

Conflict-driven learning scheme for multi-agent based intrusion detection in internet of things

Durga Bhavani Attluri, Srivani Prabhakara

Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bangalore, Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Mar 5, 2024

Revised Jun 24, 2024

Accepted Jul 2, 2024

Keywords:

Internet of things

Intrusion detection system

Multi-agent system

Reinforcement learning

Security

ABSTRACT

This paper introduces an effective intrusion detection system (IDS) for the internet of things (IoT) that employs a conflict-driven learning model within a multi-agent architecture to enhance network security. A double deep Q-network (DDQN) reinforcement learning algorithm is implemented in the proposed IDS with two specialized agents, the defender and the challenger. These agents engaged in an antagonistic adaptation process that dynamically refined their strategies through continual interaction within a custom-made environment designed using OpenAI Gym. The defender agent aims to identify and mitigate threats by matching the actions of the challenger agent, which is designed to simulate potential attacks in the environment. The study introduces a binary reward mechanism to encourage both agents to explore and exploit different actions and discover new strategies as a response to adversarial actions. The results showcase the effectiveness of the proposed IDS in terms of higher detection rate the comparative analysis also validates the effectiveness of the proposed IDS scheme with an accuracy of approximately 96%, outperforming similar existing approaches.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Durga Bhavani Attluri

Department of Computer Science and Engineering, BMS Institute of Technology and Management

Bangalore, Karnataka, India

Email: durga842004@bmsit.in

1. INTRODUCTION

The increasing connection of physical and digital worlds through internet of things (IoT) devices has unleashed many benefits across industries [1]. However, increasing interconnected devices has created a complex cybersecurity landscape. This has resulted in unique vulnerabilities, which malicious actors often exploit. These attacks can compromise critical aspects, such as privacy, integrity, and availability [2]. Traditional security measures need help adapting to the dynamic nature of the IoT and attackers' evolving tactics [3]. Resource limitations on many devices necessitate robust and lightweight security solutions [4]. intrusion detection systems (IDSs) are crucial for identifying and mitigating cyber threats in the IoT, but their dependence on known attack signatures limits their effectiveness against unknown and evolving threats [5], [6]. This limitation highlights the need to develop adaptive IDSs that detect known attacks and adapt to uncover unknown security threats and attacks.

Recent advancements in machine learning (ML) techniques and their integration into IDSs have shown promising results in the context of network security [7], [8]. However, IDS developed using supervised learning models highly depend on labelled data quality and are ineffective against evolving cyber threats. Unsupervised learning offers an alternative approach to developed IDS but suffers from high false positives and vulnerabilities to adversarial attacks [9], [10]. On the other hand, reinforcement learning (RL), an alternative ML approach, enables IDSs to adapt to evolving threats by continuously learning and exploring

the states of the network environment and adjusting its detection strategies over time. However, with the advancement of computational intelligence technology, adversaries can now modify their attack methods in response to encountered defense mechanisms.

The state-of-the-art research has presented various implementation schemes for building intelligent IDSs for wireless sensor networks (WSNs) and IoT environments [11]. It has been analyzed that most research work on network IDSs have adopted supervised learning schemes such as artificial neural network (ANN), support vector machine (SVM), k-nearest neighbor (KNN), naïve Bayes, logistic regression (LR), decision tree and different hybrid models. Bhavani and Mangla [12] modelled a deep regularize mechanism to address the class imbalance issues in a supervised learning scheme for developing effective IDS against modern attacks. The outcome shows an effective outcome in terms of accuracy and F1-score. Mohammadi *et al.* [13] implemented SVM-based IDS followed by a feature selection process to enhance the effectiveness of intrusion detection. The work of Ding *et al.* [14] presented a hybrid approach where the KNN classifier is integrated with a generative adversarial learning network (GAN) to tackle the problem of biased learning. Laghrissi *et al.* [15] applied a long short-term memory (LSTM) neural network followed by the feature reduction algorithm principal component analysis (PCA). The research is done in a similar direction presented by Imrana [16], where the Bi-directional LSTM model is implemented to develop robust network IDS. Halbouni *et al.* [17] presented an advanced learning scheme that combines the application of LSTM with convolutional neural networks (CNN) to capture both spatial and temporal characteristics of network traffic in the training phase.

The existing literature also consists of many IDS approaches using unsupervised learning techniques. Chen *et al.* [18] implemented the K-means clustering technique optimized by the ant-lion optimization approach to identify and isolate anomalies from the network effectively. Similarly, a global hierarchical clustering followed by a whale optimization approach is suggested by Wang *et al.* [19] for developing efficient and reliable IDS. Lopes *et al.* [20] demonstrated how autoencoders could be effectively employed to create a baseline of normal network behavior, subsequently enabling the detection of deviations indicative of cyber-attacks. Zhang *et al.* [21] highlighted the effectiveness of auto-encoder models, and they implemented a stacked sparse autoencoder with an improved Gaussian mixture model to detect network anomaly. Another work done by Durga and Mangla [22] focuses on using autoencoder as a feature extraction technique for developing multi-class IDS. The features of regular network traffic were extracted, and distance vector and Gini-index measure clustering were used to assign cluster id for different intrusion classes. The intrusions are classified by implementing different supervised classifiers.

