ICT Express xxx (xxxx) xxx

KICS The Korean Institute of Communications and Information Sciences

Contents lists available at ScienceDirect

ICT Express

journal homepage: www.elsevier.com/locate/icte

Deep Q-learning intrusion detection system (DQ-IDS): A novel reinforcement learning approach for adaptive and self-learning cybersecurity

Md. Alamgir Hossain [1,*]

Department of Computer Science and Engineering, State University of Bangladesh, South Purbachal, Kanchan, Dhaka 1461, Bangladesh

ARTICLE INFO

Keywords:
Reinforcement learning
Deep Q-network (DQN)
Intrusion detection system (IDS)
Self-learning cybersecurity
DQ-IDS

ABSTRACT

With the increasing sophistication of cyber threats, traditional Intrusion Detection Systems (IDS) often fail to adapt to evolving attack patterns, leading to high false positive rates and inadequate detection of zero-day attacks. This study proposes the Deep Q-Learning Intrusion Detection System (DQ-IDS), a novel reinforcement learning (RL)-based approach designed to dynamically learn network attack behaviors and continuously enhance detection performance. Unlike conventional machine learning (ML) and deep learning (DL)-based IDS models that depend on static, pre-trained classifiers, DQ-IDS employs Deep Q-Networks (DQN) with experience replay and adaptive ε-greedy exploration to autonomously classify benign and malicious network traffic. The integration of experience replay mitigates catastrophic forgetting, while adaptive exploration ensures an optimal trade-off between learning efficiency and threat detection. A reward-driven training mechanism reinforces correct classifications and penalizes errors, thereby reducing both false positive and false negative rates. Extensive empirical evaluations on real-world network datasets demonstrate that DQ-IDS achieves a detection accuracy of 97.18%, significantly outperforming conventional IDS solutions in both attack detection and computational efficiency. This work introduces a paradigm shift toward adaptive, self-learning cybersecurity systems capable of real-time, robust threat mitigation in dynamic network environments.

1. Introduction

A network intrusion is any unauthorized access, attack, or malicious activity that attempts to compromise the confidentiality, integrity, or availability of a computer network. Intrusions can involve hacking attempts, malware infections, denial-of-service (DoS) attacks, data breaches, or unauthorized access to network resources [1]. These threats can originate from external attackers (e.g., cybercriminals) or internal users (e.g., insider threats) and can cause severe security risks, including data theft, system disruption, and financial loss [2,3].

Detecting network intrusions is a highly complex and challenging task due to the dynamic and evolving nature of cyber threats. Attackers constantly develop new evasion techniques, such as polymorphic malware, encrypted payloads, and adversarial attacks, making traditional signature-based and anomaly-based Intrusion Detection Systems (IDS) ineffective against zero-day threats. Additionally, IDS models must analyze large volumes of real-time network traffic, requiring high computational efficiency while maintaining low false positive and false negative rates. ML-based IDS solutions, while promising, often suffer from concept drift, where attack patterns change over time, reducing model accuracy. Moreover, achieving a balance between security and efficiency is difficult, as overly sensitive IDS can overwhelm security teams with false alarms, while less sensitive models may fail to detect critical threats [4,5].

Despite significant advancements in Intrusion Detection Systems (IDS), existing approaches suffer from several critical limitations that hinder their effectiveness in modern cybersecurity environments. Traditional signature-based IDS rely on predefined attack patterns, making them incapable of detecting zero-day threats or novel attack variations [6,7]. While anomaly-based IDS can identify new attack behaviors, they often generate excessive false positives, leading to alert fatigue and making real-time response inefficient. Additionally, ML and DL-based IDS models require large labeled datasets, frequent retraining, and manual feature selection, making them computationally expensive

* Correspondence author.
E-mail address: alamgir.cse14.just@gmail.com.
[1] ORCID: 0000-0001-5120-2911.

https://doi.org/10.1016/j.icte.2025.05.007
Received 28 February 2025; Received in revised form 26 April 2025; Accepted 16 May 2025
Available online 18 May 2025

and less adaptable to evolving threats [8,9]. Another major gap is that existing IDS models lack continuous learning capabilities, meaning they become outdated as attack techniques evolve. Furthermore, most IDS implementations struggle with real-time detection due to high computational costs, limiting their scalability in large networks. These challenges highlight the need for an intelligent, adaptive, and resource-efficient IDS that can overcome these limitations by learning and evolving dynamically in response to emerging cyber threats [10,11].

