Enhancing intrusion detection in next-generation networks based on a multi-agent game-theoretic framework

Sai Krishna Lakshminarayana, Prabhugoud I. Basarkod
School of Electronics and Communication Engineering, REVA University, Bangalore, India

Article Info

Article history:

Received Jan 25, 2024
Revised Mar 19, 2024
Accepted Mar 21, 2024

Keywords:

Game theory
Internet of things
Intrusion detection system
Multi-agent
Network security
Next generation networks
Reinforcement learning

ABSTRACT

With cyber threats becoming increasingly sophisticated, existing intrusion detection systems (IDS) in next generation networks (NGNs) are subjected to more false-positives and struggles to offer robust security feature, highlighting a critical need for more adaptive and reliable threat detection mechanisms. This research introduces a novel IDS that leverages a dueling deep Q-network (DQN) a reinforcement learning algorithm within game-theoretic framework simulating a multi-agent adversarial learning scenario to address these challenges. By employing a customized OpenAI Gym environment for realistic threat simulation and advanced dueling DQN mechanisms for reduced overestimation bias, the proposed scheme significantly enhances the adaptability and accuracy of intrusion detection. Comparative analysis against current state-of-the-art methods reveals that the proposed system achieves superior performance, with accuracy and F1-score improvements to 95.02% and 94.68%, respectively. These results highlight the potential scope of the proposed adaptive IDS to provide a robust defense against the dynamic threat landscape in NGNs.

Corresponding Author:

Sai Krishna Lakshminarayana
School of Electronics and Communication Engineering, REVA University
Bengaluru, India
Email: sklnarayana@gmail.com

1. INTRODUCTION

The communication network landscape has witnessed a transformative shift from static, circuit-switched models to dynamic, data-centric architectures. Although traditional networks have established global connections, they faced challenges such as limited bandwidth, and scalability issues [1]. The advent of wireless sensor networks (WSNs) brought enhanced data gathering capabilities, but has been hampered by limited processing power and energy constraints [2]. The internet of things (IoT) era further changed these network paradigms, consisting vast amounts of smart devices that gathers and generate data, often disregarding security measures [3]. As a result, the security scenario has become increasingly vulnerable to advanced cyber threats [4]. In the near future, the IoT ecosystem will continue to evolve and can be regarded as called next generation networks (NGN) providing greater flexibility, and adaptability by integrating various network elements such as software-defined networking (SDN), network virtualization, cloud integration, and artificial intelligence (AI)-powered algorithms [5]. However, NGNs also brings its own vulnerabilities, as the threat landscape in NGNs is not only limited to the vast connected devices, but also includes sophisticated attack strategies. The distributed aspect of NGNs opens multiple points for attacks, and the incorporation of diverse devices presents new exploitation risks. Advanced threats like botnets, zero-day exploits, and targeted attacks present serious challenges, risking critical infrastructure disruption, data breaches, and significant economic impacts [6].

In the context of network security, traditional approaches have primarily focused on bounded defense mechanisms, such as firewalls, antivirus software, and encryption [7], [8]. These solutions, while effective in a more controlled network environment, often fall short in addressing the dynamic nature of NGNs. The traditional methods tend to be reactive rather than proactive, addressing threats only after they have breached the network. Encryption, while vital for data protection, does not protect against active intrusions targeting network infrastructure [9]. The conventional approaches are increasingly inadequate in the face of advanced persistent threats (APTs) and other sophisticated attacks that can evade standard detection mechanisms. Intrusion detection systems (IDS) have emerged as a critical component in safeguarding wireless networks. Unlike traditional security measures, IDS offer a more dynamic and proactive approach [10]. They are designed to detect and respond to unusual or suspicious activities within the network, providing an additional layer of security that can adapt to the evolving landscape of cyber threats. The advantage of IDS in NGNs is that it can continuously monitor network traffic and system activities and identify potential threats in real time. However, traditional security solutions, including traditional IDS, often struggle to keep up with the rapid development of NGN. These systems often rely on predefined rules or signatures to detect threats, an approach that becomes less effective as attack patterns evolve and become more complex [11], [12]. Additionally, the large amounts of data generated by NGNs may overwhelm traditional security methods, resulting in high false alarm rates and missed detections.

In the recent state of art work, the research trends have seen a shift towards leveraging machine learning (ML) and deep learning (DL) techniques in network security. For example, Su *et al.* [13] tackled traditional IDS limitations like low accuracy and manual feature engineering dependency by integrating DL long short-term memory (LSTM) model with an attention mechanism, enabling automatic key feature learning. Similarly, Sumadi *et al.* [14] proposed an approach to combat distributed denial of service (DDoS) attacks, combining honeypot sensors with SDN and employing semi-supervised learning with support vector machine (SVM) and adaptive boosting for attack classification. Addressing DDoS threats further, Mohammed *et al.* [15] explored the use of two neural network models with distinct configurations. Tackling the challenge of class imbalance in IDS development, Martin *et al.* [16] utilized a custom radial basis function (RBF) neural network. Saikam and Koteswararao [17] also addressed class imbalance by merging deep networks with hybrid sampling, employing generative adversarial networks (GAN), DenseNet169 for spatial feature extraction, and self attention for temporal aspects. They further applied a spike neural network for classification. Another approach by Aljehane *et al.* [18] incorporated the golden jackal optimization algorithm within an LSTM architecture for automated feature selection and optimal classification. The adoption of ML and DL approaches in IDS offer significant improvements over traditional rule-based IDSs, particularly in their ability to learn and adapt to new threats, uncovering patterns and anomalies that may indicate a security breach. However, most of these methods adopts supervised learning, highly rely on labeled data and are also prone to class imbalance problems. Dependency on extensive, well-labeled datasets can be a significant bottleneck, while imbalanced classes often introduces biases in the learning process.

Among the various AI methodologies, reinforcement learning (RL) has shown considerable promise in revolutionizing IDS for NGN. RL differs from other ML and DL approaches in its ability to learn optimal actions through trial and error interactions with a dynamic environment [19]. RL-driven IDSs are particularly well-suited to the ever-changing landscape of network security, where the system must continuously adapt to new threats. In past five years significant research works have been done in the context of RL driven IDS. Dong *et al.* [20] explored abnormal traffic detection in network security, using an autoencoder and double deep Q-network (DQN) for traffic feature reconstruction and classification, alongside K-means clustering is applied for target network training. They demonstrate effectiveness using network security lab-knowledge discovery in databases (NSL-KDD) and Aegean Wi-Fi intrusion dataset (AWID) datasets but overlook model complexity for resource-constrained devices. Lu *et al.* [21] introduce a deep self-encoding model with missing measurement weights and a unique oversampling algorithm for enhanced attack detection, though scalability remains a concern. Li *et al.* [22] address unbalanced datasets and minority attack identification through adversarial environment learning and a soft actor-critic RL algorithm, focusing on data resampling and tailored reward values for specific attacks. Ren *et al.* [23] enhance IDS efficacy by utilizing recursive feature elimination (RFE) and deep Q-learning for feature selection, though usage of RFE may omit crucial features for complex attack identification. Yu *et al.* [24] combine a DQN with a variational auto-encoder for traffic classification and unknown attack recognition in industrial IoT. The work in similar direction can be also seen by Sethi *et al.* [25]. They critique existing IDSs for inaccuracy and performance issues against unseen attacks, proposes a solution using DQN-based distributed agents with attention mechanisms and a denoising autoencoder for robustness. However, these scheme may struggle with the computational complexity introduced by the integration of DQN and auto-encoder. Benaddi *et al.* [26] employed stochastic game theory and Markov decision processes (MDP) to model the interaction between IDS and attackers. He *et al.* [27] introduces a transformative approach combining deep RL with strategies to prioritize outliers without classifying the entire dataset to prioritize outliers without classifying the entire dataset transferability and adaptability with fewer samples for intrusion detection. Feng *et al.* [28] proposed a collaborative DDoS detection method using a soft actor-critic learning model, featuring a collaborative aggregation module and a unique reward mechanism. Soltani *et al.* [29] suggest adapting DL models to changing traffic behaviors using federated learning and sequential packet labeling. Alavizadeh *et al.* [30] introduce an IDS method combining Q-learning with a deep feed-forward neural network, featuring a DQN model for auto-learning. However, this method may face challenges in real-time adaptation to rapidly evolving attacks and maintaining computational efficiency in dynamic network environments. The literature review reveals that researchers have proposed various schemes using RL-based IDS to counter sophisticated cyber attacks. However, despite many efforts, a significant research gap persists in effectively applying RL to IDS in NGN. The followings are the highlights of the significant research problem.

