E2 interface. This feedback loop allows the agents to continuously learn and adapt their strategies in response to changing network conditions and evolving jamming attacks. Our strategy overcomes the problems with data sharing and the dimensionality curse in MARL by merging FL with DRL. It enables effective jamming attack detection in O-RAN by utilizing the benefits of distributed

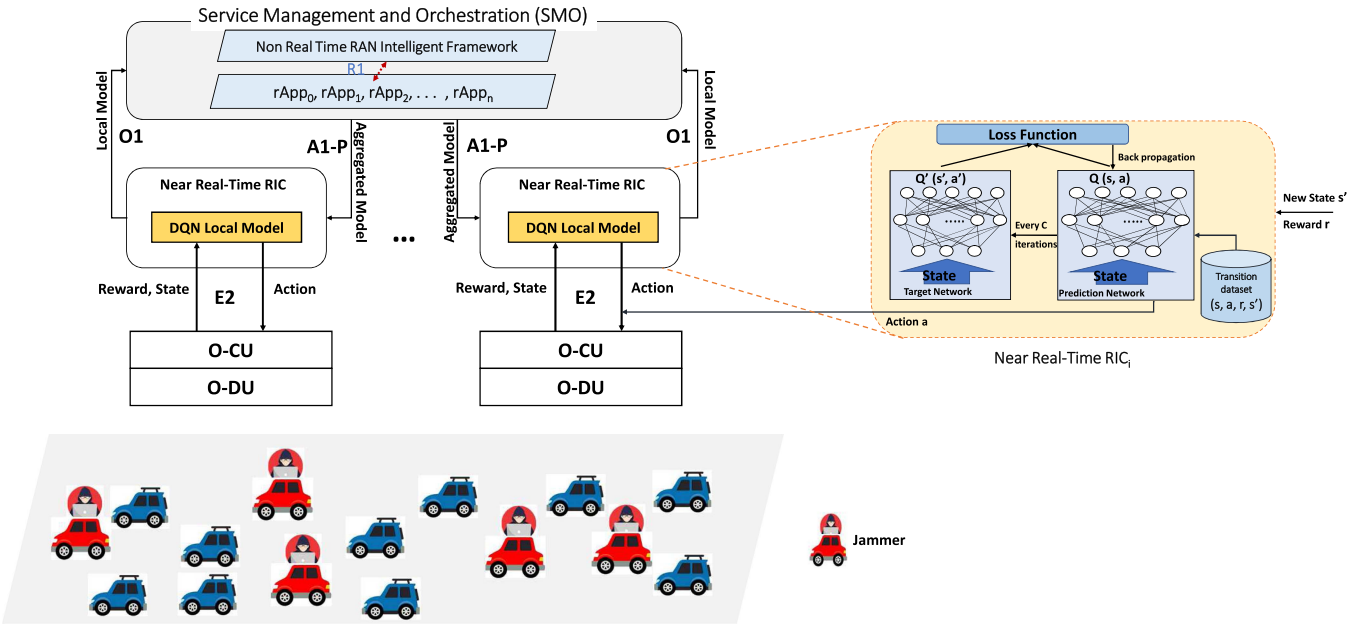

Fig. 1. Proposed O-RAN-based architecture.

learning, localized decision-making, and the expressive power of deep neural networks. Furthermore, the integration of FL and DRL ensures efficient utilization of network resources and promotes scalability in large-scale wireless networks.

B. System Model

In this subsection, we outline the mathematical approach to the problem we studied.

1) Markov Decision Process (MDP): The MDP is defined by the triplet (S_t, A_t, R_t), where S_t, A_t, and R_t represent the state space, action space, and reward function at time step t, respectively. State S_t: Represents the current information or observations available to the agent at time step t. It captures the relevant aspects of the environment, such as signal strength, network traffic, and any indications of jamming activity. Action A_t: The decision or choice made by the agent at time step t based on the current state S_t. It represents the agent's response to the detected or potential jamming activity. Reward Function $R(S_t, A_t)$: Let S_t represent the state at time step t, and A_t represent the action taken at time step t. The reward function $R(S_t, A_t)$ is defined as: computing the immediate reward from action A_t in state S_t. The reward function encourages desired behavior and penalizes undesirable actions. The reward function, $R(S_t, A_t)$, can be defined as follows:

$$R(S_t, A_t) = w_{\text{prompt}} \cdot R_{\text{prompt}}(A_t) + w_{\text{detection}} \cdot R_{\text{detection}}(S_t) + w_{\text{mitigation}} \cdot R_{\text{mitigation}}(A_t) \quad (1)$$

where w_{prompt}, $w_{\text{detection}}$, $w_{\text{mitigation}}$, and $w_{\text{ineffective}}$ are weights associated with each reward component. More specifically:

- *Prompt Action Reward ($R_{prompt}(A_t)$):* Encourages the agent to take actions promptly to mitigate or respond to jamming. It is weighted by w_{prompt}.

$$R_{\text{prompt}}(A_t) = \begin{cases} 1 & \text{if the agent takes action } A_t \text{ to encourage prompt action} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

- *Jamming Detection Reward ($R_{detection}(S_t)$):* Rewards the agent for successfully detecting jamming based on the current state. It is weighted by $w_{\text{detection}}$.

$$R_{\text{detection}}(S_t) = \begin{cases} 1 & \text{if jamming is successfully detected} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

- *Mitigation Reward ($R_{mitigation}(A_t)$):* Rewards the agent for successful actions taken to mitigate the impact of jamming. It is weighted by $w_{\text{mitigation}}$.

$$R_{\text{mitigation}}(A_t) = \begin{cases} n & \text{if the agent takes } n \text{ successful actions} \\ 0 & \text{if no successful actions are taken} \end{cases} \quad (4)$$

where n is a positive integer representing the number of successful taken actions to mitigate a jamming attack in ORAN.