To address these limitations, this research introduces a RL-Based Intrusion Detection System, which leverages Deep Q-Networks (DQN) with experience replay and adaptive exploration to detect and classify network intrusions dynamically. Unlike traditional IDS models, DQ-IDS does not rely on predefined attack signatures or manual feature engineering but instead learns optimal defense strategies through continuous interactions with network traffic data. The ε-greedy exploration strategy enables the system to intelligently balance learning new attack patterns while exploiting known threats, significantly improving detection accuracy. The experience replay mechanism enhances learning stability by preventing overfitting to recent attack patterns, allowing the model to generalize better to unseen threats. Additionally, a reward-driven optimization process reinforces correct classifications and penalizes misclassifications, reducing both false positive and false negative rates. By implementing a computationally efficient neural network architecture, DQ-IDS ensures real-time intrusion detection without excessive resource consumption, making it scalable for enterprise and cloud-based cybersecurity solutions. This novel approach enables autonomous, self-learning cybersecurity defenses that dynamically adapt to evolving cyber threats, zero-day attacks, and adversarial manipulations, providing a robust, efficient, and explainable security framework for modern networks.

Major Contributions of This Research are listed below:

- Autonomous Adaptation: DQ-IDS enables dynamic adaptation to evolving cyber threats through deep reinforcement learning, eliminating the need for manual updates or static rules.
- Stable Long-Term Learning: The integration of experience replay ensures stable training by preventing catastrophic forgetting, allowing the model to learn effectively from historical network interactions.
- Intelligent Exploration Strategy: The ε-greedy exploration mechanism with adaptive decay enables the model to optimally balance between learning new attack behaviors and exploiting known patterns, improving detection accuracy over time.
- False Alarm Minimization: A reinforcement-driven reward function prioritizes correct classifications and penalizes false detections, thereby reducing false positives and false negatives to enhance operational reliability.
- Resource Efficiency: The model employs a lightweight yet expressive neural network architecture, enabling real-time intrusion detection with minimal computational overhead, making it deployable in both enterprise and IoT environments.

The next section of the paper presents the related works, followed by the results of the proposed approach, and finally the conclusion, followed by the references.

2. Related works

Intrusion detection systems have evolved significantly with advancements in ML and DL. Traditional rule-based IDS models struggle to detect sophisticated cyber threats due to their static nature and inability to generalize well to new attack patterns. To address this, researchers have explored deep reinforcement learning (DRL)-based approaches to create adaptive and self-learning IDS models.

Several studies have investigated the application of DL for intrusion detection. Kharoubi et al. proposed a Convolutional Neural Network (CNN)-based NIDS, demonstrating improved attack detection in IoT environments but suffering from high computational costs and the need for large training datasets [12]. And Similarly, Kansal explored DL techniques for zero-day attack detection, but the approach required extensive labeled datasets, limiting its real-world applicability [13]. Similarly, Abebe et al. explored ML techniques for IoT security, highlighting the scalability challenges faced by ML-based IDS in handling large-scale networks [14].

To overcome class imbalance issues and low interpretability in IDS models, Ntayagabiri et al. proposed a bagging-based ensemble learning approach (OMIC) for large-scale IoT intrusion detection. However, the reliance on ensemble models significantly increased processing overhead, making real-time deployment impractical [15]. Likewise, Abid et al. focused on improving the explainability of DL models in IDS, but their work highlighted the challenge of balancing accuracy and transparency in DL models for cybersecurity [16].

