IEEE ComSoc IEEE Communications Society
IEEE Open Journal of the
Communications Society

Received 6 July 2023; revised 24 August 2023; accepted 31 August 2023. Date of publication 8 September 2023; date of current version 20 October 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3313352

Empowering Security and Trust in 5G and Beyond: A Deep Reinforcement Learning Approach

HAJAR MOUDOUD (Member, IEEE), AND SOUMAYA CHERKAOUI (Senior Member, IEEE)

Department of Computer and Software Engineering, Polytechnique Montréal, Montreal, QC H3T 1J4, Canada

CORRESPONDING AUTHOR: H. MOUDOUD (e-mail: hajar.moudoud@usherbrooke.ca)

ABSTRACT Recent advances in 5G and beyond have further expanded the potential of IoT applications, bringing unprecedented levels of connectivity, speed, and low latency. However, these advances come with significant security threats that can cause widespread damage. An effective approach to addressing these issues involves the integration of cutting-edge technologies like machine learning (ML), particularly deep reinforcement learning (DRL). DRL is a specialized area of ML that integrates the concepts of deep learning and reinforcement learning to create effective solutions for various tasks. In particular, DRL can facilitate the creation of intelligent security systems that can adapt to dynamic and intricate IoT applications connected to 5G and beyond networks. However, effectively implementing DRL-based intrusion detection frameworks in IoT applications connected to 5G networks poses significant challenges due to bandwidth utilization and device behavior. The data generated by IoT devices is often limited, and malicious behavior may be infrequent, making it difficult to accurately identify and train the algorithm to detect such behavior. Moreover, DRL algorithms pose a significant challenge for IoT devices constrained by limited bandwidth, as communicating large amounts of data required by DRL algorithms can cause network congestion and delay critical communications. In this article, we introduce a novel approach to improving the security of IoT applications in the 5G and beyond era by developing an intrusion detection system that employs DRL algorithms. Our approach involves a distributed Q-learning algorithm that observes the behavior of connected devices and predicts anomalous actions. Additionally, to overcome the challenges associated with bandwidth utilization and device behavior, we introduce a bandwidth allocation problem based on a reputation mechanism that allocates bandwidth to only trustworthy devices. Finally, we evaluate our proposed intrusion detection system on the selected indicators. The numerical results demonstrate that our proposed approach outperforms the referenced solutions on the selected indicators.

INDEX TERMS 5G and beyond, intrusion detection system (IDS), deep reinforcement learning (DRL), Internet of Things (IoT).

I. INTRODUCTION

THE EMERGENCE of 5G and beyond networks has accelerated the growth of IoT technology, facilitating the connection of an even greater number of devices to the Internet. According to estimates, billions of these devices will be connected by the end of 2025, taking advantage of the high speed, low latency, and wide connectivity provided by 5G networks [1]. However, the deployment of IoT networks is a complex process that can increase the risk of security vulnerabilities and give rise to sophisticated and complex security threats [2], [3]. These threats have the capacity to inflict substantial harm and jeopardize the reliability of IoT networks. Additionally, due to the vast interconnectivity and abundance of devices within these networks, they are particularly susceptible to security breaches. Device interconnectivity can create significant vulnerabilities, as a single malicious device can infect the entire network, causing battery drainage or denial of service-of-service attacks [4].

Maintaining the integrity of IoT services depends on the capability to detect and mitigate security attacks. To achieve this, it is crucial to develop advanced security mechanisms capable of effectively addressing the challenges presented by 5G and beyond networks. This ensures the secure and reliable operation of IoT systems. A commonly used approach to secure IoT networks involves implementing an Intrusion Detection System (IDS) that utilizes machine learning (ML) models to identify and respond to such threats [5]. For example, an IDS can create a profile that includes the behavior of various components, including IoT devices, servers, network traffic, and other relevant parameters. By analyzing IoT activity, events can be categorized as intrusions or anomalies using binary classification methods, akin to traditional classification problems. This approach enables the recognition of potential security breaches and anomalous behavior in IoT systems. Classical ML models employed in IDS for intrusion detection in IoT networks might have limitations in detecting new threats and zero-day attacks, as they rely on predefined rules and patterns. Consequently, these models might lack the flexibility to adapt to emerging security threats in real-time, emphasizing the necessity for more adaptive and dynamic approaches like reinforcement learning.

Reinforcement learning (RL) has the potential to be an effective ML technique for detecting and classifying security attacks. By interacting with the environment, receiving feedback, and employing a reward strategy, RL agents can enhance their decision-making processes. In contrast to classical learning methods like supervised and unsupervised learning, RL doesn't require expert knowledge and has shown potential in identifying security threats within IoT networks. Recently, several intrusion detection methods based on RL have emerged, providing an efficient solution for recognizing security threats in IoT networks [6], [7], [8], [9]. However, designing an effective reward function can prove to be a challenging and time-consuming task. Furthermore, traditional RL algorithms may not be equipped to handle the intricacies and high dimensionality of the intrusion detection problem.

To address the limitations of traditional intrusion detection systems, deep reinforcement learning (DRL) methods have emerged as a viable solution. These techniques offer the capability to effectively handle complex features and significantly enhance the overall accuracy of intrusion detection [10]. DRL harnesses the power of deep neural networks (DNNs) to adeptly tackle the challenges associated with detecting intrusions in intricate and dynamic IoT networks. This approach provides a robust means to analyze and comprehend the intricate patterns and features present in such environments, leading to improved accuracy and resilience in intrusion detection. Furthermore, DRL excels at managing the vast and unmanageable state spaces often encountered in IoT networks. It achieves this by employing deep Q-learning (DQL) and other function approximation techniques that leverage deep neural networks. By utilizing DRL, the issue of state explosion, which classic RL approaches encounter, can be overcome. Moreover, DRL has the capacity to learn from both labeled and unlabeled data, rendering it well-suited for identifying novel threats and zero-day attacks.

Establishing trust between RL agents and the devices engaged in the learning process, encompassing intrusion detection, remains a challenge. The establishment of trust is fundamental for cultivating a resilient and efficient detection mechanism, which assumes a pivotal role in guaranteeing the security of upcoming networks. Therefore, the incorporation of a distributed reputation mechanism into intrusion detection systems rooted in DRL becomes indispensable. This integration not only heightens the precision of the intrusion detection system but also bolsters the comprehensive security and dependability of the IoT network by fostering trustworthiness.

In the intricate landscape of IoT interconnections, where devices span a spectrum of trustworthiness, evaluating device reputation becomes a pivotal factor for robust intrusion detection. DRL-based intrusion detection systems are based on the collective contribution of various devices to refine their detection capabilities. Nevertheless, some devices might fall victim to compromise or harbor nefarious intentions, thus introducing skewed or deceptive data into the system. By assessing the reputation of each participant in the learning process, the system can assign greater significance to data from proven sources while discounting input from dubious origins. The integration of a distributed reputation mechanism into DRL-based intrusion detection systems marks a pivotal stride towards fortifying the prospects of future IoT networks. This integration not only amplifies the precision of intrusion detection but also increases the overall security and dependability of the IoT network.