- Overestimation in DQN: existing approaches frequently employed the DQN algorithm, which struggles with the state explosion problem and tends to overestimate Q-values. This overestimation, stemming from the max operation in Q-learning updates, can lead to suboptimal policy learning.
- Challenges in double DQN implementation: although some studies have adopted the double DQN algorithm to mitigate DQN's overestimation bias, fine-tuning its hyperparameters for effective self-learning and interaction with network environments remains a substantial challenge.
- Focus on detection performance: many IDS designs based on double DQN concentrate primarily on enhancing detection capabilities compared to Q-learning and DQN, without adequately differentiating the value of specific states from the value of actions within those states.
- Design optimization: there is a lack of comprehensive studies on optimizing double DQN to more accurately evaluate state values and improve learning efficiency. Most current works do not provide detailed optimization strategies or dependent on certain assumptions about network characteristics.
- Simulation-based environments and their limitations: the use of simulation-based approaches, such as Monte Carlo simulations, in many RL-based IDS models does not offer a realistic representation of network ecosystems. A need for more realistic, benchmarked environments, possibly using OpenAI Gym, is identified for effective agent training and evaluation.

Therefore, this paper proposes an advanced IDS framework to address the above-mentioned research gaps by utilizing a multi-agent RL approach and combined with an adaptive learning scheme leveraging game-theoretic approach. This framework is designed to detect and respond to cyber threats in complex network environments like NGNs. The key contribution of the proposed work is as follows:

- This paper presents a adversarial model through development of multi-agent algorithm utilizing a game-theoretic approach, making defender agent dynamic to continuously learn and adapt its strategies against evolving attack patterns by attacker agent, thereby enhancing the robustness and effectiveness of the IDS.
- Unlike existing works, the proposed framework adopts dueling DQN mechanism in multi-agent algorithm to overcome problem of state overestimation bias and optimizing the learning process by efficiently evaluating state values and action advantages. This approach is particularly beneficial in the dynamic environment of NGNs, ensuring more effective decision-making process.
- The proposed work also presents a customized environment leveraging the functionalities of the OpenAI Gym, a benchmark tool for simulating the complexities and dynamic nature of NGNs. The proposed custom environment represents realistic network traffic patterns and behaviors, enhancing the agents' exploration and decision-making capabilities, and leading to more effective detection strategies.

The novelty of the proposed research work is the introduction of multi-agent in adversarial learning setup, adoption of dueling DQN for enhanced adaptability in NGNs. It uniquely combines realistic adversarial modeling and a customized OpenAI Gym environment as an effective IDS in dynamic and responsive network security against evolving cyber threats.

2. METHOD

This research introduces a novel IDS design for NGNs that leverages the dynamic and adaptive capabilities of multi-agent RL. By integrating a game-theoretic framework, the proposed IDS can simulate realistic adversarial scenarios, where defender and attacker agents continuously evolve their strategies. The objective is to create an IDS that not only effectively detects cyber threats but also adapts and evolves in response to emerging threats, providing a significant enhancement over traditional IDS solutions.

2.1. Reinforcement learning

RL is a subtype of ML where an agent model interacts with an environment, observes states, and takes actions to maximize long-term rewards. The agent is the decision-making entity, while the environment regarded as task scenario mimicking problem, which provides the states and rewards that inform the agent's decisions. Figure 1 illustrates the typical architecture of this agent-environment interaction.

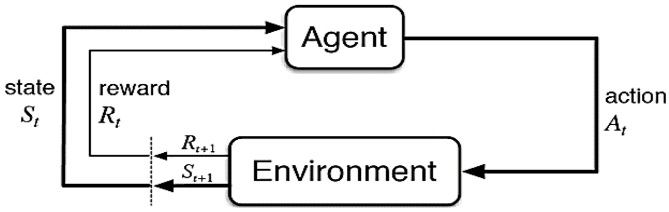

Figure 1. Typical architecture of RL

Figure 1 depicts the fundamental interaction cycle of RL, where an agent makes decisions by taking actions (A_t) in response to the environment's states (S_t) to maximize cumulative rewards (R_t). The environment post interaction evaluates the actions being taken by agent and updates the agent with new states (S_{t+1}) and corresponding rewards (R_{t+1}), facilitating a sequential decision-making problem and continuous learning process. The core operation of RL operates under the mathematical framework of MDP where the decision-making actions have stochastic consequences and feedbacks are received over time, depending on the current state and the chosen action.

The mathematical framework of MDP is characterized by tuple $M = \{S, A, P, R, \gamma\}$, where S denotes set of all possible states in the environment, A is a set of all possible actions the agent can take, P defines the probability $P(S_{t+1}|s_t, a_t)$ of transitioning from state s_t to state s_{t+1} after taking action a_t, R immediate reward r received after transitioning from state s_t to state s_{t+1} due to action a_t and γ is discount factor. The prime goal of the agent is to discover a policy π that maps states to actions, maximizing the cumulative reward over time. The policy $\pi(a \mid s)$ denotes the probability of taking an action a in state s, and the value of following a policy from a specific state is computed by the action-value function $Q^{\pi}(s, a)$ as (1).

$$Q^{\pi}(s, a) = \mathbb{E}[R_{t+1} + \gamma\, Q^{\pi}(S_{t+1}, A_{t+1})|S_t = s, A_t = a] \quad (1)$$

Where $Q^{\pi}(s, a)$ is action-value function, γ ranges between 0 and 1, determines the present value of future rewards, reflecting the trade-off between immediate and future rewards, and Q^{π} refers to optimal policy value. The optimal policy π^* is the one that maximizes the action-value function across all states.

$$\pi^* = \arg max_{\pi}\; Q^{\pi}(s, a) \quad (2)$$

To compute Q values without a model of the environment i.e., in model-free scenarios such as Q-learning, the agent learns the optimal action-value function Q^* without knowledge of P and R and seeks to learn the optimal action-value function directly using the Bellman (3).

$$Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha[r_{t+1} + \gamma\; max_{a'}\; Q(s_{t+1}, a') - Q(s_t, a_t)] \quad (3)$$

Here, $Q^{\pi}(s_t, a_t)$ is the current estimated Q-value for a given state-action pair, α is the learning rate, $r_{t+1}$is the reward received after taking an action at in state st and transitioning to state s_{t+1} and the term $max_{a'}\; Q(s_{t+1}, a')$ represents the estimate of the optimal future value. The updating process of the Q-value is iterative and continues until the policy converges to the optimal policy π^*, which dictates the best action to take in every state.