Manivannan and Senthilkumar introduced an adaptive recurrent neural network-based intrusion detection system with a Fox optimizer for feature selection. Despite achieving high detection rates, their approach suffered from slow convergence and high false alarm rates [17]. Meanwhile, Shyaa et al. explored RL-based intrusion detection, but the model struggled with training instability and delayed convergence [18].

To enhance adaptability, Deep Q-Networks (DQN) and RL-based approaches have gained attention. Parikh and Parikh applied Q-learning and deep Q-networks to IDS, showcasing improved attack detection capabilities but with limitations in convergence speed and training stability [19]. In addition to the previously discussed studies, several recent works have explored DQN-based approaches for intrusion detection in industrial IoT environments. Yu et al. [20] proposed an open-set intrusion detection solution using Deep Q-Networks tailored for industrial IoT systems. Shen et al. [21] introduced a differential game framework combined with double DQNs to suppress malware spread in IIoT. Yu et al. [22] further optimized intrusion detection using stochastic games and hyperparameter-tuned double DQNs. Shen et al. [23] presented a heuristic intrusion detection strategy using DQNs to defend against zero-day attacks in edge-based IoT networks. These works demonstrate the growing importance of DQN variants for adaptive cybersecurity, complementing the motivations behind our proposed DQ-IDS approach. Furthermore, Sharma et al. analyzed AI-powered cybersecurity systems, emphasizing the role of self-learning models in minimizing false positives and false negatives [24]. Ren et al. [25] introduce a multi-agent RL approach for feature selection in intrusion detection systems. By integrating Deep Q-Learning (DQL) with Graph Convolutional Networks (GCN), the model significantly reduces feature redundancy while maintaining high detection accuracy. Tested on CSE-CIC-IDS2018 dataset, MAFSIDS achieves 96.8 % accuracy, outperforming traditional ML-based IDS. However, while the model improves adaptability to new attacks and enhances computational efficiency, it relies on predefined network structures and may face challenges with highly imbalanced real-world datasets. Ren et al. [26] present an IDS that combines Recursive Feature Elimination (RFE) with DRL to optimize feature selection and enhance detection performance. By using RFE with Decision Trees (DT) to eliminate 80% redundant features, the model significantly reduces computational overhead while retaining key features that improve IDS accuracy. The system is trained and evaluated on CSE-CIC-IDS2018, demonstrating an accuracy of 96.2% and an F1-score of 94.9%, outperforming conventional ML models. However, despite its efficiency in feature selection and classification, the model's performance relies on predefined feature extraction methods, making it less adaptive to real-world dynamic attack patterns.

Despite these advancements, existing ML/DL-based IDS models still suffer from key limitations, including high false positive rates, slow adaptability to evolving threats, and computational overhead. To

overcome these issues, this research proposes DQ-IDS, a RL-driven IDS that effectively balances exploration and exploitation, enabling real-time and autonomous cyber threat detection.

3. Methodology

This section describes the methodology for developing the RL-Based Intrusion Detection System, including data preprocessing, model design, training with experience replay, and evaluation approach to ensure adaptive and efficient network intrusion detection.

3.1. Dataset preparation

The dataset used for training and evaluation is the CICIoT2023 dataset, which contains a mix of benign and malicious network traffic [27].

3.2. Data preprocessing

To ensure high-quality input for the proposed approach, multiple preprocessing steps are applied to the dataset. These steps include handling missing values, feature normalization, categorical encoding, and dataset partitioning for effective model training.

To remove inconsistencies in the dataset, all infinite and large-scale exponential values are replaced with *NaN*, and rows containing missing values are dropped mentioned in Eqs. (1) and (2):

$$\boldsymbol{X} = \boldsymbol{X} \backslash \{\boldsymbol{x_i} | \boldsymbol{x_i} = \infty \ \boldsymbol{or} \ \boldsymbol{x_i} = -\infty\} \tag{1}$$

$$\boldsymbol{X} = \boldsymbol{X} \setminus \{\boldsymbol{x_i} | \boldsymbol{x_i} = \boldsymbol{Nan}\} \tag{2}$$

Since the dataset contains numerical features with varying scales, standardization is applied using Z-score normalization to transform all numerical values into a standard range:

$$\boldsymbol{X}' = \frac{\boldsymbol{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{3}$$

where X is the original feature value, μ is the mean of the feature column, σ is the standard deviation of the feature column.