The higher levels of connectivity, speed, and low latency provided by 5G networks have not only facilitated a variety of novel IoT applications, but have also introduced fresh security threats, capable of rapid propagation and substantial damage. Ensuring secure communication among myriad devices within the IoT ecosystem requires optimization of resource utilization while addressing specific challenges inherent to the IoT, such as restricted bandwidth and device reliability. Consequently, the judicious selection of appropriate devices and the efficient allocation of resources become pivotal factors in realizing holistic security solutions for IoT systems.

This paper introduces a novel approach to detecting intrusions in IoT networks based on a DQL algorithm. The core of our approach involves the construction of a simulated environment, augmented by a record sampling function. This function empowers agents to continuously refine their capacity to observe and predict abnormal behavior within IoT networks. Using this methodology, we can facilitate more effective and efficient intrusion detection in IoT networks. This approach permits the creation of a self-learning system that can adapt and evolve over time, leading to more precise and efficient detection of anomalies within IoT networks. Moreover, we propose a reputation assessment mechanism aimed at identifying and neutralizing malicious devices displaying erratic behavior that could trigger intrusions. Additionally, we present a scheduling method that meticulously assigns the required bandwidth to chosen reliable devices, ensuring optimal resource allocation. This algorithm strives to minimize communication among DRL agents while prioritizing devices with trustworthy behavior, thus fostering expedited and dependable communication between devices and agents. This paper presents several notable contributions, which can be summarized as follows:

- We design a novel IDS for IoT applications connected to 5G and beyond networks. The proposed IDS utilizes DRL algorithms and capitalizes on the advanced features of 5G networks. Our approach uses a decentralized Q-learning algorithm to observe and anticipate abnormal actions exhibited by IoT devices.
- We present a reputation management approach that evaluates the trustworthiness of IoT devices by analyzing their interactions and behavior.
- We propose a resource allocation algorithm that considers the trustworthiness of devices to optimize bandwidth utilization in IoT networks. By employing these approaches, we can enhance the efficiency and security of IoT systems.
- We evaluate our proposed framework in terms of accuracy, F1-score, precision, and detection rate. The NSL-KDD dataset, which encompasses well-known attacks, was utilized to conduct comprehensive experiments aimed at evaluating the effectiveness and reliability of our system in thoroughly detecting these threats.

The remainder of this paper is organized as follows. In Section II, we provide an overview of related works. Section III establishes the foundational background for the reinforcement learning (RL) approach, while Section IV outlines the system model that we examine. Our proposed approach for assessing device reputations is presented in Section V, along with a bandwidth allocation problem that we formulate and solve. The numerical results of our experiments are presented in Section VI. Finally, we conclude the paper in Section VII.

II. RELATED WORK

In this section, we provide a summary of the pertinent literature concerning IDS and explore the application of machine learning to create an intelligent IDS.

Within the realm of safeguarding IoT systems from sophisticated security threats, IDS plays an essential role in furnishing an added layer of protection. Detecting threats in IoT systems can prove challenging and time-consuming, underscoring the significance of devising a resilient IDS [11]. Numerous research studies have explored various ML techniques for intrusion detection, including support vector machines (SVM) and self-organizing maps (SOM) [12], [13]. In a distinct investigation, Yin et al. [14] introduced recurrent neural networks (RNNs) as a deep learning approach for intrusion detection. They evaluated the efficacy of RNNs in both binary and multiclass classification scenarios, leveraging their findings to frame intrusion detection as a classification problem.

To detect and mitigate network threats, Hossain et al. [15] proposed an IDS based on Long Short-Term Memory (LSTM). Similarly, for identifying intrusion attempts, Javed et al. [16] suggested integrating a Convolutional Neural Network (CNN) with an attention-based Gated Recurrent Unit (GRU) model. Marteau [17] recommended employing the Random Partitioning Forest (RF) for collective anomaly detection in IDS. Additionally, Lan et al. [18] introduced a supervised machine learning approach that incorporates the CNN-Bidirectional LSTM (CNN-BiLSTM) algorithm alongside a threshold adjustment mechanism. Furthermore, Heartfield et al. [19] evaluated the application of unsupervised learning and Reinforcement Learning (RL) techniques for system anomaly detection in IoT devices. The proposed approach centers on monitoring shifts in device behavior to tailor the decision function of underlying anomaly categorization models. However, the authors did not consider how model hyperparameters and Q parameters would evolve as the learner advanced within the proposed method. Particularly in the context of large-scale IoT, the Q parameter could be employed to tackle the challenge of state explosion.

To enhance the performance of IDS, RL approaches have been employed across multiple systems. These methodologies are well-suited for the IoT environment, as they empower systems to adjust their actions based on continuous feedback to maximize rewards. Several studies employ Deep Reinforcement Learning (DRL) algorithms to conduct intrusion detection in either real or simulated environments [14], [20], [21]. Iannucci et al. [14] introduced an intrusion response DRL approach rooted in a stateful Markov Decision Process, which redefines the landscape in terms of system scale and attack scope. Alavizadeh et al. [22] explored the utilization of DQL, a fusion of RL and a deep forward neural network, for network intrusion detection. Their strategy harnesses automated trial and error and ongoing learning to effectively detect diverse intrusion types, thus bolstering detection capabilities. However, while RL has demonstrated its effectiveness in detecting intrusions within IoT networks, prior research has tended to overlook the potential impact of malicious devices and agent reliability on intrusion detection accuracy. The presence of a malicious RL agent within a network can result in network saturation or the creation of a single point of failure, potentially compromising the reliability and precision of intrusion detection.

III. PRELIMINARIES

This section briefly covers the fundamentals of DRL. Subsequently, we provide a comprehensive overview of the dataset employed in our study.

A. OVERVIEW OF REINFORCEMENT LEARNING

A Markov Decision Process (MDP) serves as a mathematical framework employed to model decision-making processes within scenarios encompassing both partial randomness and agent-controlled factors. This framework involves discrete-time stochastic control, where an agent engages with its environment by selecting a viable action, accounting for the initial state of the environment. Subsequently, the environment undergoes state transitions according to a probability distribution, and the agent receives a reward contingent upon the effectiveness of its chosen action.

RL constitutes a subfield of ML that employs the MDP framework to acquire optimal decision-making policies through iterative interactions with the environment. In situations where rewards or transition probabilities are uncertain, RL leverages the MDP to characterize the environment. The primary objective of an RL agent is to uncover an optimal policy that establishes a mapping from states to actions, thereby enabling it to select the most suitable action based on its present state.