Recent research work has also shown a greater interest towards exploring the application of RL in the context of network anomaly and intrusion recognition. Lopez-Martin *et al.* [23] discussed RL and its applicability to network IDS. The authors have empirically shown how RL agents could dynamically adjust their policy in response to the detected anomalies. Yaseen and Saadi [24] presented a deep Q-learning driven IDS to detect and counter denial-of-service (DoS) attack. Suratkar *et al.* [25] suggested an adaptive honeypot system using Q-learning for severity analysis of cyber-attacks. It significantly enhances honeypots' deception and intelligence-gathering capabilities, providing insights to boost real-world cybersecurity defenses. The work by Kim and Park [26] suggested an online IDS scheme using the joint approach of deep auto-encoder and Q-network to predict and classify network threats in real-time accurately.

Cardellini *et al.* [27] implemented a deep Q-learning network (DQN) to process complex input spaces and perform efficient action selection, enhancing the IDS's ability to detect and mitigate threats in real time. Recent research also focuses on multi-agent-based IDS based on a game theory concept to model learning processes in cooperative or non-cooperative approaches. Shamshirband *et al.* [28] presented a design of fuzzy logic and Q-learning-based cooperative game theory model to identify denial-of-service attacks and prevention of genuine nodes in wireless networks. Similarly, a Bayesian game theory concept is applied by Liang *et al.* [29] to develop a self-learning IDS framework using the DDQN algorithm to locate intrusions in vehicular Ad-hoc networks effectively. In the study of Khoury and Nassar [30], a hybrid approach is presented, integrating a game theory model with a Q-learning algorithm to derive optimal attack sequences and corresponding optimal defense strategies to train IDS for cyber-physical network security. Hence, it is evident from the above discussion that various IDS solutions exist based on supervised, unsupervised, and reinforcement learning methodologies. Table 1 summarizes the above-discussed literature to offer a quick insight into their advantages and limitations.

It has been analyzed based on literature exploration and analysis that to date, none of the existing methods is a standard and foolproof concept, and since the IoT ecosystem is dynamic and heterogeneous, developing a single or universal security mechanism is very challenging. The potential research problems being identified are as follows:

- Most existing IDSs using supervised learning heavily depend on high-quality labelled datasets. This dependency limits the IDS's ability to detect zero-day attacks, as these require the identification of previously unknown attack signatures.
- Unsupervised learning-based IDS can address zero-day attacks. Yet, their effectiveness is frequently compromised by high false-positive rates, especially when anomalies' features are identical to normal traffic patterns.
- Most research using RL applications in IDS design is based on a Q-learning algorithm, which suffers from the state explosion problem. The exponential increase in possible states due to network complexity makes it impractical to maintain a comprehensive Q-table.
- The application of game theory and multi-agent systems is also well explored. However, most of them are implemented using Q-learning and deep Q-network (DQN). Both tend to overestimate Q-values, leading to suboptimal policy choices and affecting the system's accuracy and reliability.
- It has also been identified that previous work on IDS development is more focused on achieving higher accuracy only and ignored optimization in the higher computational demands.

Table 1. Summary of the existing literature in the context of network IDS

Citations	Problem context	Solution approach	Remark
[12]	Class imbalance issue in supervised IDS	Contrastive learning and deep regularization scheme	Self-supervised, un-biased learning, but may suffers in identifying unknown threats
[13]	Detection of evolving threats	An empirical study of SVM classifiers for IDS	Limited adaptability of SVM to dynamic IoT environments
[14]	Class imbalance and classification errors in IDS	KNN and generative learning model	Implementation of generative models may not fully capture complex attack patterns
[15]	Feature optimization	LSTM and PCA	Assumption of linearity, loss of Interpretability and highly sensitive to outliers
[16]	High false alarm rates	Bidirectional long-short-term-memory (Bi-LSTM) model	Reduced false alarm rate but higher computational demands
[17]	Evolving attack vectors and expanding network sizes	Hybrid CNN-LSTM model	Resource-intensive and may not generalize across all scenarios
[18]	K-means convergence to local optima	A hybrid clustering algorithm	Specialized nature may limit its adaptability to different context
[19]	Overloading and service un-availability	Whale optimization and hierarchical clustering	Needs effective benchmarking
[20]	Training in lack of enough data samples	Autoencoder neural network	Higher complexity and resource demands for implementation
[21]	High dimensionality and lack of labeled datasets in IDS.	Stacked sparse autoencoder with improved Gaussian mixture model	Complexity in optimizing joint parameters for varied datasets.
[22]	Feature modelling	Autoencoders and distance function-based clustering	Improved classification outcome
[23]	Performance enhancement in IDS	Deep reinforcement learning (DRL)	Difficulty in designing a universal reward function
[24], [25]	Intrusion detection	Q-learning	Adversaries may still develop countermeasures against presented solution
[26]	Detection of advanced cyber-threat	Deep auto-encoder with Q-network	Demands higher computational resources
[27]	Curse of dimensionality	Deep Q-Networks with transfer learning.	Transfer learning effectiveness can vary based on the system's change dynamics
[28]	DoS attack detection in WSN	Cooperative game-based fuzzy Q-learning	Needs more optimization
[29], [30]	Mitigating new cybersecurity threats.	Game theory and Q-learning based multi-agent system	Achieves robust performance but potential scalability issues

Therefore, this paper introduces a novel IDS based on multi-agent architecture incorporating defender and challenger agents. Both agents are precisely designed using a double deep Q-network (DDQN) RL algorithm within a conflict-driven learning framework that facilitates an antagonistic adaptation approach wherein each agent iteratively refines its action policy in response to the evolving strategies of its counterpart. The concept of antagonistic adaptation refers to the competitive relationship that enhances the learning and adaptation of agent capabilities, enabling the IDS to detect better and counteract emerging threats. The overall contribution of this paper can be highlighted as follows:

- This paper introduces a novel conflict-driven learning model within a multi-agent algorithm. This model employs an antagonistic adaptation approach, enabling the defender agent to dynamically evolve and adjust its defensive strategies against evolving attack patterns simulated by the challenger agent.
- The proposed multi-agent-driven IDS leverages the DDQN algorithm to address state overestimation bias and optimize the learning process.