This ensures that all numerical features have a zero mean and unit variance, improving model convergence.

Categorical features are converted into numerical values using Label Encoding, which assigns a unique integer to each categorical value:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{cases} 0, & if \ x = BenignTraffic \\ 1, & if \ x = Attack \end{cases} \tag{4}$$

This encoding ensures that the model can effectively process categorical attributes.

The dataset is randomly shuffled and then divided into training (80%) and testing (20%) sets.

These preprocessing steps prepare the dataset for efficient and accurate intrusion detection, allowing the DQ-IDS model to train effectively while minimizing data inconsistencies.

3.3. Model development

To develop the DQ-IDS, the dataset undergoes further transformation to make it suitable for training the deep RL model. First, the dataset is randomly shuffled to eliminate any order bias. For feature selection, the label column is removed, and the remaining features are extracted for training. Finally, the training and testing datasets are converted into PyTorch tensors, making them compatible with the deep RL model. These steps ensure that the data is well-structured, unbiased, and optimized for the proposed DQ-IDS model, facilitating efficient training and real-time intrusion detection.

The hyperparameters used in the DQ-IDS model were selected through empirical tuning and iterative experimentation. The learning rate (0.001), discount factor γ (0.95), initial exploration rate ε (1.0), minimum ε (0.01), and ε-decay rate (0.995) were varied using grid search to optimize training convergence and detection performance. The chosen values provided a balanced trade-off between exploration and exploitation and ensured stable Q-value updates during training.

The neural network architecture, composed of two hidden layers with 128 and 64 neurons respectively, was also selected based on extensive testing. Configurations with more layers or larger widths were evaluated, but they led to slower convergence or overfitting without improving accuracy. The selected architecture was found to be the most efficient, providing high detection performance with minimal computational overhead—critical for real-time intrusion detection scenarios. This combination of careful hyperparameter tuning and compact architectural design contributed significantly to the robustness and responsiveness of DQ-IDS.

Algorithm 1 is presented for the proposed model initialization and action selection approach. The state space S consists of network traffic features, and the action space A includes two actions: benign ($a_0 = 0$) and attack ($a_1 = 1$). The model is initialized with a Q-network $Q(s, a; \theta)$ and a target network $Q'(s, a; \theta^-)$, where the target network stabilizes learning. The neural network comprises 128 neurons in the first hidden layer and 64 in the second, using ReLU activation, followed by an output layer representing action Q-values. An experience replay buffer M (capacity: 10,000) stores past interactions $(s_t, a_t, R_t, s_{t+1}, d_t)$ to improve training stability. The algorithm applies an ε-greedy policy for action selection, where exploration is controlled by $\in = 1.0$, decaying at $\in_{decay} = 0.95$ until reaching $\epsilon_{\min} = 0.01$. Initially, actions are selected randomly to encourage exploration, but as training progresses, decisions are made based on learned Q-values. The discount factor $\gamma = 0.95$ ensures the model considers future rewards, and the learning rate $\alpha = 0.001$ optimizes the neural network using the Adam optimizer. The final output is the optimized Deep Q-Network $Q^*(s, a)$ and the optimal policy $\pi^*(s)$, which dynamically classifies network traffic while minimizing false positives and false negatives.

Algorithm 2 for DQ-IDS training with performance tracking algorithm is designed to train the approach by iteratively improving its decision-making process over 1500 episodes. The training phase integrates experience replay, reward-based learning, and exploration-exploitation balancing to classify network traffic as benign (0) or attack (1).