To incorporate penalties, let $P(S_t, A_t)$ denote the penalty function penalizing undesirable actions or delays in response. Penalty Function $P(S_t, A_t)$: Calculates the penalty for taking action A_t in state S_t. The penalty function discourages undesirable behavior and delays in responding to jamming. The penalty function can be defined as:

$$P(S_t, A_t) = w_{\text{Ineffective}} \cdot P_{\text{Ineffective}}(A_t) + w_{\text{delay}} \cdot P_{\text{delay}}(S_t, A_t) \quad (5)$$

where $w_{\text{Ineffective}}$ and w_{delay} are weights associated with each penalty component.

- *Ineffective Action Penalty ($R_{ineffective}(A_t)$):* Ineffective Action Penalty: Penalizes the agent for taking actions that fail to effectively address the jamming. It is weighted by $w_{\text{ineffective}}$. It is be defined as:

$$R_{\text{ineffective}}(A_t) = \begin{cases} -m & \text{if the agent takes } m \text{ ineffective} \\ & \quad \text{actions} \\ 0 & \text{if no ineffective actions are taken} \end{cases} \tag{6}$$

  where m is a positive integer representing the number of ineffective taken actions.
- *Delay Penalty:* Penalizes the agent for delays in responding to jamming. It encourages prompt action and is weighted by w_{delay}.

Thus, the total Reward $R_{\text{total}}(S_t, A_t)$ that takes into account both the reward and penalty components is calculated as the sum of the reward function and penalty function, as follows:

$$R_{\text{total}}(S_t, A_t) = R(S_t, A_t) - P(S_t, A_t) \tag{7}$$

2) Multi-Agent Reinforcement Learning: MARL stands for Multi-Agent Reinforcement Learning, which extends reinforcement learning to the setting where multiple agents interact in a shared environment.

In MARL, there are N agents, and the environment is represented as a Markov Game, which is an extension of the Markov Decision Process (MDP). The Markov Game is defined by a tuple $(S, A_1, \ldots, A_N, P, R_1, \ldots, R_N, \gamma)$, where:

S is the set of states, A_i is the set of actions available to agent i, $P(s, a_1, \ldots, a_N, s')$ is the state transition probability, representing the probability of transitioning from state s to state s' when agents take actions $a_1, \ldots, a_N$, $R_i(s, a_1, \ldots, a_N, s')$ is the reward function for agent i, representing the immediate reward obtained by agent i when transitioning from state s to state s' while agents take actions $a_1, \ldots, a_N$, γ is the discount factor, determining the importance of future rewards.

The goal in MARL is to find a joint policy $\pi_1(a_1|s), \ldots, \pi_N(a_N|s)$ for all agents that maximizes the expected cumulative reward over time. This can be represented by the joint value function $V^{\pi}(s)$, which is the expected cumulative reward starting from state s and following the joint policy π, and the joint action-value function $Q^{\pi}(s, a_1, \ldots, a_N)$, which is the expected cumulative reward starting from state s, with agents taking actions $a_1, \ldots, a_N$, and then following the joint policy π.

The joint value function and joint action-value function can be defined using the Bellman equations:

$$\begin{aligned} V^{\pi}(s) = \sum_{a_1 \in A_1} \cdots \sum_{a_N \in A_N} \Big[& R_1(s, a_1, \ldots, a_N) \\ & + \ldots + R_N(s, a_1, \ldots, a_N) \\ & + \gamma \sum_{s' \in S} P(s, a_1, \ldots, a_N, s') V^{\pi}(s') \Big] \end{aligned} \tag{8}$$

The optimal joint value function $V^*(s)$ and optimal joint action-value function $Q^*(s, a_1, \ldots, a_N)$ can be defined as:

$$\begin{aligned} V^*(s) = \max_{a_1 \in A_1} \cdots \max_{a_N \in A_N} \Big[& R_1(s, a_1, \ldots, a_N) \\ & + \ldots + R_N(s, a_1, \ldots, a_N) \\ & + \gamma \sum_{s' \in S} P(s, a_1, \ldots, a_N, s') V^*(s') \Big] \end{aligned} \tag{9}$$

$$\begin{aligned} Q^*(s, a_1, \ldots, a_N) = & R_1(s, a_1, \ldots, a_N) + \ldots \\ & + R_N(s, a_1, \ldots, a_N) \\ & + \gamma \sum_{s' \in S} P(s, a_1, \ldots, a_N, s') \\ & \ldots \max_{a'_N \in A_N} Q^*(s', a'_1, \ldots, a'_N) \end{aligned} \tag{10}$$