RL algorithms employ iterative processes to enhance the decision-making aptitude of the agent over time. The agent engages in exploratory actions within the environment, garnering feedback in the form of rewards, and employs this input to refine its policy. Via repetitive interactions, the agent progressively discerns actions that are prone to yield substantial rewards, ultimately converging towards an optimal policy that maximizes its anticipated cumulative reward across time.

A Markov Decision Process (MDP) constitutes a mathematical model defined by a tuple (S, A, P, r), in which S signifies the set of possible states, A represents the set of potential actions, P symbolizes the transition probability function, and r embodies the reward function. In contrast, a Markov chain refers to a probabilistic model capturing the sequence of events or states, wherein the transition probability between states relies solely on the current state and does not consider previous states. This is referred to as the Markov property, signifying that future states are contingent solely upon the present state and not the past ones. The probability of transitioning from state s_t to state s_{t+1}, given the state sequence $s_1, s_2, \ldots, s_t$, can be denoted as $P(s_{t+1}|s_1, \ldots, s_t)$. Alternatively, this probability can be expressed as the likelihood of state s_{t+1} being equal to j, given that the current state s_t is equal to i, represented as $P(s_{t+1} = j|s_t = i)$. This conditional probability is denoted by $p_{i,j}$, where $p_{i,j}$ signifies the probability of transitioning from state i to state j.

The transition probability matrix $\mathbb{T}$ is employed to depict the probabilities associated with all feasible transitions among states. The transition probability $p_{i,j}$ adheres to the condition that the total probabilities for all potential transitions from state i sum up to 1. The transition matrix $\mathbb{T}$ takes the form of a square matrix with dimensions $n \times n$, where each entry $T_{i,j}$ signifies the probability of transitioning from state i to state j. The matrix rows correspond to the current state, while the columns pertain to potential subsequent states. The probability of transitioning from state i to state j is represented by $T_{i,j} = p_{i,j}$. The stipulation that the sum of probabilities for all conceivable transitions from state i equates to 1 can be expressed as:

$$\mathbb{T} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix} \tag{1}$$

Within the realm of RL, the association between states (s) and actions (a) is forged through the application of a policy function denoted as π. This policy function holds the responsibility of establishing the connection between states and actions, thus steering the decision-making process of an RL agent. Fundamentally, the policy function π functions as a mechanism that maps each state s to an associated action, thereby governing the behavior of the agent within a specific state:

$$\pi(s|a) = P[A_t = a|S_t = s], \tag{2}$$

Similarly, when an agent encounters a specific state s, it evaluates the efficacy of its chosen action a by employing a function referred to as the state value function. Denoted as $v_\pi(s)$, the state value function quantifies the expected cumulative reward attainable to the agent over a temporal span, originating from state s and adhering to its policy π:

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \alpha^k R_{t+k+1}|S_t = s\right], \forall s \in S, \tag{3}$$

In equation (3), R_{t+k+1} symbolizes the reward value at a future instance, specifically at time $t + k + 1$. These rewards are discounted utilizing a factor denoted as $0 < \alpha < 1$. The discount factor γ holds significant importance in steering the decision-making process of the RL agent. It plays a crucial role in aiding the agent to make informed decisions and determine the most favorable course of action. Its role is to strike a balance between the significance of future rewards and the immediate rewards gained from the current state s. In the domain of RL, the agent strives to maximize its cumulative reward across time. However, given the uncertainty of future rewards, the agent must factor in the potentiality of receiving lower or higher rewards in the future, as illustrated below:

$$G = \max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t R_t|S_t = s\right], \tag{4}$$

Q-learning stands as a widely adopted RL technique, focusing on a function called the Q-function to gauge the merits of a state-action pair. The chief objective of Q-learning involves ascertaining the optimal policy within a provided MDP by learning the optimal Q-function. This achievement is realized through the use of value iteration updates and the Bellman equation. The Q-function is instrumental in calculating the anticipated cumulative reward attainable to an agent through the execution of a specific action a in a given state s, followed by adherence to the prevailing policy. With each iteration, the Q-function undergoes updating to approximate the optimal Q-function with greater precision. This, in turn, leads to better decision-making on the part of the agent:

$$Q(s, a) = \sum_{s'} P_a\big[s, s'\big]\big(R\big(s, s', a\big) + \alpha v\big(s'\big)\big), \tag{5}$$

In Q-learning, the Q function is updated through the utilization of value iteration updates, which take into account several components. These include the transition probability $P_a[s, s']$ from state s to state s', the reward $R(s, s', a)$ acquired from the transition, and the prevailing value function of state s', indicated as $\nu(s)$. The equation encapsulating the value iteration update employed in Q-learning can be formulated as follows:

$$\begin{aligned} Q^{\text{new}}(s_t, a_t) &\leftarrow Q(s_t, a_t) \\ &+ \alpha\left(R_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right), \end{aligned} \tag{6}$$

In this equation, the updated value $Q^{\text{new}}(s_t, a_t)$ for the state-action pair (s_t, a_t) is obtained by adding the learning rate α multiplied by the temporal difference error. The temporal difference error is computed as the disparity between the immediate reward R_t and the discounted maximum expected future reward $\gamma \max_a Q(s_{t+1}, a)$, subtracted from the present value $Q(s_t, a_t)$. The discount factor γ determines the significance of future rewards compared to immediate ones.

For evaluating the anticipated reward concerning a specific state-action pair, the Q-learning algorithm employs a Q-table housing Q-values. Nonetheless, this strategy proves unsuitable for IoT contexts, where numerous agents might be engaged, given the substantial memory demands associated with Q-table storage.

In tackling this challenge, the realm of DRL has emerged as a powerful methodology. It leverages function approximation and representation learning to glean valuable insights from compressed, lower-dimensional data. This technique empowers agents to glean knowledge and generalize proficiently within intricate, high-dimensional environments, without imposing an excessive burden on memory resources.

B. OVERVIEW OF DEEP REINFORCEMENT LEARNING

Conversely, DRL can overcome the constraints of Q-learning by harnessing deep neural networks to approximate the Q-function, thereby enabling adept management of high-dimensional state spaces. This methodology is recognized as Deep Q-Networks (DQN), seamlessly melding Q-learning with deep neural networks to empower agents in acquiring knowledge from sensory inputs characterized by high dimensions.

Additionally, DRL algorithms like DQN can also incorporate alternative exploration policies, such as softmax action selection or bootstrapped ensembles, to navigate the environment with heightened efficiency, all the while maintaining a delicate equilibrium between exploration and exploitation. This capacity accelerates the agent learning process and enhances his performance within IoT applications.