- The paper introduces a custom RL environment developed using the OpenAI Gym toolkit. This environment accurately replicates realistic network traffic and communication scenarios, enhancing the agents' self-exploration capability and decision policies toward more sophisticated detection strategies.

The uniqueness and novelty of the proposed work are the integration of DDQN algorithms within a multi-agent architecture, combined with the concept of antagonistic adaptation towards designing dynamic and effective IDS solutions in cybersecurity, mitigating the rapidly evolving threats in network environments. The usage of DDQN algorithm in the agent design can enhance its ability to handle complex, high-dimensional state spaces typical in network environments, thereby mitigating the state explosion problem often encountered in traditional RL algorithms. The proposed scheme facilitates the detection and mitigation of known cyber threats and also it enables IDS to evolve in response to emerging sophisticated attack strategies dynamically.

2. METHOD

The proposed study introduces a modelling of effective network IDS under conflict-driven multi-agent framework, utilizing DDQN RL algorithms. This study also develops a custom RL environment to simulate effective networking environment and offer a better interaction and learning experiences to proposed agent algorithms. The prime aim is to offer a highly responsive and adaptive IDS in dynamic network environments, particularly addressing sophisticated and evolving cyber-threats in the IoT ecosystem. The high-level architecture of the proposed system is outlined in Figure 1.

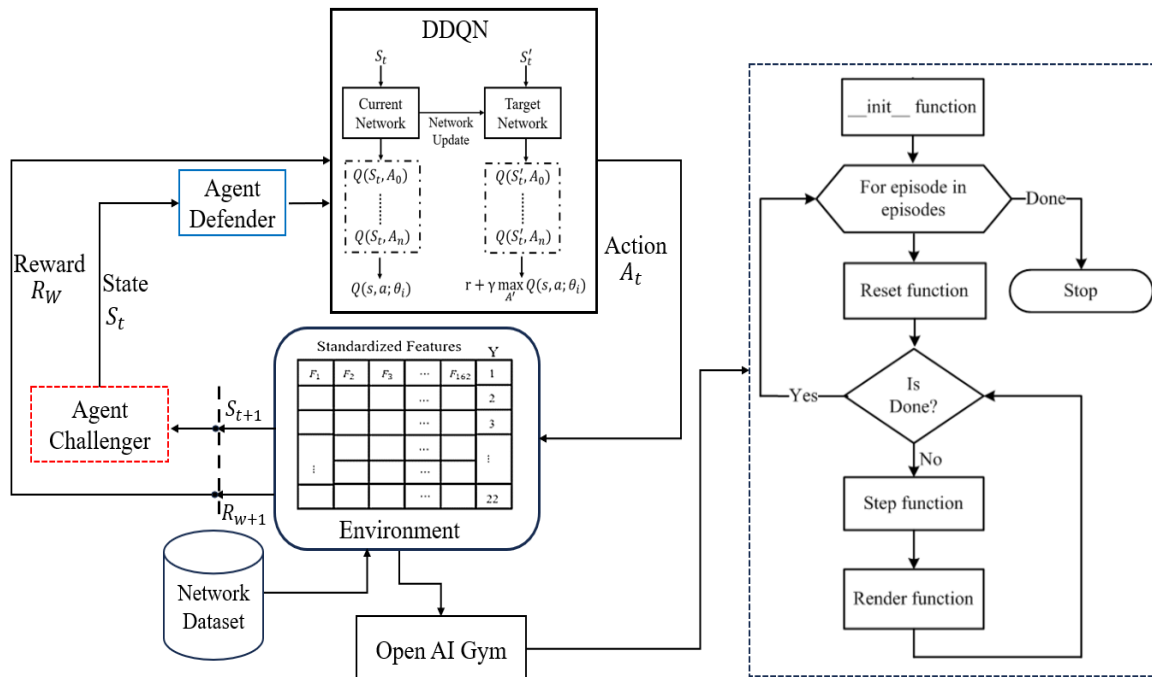


Figure 1. Illustrating a high-level architecture of the proposed multi-agent driven IDS

Figure 1 depicts the schematic architecture and workflow overview of the proposed IDS system. The multi-agent framework includes two agent models, called defender and challenger. The primary role of the defender agent is to identify and mitigate potential threats. The challenger agent simulates potential attack strategies, enhances the learning capabilities of the defender agent network, and improves its decision-making process by continuously providing adversarial scenarios for the defender to counterattack. Here, both agents are combined with decision-making policies using the DDQN algorithm to define actions that classify network activity as normal or potentially malicious. Both use a similar DDQN architecture consisting of two deep neural networks, the current and target networks. The target network stabilizes learning by providing a fixed baseline to update the Q-value in the current network. The network update process utilizes a combination of reward signals and a max operator applied to Q-values to iteratively improve the policy. The execution of the proposed agent models is carried out in the proposed custom environment, which is designed

using features of OpenAI Gym, such as environment initialization functionality and execution of the agent in multiple episodes in an iterative manner. In each episode, the system performs a reset function to initialize the environment, and the process continues until the termination condition is met. Step functions represent the agent's interaction with the environment, taking actions and receiving new states and rewards. Finally, rendering functions are used for visualization and logging purposes. This custom environment was developed to simulate real-world network traffic scenarios from the network dataset NSL-KDD, which consists of various network traffic patterns, including benign and malicious data points. The dataset used in the study was first subjected to a pre-processing stage, where features were normalized to ensure consistency and promote more effective learning by the agent. The proposed custom environment is an optimal interaction platform for the multi-agent system, providing agents with a consistent and controlled setup to interact with the dataset and learn from their experience.