2.2. Dueling deep Q-network

The proposed study adopts the dueling DQN algorithm for agent modeling, aiming to refine the estimation of state-action value functions (Q(s,a)) in complex environments like NGNs and IoT, characterized by rich state-action spaces. In the context of RL-based IDS, existing approaches predominantly utilized DQN or double DQN. The DQN model, while widely used, has a tendency to overestimate Q-values, which can lead to the formulation of suboptimal policies. Double DQN, though it addresses this overestimation by using two neural networks (current and target Q-networks), does not effectively discriminate between the value of states and actions. Additionally, it faces slow convergence and not much efficient to capture latent or minute differences in state-action pairs. Dueling DQN is preferred for its unique ability to differentiate between state values and action advantages, thereby precisely estimates state-action value which is crucial in complex settings like NGNs, where precise assessment of each state-action pair is fundamental to effective intrusion detection. Figure 2 presents the architecture of dueling DQN, illustrating how the algorithm splits the estimation process into two streams: one for evaluating the state value and another for assessing the advantage of actions.

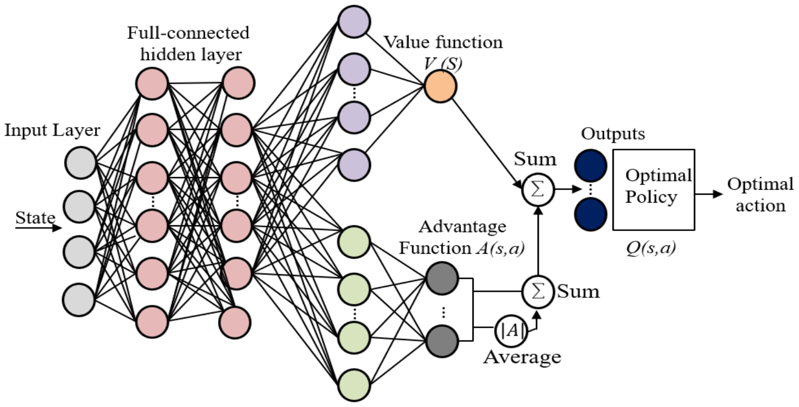

Figure 2. Architecture of dueling DQN

As depicted in Figure 2, the dueling DQN comprises an input layer that takes the current state of the environment as input. Afterwards, a fully-connected hidden layer is responsible for extracting features from the input state. Further it decomposes into two sub-networks for the separate estimation of state values V(a) and action advantages F(s,a). The sub-network subjected to value function V(s) represents the scalar value of the state and computes the value function, which estimates how good it is to be in a given state s, regardless of the action taken. The second sub-network F(s,a) computes the advantage function for each action a given the states using (4).

$$A(s,a;\theta,\alpha) = Q(s,a;\theta,\alpha,\beta) - V(s;\theta,\beta) \tag{4}$$

Where θ denotes the parameters of the shared neural network layers, while α and β are the parameters of the advantage and value function, respectively. The advantage function indicates the relative benefit of taking a particular action compared to the average action in the current state. Before producing the final Q-value for a state-action pair, the network aggregates the value and advantage estimates using (5).

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\alpha)\right) \tag{5}$$

Where A represents the action space, and $|\mathcal{A}|$ is the number of actions. To stabilize the learning process and ensure the identifiability of the value function, the advantage function estimator is modified by subtracting the mean advantage of all possible actions. The proposed study aims to maximize the expected cumulative reward from a state s, represented by the value function $V^{\pi}(s)$ under a policy π, expressed as (6).

$$V^{\pi}(s) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^{k} \mathcal{R}(S_{t+k}, A_{t+k})|S_t = s] \tag{6}$$

Hence, the research problem is to optimize the parameters θ, α, β to maximize the expected cumulative reward by learning an optimal policy $\pi*$. This is subject to the constraints imposed by the dueling DQN architecture for the estimation of Q-values. The optimization problem can be formulated as (7) and (8).

$$max_{\infty,\alpha,\beta} V^{\pi}(s) \tag{7}$$

$$\theta,\alpha,\beta \leftarrow \theta,\alpha,\beta + \eta\, \nabla_{\theta,\alpha,\beta} V^{\pi}(s) \tag{8}$$

The (8) shows that during the learning process, parameters θ, α, β are adjusted through gradient ascent on the expected cumulative reward, where η is the learning rate. The learning process is constrained by the architecture of dueling DQN, which separates the estimation of the value of being in a state from the estimation of the advantage of taking specific actions in that state. To achieve the optimization objectives, the parameters are tuned by minimizing the loss function.

$$\mathcal{L}(\theta,\alpha,\beta) = \mathbb{E}\left[\left(y_i - Q(s,a;\theta,\alpha,\beta)\right)^2\right] \tag{9}$$

Where y_i is the target Q-value calculated using a variant of the Bellman (9). The optimization is considered converged when updates to θ, α, β result in minimal changes to the value function $V^{\pi}(s)$, indicating that the policy π is near-optimal or optimal.

2.3. Game theoretic framework

The proposed study presents a multi-agent IDS system incorporating game theory to model strategic interactions between cyber attackers and defenders. Within this IDS framework, cyber conflict is conceptualized as a game where the attacker seeks to compromise the system, and the defender aims to prevent breaches. The defender and attacker use their respective utility functions to evaluate and optimize their strategies. In the proposed dueling DQN based IDS implementation, both the attacker and defender agents are trained to maximize their respective utility functions. The defender's agent is trained to minimize the impact of attacks, while the attacker's agent aims to find successful attack strategies. The interaction between the defender and attacker is dynamic, allowing for continuous adaptation and learning. The dueling DQN estimates state-action value functions for both for both the attacker and defender, expressed as (10).

$$\begin{cases} Defender\colon Q_d(s,a;\theta_d,\alpha_d,\beta_d) \\ Attacker\colon Q_a(s,a;\theta_a,\alpha_a,\beta_a) \end{cases} \tag{10}$$

Where, Q_d and Q_a are the action-value function for the defender and attacker, respectively. Both represents the expected reward for taking a specific action a in a given state s, based on the current policy. This value is an estimation of the total amount of reward (cumulative reward) the defender and attacker can expect to accumulate over the future, starting from state s and taking an action a. Therefore, the objective for each agent is to maximize its expected cumulative reward, and the (10) can be modified as (11).

$$\begin{cases} Defender\colon V_d{}^{\pi}(s) = \mathbb{E}[\sum_{k=0}^{\infty}\gamma^k \mathcal{R}_t^d(S_{t+k},A_{t+k})|S_t = s] \\ Attacker\colon V_a{}^{\pi}(s) = \mathbb{E}[\sum_{k=0}^{\infty}\gamma^k \mathcal{R}_t^a(S_{t+k},A_{t+k})|S_t = s] \end{cases} \tag{11}$$

Where, $\mathcal{R}_d$ and $\mathcal{R}_a$ are the reward functions for the defender and attacker, respectively. The parameters θ_d,α_d,β_d for the defender and $\theta_a,\alpha_a,\beta_a$ for the attacker are adjusted using gradient ascent described in the above (8) on the expected reward, enabling each agent to learn and adapt its strategy.