The optimal joint policy $\pi^*(a_1|s), \ldots, \pi^*(a_N|s)$ can be obtained by selecting the actions that maximize the joint action-value function for each agent:

$$\pi_i^*(a_i|s) = \arg\max_{a_i \in A_i} Q^*(s, a_1, \ldots, a_N) \tag{11}$$

3) Federated Deep Reinforcement Learning: We consider a federated learning scenario with K clients denoted by subscript $i = 1, 2, \ldots, K$ and a central server. - Each client i has its local dataset D_i consisting of sequences of states, actions, rewards, and next states, given as $D_i = \{(s_{i,t}, a_{i,t}, r_{i,t}, s'_{i,t})\}_{t=1}^{T_i}$. The goal is to train a global policy p that can effectively detect and mitigate jamming attacks in O-RAN.

Each client i trains its local policy p_i using its local dataset D_i and deep reinforcement learning algorithms such as Deep Q-Network (DQN) or Proximal Policy Optimization (PPO). The local policy p_i is trained to maximize the expected cumulative rewards by iteratively updating its parameters θ_i based on the observed rewards. The update rule for the local policy parameters can be expressed as:

$$\begin{aligned} \theta_i \leftarrow \theta_i + \alpha \nabla_{\theta_i} \sum_{t=1}^{T_i} \Big(& r_{i,t} + \gamma \max_a Q_{\theta_i}(s'_{i,t}, a) \\ & - Q_{\theta_i}(s_{i,t}, a_{i,t}) \Big), \end{aligned} \tag{12}$$

where α is the learning rate, γ is the discount factor, ∇_{θ_i} represents the gradient of the local policy, $Q_{\theta_i}(s, a)$ is the action-value function parameterized by θ_i, and $a_{i,t}$ is the action taken by client i at time t.

Then, after local training, the central server aggregates the local policies to obtain the global policy p by averaging their parameters:

$$\theta = \frac{1}{K} \sum_{i=1}^{K} \theta_i, \tag{13}$$

where θ represents the parameters of the global policy.

Algorithm 1: Federated Deep Q-Learning for Jamming Attack Detection.

Initialize: global replay buffer $\mathcal{D}$
Initialize: global Q-networks with parameters $\theta_1, \theta_2, \ldots, \theta_N$ for each client
For : round $= 1$ to R
For : each client c in parallel
1: Receive global Q-network parameters θ_c from the server;
2: Initialize local replay buffer $\mathcal{D}_c$;
3: Initialize states $s_1, s_2, \ldots, s_{N_c}$;
While : $s_1, s_2, \ldots, s_{N_c}$ are not terminal
For : each agent $i = 1$ to N_c
4: Select action a_i for agent i using an exploration strategy (e.g., epsilon-greedy);
5: Execute joint action $\mathbf{a} = (a_1, a_2, \ldots, a_{N_c})$, observe rewards $\mathbf{r} = (r_1, r_2, \ldots, r_{N_c})$ and next states $\mathbf{s}' = (s'_1, s'_2, \ldots, s'_{N_c})$;
6: Compute total reward $R_{\text{total}}(S_t, A_t) = R(S_t, A_t) - P(S_t, A_t)$;
7: Store transition $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$ in local replay buffer $\mathcal{D}_c$;
8: Sample a mini-batch of experiences from $\mathcal{D}_c$;
For : each agent $i = 1$ to N_c
For : each experience $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$ in the mini-batch
9: Compute target Q-value: $Q^i_{\text{target}}(\mathbf{s}, \mathbf{a}) = r_i + \gamma \cdot \max_{\mathbf{a}'_i} Q(\mathbf{s}', \mathbf{a}'_i; \theta_c)$;
10: Compute predicted Q-value: $Q^i_{\text{predicted}}(\mathbf{s}, \mathbf{a}) = Q(\mathbf{s}, \mathbf{a}; \theta_c)$;
11: Compute loss: $L^i(\theta_c) = (Q^i_{\text{target}}(\mathbf{s}, \mathbf{a}) - Q^i_{\text{predicted}}(\mathbf{s}, \mathbf{a}))^2$;
12: Update local Q-network parameters: $\theta_c \leftarrow \theta_c - \alpha \cdot \nabla_{\theta_c} L^i(\theta_c)$;
13: Update states: $\mathbf{s} \leftarrow \mathbf{s}'$;
14: Send local Q-network parameters θ_c to the server;
15: Send local replay buffer $\mathcal{D}_c$ to the server;
16: Aggregate and update global Q-network parameters θ using federated averaging: $\theta \leftarrow \frac{1}{N} \sum_{c=1}^{N} \theta_c$;

Algorithm 1 summarizes the process of FMARL for jamming attack detection in O-RAN. Each client trains a local policy using its own data, and the central server aggregates the local policies by averaging their parameters to obtain a global policy. The global policy is then used to detect and mitigate jamming attacks in O-RAN.