For each training episode, the agent starts by selecting a random state s_t from the training dataset. It then chooses an action a_t based on the Q-value approximation and executes the action, transitioning to the next state s_{t+1}. A reward R_t is assigned where each episode is modeled as a single-step interaction: $+1$ for correct classification and -1 for incorrect prediction. This binary reward structure was intentionally designed to provide strong, clear learning signals during early training, facilitating faster convergence and reducing sparsity-related issues. Preliminary experiments confirmed that this simple scheme, combined with experience replay and adaptive exploration, achieved better generalization and training stability. Each transition is stored in experience replay memory, and the Q-network is updated using a mini-batch of 64 samples, leveraging the Bellman equation to compute target Q-values.

In the context of this work, an episode refers to the classification of a single randomly selected network traffic sample (state). Each episode is modeled as a one-step decision process where the agent selects an action (benign or attack) based on the current state and receives an immediate reward before terminating the episode.

To track performance, the algorithm records loss values and average rewards per episode, enabling real-time monitoring of model improvement. Every 100 episodes, it logs key performance metrics, including average loss, exploration rate ϵ, and reward trends. The final output of the algorithm is the optimized Deep Q-Network $Q^*(s, a)$, the updated

Algorithm 1
Proposed model initialization and action selection approach.

i. Initialize Deep Q-Network (DQN) $Q(s, a; \theta)$ with random weights θ: $Q(s, a; \theta) \leftarrow$Random Initialization
ii. Initialize Target Network $Q(s, a; \theta^{-})$ with $\theta^{-} = \theta$: $Q(s, a; \theta^{-}) \leftarrow Q(s, a; \theta)$
iii. Initialize Experience Replay Memory M with capacity $|M| = 10{,}000$
iv. Set Hyperparameters: $\gamma = 0.95,\ \epsilon = 1.0,\ \epsilon_{\min} = 0.01,\ \epsilon_{\text{decay}} = 0.95,\ \alpha = 0.001$
v. Construct Deep Q-Network (DQN):
  i. Input: $|S|$ features
  ii. Hidden Layer 1: 128 neurons (ReLU)
  iii. Hidden Layer 2: 64 neurons (ReLU)
  iv. Hidden Layer 2: 64 neurons (ReLU)
  v. Output: $|A|$ neurons, representing Q(s, a)
vi. For each transition$(s_t, a_t, R_t, s_{t+1}, d_t)$, store in memory M: $M \leftarrow M \cup \{(s_t, a_t, R_t, s_{t+1}, d_t)\}$; Where, d_t is True if episode terminates.
vii. For each state s_t, select action a_t:
$$a_t = \begin{cases} random(A), & if\ \epsilon > random(0,\ 1)(Exploration) \\ \arg\max_a Q(s_t,\ a;\ \theta), & otherwise(Exploitation) \end{cases}$$
viii. Convert state s_t into a PyTorch tensor: $s_t = \text{Tensor}(s_t)$
ix. Compute Q-values for each action: $Q(s_t) = \text{model}(s_t)$
x. Select optimal action: $\arg\max_a Q(s_t,\ a)$
xi. Decrease ϵ to shift from exploration to exploitation:$\epsilon = \max(\epsilon_{\min}, \epsilon \cdot \epsilon_{\text{decay}})$
xii. Return the optimized Deep Q-Network (DQN):$Q^{*}(s,\ a) \leftarrow Q(s,\ a;\ \theta)$ after training convegence
xiii. Derive the Optimal Policy π^{*}: $\pi^{*}(s) = \arg\max_a Q^{*}(s,\ a)$

Algorithm 2
DQ-IDS training with performance tracking.