2.4. Dataset adopted

The dataset adopted in the proposed study is the NSL-KDD dataset, a labeled dataset widely adopted in the research community and established as a standard benchmark for network intrusion detection. Frequently employed to evaluate the performance of various IDS, the NSL-KDD dataset contains millions of labeled data points, each representing network connections categorized into normal behavior and various attack types (DoS, U2R, R2L, and Probe). The dataset mirrors real-world network traffic, consisting a mix of normal and anomalous activities, and features are derived from TCP/IP connection characteristics, providing a realistic challenge for detection algorithms. Moreover, this dataset is used to model the RL environment using Open-AI Gym functionalities. Table 1 summarizes the characteristics and distribution of class samples.

Table 1. NSL-KDD data-record classes

Categories	Definitions	Samples #
Normal (0)	Typical user behavior on the network	77054
DoS (1)	Attacks aimed at service unavailability	53385
Probe (2)	Network scanning and vulnerability mapping attempts	14077
R2L (3)	Unauthorized access attempts by a remote machine to gain local user privileges	3749
U2R (4)	Unauthorized access attempts by users to gain root privileges	252

2.5. Proposed system implementation

In the above sections, the study discusses core technologies and design considerations adopted in the proposed multi-agent driven IDS. The system utilises a duel DQN architecture for NGNs. In addition, this section details implementation methodology emphasizing proposed customized environment and multi-agent modelling.

2.5.1. Customized environment using Open-AI Gym for IDS simulation in NGNs

The proposed system introduces NGNEnv, a custom environment for IDS that harnesses the OpenAI Gym interface to simulate the intricate dynamics of NGNs. Open-AI Gym is an open-source and benchmarked tool for developing and comparing RL algorithms [31]. It offers a standardized set of environments for implementing and testing various RL models. This simulation uses the NSL-KDD dataset to create a realistic network traffic environment for the IDS. Incorporating Open-AI Gym in our study enhances the development and evaluation of advanced IDS models. Its ability to simulate realistic network environments, combined with the flexibility to incorporate complex game-theoretic approaches, providing a consistent interface for RL algorithms, making it easier to compare the effectiveness of different models under similar conditions. The modelling of the environment consists of following core components.

- State space**:** The state space, S, includes normalized features representing network traffic data such that $S = \{s_1, s_2, s_3 \dots, s_{162}\} \in \mathbb{R}^n$, where each s_i is a normalized and standardized data samples representing network traffic with n number of features in the NSL-KDD dataset.
- Action space:The discrete action space A, for the defender agent includes five possible actions corresponding to IDS classifications, such that A_d={0,1,2,3,4} where each action represents a classification decision by the IDS on attack categories as shown in Table 1. Similarly, the attacker agent's action space comprises 23 discrete actions such that: $\mathrm{A}_a = \{0,1,2, \dots, 22\}$ as shown in Table 2.

Table 2. Number of actions taken by attacker agent A_d

Action index	Attack type	Description
0	normal	Normal network traffic
1	back	Denial-of-service attack that floods target with reflected packets
2	land	Denial-of-service attack with source and destination addresses set to same host
3	neptune	Denial-of-service attack flooding with UDP packets from random source ports
4	pod	Denial-of-service attack flooding with TCP packets from random source ports
5	smurf	Denial-of-service attack using ICMP echo request from spoofed source address
6	teardrop	Denial-of-service attack fragmenting TCP packet to trigger reassembly errors
7	ipsweep	Reconnaissance attack scanning for active hosts on a network
8	nmap	Port scanning tool for identifying open ports and services
9	portsweep	Reconnaissance attack scanning for open ports on a network
10	satan	Port scanning tool with additional service identification capabilities
11	ftp_write	Malicious attempt to write to a file on an FTP server
12	guess_passwd	Brute-force attack attempting to guess user passwords
13	imap	Unauthorized access attempt to an IMAP server
14	multihop	Indirect attack routing through multiple machines to hide attacker origin
15	phf	Port scan using SYN packets with various flags set
16	spy	Attempt to gain unauthorized access to a system for information gathering
17	warezclient	Downloading pirated software content
18	warezmaster	Distributing pirated software content
19	buffer_overflow	Attempt to exploit a buffer overflow vulnerability for code execution
20	loadmodule	Attempt to load and execute malicious code as a kernel module
21	perl	Attempt to execute malicious Perl script
22	rootkit	Attempt to install a rootkit for unauthorized access and control

- Reward mechanism: The reward function $\mathcal{R}_t$ is designed to incentivize each agent for their action. The study adopts binary reward scheme, whether the defender's action matches the attacker's action or not. The rewards take the value 1 for a correct match and 0 otherwise. Mathematically expressed as (12) and (13).

$$\text{Defender's Reward:} \mathcal{R}_t^d \leftarrow \begin{cases} 1 \text{ if defender\_actions} = \text{attack\_actions} \\ 0 \qquad\qquad\qquad\qquad \text{Otherwise} \end{cases} \tag{12}$$

$$\text{Attacker's Reward: } \mathcal{R}_t^a \leftarrow \begin{cases} 1 \text{ if defender\_actions} \neq \text{attack\_actions} \\ 0 \qquad\qquad\qquad\qquad \text{Otherwise} \end{cases} \tag{13}$$

The proposed NGNEnv also consists of two modules such as episode termination and reset functionality. The environment simulation runs for a certain number of steps, each step corresponding to a single row or event in the dataset. An episode ends when all the events in the dataset have been presented to the agent, which simulates the process of monitoring network traffic over a period. The reset function is responsible for reinitializing the environment to its starting condition, which allows the agent to start learning a new cycle of the network traffic monitoring.

2.5.2. Proposed multi-agent model using duelling deep Q-network

The proposed environment NGNEnv simulates the multi-agent system with defender agent that detects intrusions and an attacker agent for adversarial attempts, both utilizes dueling DQN architecture to evolve optimal counter strategies. Algorithm 1 outlines the computational steps for this adversarial IDS framework in NGNs. The operational parameters for both agents include a standardized input features