4) Jamming Attack Mitigation: A Resource Blocks (RB) Allocation Problem: The global policy p is deployed in O-RAN to detect and mitigate jamming attacks. Each client i can use the global policy p to make decisions based on the observed states and take appropriate actions to counteract jamming attacks. We formalize the problem of jamming attack mitigation as a problem of RB Allocation in ORAN, where only trustworthy and reliable devices that have high reputation scores can be allowed to use the maximum allowed RBs. In the context of ORAN, RBs represent the available communication resources, and the goal is to allocate these RBs to reliable devices in a way that mitigates the impact of jamming attacks. Only trustworthy and reliable devices with high reputation scores are allowed to utilize the RBs.

The formulation for the RB allocation problem with reputation scores, penalty factors, weight factors, scaling factors, cost factors, load balancing, and additional constraints can be expressed as follows:

Objective:

$$\max \sum_{i=1}^{N} w_i \cdot C_i \cdot B_i$$

Subject to:

$$B_i \geq 0 \quad \forall i$$

$$\sum_{i=1}^{N} B_i \leq B_{\text{total}} \quad \text{(total available bandwidth constraint)}$$

$$B_i \leq B_{\max} \quad \forall i \quad \text{(maximum bandwidth constraint per device)}$$

$$B_i \leq w_i \cdot P_i \cdot S \cdot R_i \cdot B_{\text{total}} \quad \forall i \quad \text{(with penalty/scaling factors)}$$

$$B_i - B_j \leq B_{\text{diff}} \quad \forall i, j \quad \text{(load balancing constraint)}$$

$$\sum_{i=1}^{N} w_i = 1 \quad \text{(weight normalization constraint)}$$

$$R_i \geq R_{\min} \quad \forall i \quad \text{(minimum reputation score constraint)}$$

$$R_i \leq R_{\max} \quad \forall i \quad \text{(maximum reputation score constraint)}$$

$$B_i \geq B_{\min} \quad \forall i \quad \text{(minimum bandwidth allocation constraint)}$$

where C_i represents the cost factor for device i reflecting the cost implications of bandwidth usage; S is the scaling factor that adjusts the influence of reputation scores on bandwidth allocation; B_{diff} is the maximum allowed difference in bandwidth allocation among devices, promoting load balancing; B_{total} is the total available bandwidth; $B_{\max}$ represents the maximum bandwidth allocation per device; $R_{\min}$ and $R_{\max}$ are the minimum and maximum reputation scores, respectively; and $B_{\min}$ represents the minimum bandwidth allocation for each device.

The objective function aims to maximize the overall system performance considering the weighted combination of bandwidth allocation, reputation scores, and cost factors. This mechanism can effectively allocate resource blocks to devices based on their reputation scores, taking into account penalty factors, weight factors, scaling factors, cost factors, load balancing, and additional constraints. The iterative optimization process adjusts the resource block allocations until convergence, optimizing the system performance while considering reputation-based constraints and other considerations (see Algorithm 2).

Also, we introduce a scaling factor to adjust the overall impact of reputation scores on bandwidth allocation. Let S represent the scaling factor. By multiplying the reputation score of each device by S, we can control the influence of reputation scores in the equation. This factor allows for fine-tuning the balance between reputation-based allocation and other considerations.

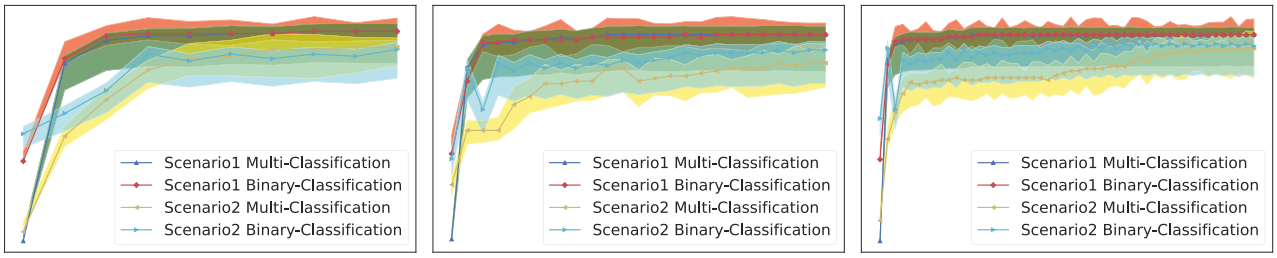

Fig. 2. Accuracy of our proposed framework for (a) 10 rounds; (b) 25 rounds; and (c) 50 rounds.

The reputation-based constraint with the scaling factor becomes:

$$\forall i \quad B_i \leq w_i \cdot P_i \cdot S \cdot R_i \cdot B_{\text{total}} \tag{14}$$

We also incorporate a cost factor associated with bandwidth allocation. Let C_i represent the cost factor for device i. By multiplying the bandwidth allocation of each device by its cost factor, we can account for the cost implications of bandwidth usage. The objective function with the cost factor becomes:

$$\max \sum_{i=1}^{N} w_i \cdot C_i \cdot B_i \quad \text{(objective function with cost factor)}$$

To promote load balancing, we can introduce a constraint that limits the difference in bandwidth allocation among devices. Let B_{diff} represent the maximum allowed difference in bandwidth allocation. The load balancing constraint can be defined as:

$$B_i - B_j \leq B_{\text{diff}} \quad \forall i, j \quad \text{(load balancing constraint)}$$

This constraint ensures that the difference in bandwidth allocation between any two devices does not exceed a specified threshold, promoting a more equitable distribution of resources.