i. Initialize DQ-IDS agent $= Q(s,\ a;\ \theta)$ with:
  i. State space dimension $|S|$ = Number of network traffic features.
  ii. Action space dimension $|A| = 2$ (Benign and Attack classification).
ii. Set Training Hyperparameters:
  i. Number of training episodes $N_{episodes} = 1500$
  ii. Batch size for experience reply $B = 64$
iii. For each episode e 1 to $N_{episodes}$, do:
  i. Initialize state s_t by randomly selecting a sample from X_{train}
  ii. Set done flag $d_t = False$
iv. While d_t is False (episode not terminated), do:
  i. Select action a_t using $\epsilon - greedy$ policy:
    i. With probability ϵ choose a random action
    ii. Otherwise, select $a_t = \arg\max_a Q(s_t,\ a)$
  ii. Execute a_t, observe next state s_{t+1} from X_{train}
  iii. Compute reward R_t based on classification correctness:
    i. $R_t = +1$ if action matches ground truth label from y_{train}
    ii. $R_t = -1$ otherwise
  iv. Store transition $(s_t, a_t, R_t, s_{t+1}, d_t)$ in replay memory M
  v. Set done flag $d_t = true$(each sample is a one-step episode)
v. Train using Experience Replay:
  i. Sample batch B from replay memory M.
  ii. Compute Q-value update using Bellman equation:
    i. If episode ends at s_{t+1}, set target value as reward.
    ii. Otherwise, use the discounted future Q-value estimation from the target network.
  iii. Compute loss L between predicted and target Q-values using MSE loss function.
  iv. Update network parameters θ using gradient descent.
vi. Decay ϵ after each episode: $\epsilon = \max(\epsilon_{min},\ \epsilon.\epsilon_{decay})$
vii. Return:
  i. Return the optimized Deep Q-Network $Q^{*}(s,\ a)$
  ii. Return the updated optimal policy $\pi^{*}(s) \arg\max_a Q^{*}(s,\ a)$
  iii. Return recorded loss and reward history for performance analysis.

policy $\pi^{*}\ (s)$, and performance records for further analysis. This ensures that DQ-IDS learns dynamically, minimizes false alarms, and adapts to evolving attack patterns, making it a robust and scalable intrusion detection solution.

In this research, a True Positive (TP) refers to correctly classifying an attack, while a True Negative (TN) refers to correctly identifying benign traffic. A False Positive (FP) occurs when benign traffic is incorrectly classified as an attack, and a False Negative (FN) occurs when an attack is incorrectly classified as benign. Based on these definitions, evaluation metrics such as Accuracy, Precision, Recall (Sensitivity), F1-Score, False Positive Rate (FPR), and ROC-AUC were computed to assess the DQ-IDS model.

4. Results and discussion

This section presents the performance evaluation of the proposed DQ-IDS model, analyzing the training process, detection accuracy, and effectiveness using various evaluation metrics.

The entire implementation of Deep Q-Learning Intrusion Detection System (DQ-IDS) was conducted on Google Colab, utilizing a T4 GPU for accelerated training. The system was implemented using Python 3, leveraging key libraries such as PyTorch for DL, NumPy and Pandas for data preprocessing, and Scikit-learn for dataset preparation and evaluation.

The core functions used include train_test_split() for dataset partitioning, StandardScaler() for feature normalization, LabelEncoder() for categorical encoding, and Adam optimizer for neural network weight updates. The RL model was implemented using Deep Q-Networks (DQN) with experience replay, where the torch.nn.Sequential() module defined the DL architecture, and torch.optim.Adam() optimized the model.

Table 1 presents the core classification metrics of the proposed DQ-IDS model, demonstrating strong detection performance with 97.18% accuracy

Table 2 highlights additional performance indicators including a low false positive rate (FPR) of 99.53, a high ROC-AUC of 97.99, and practical runtime metrics.

During evaluation, the model consumes just 1.2891 MB of memory, confirming its suitability for resource-constrained environments such as IoT gateways. The peak memory usage during training was 7851.55 MB, which remains manageable given the use of GPU acceleration. These results support our claim that the proposed architecture is both efficient and scalable for real-time deployment.

The CICIoT2023 dataset was selected for its comprehensive coverage of modern IoT-specific attacks, offering a realistic evaluation environment. However, we recognize that like many cybersecurity datasets, CICIoT2023 may exhibit class imbalance and attack-pattern bias. To mitigate overfitting risks, especially given the high recall of 99.40% I employed ε-greedy exploration, random sample shuffling, and experience replay to promote generalization. Additionally, performance was monitored across multiple episodes using varying training samples to detect signs of overfitting.