$X \in \mathbb{R}^{m\times n}$ and label $Y \in \{0,1,2 \ldots 22\}^m$ along with the exploration rate (ε), discount factor (γ), and an experience replay mechanism. After successful execution, it returns optimal policy value for taking best actions.

```
Algorithm 1: Adversarial learning using dueling DQN for intrusion detection
Input: $\mathcal{D}$ dataset with features $X \in \mathbb{R}^{m\times n}$ and $Y \in \{0,1,2 \ldots 22\}^m$, where $m$ number of features and $n$ number of samples; $\Theta$ parameters for dueling DQN model for defender ($\Theta_d$) and attacker $\Theta_a$; $\gamma$ discount factor for future reward for both attacker $\gamma_a$ and defender $\gamma_d$ and $\varepsilon$ initial exploration $\varepsilon_d$ and $\varepsilon_a$.
Output: Optimal policy $\pi_d^*$ for defender and $\pi_a^*$ for the attacker
Start
    1. Environment Initialization:
       Standardize features in D to obtain state space S with $S \subset \mathbb{R}^n$
         Define action space:
           Defender $A_d = \{0, \cdots, 4\}$
           Attacker $A_a = \{0, \cdots, 22\}$
         Initialize dueling DQN network parameter $\Theta_d$and $\Theta_a$ with random weights
    2. For each episode$e = 1$ to $E$:
         Initialize the initial environment state $s_0$ by selecting a random sample form $\mathcal{D}$.
       For each timestep$t = 1$ to $T$:, where T is the maximum number of timesteps per episode
       Select Actions:
            For the defender agent : $a_t^d = \begin{cases} \text{random } (A_d) & \text{with probability } \varepsilon_d \\ \arg\max_{a\in A_d} Q_d(s_t, a; \Theta_d) & \text{otherwise} \end{cases}$
            For the attacker agent : $a_t^a = \begin{cases} \text{random } (A_a) & \text{with probability } \varepsilon_d \\ \arg\max_{a\in A_a} Q_a(s_t, a; \Theta_a) & \text{otherwise} \end{cases}$
            Apply action $a_t^d$ and $a_t^a$ to the environment and receive new state ($s_{t+1}$) and rewards $r_t^d$, $r_t^a$
       Update state: $S_t \leftarrow s_{t+1}$
       Store Transition:$(s_t, a_t^d, r_t^d, s_{t+1})$ in replay buffer $D_d$ and $(s_t, a_t^a, r_t^a, s_{t+1})$ in buffer $D_a$.
       Sample minibatch and Update Dueling DQN Models:
       Sample minibatch from $D_d$ and $D_a$
       Compute the largest Q-values for the $s_{t+1}$ for both agents using the temporal difference target
       $y_t^d = r_t^d + \gamma_d {}^{max}_{a'} Q_d(S_{t+1}, a'; \Theta_d^-) \& y_t^a = r_t^a + \gamma_a {}^{max}_{a'} Q_a(S_{t+1}, a'; \Theta_a^-)$
            Perform gradient descent step to update $\Theta_d$ and $\Theta_a$ by minimizing the expected loss :
       $\Theta_d \leftarrow \Theta_d - \alpha_d \times \nabla \Theta_d \mathcal{L}(Q_d(s_t, a_t^d; \Theta_d) y_t^d) \& \Theta_a \leftarrow \Theta_a - \alpha_a \times \nabla \Theta_a \mathcal{L}(Q_a(s_t, a_t^a; \Theta_a) y_t^a)$
       Update Target Networks:
            Every $T$ steps, update the target network parameters using the soft update rule
       $\Theta_d^- \leftarrow \tau\Theta_d + (1-\tau)\Theta_d^- \& \Theta_a^- \leftarrow \tau\Theta_a + (1-\tau)\Theta_a^-$
         Policy Improvement: Update the ∈-greedy exploration rates for both the defender and attacker
       $\epsilon_d \leftarrow max(\in_{d,min}, \in_d \times decay_d) \& \epsilon_a \leftarrow max(\in_{a,min}, \in_a \times decay_a)$
                Where $\in_{d,min}$ and $\in_{a,min}$ are the minimum exploration rates
Terminal State Check: If $S_{t+1}$ is a terminal state or $t$ equals the maximum number of timesteps $T$
           Reset the environment: $S_t + 1 \leftarrow reset()$ and proceed the next episode if any.
    3. End of Episode and Convergence check
    4. Evaluate policy performance and adjust parameters if necessary.
    5. Check convergence for both $\Theta_d$ and $\Theta_a$ over episodes and extract final policies if convergence criteria are met: $\pi_d^* \leftarrow {}^{\arg max}_{a\in A_d} Q_d(s, a; \Theta_d)$;$\pi_a^* \leftarrow {}^{\arg max}_{a\in A_a} Q_a(s, a; \Theta_a)$// This represents the optimal policies for deciding suitable action by both defender and attacker agents, respectively.
End
```

The Algorithm 1 presents the implementation steps for a proposed multi-agent model utilizing the dueling DQN architecture to establish a robust IDS in NGNs. The algorithm takes as input a standardized dataset with features X and label Y. It also requires the initialization of various other parameters such as random weights Θ_d and Θ_a for the defender and attacker dueling DQN models, respectively, as well as the discount factors γ_a and γ_a, and the initial exploration rates ε_d and ε_a. The entire computing steps are structured to train agent models in episodes and timesteps within an adversarial learning step, where at each step, actions are chosen based on an ε-greedy policy that balances exploration with exploitation. The actions taken by both the defender and attacker agents lead to new state observations and rewards, which are then used to update the agents' strategies through a process of RL. The transitions observed at each timestep are stored in replay buffers, from which minibatches are sampled to update the dueling DQN models. These updates are made by computing the largest Q-values for the subsequent state, using temporal difference targets, and performing gradient descent to minimize expected loss, thereby refining the decision-making policies. A key aspect of the algorithm is the update of target networks, which occurs every T timesteps to stabilize the Q-value predictions. This step uses a soft update rule to blend the current network parameters with those of the target networks, preventing the rapid propagation of errors through the Q-value estimations. The exploration rates are also adjusted over time to reduce the likelihood of random actions as the models become more capable, thus focusing on exploiting the learned policies. The final steps of the algorithm accounts for terminal states within the environment, which signify the end of an episode and a reset of environment to begin anew states_{t+1}. However, when the condition s_{t+1} a terminal state is not met, the agent simply proceeds to the next timestep within the current episode. This is the typical loop within an episode of training, where the agent continually interacts with the environment until it reaches a terminal state or the maximum number of allowed timesteps. This iterative process ensures continuous learning and adaptation of the agents to the dynamics of the network traffic, exploring the complex interactions and decision-making processes necessary for maintaining network integrity and security. Figure 3 illustrates the flowchart of process described in Algorithm 1, summarizing the systematic and iterative nature of training dueling DQN agents in an adversarial setting from initialization through to policy evaluation and convergence.