In Q-learning, a table housing state-action pairs is crafted to ascertain the optimal action corresponding to a given state. To navigate the realm of potential rewards, Q-learning commonly employs the ϵ-greedy approach as an exploration policy. This involves randomly selecting an action with a probability of ϵ. However, constructing a Q-table and pinpointing the optimal policy can be computationally taxing, and the state space may not be exhaustively explored, leading to infrequent visits to certain states.

Deep Q-learning (DQL) represents a ML technique that advances upon the Q-learning approach by employing neural networks to approximate the Q-function in lieu of a Q-table. This methodology proves particularly efficacious for resolving problems characterized by high-dimensional state spaces and action spaces, in situations where maintaining a table for every conceivable state-action pair becomes impractical. Within the framework of DQL, the neural network takes the state as input and generates the estimated Q-values for each action. Subsequently, the agent selects the action associated with the highest Q-value, an action anticipated to yield the maximum cumulative reward. Through neural networks, DQL imbibes the ability to learn from an abundance of state-action pairs and generalize to unfamiliar scenarios. This quality makes it a potent technique for addressing intricate reinforcement learning conundrums.

C. NSL-KDD DATASET DESCRIPTION

This NSL-KDD dataset encompasses 41 attributes, denoted as states, categorized as either reliable or linked to specific attack types (classes) [23]. To render the NSL-KDD dataset compatible with DRL, which only operates with numerical or floating-point values, we implemented the one-hot encoding technique for dataset pre-processing. The NSL-KDD dataset encompasses four attack types: Denial of Service (DoS), Probe, Root to Local (R2L), and Unauthorized to Root (U2R). The binary attributes were converted into numerical values via the one-hot encoding technique, which enabled their utilization within the DRL algorithm.

IV. SYSTEM MODEL

In the context of cutting-edge wireless communication technologies like 5G and beyond, we investigate an IoT environment comprising devices capable of both honest and malicious behavior. Our system model is designed to detect intrusions in 5G and subsequent wireless networks (see Fig. 1). Within this model, wireless devices establish connections with the network edge either through edge servers or base stations, both of which function as DRL agents. These agents gather data and employ the DRL model detailed in subsequent sections to formulate decisions. A DRL agent, a subtype of RL agent, can train the model to anticipate forthcoming rewards and recognize potential intrusions or attacks. The IoT node, functioning as an agent—such as a base station or server equipped with adequate computational capabilities—interacts with its surroundings by observing and making decisions through actions.

The process of intrusion detection involves two levels. Firstly, a distributed trust management system is established to meticulously select trustworthy devices for network communication. The trustworthiness of each device is assessed through reputation evaluation. Dedicated nodes, like servers or base stations, gather transmitted data and make determinations using the subsequently developed DRL model.

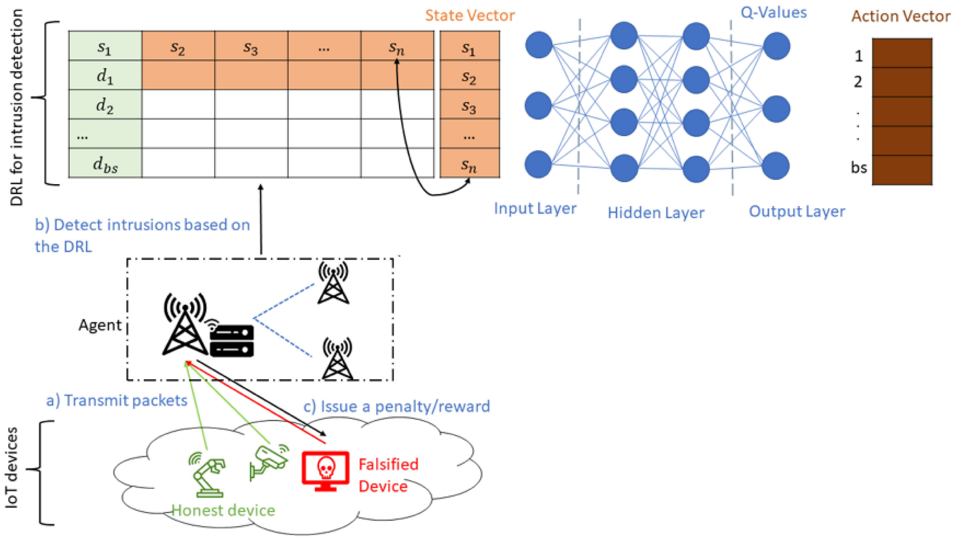


FIGURE 1. System model demonstrating DQN's Q-value output and action calculation via argmax Q for current state in a DRL intrusion detection system employed in 5G and Beyond IoT Applications.

An RL agent, referred to as a DRL agent, can train the model to anticipate forthcoming rewards and identify potential intrusions or attacks. The DRL agents exclusively engage in communication with the most trustworthy devices that satisfy the necessary bandwidth requirements.

Secondly, we propose the adoption of a DRL approach to detect intrusions in the network. During the training phase, the agent actively explores the available action space by employing an exploration policy, such as the ϵ-greedy strategy. This policy empowers the agent to make decisions: either choosing a random action with a probability of ϵ or selecting the action with the highest value function using the greedy approach with a probability of $1-\epsilon$. The DQN algorithm, which extends traditional RL techniques, estimates the Q-value by considering a generalized state-action pair with the function $Q(s, a)$. In our system, the DQN agent employs a DNN approximation to calculate the Q-value, utilizing both the environment's states and actions, with a specific emphasis on leveraging the NSL-KDD dataset. Given the dataset's extensive array of features and the batch size employed during the DQN process, storing Q-values for every state-action pair in a Q-table becomes impractical. Therefore, we propose the adoption of the DQL process for intrusion detection, which employs neural networks to approximate the Q-value of each state-action pair. Further elaboration on our proposed DQL methodology can be found in Section V-C.

V. DEEP REINFORCEMENT LEARNING FOR SECURING IOT NETWORKS IN THE 5G AND BEYOND ERA

In this section, we will delve into the details of our proposed IDS designed to enhance the security of 5G and beyond networks. The implementation of our IDS involves a multistep process. We begin by outlining a decentralized approach to establishing trust among IoT devices. Secondly, we employ resource allocation strategies to make informed decisions regarding IoT device selection.