By incorporating these additional factors and constraints, we can further tailor the equation to account for scaling factors, cost considerations, load balancing, and other relevant aspects of the IoT system, enabling a more comprehensive and fine-grained bandwidth allocation based on reputation scores.

IV. Implementation

In this section, we present the implementation and evaluation steps for our proposed framework. Initially, we assess the performance of the framework and then provide a comparison with some recent works.

To implement our proposed FRL framework, we utilize Pysyft [32], a privacy-centric deep learning library based on PyTorch. In our testing environment, multiple agents conduct local training, each specializing in a specific domain and utilizing a set of deployed wireless devices. To evaluate the feasibility and effectiveness of our framework, we employ the WSN-DS dataset [33] for intrusion detection in wireless sensor networks. In our study, we considered two scenarios (i.e., binary scenario and multi-class scenario). In the multi-class scenario, predictions are made from a set of multiple labels (i.e., Blackhole, Greyhole, Flooding, and Scheduling), whereas the binary scenario involves distinguishing between two labels (Normal and Attack). The

Algorithm 2: Resource Block Allocation in Open RAN Based on Reputation Scores and Deep Reinforcement Learning.

Input: Total available resource blocks (RB_{total}), maximum resource blocks per device ($RB_{\max}$), minimum resource block allocation ($RB_{\min}$), reputation scores based on past requests (R_i), penalty factors (P_i), weight factors (w_i), scaling factor (S), cost factors (C_i), maximum difference in resource block allocation (RB_{diff})

Output: Optimal resource block allocation (RB_i) for each Open RAN device

Normalize weight factors (w_i) to ensure $\sum_{i=1}^{N} w_i = 1$; Normalize reputation scores (R_i) between 0 and 1 ; Initialize resource block allocations (RB_i) for all devices as 0;

while *Resource block allocation not converged* **do**
  Update reputation scores based on past requests using deep reinforcement learning ;
  for *each device* i **do**
    Calculate reputation-based resource block allocation: $RB_i^{\text{rep}} = w_i \cdot P_i \cdot S \cdot R_i \cdot RB_{\text{total}}$;
    Calculate cost-based resource block allocation: $RB_i^{\text{cost}} = w_i \cdot C_i \cdot RB_i$;
    Set the device's resource block allocation: $RB_i = \min(RB_i^{\text{rep}}, RB_i^{\text{cost}}, RB_{\max})$;
    Enforce the minimum resource block allocation: $RB_i = \max(RB_i, RB_{\min})$;
  end
  Check load balancing and adjust resource block allocations if needed ;
end

TABLE I
WSN-DS Labels

Label	Multi-class	Binary
Normal	0	0
Blackhole	1	1
Flooding	2	1
Grayhole	3	1
Scheduling (TDMA)	4	1

TABLE II
Considered Scenarios

Scenarios	Type	Rounds	Local epochs
1	Binary	10,25,50	1-5
2	Multi-class	10,25,50	1-5

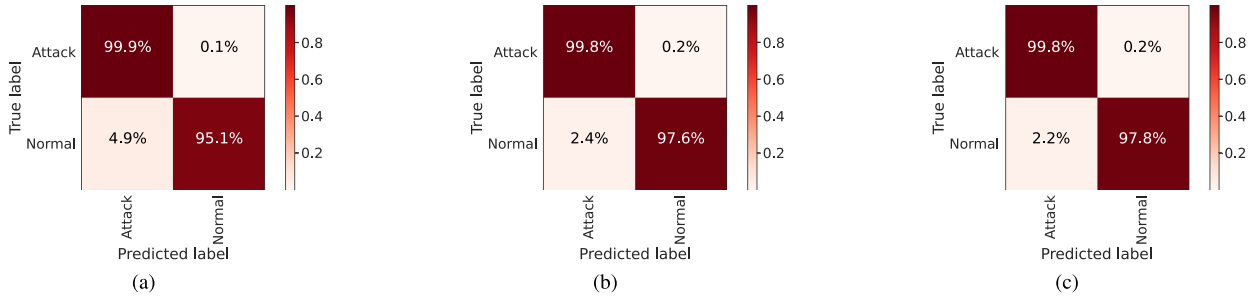

Fig. 3. Confusion matrices of our proposed framework for binary classification during (a) 10 rounds; (b) 25 rounds; and (c) 50 rounds.