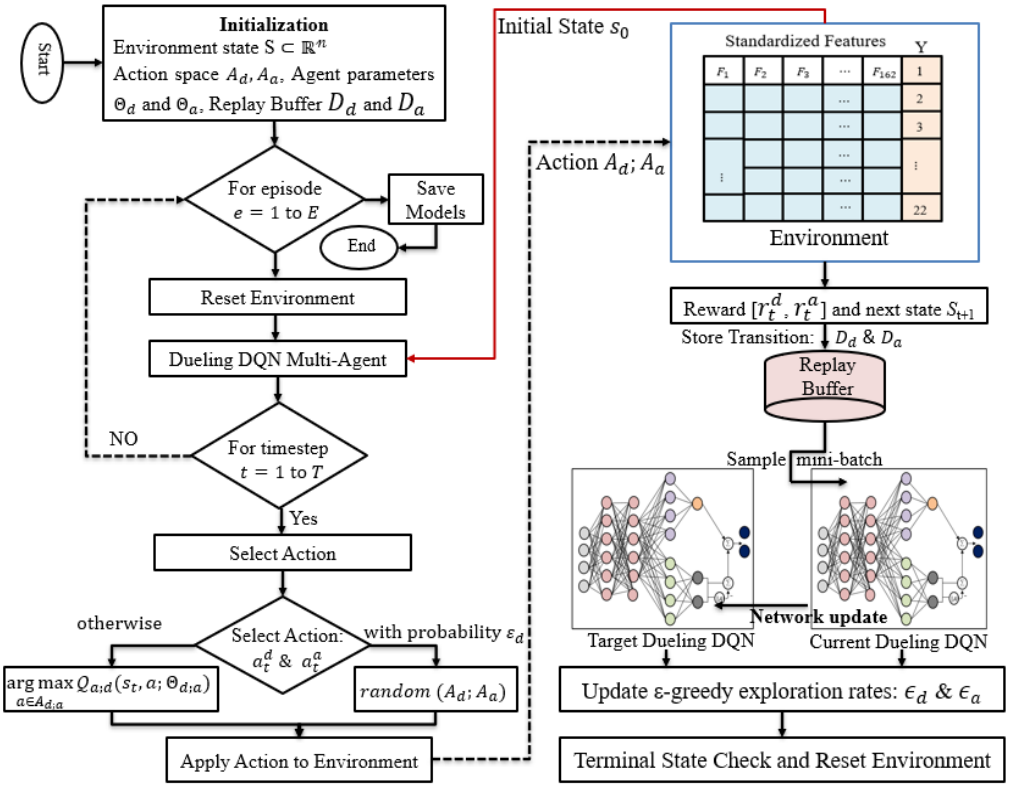
Start
Initialization
Environment state S ⊂ $\mathbb{R}^n$
Action space A_d, A_a, Agent parameters
Θ_d and Θ_a, Replay Buffer D_d and D_a
For episode e = 1 to E
Save Models
End
Reset Environment
Dueling DQN Multi-Agent
NO
For timestep t = 1 to T
Yes
Select Action
otherwise
Select Action: a_t^d & a_t^a
with probability ε_d
$\arg\max_{a \in A_{d;a}} Q_{a;d}(s_t, a; \Theta_{d;a})$
$random\ (A_d; A_a)$
Apply Action to Environment
Initial State s_0
Action $A_d; A_a$
Standardized Features
Y
F_1
F_2
F_3
…
F_{162}
1
2
3
⋮
22
Environment
Reward $[r_t^d, r_t^a]$ and next state S_{t+1}
Store Transition: D_d & D_a
Replay Buffer
Sample mini-batch
Network update
Target Dueling DQN
Current Dueling DQN
Update ε-greedy exploration rates: ϵ_d & ϵ_a
Terminal State Check and Reset Environment

Figure 3. Operational Flowchart for dueling DQN-based adversarial learning in IDS

3. RESULTS AND DISCUSSION

The proposed dueling DQN-based defender and attacker agents were designed and developed using the Python programming language within the Anaconda environment. The experiments were conducted on a Windows 11 64-bit operating system with an NVIDIA GTX 1650 GPU. The dueling DQN agents employ a DL architecture to estimate the action-value function. The defender agent consists of one input layer, three hidden layers, each with 128 neurons, and an output layer with 5 neurons representing intrusion detection actions. In contrast, the attacker agent is configured with one input layer and 2 hidden layers, each containing 128 neurons, and an output layer with 23 neurons representing various attack strategies. The discount factor for future reward for both attacker γ_a and defender γ_d is initialized to 0.001 and ε initial exploration for both the agent ε_d and ε_a is set to 0.9. The training of both agents is conducted over 200 episodes, with each episode consisting of 100 iterations (timesteps). The performance of the proposed IDS is evaluated based on reward and loss rates throughout episodes, as well as accuracy and F1-score metrics.

Figure 4 illustrates the cumulative rewards for both defense and attack agents across 200 episodes. Based on the careful observation of graph trends it can be analyzed that the defense reward rapidly increases and stablizes early in the training process. This suggests that the defender agent quickly learns effective strategies for detecting intrusions, indicating a robust defense strategy over time. While, the attacker agent's reward shows an initial increase but then it decrease at a lower rewards value throughout the remaining episodes. The defender's higher cumulative reward throughout the majority of the episodes suggests that the defense strategies being learned and employed are effectively mitigating the attack strategies. This analysis shows that the effectiveness of the dueling DQN algorithm in training the defender agent to adapt and respond to evolving threats in a dynamic adversarial environment.

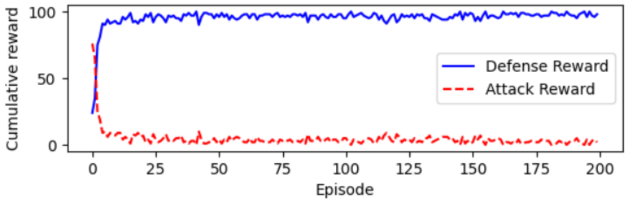

Figure 4. Cumulative reward Vs episodes

Figure 5 presents the loss trends of the defender and attack agents over 200 training episodes. It can be observed the loss curve for both the agents during training shows a sharp decline in the initial episodes, indicating rapid learning and improvement in the performance of both agents. As the episodes progress, both loss measures exhibit a convergence towards a minimal value, with slight fluctuations that reflects the process of ongoing learning and adaptation of the agents to each other's strategies. It can be also seen that the loss curve for defender stabilizes quickly indicating defender agent's ability to reliably predict and counteract attack strategy.

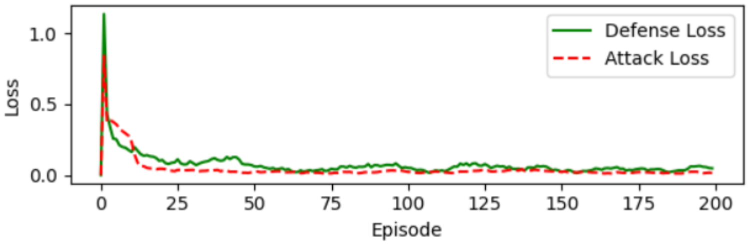

Figure 5. Training loss over episodes

Figure 6 depicts a heat map representing the intensity of different types of network attacks as well as normal traffic over the 200 training episodes. For instance, the brighter yellow squares in the 'DoS' and 'R2L' columns suggest a higher intensity of these attacks at certain epochs, such as the 20th and 120th for 'DoS' and the 80th and 160th for 'R2L'. As observed that the consistently darker shades in the U2R column suggest that this type of attack occurs less frequently across all epochs. This heat map indicates a dynamic interplay between the defense mechanisms of the IDS and the attack strategies employed by the adversarial learning step.

Figure 7 presents a statistical analysis of the proposed IDS, showcasing its effectiveness in accurately identifying instances across various classes. For normal traffic, the proposed IDS successfully classified 8,531 out of 9,712 instances. This indicates a substantial robustness of the proposed system in distinguishing benign activities from malicious ones. In the context of DoS attacks, the IDS correctly identified 6,647 out of 7,458 instances. However, the system also recorded 409 false negatives and 400 false positives in this category. The probe attack with a total of 2,421 data samples, the proposed IDS system correctly identified 2,146 as probe and for R2L attacks the IDS correctly classified 2,450 out of 2,753 data samples. The U2R attack class, being the least frequent with only 200 instances, and 148 correctly identified by the IDS and 25 false negatives and 27 false positives are particularly concerning given the lower sample size. Despite the challenge of class imbalance, the proposed IDS is robust in its detection capabilities.