A. DEVICE REPUTATION EVALUATION

We propose a method for evaluating a device's Reputation Score (RS) based on its previous interactions with other devices, which offers several advantages. Firstly, it incentivizes desirable device behavior by establishing a trust management system that rewards normal behavior (NB) while penalizing abnormal behavior (AB). Secondly, it mitigates the influence of malicious devices by allocating bandwidth resources exclusively to trustworthy devices with high RS. The RS is initially assigned a positive value to each device, which can increase or decrease based on its behavior. This score is used to assess the device's trustworthiness and determine its eligibility for resource allocation. To calculate the average reputation, the RS of all devices is summed and divided by the total number of devices. This calculation provides a measure of the overall trustworthiness of the devices in the network, as shown below:

$$RS_{avg} = \frac{\sum_{j=1}^{n} RO_{jk}}{n}, \tag{7}$$

where RO represents the reputation of a device j as perceived by a device k based on their past interactions. The RS can therefore be defined for each device as follows, in accordance with the type of behavior it exhibits:

$$RS = \frac{\sum_{k=1}^{n} RO_{jk}\left(1 - |RS_{avg} - RO_{jk}|\right)}{\sum_{k=1}^{m} 1 - |RS_{avg} - RO_{jk}|}, \tag{8}$$

The evaluation process assesses the reliability of a recommendation made by a device k regarding a device j by measuring the deviation from the average recommendation value. In response to their atypical behavior, devices demonstrating abnormal actions experienced a reduction in their reputation scores, resulting in the following update equation:

$$RS^{\text{new}} = \beta\left(1 - \frac{ND}{NT}\right)R^{\text{old}}, \tag{9}$$

where β falls within the range of [0,1]. The weight assigned to each device's reputation score depends on the context of the IoT environment and is calculated using the frequency of device connections (ND) and the total number of device interactions (NT) over a given period.

B. RESOURCE ALLOCATION PROBLEM

In response to the resource constraints often faced in IoT contexts, our approach revolves around the targeted allocation of bandwidth exclusively to devices with proven trustworthiness and reliability. This strategy not only optimizes learning outcomes but also minimizes potential delays. To illustrate, let's consider a scenario where there exists a total bandwidth of B Hz within an Orthogonal Frequency Division Multiple Access (OFDMA) system. In this setup, each individual device, denoted as k, is allotted a specific share of the overall available bandwidth, denoted as $\alpha_k \in [0, 1]$. This division of bandwidth influences the achievable rate of communication for device k when interacting with the base station (BS). The relationship between the device and the BS is characterized by the channel gain, represented as g_k.

Given the paramount importance of meticulous criterion selection and the inherent challenges stemming from communication limitations, we formulate the following problem:

$$
\begin{aligned}
\underset{x,\alpha}{\text{minimize}} \quad & \sum_{k=1}^{m} x_k RS_k && \text{(10a)}\\
\text{subject to} \quad & \sum_{i} \alpha_i \leq 1 && \text{(10b)}\\
& 0 \leq \alpha_k \leq 1, \qquad \forall k \in [1, n] && \text{(10c)}\\
& x_{i,j} t_j \leq T && \text{(10d)}\\
& \sum_{j=1}^{n} RS_j \geq R_{min}, \qquad \forall k \in [1, n] && \text{(10e)}\\
& x_k \in \{0, 1\}, \qquad \forall k \in [1, n]. && \text{(10f)}
\end{aligned}
$$

The objective of the problem (10a) is to allocate bandwidth solely to devices exhibiting high reputation scores, indicative of their reliability. Constraints (10b) and (10c) enforce that the allocated bandwidth fractions remain within the interval of 0 to 1, with their cumulative sum abiding by the total bandwidth budget. Constraint (10d) ensures that the chosen devices successfully complete their model training and data uploading within the designated blockchain deadline of T. Constraints (10e) guarantee that the selected reliable devices for bandwidth allocation meet or exceed the specified minimum requirement of R_{min}. Lastly, constraints (10f) define the optimization variables.

The solution to problem (10a) presents a challenge due to its equivalence to the NP-hard knapsack problem. The problem's objective is to maximize the count of trustworthy devices while adhering to bandwidth limitations, where device trustworthiness is determined by reputation scores. The constraints ensure that the chosen devices accomplish their tasks within the deadline and that the total sum of allocated bandwidth fractions remains within the bandwidth budget. To surmount this complexity, we introduce Algorithm 1, a classical greedy knapsack algorithm, as a proposed solution.

Algorithm 1 Bandwidth Allocation Algorithm

```
Input: A queue of K IoT devices waiting for bandwidth allocation B;
Output: Array α, Allocated bandwidth for the devices x = [x_1, ..., x_K];
Order devices according to their reputation parameter (R) decreasingly and index them from 1 to K;
for k = 1, ..., K do
    x_k ← 0;
end for
A ← B;
k ← 1;
while A ≥ 0 and k ≤ K do
    if A − α_k B ≥ 0 then
        x_k ← 1;
        A ← A − α_k B;
        α_k ← α_k;
        k ← k + 1;
    else
        α_k ← A/B;
        A ← 0;
        x_k ← 1;
    end if
end while return x and α
```

C. DRL FORMULATION

We frame the intrusion detection problem as an MDP and leverage DRL techniques to tackle it within the IoT context. The MDP is structured around three primary components, outlined as follows:

1) The state space serves as the information source for the agent to base its decisions upon. In the context of intrusion detection, we make use of the NSL-KDD dataset, where each column represents a distinct feature. As a result, we define the state s_i for our DNN as F_i, where F_i corresponds to a feature within the dataset F.
2) The collection of choices available to an agent, determined by the information provided by the environment, is referred to as the action space. In the pursuit of identifying potential intrusions or attacks, the agent compiles a roster of plausible actions within a defined time window. Actions are chosen by the agent by considering a predetermined-sized mini-batch, and the output of the DNN is juxtaposed with the Q-value to ascertain the occurrence of an attack.
3) The reward function embodies the input from the environment concerning the action executed by the agent. Its design is geared towards motivating the agent to accurately identify attacks, assigning a reward of +1 for correct detection and −1 for misjudgment. The reward value is additionally refined using the

classifier's prediction probability and the acquired Q-values, all in pursuit of enhancing the overall classifier performance.

The system model employed for training the DQN in the context of intrusion detection encompasses several key components and sequential stages. Its objective is to construct a dependable model with the competence to accurately identify intrusions within a given environment.

The primary step involves generating essential parameters for the DNN, encompassing the network architecture, behaviors of IoT devices, and packet sizes. These parameters are instrumental in shaping the DQN's capacity for learning and making informed decisions.

The training procedure unfolds across numerous episodes, each delineating a series of interactions between the agent and its environment. At the inception of each episode, the initial state is updated to establish the foundational context for the agent's engagement with the surroundings.

Within each episode, the training algorithm enters an internal loop comprising individual steps. This algorithm operates with two distinct strategies for action selection: exploitation and exploration. Exploitation is favored with a likelihood of $(1 - \epsilon)$, wherein the algorithm predicts the current state and calculates the corresponding action vector based on the acquired policy. Exploration is employed with a likelihood of ϵ, enabling the algorithm to select a batch size and determine an action vector that fosters exploration within the environment.

Following action selection, the algorithm proceeds to compute the target Q-value, which encapsulates the projection of future rewards. Subsequently, the reward function is computed based on the prevailing state and the chosen action. In tandem, the algorithm evaluates the discrepancy between the predicted Q-value and the target Q-value, facilitating the DQN's assimilation of these differences to enhance its predictive prowess.