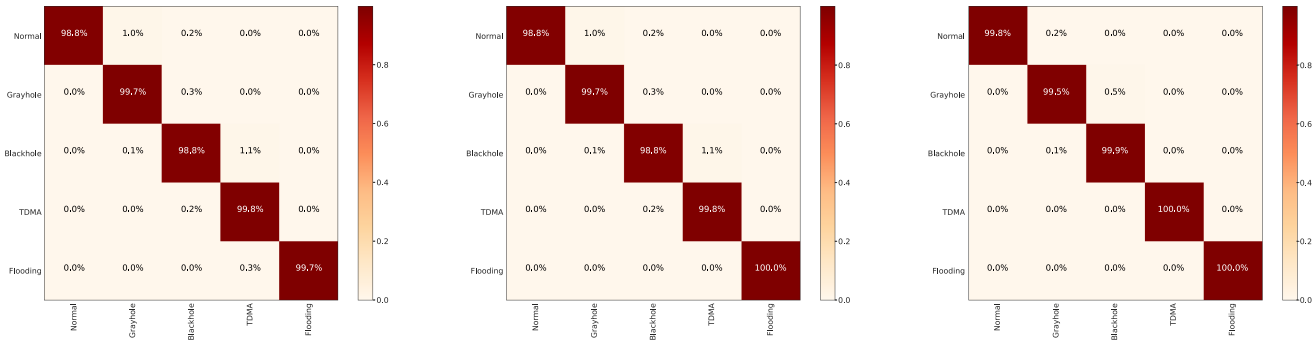

Fig. 4. Confusion matrices of our proposed framework for Multi-class classification for (a) 10 rounds; (b) 25 rounds; and (c) 50 rounds.

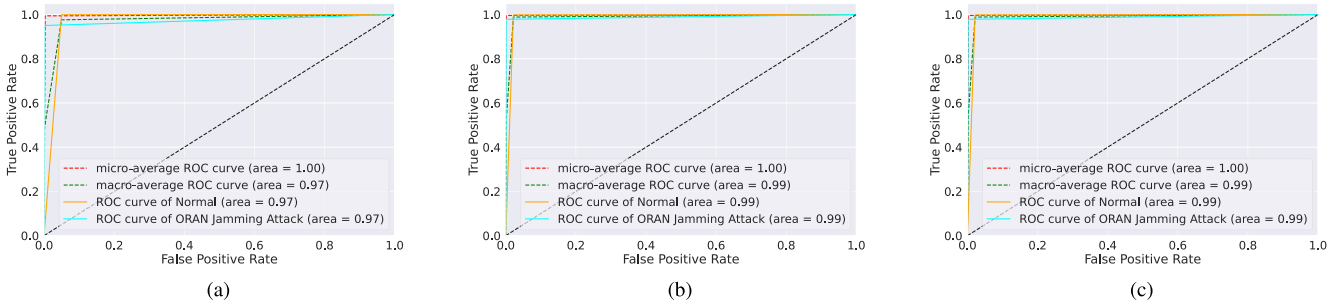

Fig. 5. ROC Curves of our proposed framework for Binary classification for (a) 10 rounds; (b) 25 rounds; and (c) 50 rounds.

main objective is to assess performance in diverse classification tasks, with a particular focus on addressing the challenge of non-independently and identically distributed (non-IID) data (see Table II). Considering the wide range of values in the input features of the WSN-DS dataset, we apply a standardization technique to normalize the feature values.

Fig. 2(a), (b), and (c) show the obtained accuracy of the trained model across the four Real-Time RICs during 10, 25, and 50 rounds of training, respectively. We achieve a high accuracy of 99% for both scenarios. This highlights the framework's high accuracy and privacy in detecting jamming attacks. Figs. 3 and 4 depict the confusion matrices for our proposed framework in binary classification and multi-class classification, respectively. In both scenarios, 99% of attack traffic was accurately classified as such, indicating that the proposed framework is well-trained and effectively detects jamming attacks.

A. *Performance Evaluation*

We evaluated our proposed framework using various metrics including, accuracy, precision, recall, F1-score, and AUC. The ROC curve illustrates the model's trade-off between sensitivity and specificity, while the confusion matrix provides a detailed summary of classification results. AUC measures binary classifier performance on a scale of 0 to 1, with higher values indicating better performance. Fig. 5 shows the ROC curves of our proposed framework for binary classification setup. A micro-average ROC curve is used when classes are imbalanced, while a macro-average ROC curve is useful for balanced classes. For 10, 25, and 50 rounds, we achieved high-performance scores of 99%, 99%, and 100% respectively.

Table III and Fig. 6 show the performance metrics of our proposed framework and AI benchmarks using the WSN-DS dataset, including some recent AI-based solutions (i.e.,

TABLE III
PERFORMANCE METRICS OF OUR PROPOSED FRAMEWORK AND BENCHMARKS

Methods	Accuracy	Precision	Recall	F1	Time (second)
NB	0.72	NA	NA	NA	NA
MLP	0.7	NA	NA	NA	NA
Adaboost	0.94	0.84	0.85	0.84	109
Gradient Boost (GB)	0.98	0.97	0.97	0.97	2880
Our Proposed Framework	**0.99**	**0.99**	**0.99**	**0.99**	**60**

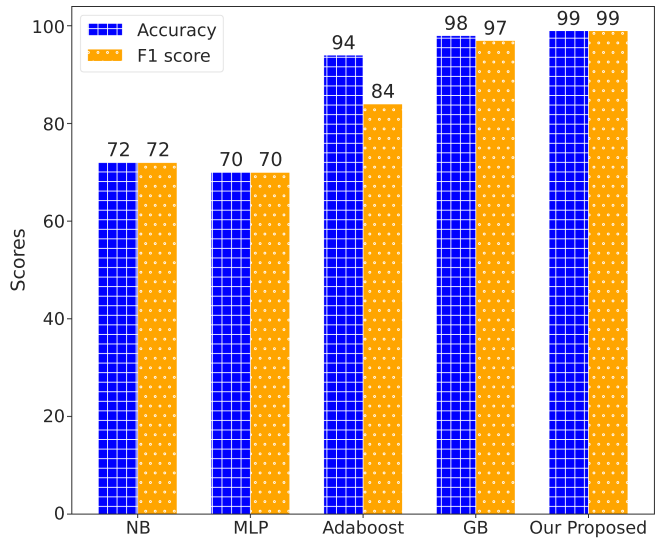

Fig. 6. Performance metrics of our proposed framework and benchmarks.