2.1. Agent and environment interaction

The agent-environment interaction within an IDS can be formulated as a Markov decision process (MDP), described by a tuple $(S_t, A_t, P_{sa}, R_w, \gamma)$, where S_t is a finite set of states representing distinct observable configurations of the network environment, wherein any states $\in S_t$ includes variables such as traffic volume, packet characteristics, user commands, and protocol anomalies indicative of the network's current behaviour. The variable A_t is a finite set of actions available to the agent, with any action $a \in A_t$ consisting of operational responses like initiating alerts and connection termination. The next variable P_{sa} denotes the state transition probability that defines the dynamics of the environment, such that $P_{sa}(s' | s, a)$ is the probability of transitioning to the next state s' or $s + 1$ from state s upon taking action a . For each action, the agent gets feedback in the form of a reward such that: $S \times A \rightarrow R$ is the reward function where $R(s, a, s')$ quantifies the immediate reward received after transitioning from state s to state s' as a result of action a . This function is designed to encourage the agent to improve its action policy, and for each optimal action being executed (accurate identification of threats), it receives a positive reward R^+ and a penalty R^- will be given in case of false positives or non-detections. Rewards are assigned based on the correctness of the intrusion detection. A high reward is given for correctly identifying an intrusion, a lesser reward for detecting an intrusion, and a negative reward for false positives or missed detections.

The variable $\gamma \in [0,1]$ is the discount factor determining the present value of future rewards, thereby influencing the agent's policy for appropriate actions. The policy $\pi(a | s)$ is a strategy that the agent employs to determine which action to take given state s , $\pi: S \rightarrow A$. It is usually a probability distribution over the actions. The value function $V^\pi(s)$ estimates the expected return (cumulative discounted rewards) from the state s . Numerically, the $V^\pi(s)$ is defined as (1), (2):

$$V^\pi(s) = E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_{t=s}] \quad (1)$$

$$Q(s, a) = E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_{t=s}, A_{t=a}] \quad (2)$$

In (1) $V^\pi(s)$ indicates long-term reward an agent expects from a current state (s), effectively quantifying the potential future benefits of adhering to policy π . On the other hand, action-value function $Q(s, a)$ in (2) estimates the immediate reward and future rewards based on taking a specific action (a) in the current state (s), according to the current policy π . Although it considers future rewards into its calculation, but its primary focus is to measure the immediate impact of an action, and ensure correctness of short-term decisions with long-term objectives. Both functions $V^\pi(s)$ and $Q(s, a)$ ensures that the agent to develop a balanced policy between immediate response to threats and the set of optimal future states.

2.2. Double deep Q-networks

The agents are modeled using the DDQN algorithm, which employs a dual deep network architecture. The current network ($Q_{current}$) selects the best action based on current Q-values and the target network (Q_{target}) estimates the expected future Q-value for the chosen action. This dual network approach separates the evaluation of the current state-action from the estimation of the subsequent state's Q-value, addressing a limitation in the standard DQN. The DDQN model offers improved stability in agent training and effectively mitigates the issue of Q-values overestimation often encountered in DQN and Q-learning algorithms. Figure 2 presents the DDQN architecture used in the design of the proposed agent model.

The current network, $Q_{current}(S_t, a_t)$, is responsible for estimating the Q-value based on the current state S_t and action a_t . This network directly influences the actions taken by the agent, as it steers the policy decisions. Actions are selected using an ϵ -greedy policy, which ensures a balance between exploration of new actions and exploitation of known rewards. Mathematically, the action-selection policy can be represented as:

$$\begin{cases} \text{random action} \\ \arg \max_a Q_{\text{current}}(S_t, a_t) \end{cases} \quad \begin{cases} \text{with probability } \epsilon \\ \text{with probability } (1 - \epsilon) \end{cases} \quad (3)$$