The agent's parameters, encompassing the weights and biases of the DNN, undergo updates via backpropagation, a mechanism that employs the computed loss. This iterative refinement process aims to elevate the DQN's performance and bolster its efficacy in detecting intrusions.

VI. PERFORMANCE EVALUATION

In this section, we present a comprehensive overview of the numerical results obtained through our simulations. We begin by outlining the simulation setup and its corresponding configuration. Subsequently, we proceed to assess the performance of our proposed system model and contrast it with prior research outcomes. Lastly, we delve into an analysis of the influence of malicious behavior on the overall performance of the system.

A. EXPERIMENT SETUP

To conduct our simulations, we utilized the NSL-KDD datasets and employed a min-max normalization technique to normalize both the training and test datasets within the range of [0, 1].

Our proposed approach for intrusion detection in IoT devices is based on a four-layer DRL architecture utilizing relu activation. The input layer, situated at the top, comprises neurons capturing environmental variables. The final layer, which represents the Q-values for each category of intrusion/attack, constitutes the output. The two intermediary layers serve as hidden layers that contribute to the training process.

For our network simulation, we considered a 500-square-meter area centered around a base station. This network simulation resembled both mobile and cellular networks, accommodating a total of m devices uniformly distributed across the square. Such network configurations are commonly encountered in applications within smart cities where wireless connectivity is established.

Furthermore, we employed two fundamental metrics to gauge the model's performance: detection accuracy and F1 score. Detection accuracy quantifies the percentage of accurately identified attacks, while the F1-score provides a balanced measure by combining precision and recall.

The following formulas are applied to calculate the detection accuracy and F1 score, where TP represents "true positives" (intrusions correctly identified as intrusions), FN denotes "false negatives" (intrusions misclassified as reliable device behaviors), FP signifies "false positives" (reliable device behaviors incorrectly labeled as intrusions), and TN corresponds to "true negatives" (reliable device behaviors correctly identified as such).

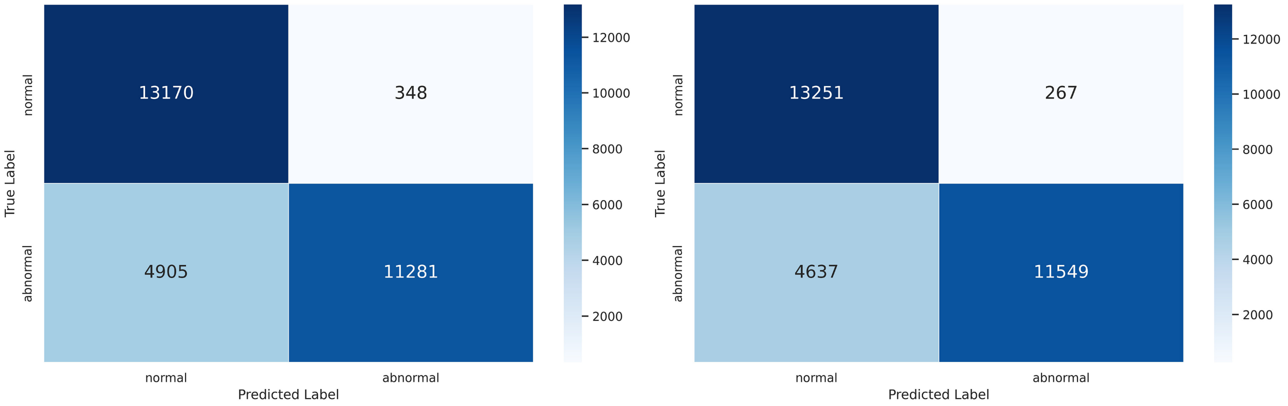
B. EVALUATION RESULTS

The confusion matrices for our DQL model after 25 and 50 training epochs are illustrated in Fig. 2(a) and Fig. 2(b), respectively. These matrices provide insights into the model's performance by depicting its ability to distinguish between intrusions and consistent device behaviors accurately.

Following 25 training iterations, our DQL model exhibited a commendable intrusion detection rate of $11,281$, effectively identifying instances of unauthorized access. Furthermore, it displayed a high precision of $13,170$ in correctly identifying the behavior of the regular and reliable devices. However, there were occurrences of 348 instances where reliable device behaviors were incorrectly classified as intrusions and $4,905$ instances where intrusions were incorrectly labeled as reliable device behaviors.

Fig. 3(a) and Fig. 3(b) display the Receiver Operating Characteristic (ROC) curves of our DQL model. These curves visually represent the trade-off between the true positive rate (sensitivity) and the false positive rate. The area under the ROC curve offers valuable insight into the effectiveness of our DQL model in distinguishing between dependable and undependable device behaviors.

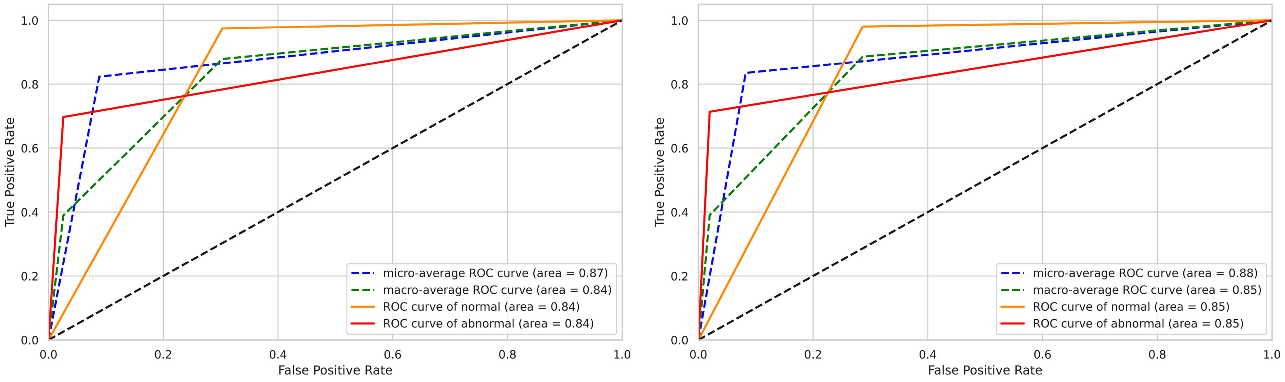
For the 25-epoch training, our DQL model achieves an AUC of 0.87, affirming its competence in accurately categorizing device behaviors. Similarly, with the 50-epoch



(a)

(b)