Naive Bayes (NB) [29], Multilayer Perceptron (MLP) [34], Adaboost [35], and Gradient Boost (GB) [35]) using the same WSN-DS dataset. our proposed framework achieves the highest performance on the WSN-DS dataset, achieving an accuracy, precision, recall, and F1 score of 99% each. Moreover, our proposed framework only requires 60 seconds of training time to achieve these results, which is relatively fast compared to other state-of-the-art approaches. These findings indicate that our proposed framework is an effective algorithm for intrusion detection in WSNs, and it outperforms other existing approaches in terms of accuracy and efficiency. This makes it a promising candidate for real-world applications where detecting intrusions in WSNs is critical.

V. CONCLUSION

This paper proposed a novel Federated Deep Reinforcement Learning framework for effective jamming attack mitigation in Open RAN. The proposed approach leverages the strengths of federated learning and deep reinforcement learning to overcome the challenges of limited data, slow convergence, and vulnerability to zero-day attacks faced by AI systems. First, we proposed a deep Q-learning model to enable adaptive decision-making and detect sophisticated jamming attack patterns. Second, we designed a distributed federated learning approach, where the proposed model is trained on distributed devices while preserving privacy. The locally trained models were then aggregated at a central server using federated averaging. We evaluated the proposed framework under different settings and proposed a resource block allocation algorithm to optimize system performance considering device reputation. Extensive simulations showed that the proposed framework achieves superior detection accuracy, efficient resource utilization, and high scalability.

REFERENCES

[1] M. Liyanage, A. Braeken, S. Shahabuddin, and P. Ranaweera, "Open RAN security: Challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 214, 2023, Art. no. 103621. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804523000401

[2] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 767–809, Secondquarter 2022.

[3] B. Brik, K. Boutiba, and A. Ksentini, "Deep learning for B5G open radio access network: Evolution, survey, case studies, and challenges," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 228–250, 2022.

[4] P. Zhou, Q. Yan, K. Wang, Z. Xu, S. Ji, and K. Bian, "Jamsa: A utility optimal contextual online learning framework for anti-jamming wireless scheduling under reactive jamming attack," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1862–1878, Jul.–Sep. 2020.

[5] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 48–59, Mar. 2018.

[6] L. Yang and C. Wen, "Optimal jamming attack system against remote state estimation in wireless network control systems," *IEEE Access*, vol. 9, pp. 51679–51688, 2021.

[7] A. Garnaev, Y. Liu, and W. Trappe, "Anti-jamming strategy versus a low-power jamming attack when intelligence of adversary's attack type is unknown," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 1, pp. 49–56, Mar. 2016.

[8] A. A. Elsaeidy, A. Jamalipour, and K. S. Munasinghe, "A hybrid deep learning approach for replay and DDoS attack detection in a smart city," *IEEE Access*, vol. 9, pp. 154864–154875, 2021.

[9] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," *IEEE Wireless Commun.*, vol. 14, no. 5, pp. 85–91, Oct. 2007.

[10] C. D. Alwis et al., "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 836–886, 2021.

[11] Z. A. El Houda, B. Brik, and L. Khoukhi, "Ensemble learning for intrusion detection in SDN-based zero touch smart grid systems," in *Proc. IEEE 47th Conf. Local Comput. Netw.*, 2022, pp. 149–156.

[12] B. Kannhavong, H. Nakayama, N. Kato, Y. Nemoto, and A. Jamalipour, "Analysis of the node isolation attack against OLSR-based mobile ad hoc networks," in *Proc. IEEE Int. Symp. Comput. Netw.*, 2006, pp. 30–35.

[13] G. Raja, N. D. Philips, R. K. Ramasamy, K. Dev, and N. Kumar, "Intelligent drones trajectory generation for mapping weed infested regions over 6G networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7506–7515, Jul. 2023.

[14] S. Anbalagan, G. Raja, S. Gurumoorthy, R. D. Suresh, and K. Dev, "IIDS: Intelligent intrusion detection system for sustainable development in autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15866–15875, Dec. 2023.

[15] D. Liu, F. Sun, W. Wang, and K. Dev, "Distributed computation offloading with low latency for artificial intelligence in vehicular networking," *IEEE Commun. Standards Mag.*, vol. 7, no. 1, pp. 74–80, Mar. 2023.

[16] Z. A. El Houda, H. Moudoud, B. Brik, and L. Khoukhi, "Securing federated learning through blockchain and explainable AI for robust intrusion detection in IoT networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2023, pp. 1–6.