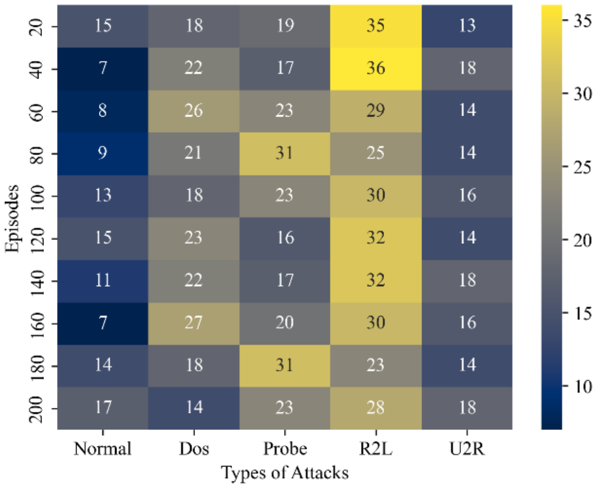

Figure 6. Intensity of attacks over epoch

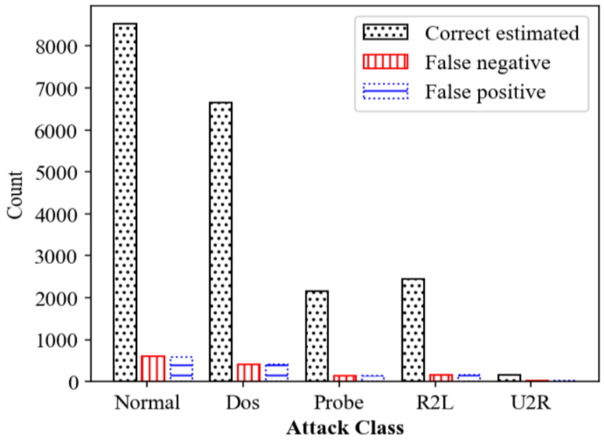

Figure 7. Illustrates statistical outcome of the proposed IDS system

In the comparative analysis presented in Table 3, the proposed IDS showcases a significant advancement over existing methods. The proposed system achieves a high accuracy of 95.02% and an F1-score of 94.68%, indicating not only its capability to correctly classify intrusions but also to maintain a balance between precision and recall. The existing DQN RL approach [30] lacks adaptive learning, which is a critical shortfall in the context of evolving cyber threats, resulting in lower accuracy and F1-score. The existing scheme adaptive education (AE)-RL [32], despite incorporating adaptive learning, does not achieve the high performance metrics of the proposed system, potentially due to its simplified simulation of adversarial interactions which may not adequately represent the complex dynamics of intrusions to learn by an defense agent leading to fail in capturing intrusion in unseen data. The AESMOTE [33] approach, while addressing class imbalance and including adaptive learning, still falls short of the proposed system's performance. This could be due to the limitations in its environmental modeling particularly the data sampling approach that balances attack classes through artificial data generation in training instances.

Table 3. Comparative analysis

Existing approaches	Adaptive learning	Multi-agent approach	Accuracy	F1-Score
DQN RL [30]	No	No	0.7807	0.8141
AE-RL [32]	Yes	Yes	80.16	79.40
AESMOTE [33]	Yes	Yes	0.82	82.4
Proposed (Dueling DQN)	Yes	Yes	0.9502	0.9468

The reason behind achieving better performance by the proposed system are multiple. The primary reason is the incorporation of advanced dueling DQN mechanisms, which allow for a refined estimation of state values and action advantages, significantly reducing the overestimation bias that is common in other models. Furthermore, the system's adaptability is enhanced through the use of a customized OpenAI Gym environment, which provides a realistic and dynamic platform for the agents to learn and evolve. The multi-agent adversarial learning framework further ensures that the IDS is continuously tested against an actively adapting adversary, where attackers are constantly developing new strategies.

4. CONCLUSION

This paper has introduced an novel IDS employing a RL algorithm dueling DQN within a multi-agent adversarial framework. The proposed system has demonstrated superior performance in detecting network intrusions suitable for NGNs, as evidenced by the comparative analysis which benchmarks the system against current state-of-the-art methods. The integration of adaptive learning has proven to be significant that allows the system to evolve and become dynamic to the ever-changing tactics employed by cyber adversaries. Another unique contribution is the utilization of a customized OpenAI Gym environment, which simulates realistic network traffic behaviors, providing a robust platform for our agents to learn and make decisions. The dynamic nature of the multi-agent adversarial learning setup ensures that the defender agent is not only equipped to deal with current threat patterns but is also continually refining its strategies to respond to emerging threats. Future work will focus on refining the system’s learning algorithms, exploring the integration of other AI techniques, and expanding the system’s applicability to diverse network problem such as routing, security and bandwidth optimization.

REFERENCES

[1] J. Jayakumari and K. Karagiannidis, “Advances in communication systems and networks,” *Lecture Notes in Electrical Engineering*, vol. 656, Singapore: Springer, 2020, doi: 10.1007/978-981-15-3992-3.

[2] J. Amutha, S. Sharma, and J. Nagar, “WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues,” *Wireless Personal Communications*, vol. 111, no. 2, pp. 1089–1115, Mar. 2020, doi: 10.1007/s11277-019-06903-z.

[3] A. Whitmore, A. Agarwal, and L. D. Xu, “The internet of things—A survey of topics and trends,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, Apr. 2015, doi: 10.1007/s10796-014-9489-2.

[4] A. Attkan and V. Ranga, “Cyber-physical security for IoT networks: A comprehensive review on traditional, blockchain and artificial intelligence based key-security,” *Complex & Intelligent Systems*, vol. 8, no. 4, pp. 3559–3591, Aug. 2022, doi: 10.1007/s40747-022-00667-z.

[5] A. Mishra, A. V. Jha, B. Appasani, A. K. Ray, D. K. Gupta, and A. N. Ghazali, “Emerging technologies and design aspects of next generation cyber physical system with a smart city application perspective,” *International Journal of System Assurance Engineering and Management*, vol. 14, no. S3, pp. 699–721, Jul. 2023, doi: 10.1007/s13198-021-01523-y.

[6] K. Gupta and S. Shukla, “Internet of things: Security challenges for next generation networks,” in *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, IEEE, Feb. 2016, pp. 315–318, doi: 10.1109/ICICCS.2016.7542301.

[7] Ö. Aslan, S. S. Aktuğ, M. O. -Okay, A. A. Yilmaz, and E. Akin, “A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions,” *Electronics*, vol. 12, no. 6, Mar. 2023, doi: 10.3390/electronics12061333.

[8] A. K. Dangi, K. Pant, J. A. -Beltran, N. Chakraborty, S. V. Akram, and K. Balakrishna, “A review of use of artificial intelligence on cyber security and the fifth-generation cyber-attacks and its analysis,” in *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, IEEE, Jan. 2023, pp. 553–557, doi: 10.1109/AISC56616.2023.10085175.

[9] Shruti, S. Rani, D. K. Sah, and G. Gianini, “Attribute-based encryption schemes for next generation wireless IoT networks: A comprehensive survey,” *Sensors*, vol. 23, no. 13, Jun. 2023, doi: 10.3390/s23135921.