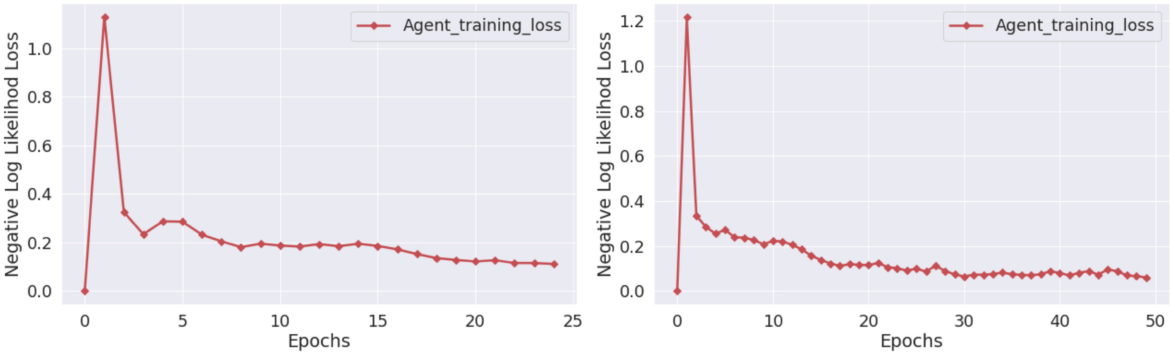
FIGURE 2. **Confusion matrices on $NSL-KDDTest^{+}$ for: (a) 25 epochs case; and (b) 50 epochs case.**



(a)

(b)

FIGURE 3. **ROC curves on $NSL-KDDTest^{+}$ for: (a) 25 epochs case; and (b) 50 epochs case.**



(a)

(b)

FIGURE 4. **Model loss on $NSL-KDDTest^{+}$ for: (a) 25 epochs case; and (b) 50 epochs case.**

training, the AUC further increases to 0.88, underscoring the enhanced performance of the model in distinguishing between dependable and undependable behaviors.

In Fig. 4(a) and Fig. 4(b) we showcase the model loss of our DQL model over 25 and 50 epochs, respectively. The trend is evident—loss values consistently decrease as epochs advance, eventually converging to nearly zero during the 50-epoch training duration. This trend signifies the model's proficiency in effectively learning from its environment.

Fig. 5(a) and Fig. 5(b) present the agent's rewards. It is particularly notable that the cumulative rewards achieved by the agent exhibit a consistent increase with each epoch, culminating in their peak towards the conclusion of the learning phase. This trend robustly underscores our DQL model's capacity for learning (referred to as the Agent), as it continually adapts to its surroundings and garners greater rewards over time.

We compare our proposed system model with state-of-the-art. The comparison study encompassed the following

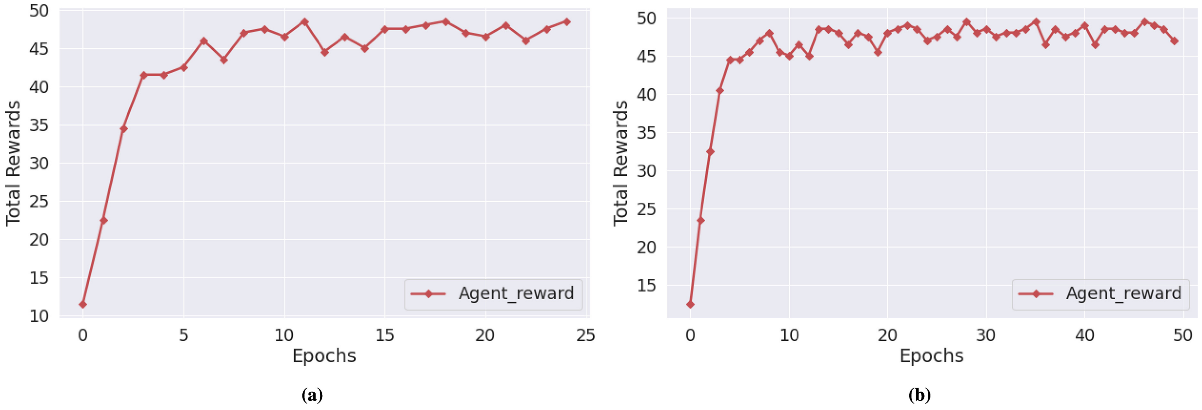


FIGURE 5. Agent reward on *NSL – KDDTest*$^+$ for: (a) 25 epochs case; and (b) 50 epochs case.

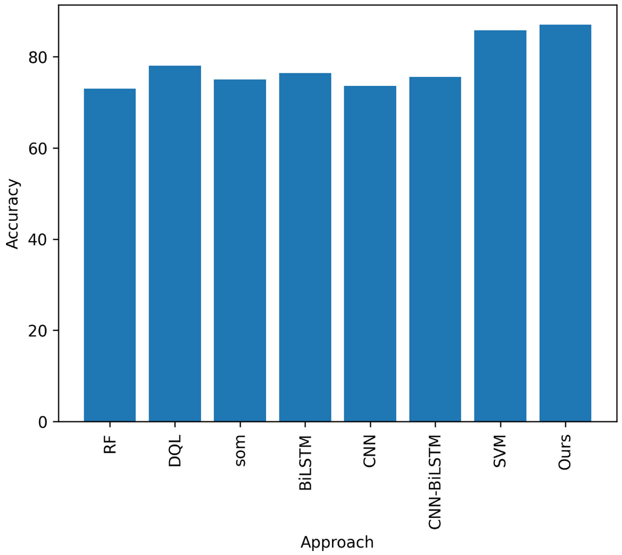


FIGURE 6. Comparison of accuracy among various Machine Learning models using NSL-KDD Dataset.

models: Random Forest (RF) and DQL models [22], Self Organization Map (SOM) [13], Bidirectional LSTM (BiLSTM) [24], Convolutional Neural Network (CNN) [16], CNN-BiLSTM [18], and Support Vector Machines (SVM) [12]. These models were evaluated to gauge their performance and efficacy within the specified context. Fig. 6 graphically represents the assessment of model performance based on accuracy using the NSL-KDD dataset, encompassing both the aforementioned machine learning models and our proposed DQL model. Additionally, Fig. 7 shows the evaluated models including: Random Forest [17], DQL [22], Self Organization Map [13], Bidirectional LSTM [24], CNN [16], CNN-BiLSTM [18], Support Vector Machines [12], and our proposed model. Fig. 7 offers valuable comparative insight into the performance of various models, effectively determining the efficacy of our proposed model in relation to contemporary approaches. Notably, the DQL approach, encompassing both our model and the one mentioned in [22], consistently outperforms other methodologies on the NSL-KDD dataset. Impressively, these DQL-based models achieved F1-scores of 80.84% and 94%, respectively.