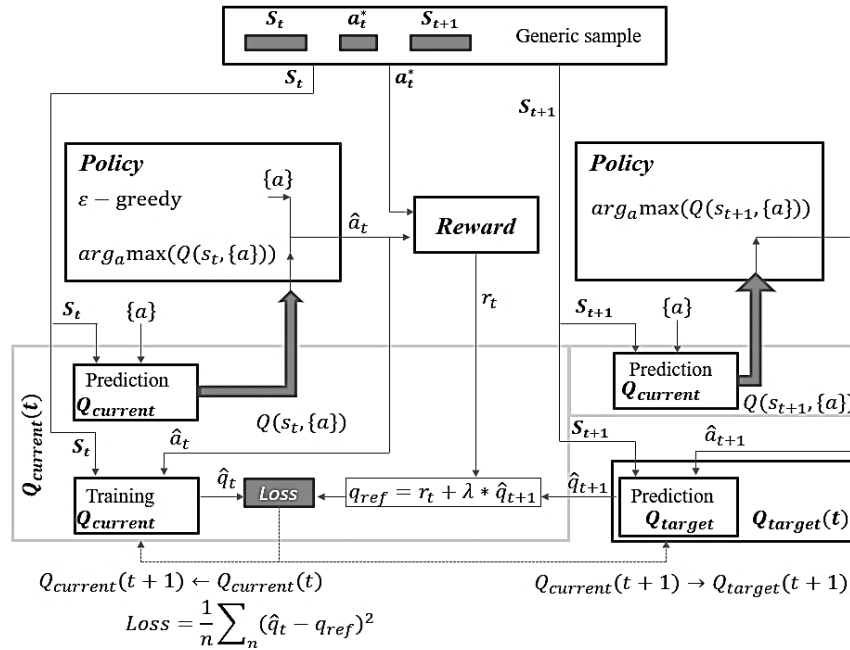


Figure 2. Typical architecture of the DDQN network

Once an action is executed, the agent receives a reward r_t and transitions to a new state S_{t+1} . These experiences are encapsulated in a tuple (S_t, a_t, r_t, S_{t+1}) and stored for training the network. The target network, Q_{target} , provides a stable reference for the future Q-value, used to compute the loss for updating Q_{current} . This target value is periodically updated to reflect the latest knowledge of the agent while avoiding rapid fluctuations. The DDQN loss function is then computed as the mean squared error between the current Q-value and the reference Q-value, Q_{ref} , which is the sum of the immediate reward and the discounted maximum future Q-value as estimated by the target network. The learning mechanism implemented in the agent model basically uses experience replay memory to recollect information, obtain fresh perspectives, and improve decision-making. This memory requires sampling mini-batches from various memory pools in order to learn. By doing this, the training becomes more stable by reducing the correlation between subsequent learning updates. Also, by minimizing TD errors i.e., the discrepancy between the estimated future reward and the current estimate, an update function is utilized that trains the neural network. These inaccuracies are computed using the rewards and forecasts for the upcoming state. The agent's policy network is adjusted in this way to move toward ideal Q-values. Further an epsilon-greedy strategy governs the selection of actions, striking a balance between exploitation (taking the best available action) and exploration (trying new actions).

2.2. Proposed custom environment

As already discussed, that the proposed environment mimics the network scenario of the NSL-KDD dataset. This network dataset is a refined version of the KDD'99 dataset widely used for the evaluation of network IDS. This dataset is considered to be good for training and testing IDSs since it covers a wide range of network traffic properties, including both benign and malicious activity. The environment consists of the following components.

- State space: This component of the environment includes normalized features (e.g., min-max scaling) representing network traffic data. The size of the observation space matches the number of features in the NSL-KDD dataset, which characterize the traffic patterns and behaviors in the network.
- Action space: The action space is discrete, consisting of five actions that correspond to different classifications the IDS can make, such as no alert (indicating normal traffic) and alert for different attack types such as DoS, Probe, remote-to-local (R2L), and user-to-root (U2R) attacks.

- Reward: Rewards are calculated to encourage both agents to improve their action policy and decision-making capability. Correct identification of intrusions yields a positive reward for the defender and a negative reward for the challenger. If the defender makes incorrect identifications (i.e., false positives and false negatives), it will receive a negative reward, and the challenger agent will receive a positive reward. A similar process is considered for the challenger agent. This encourages the agent to improve its detection accuracy over time.
- Transition dynamics: The transition dynamics are implemented in the step method provided by Open-AI Gym. This method updates the environment's state based on the chosen action, calculates the corresponding reward using the reward function, and determines if the episode has terminated. This component is implemented to model the real-world consequence of an action and provides immediate feedback to the agent.
- Episode termination: The environment simulation runs for several steps, each 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, simulating the process of monitoring network traffic over a period.
- Reset functionality: The reset function is responsible for reinitializing the environment to its starting condition, which allows the agent to start learning anew.

2.3. Agents action

The first agent i.e., the challenger initiates different attack strategies against the network. This attacker agent has a discrete action space consisting of five options such as DoS, Probe, R2L, and U2R attacks. The defender agent represents the IDS where it first learns to detect and respond to these potential attacks by continuously monitoring network traffic and identifying anomalies or suspicious patterns. Both agents undergo an initialization process before they operate in the simulation environment. This process involves defining their action repertoires and understanding the observation space. Essentially, the agents calibrate themselves to comprehend the different types of network data they will encounter and the possible responses they can take. This initialization also sets vital operational parameters for the agents, such as exploration rate, discount factor and experience replay memory.

2.4. Reward formulation

The study considers designing reward functions for both agents that accurately reflect each agent's success in their respective roles. The defender receives a reward of 1 if it is the same as the challenger's action, indicating successful defense against an attack. Otherwise, it receives a reward of 0. On the other hand, the challenger agent receives a reward of 1 if the defender's action does not match its action, indicating a successful breach. Otherwise, it receives a reward of 0. Also, if the attack is detected and mitigated, the attacker receives a negative reward. Additionally, if the attack generates much noise leading to detection, the penalty could be higher to discourage brute force or poorly executed attacks. The reward functions for both defender and challenger agent are numerically represented in (7) and (8), respectively.

$$\mathcal{R}_w^d \leftarrow \begin{cases} 1 & \text{if } A_t^d = A_t^c \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

$$\mathcal{R}_w^c \leftarrow \begin{cases} 1 & \text{if } A_t^d \neq A_t^c \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

The defender is rewarded for accurately predicting and matching the challenger's actions, thus successfully defending against simulated attacks. In contrast, the challenger is rewarded for successfully deceiving the defender, encouraging it to develop more sophisticated attack patterns that the defender must learn to counteract. This binary reward scheme forms the basis for antagonistic adaptation, driving the conflict-driven learning process that continuously enhances the capabilities of both agents.