[10] A. K. Sangaiah, A. Javadpour, and P. Pinto, “Towards data security assessments using an IDS security model for cyber-physical smart cities,” *Information Sciences*, vol. 648, Nov. 2023, doi: 10.1016/j.ins.2023.119530.

[11] S. Gaba *et al.*, “A systematic analysis of enhancing cyber security using deep learning for cyber physical systems,” *IEEE Access*, vol. 12, pp. 6017–6035, 2024, doi: 10.1109/ACCESS.2023.3349022.

[12] C. Alippi and S. Ozawa, “Computational intelligence in the time of cyber-physical systems and the internet of things,” in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, 2019, pp. 245–263, doi: 10.1016/B978-0-12-815480-9.00012-8.

[13] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, “BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset,” *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[14] F. D. S. Sumadi, C. S. K. Aditya, A. A. Maulana, S. Syaifuddin, and V. Suryani, “Semi-supervised approach for detecting distributed denial of service in SD-honeypot network environment,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 3, pp. 1094–1100, Sep. 2022, doi: 10.11591/ijai.v11.i3.pp1094-1100.

[15] A. J. Mohammed, M. H. Arif, and A. A. Ali, “A multilayer perceptron artificial neural network approach for improving the accuracy of intrusion detection systems,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 4, pp. 609–615, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp609-615.

[16] M. L. -Martin, A. S. -Esguevillas, J. I. Arribas, and B. Carro, “Network intrusion detection based on extended RBF neural network with offline reinforcement learning,” *IEEE Access*, vol. 9, pp. 153153–153170, 2021, doi: 10.1109/ACCESS.2021.3127689.

[17] J. Saikam and C. Koteswararao, “EESNN: Hybrid deep learning empowered spatial–temporal features for network intrusion detection system,” *IEEE Access*, vol. 12, pp. 15930–15945, 2024, doi: 10.1109/ACCESS.2024.3350197.

[18] N. O. Aljehane *et al.*, “Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security,” *Alexandria Engineering Journal*, vol. 86, pp. 415–424, Jan. 2024, doi: 10.1016/j.aej.2023.11.078.

[19] M. L. -Martin, B. Carro, and A. S. -Esguevillas, “Application of deep reinforcement learning to intrusion detection for supervised problems,” *Expert Systems with Applications*, vol. 141, Mar. 2020, doi: 10.1016/j.eswa.2019.112963.

[20] S. Dong, Y. Xia, and T. Peng, “Network abnormal traffic detection model based on semi-supervised deep reinforcement learning,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4197–4212, Dec. 2021, doi: 10.1109/TNSM.2021.3120804.

[21] Y. Lu, Y. Kuang, and Q. Yang, “Intelligent prediction of network security situations based on deep reinforcement learning algorithm,” *Scalable Computing: Practice and Experience*, vol. 25, no. 1, pp. 147–155, Jan. 2024, doi: 10.12694/scpe.v25i1.2329.

[22] Z. Li, C. Huang, S. Deng, W. Qiu, and X. Gao, “A soft actor-critic reinforcement learning algorithm for network intrusion detection,” *Computers & Security*, vol. 135, Dec. 2023, doi: 10.1016/j.cose.2023.103502.

[23] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, “ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model,” *Scientific Reports*, vol. 12, no. 1, Sep. 2022, doi: 10.1038/s41598-022-19366-3.

[24] S. Yu *et al.*, “Deep Q-network-based open-set intrusion detection solution for industrial internet of things,” *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 12536–12550, Apr. 2024, doi: 10.1109/JIOT.2023.3333903.

[25] K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, “Attention based multi-agent intrusion detection systems using reinforcement learning,” *Journal of Information Security and Applications*, vol. 61, Sep. 2021, doi: 10.1016/j.jisa.2021.102923.

[26] H. Benaddi, K. Ibrahimi, A. Benslimane, M. Jouhari, and J. Qadir, “Robust enhancement of intrusion detection systems using deep reinforcement learning and stochastic game,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 11089–11102, Oct. 2022, doi: 10.1109/TVT.2022.3186834.

[27] M. He, X. Wang, P. Wei, L. Yang, Y. Teng, and R. Lyu, “Reinforcement learning meets network intrusion detection: A transferable and adaptable framework for anomaly behavior identification,” *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 2477–2492, Apr. 2024, doi: 10.1109/TNSM.2024.3352586.

[28] Y. Feng, W. Zhang, S. Yin, H. Tang, Y. Xiang, and Y. Zhang, “A collaborative stealthy DDoS detection method based on reinforcement learning at the edge of internet of things,” *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 17934–17948, Oct. 2023, doi: 10.1109/JIOT.2023.3279615.

[29] M. Soltani, K. Khajavi, M. J. Siavoshani, and A. H. Jahangir, “A multi-agent adaptive deep learning framework for online intrusion detection,” *Cybersecurity*, vol. 7, 2024, doi: 10.1186/s42400-023-00199-0.

[30] H. Alavizadeh, H. Alavizadeh, and J. J. -Jaccard, “Deep Q-learning based reinforcement learning approach for network intrusion detection,” *Computers*, vol. 11, no. 3, Mar. 2022, doi: 10.3390/computers11030041.

[31] P. Palanisamy, *Hands-on intelligent agents with OpenAI Gym: Your guide to develop AI agents using deep reinforcement learning*. Birmingham, United Kingdom: Packt Publishing, 2018.

[32] G. Caminero, M. L. -Martin, and B. Carro, “Adversarial environment reinforcement learning algorithm for intrusion detection,” *Computer Networks*, vol. 159, pp. 96–109, Aug. 2019, doi: 10.1016/j.comnet.2019.05.013.

[33] X. Ma and W. Shi, “AESMOTE: Adversarial reinforcement learning with SMOTE for anomaly detection,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 943–956, Apr. 2021, doi: 10.1109/TNSE.2020.3004312.

BIOGRAPHIES OF AUTHORS

Sai Krishna Lakshminarayana working as associate professor in the Department of Electronics and Communication Engineering at Mother Theresa Institute of Engineering and Technology, Palamaner, Chittoor Dist, Andhra Pradesh, India. He received B.E. degree in Electronics and Telecommunication from AIT, Pune, M.Tech. in Digital Electronics and Communication from AMCE, Bangalore. He is a member of ISTE and pursuing Ph.D. from REVA University, Bangalore, India. He is having more than 18 years of experience in academics. His areas of interest are network security, wireless communication, machine learning, and computer communication network. He can be contacted at email: sklnarayana@gmail.com.

Prabhugoud I. Basarkod received B.E. degree in Electronics and Communication from National Institute of Engineering, Mysore, M.E. degree in Electronics from the BMS college of Engineering, Bangalore and M. S. in Software Systems from Birla Institute of Technology and Science, Pillani and completed his Ph.D. in Kuvempu University, Shankaragatta, Shimoga, Karnataka, India. He is currently working as a Professor in Department of Electronics and Communication at REVA University, Bangalore. He is having a teaching experience of thirty-three years and his areas of interest are wireless communication and computer networking. He is a member of ISTE (MISTE, India), member of IEEE (MIEEE, USA). He can be contacted at email: basarkodpi@reva.edu.in.