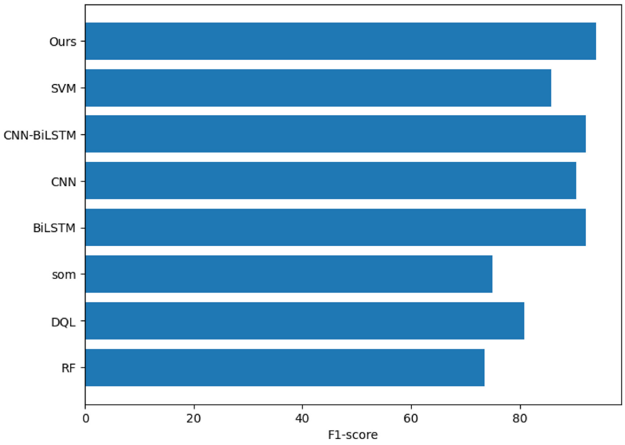


FIGURE 7. Comparison of F1-scores for system and state-of-the-art models using NSL-KDD database.

VII. CONCLUSION

In this paper, we introduced a novel approach to detect intrusions in IoT networks by leveraging a distributed Q-learning algorithm. Our approach aimed to enable agents to continually learn and enhance their ability to detect normal and anomalous IoT behaviors. To facilitate this, we devised a pseudo-environment and record sampling mechanism, culminating in the creation of a self-learning system that could adapt and improve over time, leading to more precise and efficient anomaly detection. Additionally, we proposed a distributed reputation assessment mechanism to effectively identify and neutralize malicious devices. To optimize bandwidth utilization while considering device reputation, we presented an iterative algorithm involving device selection and subsequent bandwidth allocation. Lastly, we empirically evaluated the effectiveness of our approach through experimentation and rigorous comparison with existing methods. Our results demonstrated that our proposed approach surpassed state-of-the-art techniques in terms of both accuracy and efficiency.

However, there are still numerous unresolved matters that warrant thorough examination within this domain. Primarily, one significant aspect involves exploring more advanced Q-learning techniques that adapt to changing environments and dynamic behaviors. This could involve the incorporation of techniques like deep reinforcement learning. Additionally, exploring the applicability of transfer learning techniques in the context of intrusion detection can be a valuable avenue for future research. Transfer learning allows models trained on one task or domain to be leveraged for improved performance on a related but different task or domain.

REFERENCES

[1] A. Rachedi, M. H. Rehmani, S. Cherkaoui, and J. J. P. C. Rodrigues, "IEEE access special section editorial: The plethora of research in Internet of Things (IoT)," *IEEE Access*, vol. 4, pp. 9575–9579, 2016.

[2] H. Moudoud, L. Khoukhi, and S. Cherkaoui, "Prediction and detection of FDIA and DDoS attacks in 5G enabled IoT," *IEEE Netw.*, vol. 35, no. 2, pp. 194–201, Mar./Apr. 2021.

[3] 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. (LCN)*, 2022, pp. 149–156.

[4] G. Raja, A. Ganapathisubramaniyan, G. Anand, and Gowshika, "Intrusion detector for blockchain based IoT networks," in *Proc. 10th Int. Conf. Adv. Comput. (ICoAC)*, 2018, pp. 328–332.

[5] Z. A. E. Houda, B. Brik, and L. Khoukhi, ""Why should I trust your IDS?": An explainable deep learning framework for intrusion detection systems in Internet of Things networks," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1164–1176, 2022.

[6] H. Alavizadeh, J. Jang-Jaccard, and H. Alavizadeh, "Deep Q-learning based reinforcement learning approach for network intrusion detection," 2021, *arXiv:2111.13978*.

[7] K. Sethi, R. Kumar, D. Mohanty, and P. Bera, "Robust adaptive cloud intrusion detection system using advanced deep reinforcement learning," in *Proc. Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng.*, 2020, pp. 66–85.

[8] Z. A. El Houda, B. Brik, and S.-M. Senouci, "A novel IoT-based explainable deep learning framework for intrusion detection systems," *IEEE Internet Things Mag.*, vol. 5, no. 2, pp. 20–23, Jun. 2022.

[9] 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.

[10] H. Moudoud, Z. Mlika, L. Khoukhi, and S. Cherkaoui, "Detection and prediction of FDI attacks in IoT systems via hidden Markov model," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 2978–2990, Sep./Oct. 2022.

[11] Z. A. El Houda, L. Khoukhi, and A. S. Hafid, "Bringing intelligence to software defined networks: Mitigating DDoS attacks," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2523–2535, Dec. 2020.

[12] 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.

[13] 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.*, Jun. 2022, pp. 491–494.

[14] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[15] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, vol. 8, pp. 185489–185502, 2020.

[16] A. R. Javed, S. U. Rehman, M. U. Khan, M. Alazab, and T. Reddy G, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1456–1466, Apr.-Jun. 2021.

[17] P.-F. Marteau, "Random partitioning forest for point-wise and collective anomaly detection—Application to network intrusion detection," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2157–2172, 2021.

[18] M. Lan, J. Luo, S. Chai, R. Chai, C. Zhang, and B. Zhang, "A novel industrial intrusion detection method based on threshold-optimized CNN-BiLSTM-attention using ROC curve," in *Proc. 39th Chin. Control Conf. (CCC)*, Jul. 2020, pp. 7384–7389.

[19] R. Heartfield, G. Loukas, A. Bezemskij, and E. Panaousis, "Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1720–1735, 2021.

[20] 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, 3rd Quart., 2019.

[21] L. Nie et al., "Intrusion detection in green Internet of Things: A deep deterministic policy gradient-based algorithm," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 778–788, Jun. 2021.

[22] H. Alavizadeh, J. Jang-Jaccard, and H. Alavizadeh, "Deep Q-learning based reinforcement learning approach for network intrusion detection," Nov. 2021, *arXiv:2111.13978*.

[23] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on NSL-KDD dataset," *IEEE Access*, vol. 9, pp. 140136–140146, 2021.

[24] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, Jun. 2021.

HAJAR MOUDOUD (Member, IEEE) received the B.Eng. degree in software engineering from the Mohammadia School of Engineers, Rabat, Morocco, in 2018, the first Ph.D. degree in computer engineering from the University of Sherbrooke, Canada, in 2022, and the second Ph.D. degree in computer engineering from the University of Technology of Troyes, France, in 2022. Her research interests include security of Internet of Things, applied machine/deep learning for intrusion detection system, and integration with blockchain.

SOUMAYA CHERKAOUI (Senior Member, IEEE) is a Full Professor with the Department of Computer and Software Engineering, Polytechnique Montréal. She has authored numerous conference and journal papers. Her work was awarded with recognitions, including several best paper awards at IEEE conferences. She has been on the editorial boards of several IEEE journals. She is an IEEE ComSoc Distinguished Lecturer and a Professional Engineer in Canada and has served as the Chair of the IEEE ComSoc IoT-Ad Hoc and Sensor Networks Technical Committee.