2.5. Conflict-driven learning

The conflict-driven learning model is formulated within the reinforcement learning (RL) framework, where agents learn optimal behaviors through interactions with the environment. The environment E is defined by a set of states S , actions A , and a reward function $R: S \times A \rightarrow R$. At each time step t , the agents observe a state $S_t \in S$, take an action $A_t \in A$, and receive a reward R_t based on the action's effectiveness. For the defender agent AD and the challenger agent AC , the RL objective is to maximize the cumulative reward over time, which is formalized as (6):

$$\max_{\pi_D, \pi_C} E[\sum_{t=0}^{\infty} \gamma^t R_t(S_t, A_t)] \quad (6)$$

where π_D and π_C represent the policies of the defender and challenger, respectively, and γ is the discount factor that balances the importance of immediate and future rewards. On the other hand, antagonistic adaptation in this model describes how each agent adapts its policy in response to the perceived threat or challenge presented by the other agent's actions. This can be formulated as a min-max problem in game theory, where each agent seeks to maximize its payoff while minimizing the opponent's payoff, numerically formulated as (7):

$$\max_{\pi_D} \min_{\pi_C} E_{\pi_D, \pi_C} [\sum_{t=0}^{\infty} \gamma^t R_t(S_t, A_{D_t}, A_{C_t})] \quad (7)$$

The equation (7) presents the core process of conflict-driven learning in the proposed a multi-agent system, where the defender aims to optimize its rewards while simultaneously restricting the challenger's benefits. Therefore, learning process becomes a strategic game where, each agent continuously updates its policy π_D for the defender and π_C for the challenger, using feedback from each interaction to recalibrate their actions. This iterative learning and interaction offer more precise and effective strategies, which allow defender agent not only to learn from its own experiences but also from the adaptations of its adversary (challenger agent).

3. RESULT ANALYSIS

The design and development of the proposed RL-driven IDS is carried out using Python programming language executed in Anaconda distribution installed on Windows core i7 machine configured with 16 GB RAM with NVIDIA RTX1650 GPU support. The study considers similar deep learning architecture for both agents where defender model consists of three hidden layers each with 200 neurons and challenger agent has three hidden layers with 150 neurons. The training of the proposed multi-agent system is carried out in an iterative manner with 100 episodes. The validation of the proposed IDS system is done considering the NSL KDD dataset considering different performance metrics for both attacker and defender agent.

3.1. Performance analysis

Figure 3 presents performance analysis of challenger agent concerning its action towards generating adversarial scenario to challenge defender agent. Based on the graphical analysis it can be seen that several attacks generated by attacker agents over progressive epochs in last episode. The analysis suggests that the challenger agent successfully launched varieties of attacks which were quite random and uncertain. Therefore, it presents a challenging situation for defender to tackle it.

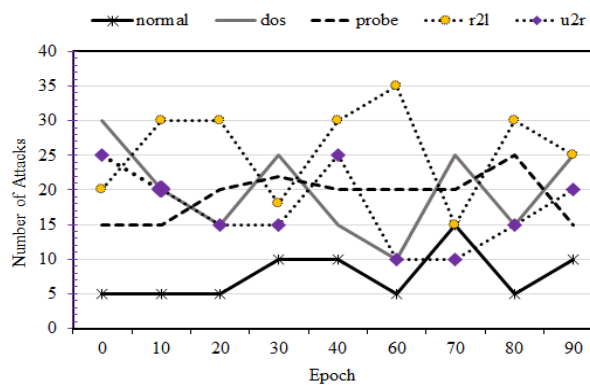


Figure 3. Analysis of attack generation by attacker agent during training

It can be seen that the number of normal traffic remains relatively stable across the epochs, with minor fluctuations, except for a peak at epochs 30, 40 and 70. In case of DoS attack, it has generated attacks scenario with more variability with high and low pattern repetitively over progressive epochs. The Probe attacks have less variation compared to Dos attacks, while R2L attack count shows significant fluctuation. The U2R attack count also fluctuates, beginning with around 25, dropping to 15. The pattern suggests that challenger agents have dynamically simulated the attacks against defender agents and frequent changes or fluctuation in response to changing defense mechanisms.

Figure 4 shows the cumulative rewards for the defender and challenger agents over 100 rounds. Based on careful observation of the chart trends, it can be analyzed that the defender reward increases rapidly and stabilizes early in the training process. It can be analyzed from the above findings that the defender agents were able to quickly learn optimal strategies against the intrusions over time. On the other hand, the challenger agent's reward initially appears to increase, but it also decreases at a lower reward value for the remaining episodes. The next Figure 5 shows the performance results of an IDS towards classifying network traffic into normal activity and different network attacks (DoS, Probe, R2L, U2R).