[17] Z. A. El Houda, B. Brik, A. Ksentini, and L. Khoukhi, "A MEC-based architecture to secure IoT applications using federated deep learning," *IEEE Internet Things Mag.*, vol. 6, no. 1, pp. 60–63, Mar. 2023.

[18] D. Dik and M. S. Berger, "Open-RAN fronthaul transport security architecture and implementation," *IEEE Access*, vol. 11, pp. 46185–46203, 2023.

[19] R. Bitton et al., "Adversarial machine learning threat analysis in open radio access networks," 2022, *arXiv:2201.06093*.

[20] M. Mohammadi et al., "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021, Art. no. 102983.
[21] M. Nair, T. Cappello, S. Dang, and M. A. Beach, "RF fingerprinting of LoRa transmitters using machine learning with self-organizing maps for cyber intrusion detection," in *Proc. IEEE/MTT-S Int. Microw. Symp., IMS 2022*, 2022, pp. 491–494.
[22] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, Third Quarter 2019.
[23] S. Iannucci, V. Cardellini, O. D. Barba, and I. Banicescu, "A hybrid model-free approach for the near-optimal intrusion response control of non-stationary systems," *Future Gener. Comput. Syst.*, vol. 109, pp. 111–124, Aug. 2020.
[24] H. Alavizadeh, J. Jang-Jaccard, and H. Alavizadeh, "Deep Q-Learning based reinforcement learning approach for network intrusion detection," *Computers*, vol. 11, Nov. 2021, Art. no. 41.
[25] Z. A. E. Houda, A. S. Hafid, and L. Khoukhi, "MiTFed: A privacy preserving collaborative network attack mitigation framework based on federated learning using SDN and blockchain," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 4, pp. 1985–2001, Jul.–Aug. 2023.
[26] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–36, 2021.
[27] Y.-A. Xie et al., "Securing federated learning: A covert communication-based approach," *IEEE Netw.*, vol. 37, no. 1, pp. 118–124, Jan./Feb. 2023.
[28] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4298–4311, Apr. 2020.
[29] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.
[30] B. Yin, H. Yin, Y. Wu, and Z. Jiang, "FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6348–6359, Jul. 2020.
[31] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 911–926, 2020.
[32] "Pysyft," 2024. [Online]. Available: https://github.com/OpenMined/PySyft
[33] A. Iman, A.-K. Bassam, and A.-A. Mousa, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, 2016, Art. no. 4731953.
[34] S. Ismail and H. Reza, "Evaluation of naïve Bayesian algorithms for cyber-attacks detection in wireless sensor networks," in *Proc. IEEE World AI IoT Congr.*, 2022, pp. 283–289.
[35] M. Nouman, U. Qasim, H. Nasir, A. Almasoud, M. Imran, and N. Javaid, "Malicious node detection using machine learning and distributed data storage using blockchain in WSNs," *IEEE Access*, vol. 11, pp. 6106–6121, 2023.

Zakaria Abou El Houda (Member, IEEE) received the Ph.D. degree in computer science from the University of Montreal, Montreal, QC, Canada, in 2021, and the second Ph.D. degree in computer engineering from the University of Technology of Troyes, Troyes, France, in 2021. He is currently a Professor with the Energy, Materials, and Telecommunications Center, National Institute of Scientific Research (INRS), Québec City, QC, Canada. He is also a Member of the INRS-UQO Joint Research Unit in Cybersecurity. Prior to joining INRS, he was a research scientist in various institutions, contributing to significant research projects on the application of machine learning for intrusion detection systems, and studying the explainability and robustness of these systems. His research interests include applied AI for intrusion detection systems, security in distributed/federated machine learning, and Blockchain for network security.

Hajar Moudoud (Member, IEEE) received the B.Eng. degree in software engineering from the Mohammadia School of Engineers, Rabat, Morocco, in 2018, the Ph.D. degree in computer engineering from the University of Sherbrooke, Sherbrooke, QC, Canada, in 2022, and the second Ph.D. degree in computer engineering from the University of Technology of Troyes, Troyes, France, in 2022. Her research interests include the security of the Internet of Things, applied machine/deep learning for intrusion detection systems, and leveraging blockchain to enhance the security of next-generation networks (5G and beyond/6G).

Bouziane Brik (Senior Member, IEEE) received the Engineer degree (Ranked First) in computer science and the M.Sc. and Ph.D. degrees from Laghouat University, Laghouat, Algeria, in 2010, 2013, and 2017, respectively. He is currently an Assistant Professor with the Computer Science Department, Computing and Informatics College, Sharjah University, Sharjah, UAE. Before joining Sharjah University, he was an Assistant Professor with the DRIVE Department, Bourgogne University, Dijon, France. He was also a Postdoc with the CESI School, University of Troyes, Troyes, France, and Eurecom Research Institute, France. He has been working on resources management and security challenges of 5G network slicing in the context of H2020 European projects, including MonB5G, 5GDrones, InDiD, and 5G-INSIGHT. His research interests include 5G and beyond networks, explainable AI, and machine/deep learning for wireless networks. He is an active member of many conferences' organizing committees, such as Globecom, WCNC, ICC, GIIS, EAI, and EAI CICom. He actively organized different special issues in prestigious journals and conference workshops.