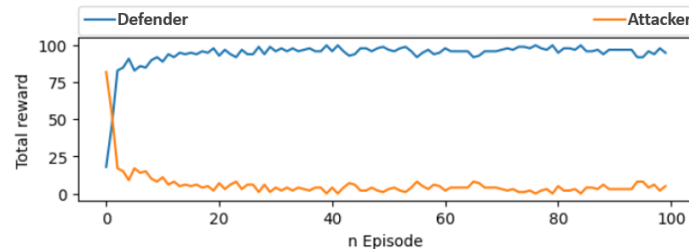


Figure 4. Analysis of reward vs episodes

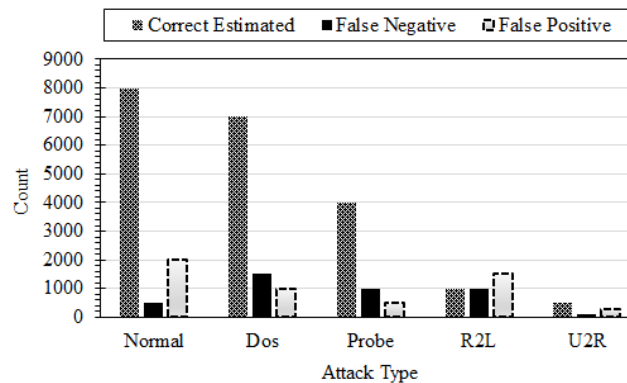


Figure 5. Analysis of detection outcome

The statistical outcome presented in Figure 5 demonstrates that the proposed defender agent IDS, effectively detected and classified correct instances of network traffic with minimal false negatives and false positives. For normal traffic, the proposed IDS successfully classified 8,131 out of 9,000 instances, while misclassification rate for other attack classes are low. This indicates the robustness of the proposed system in distinguishing benign activities from malicious ones. The reason behind mis-classified instances is due to the complex nature of dataset which is subjected to heavy class imbalance among normal and attacks instances. Although the performance achieved by the proposed defender IDS is quite considerable with high detection rate. Table 2 presents a comparative analysis for the proposed system validation against similar existing approaches. The proposed system achieves an accuracy of 95.06% outperforming other similar approaches DQN approach [31] and adversarial environment (AE) RL [32].

Table 2. Comparative analysis

Existing approaches	Multi-agent model	Accuracy
DQN [31]	No	78.07
AE-RL [32]	Yes	80.16
Proposed (DDQN)	Yes	95.06

3.2. Discussion and implications

The core findings obtained from the performance analysis suggests that the proposed RL-driven IDS indicated a significant achievement in intrusion detection. The primary reason behind achieving better performance by the proposed system is the incorporation of DDQN with antagonistic adaptation mechanisms.

This approachable the model to become more precise in the estimation of state values and action advantages, thereby reducing the overestimation bias that is common in Q-learning based IDS models. Furthermore, the system's adaptability is enhanced through customized environment, which provides a realistic and dynamic interaction to an agent towards optimal learning and evolving strategies. The conflict-driven learning scheme further ensures that the IDS is continuously tested against an actively adapting adversary, where attackers constantly develop new strategies. The defender agent demonstrated its ability to quickly learn and stabilize effective defense strategies, as evidenced by the cumulative rewards analysis. On the other hand, the challenger agent, while initially successful in generating adversarial scenarios, did not sustain an upward trajectory in rewards, suggesting that the defender was learning and adapting to the evolving attack patterns effectively. A key advantage of our system is the robustness demonstrated against a variety of attack types. For instance, the defender's consistent performance in identifying normal traffic with high accuracy, and the less variability in response to Probe attacks as compared to DoS, suggests an adaptability that is crucial in real-world applications with evolving and dynamic nature of cyber threats.

4. CONCLUSION

This paper has introduced a novel IDS employing an RL algorithm DDQN within a conflict-driven learning framework. The system facilitates a robust antagonistic adaptation approach between agents, the defender and the challenger designed to enhance and evolve network security protocols continuously. Another unique contribution is the use of a customized OpenAI Gym environment, which simulates real-life network traffic behavior and provides a powerful platform for our agents to learn and make decisions. Comparative analysis shows that the proposed system exhibits superior performance in network intrusion detection suitable for IoT, which is benchmarked against existing similar work. The proposed IDS scheme can be implemented in real-world IoT application like smart healthcare, smart farming, and many more to complement data security schemes such as data encryption to ensure both network availability as well as data integrity and confidentiality. Future work will focus on enhancing the current algorithm based on the exploration of advanced feature engineering and other deep learning techniques. Also, the scope of the proposed work can be extended to addressing other different network problems such as routing, security, and bandwidth optimization.





REFERENCES

- [1] R. Ahmad and I. Alsmadi, "Machine learning approaches to IoT security: a systematic literature review," *Internet of Things*, vol. 14, Jun. 2021, doi: 10.1016/j.iot.2021.100365.
- [2] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019, doi: 10.1109/ACCESS.2019.2924045.
- [3] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019, doi: 10.1109/COMST.2019.2910750.
- [4] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller, "Landscape of IoT security," *Computer Science Review*, vol. 44, May 2022, doi: 10.1016/j.cosrev.2022.100467.
- [5] K. Albulayhi, A. A. Smadi, F. T. Sheldon, and R. K. Abercrombie, "IoT intrusion detection taxonomy, reference architecture, and analyses," *Sensors*, vol. 21, no. 19, Sep. 2021, doi: 10.3390/s21196432.
- [6] A. Aburomman and M. Bin Ibne Reaz, "Review of IDS development methods in machine learning," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 5, pp. 2432–2436, Oct. 2016, doi: 10.11591/ijece.v6i5.pp2432-2436.
- [7] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsubihany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences*, vol. 12, no. 10, May 2022, doi: 10.3390/app12105015.
- [8] A. Fatani, A. Dahou, M. A. A. Al-qaness, S. Lu, and M. A. Abd Elaziz, "Advanced feature extraction and selection approach using deep learning and Aquila optimizer for IoT intrusion detection system," *Sensors*, vol. 22, no. 1, Dec. 2021, doi: 10.3390/s22010140.
- [9] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges," *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 3211–3243, Jun. 2021, doi: 10.1007/s11831-020-09496-0.
- [10] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 1134–1141, Feb. 2023, doi: 10.11591/ijece.v13i1.pp1134-1141.
- [11] R. Gopi *et al.*, "Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things," *Multimedia Tools and Applications*, vol. 81, no. 19, pp. 26739–26757, Aug. 2022, doi: 10.1007/s11042-021-10640-6.
- [12] A. D. Bhavani and N. Mangla, "A novel approach of intrusion detection system for IoT against modern attacks using deep learning," in *Lecture Notes in Networks and Systems*, 2024, pp. 172–182.
- [13] M. Mohammadi *et al.*, "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *Journal of Network and Computer Applications*, vol. 178, Mar. 2021, doi: 10.1016/j.jnca.2021.102983.
- [14] H. Ding, L. Chen, L. Dong, Z. Fu, and X. Cui, "Imbalanced data classification: a KNN and generative adversarial networks-based hybrid approach for intrusion detection," *Future Generation Computer Systems*, vol. 131, pp. 240–254, Jun. 2022, doi: 10.1016/j.future.2022.01.026.





- [15] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00448-4.
- [16] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, Dec. 2021, doi: 10.1016/j.eswa.2021.115524.
- [17] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022, doi: 10.1109/ACCESS.2022.3206425.
- [18] J. Chen, X. Qi, L. Chen, F. Chen, and G. Cheng, "Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection," *Knowledge-Based Systems*, vol. 203, Sep. 2020, doi: 10.1016/j.knsys.2020.106167.
- [19] L. Wang, L. Gu, and Y. Tang, "Research on alarm reduction of intrusion detection system based on clustering and whale optimization algorithm," *Applied Sciences*, vol. 11, no. 23, Nov. 2021, doi: 10.3390/app112311200.
- [20] I. O. Lopes, D. Zou, I. H. Abdulqadder, F. A. Ruambo, B. Yuan, and H. Jin, "Effective network intrusion detection via representation learning: a denoising AutoEncoder approach," *Computer Communications*, vol. 194, pp. 55–65, Oct. 2022, doi: 10.1016/j.comcom.2022.07.027.
- [21] T. Zhang, W. Chen, Y. Liu, and L. Wu, "An intrusion detection method based on stacked sparse autoencoder and improved gaussian mixture model," *Computers & Security*, vol. 128, May 2023, doi: 10.1016/j.cose.2023.103144.
- [22] A. B. Durga and N. Mangla, "A novel network intrusion detection system based on semi-supervised approach for IoT," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023, doi: 10.14569/IJACSA.2023.0140424.
- [23] M. Lopez-Martin, B. Carro, and A. Sanchez-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.
- [24] H. S. Yaseen and A. Al-Saadi, "Q-learning based distributed denial of service detection," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 972–986, Feb. 2023, doi: 10.11591/ijece.v13i1.pp972-986.
- [25] S. Suratkar *et al.*, "An adaptive honeypot using Q-Learning with severity analyzer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, pp. 4865–4876, Oct. 2022, doi: 10.1007/s12652-021-03229-2.
- [26] C. Kim and J. Park, "Designing online network intrusion detection using deep auto-encoder Q-learning," *Computers & Electrical Engineering*, vol. 79, Oct. 2019, doi: 10.1016/j.compeleceng.2019.106460.
- [27] V. Cardellini *et al.*, "irs-partition: an intrusion response system utilizing deep Q-networks and system partitions," *SoftwareX*, vol. 19, Jul. 2022, doi: 10.1016/j.softx.2022.101120.
- [28] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah, and A. Abraham, "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 228–241, Jun. 2014, doi: 10.1016/j.engappai.2014.02.001.
- [29] J. Liang, M. Ma, and X. Tan, "GaDQN-IDS: A novel self-adaptive IDS for VANETs based on Bayesian game theory and deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12724–12737, Aug. 2022, doi: 10.1109/TITS.2021.3117028.
- [30] J. Khoury and M. Nassar, "A hybrid game theory and reinforcement learning approach for cyber-physical systems security," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, Apr. 2020, pp. 1–9, doi: 10.1109/NOMS47738.2020.9110453.
- [31] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-learning based reinforcement learning approach for network intrusion detection," *Computers*, vol. 11, no. 3, Mar. 2022, doi: 10.3390/computers11030041.
- [32] G. Caminero, M. Lopez-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.

BIOGRAPHIES OF AUTHORS



Durga Bhavani Attluri     working as assistant professor in the Department of CSE in BMS Institute of Technology and Management. Her area of interest are internet of things and machine learning. She has received a grant from VTU for the project pressure ulcer detection and prevention using neural networks. She can be contacted at email: durga842004@bmsit.in.



Srivani Prabhakara     as assistant professor in the Department of CSE in BMS Institute of Technology and Management. She graduated her PhD from VTU on the subject IoT and machine learning in hydroponic system. Her area of interest are internet of things and machine learning. She has received funds from VTU and KSCST for the various projects. She can be contacted at email: srivanicse@bmsit.in.