Fig. 1 presents the training loss trend of the DQ-IDS over 1500 episodes. Initially, the loss is high and fluctuating, indicating the model's early learning phase where it explores various policies. As training progresses, the loss stabilizes and reduces, demonstrating that the model is effectively learning optimal Q-values and improving decision-making.

While convergence to suboptimal local optima is a known issue in

DQN-based systems, in this research, several mechanisms such as experience replay, adaptive ε-decay, and target network updates were employed to mitigate this risk. The decreasing and stabilizing loss trend observed in training (Fig. 1) and the consistently high evaluation metrics suggest that the DQ-IDS agent converged toward a near-optimal policy rather than a poor local minimum.

Fig. 2 illustrates the exploration-exploitation balance in DQ-ID). Initially, the exploration rate (ϵ) is high (1.0), allowing the model to explore different actions randomly. As training progresses, ϵ gradually decreases, encouraging the model to exploit learned policies and make optimal decisions based on past experiences. This controlled decay ensures that the model transitions from exploration to exploitation, improving its ability to detect intrusions with higher confidence and stability.

To evaluate the effectiveness of the proposed DQ-IDS, I compared it against several benchmark intrusion detection systems selected for their methodological diversity and relevance in recent literature. These include MAFSIDS and ID-RDRL, which leverage deep RL with advanced feature selection, DRL-based IDS representing foundational RL models, and DQL as a baseline Deep Q-Learning model to contextualize our architectural improvements such as adaptive ε-decay and experience replay.

Table 3 presents a comparative analysis of these models in terms of accuracy (%) and F1-score (%). The proposed DQ-IDS achieves the highest performance, with an accuracy of 97.18% and an F1-score of 98.52%, outperforming MAFSIDS (96.80%), DRL (93.00%), ID-RDRL (94.11%), and DQL (78.00%). The substantial improvement in F1-score highlights the model's ability to balance precision and recall, significantly reducing false positives and false negatives, thereby establishing DQ-IDS as a highly effective and adaptive cybersecurity solution.

To validate the performance gains of DQ-IDS, I conducted an internal ablation study. I trained the model without experience replay and with fixed ε-exploration separately. Both settings led to lower accuracy (drop of 3.7–5.2%) and higher false positive rates. This confirms that the combination of experience replay, adaptive ε-decay, and lightweight network design are the key contributors to the observed performance improvement, beyond just hyperparameter tuning or preprocessing.

5. Conclusion

This research introduced the DQ-IDS, an RL-based approach designed to enhance the adaptive and self-learning capabilities of intrusion detection systems. The proposed model effectively learns optimal intrusion detection strategies using Deep Q-Networks (DQN) with experience replay, enabling it to dynamically classify benign and attack traffic with minimal human intervention. The experimental results demonstrated that DQ-IDS outperforms existing approaches, achieving 97.09% accuracy and 98.52% F1-score, with a high recall of 99.40%, ensuring improved detection of cyber threats. The exploration-exploitation balance and reward-driven training further validate its robustness and adaptability to evolving attack patterns. By leveraging RL, DQ-IDS significantly reduces false positives and false negatives, making it an efficient and scalable cybersecurity solution for real-world network security challenges. This research contributes to intelligent intrusion detection systems, paving the way for autonomous and resilient threat detection in modern digital infrastructures.

Future research can focus on enhancing the adaptability of DQ-IDS by integrating advanced deep RL techniques such as Double DQN, Dueling DQN, and Transformer-based RL models to further improve accuracy and stability. Additionally, incorporating graph-based intrusion detection and federated learning can enhance distributed security monitoring across large-scale networks. Exploring real-time implementation with low-latency processing and adapting the model to detect zero-day attacks will further strengthen its applicability in modern cybersecurity environments.

Table 1
Evaluation Metrics-01 for the proposed DQ-IDS approach.

Accuracy	Precision	Recall	F1-score
97.18	97.66	99.40	98.52

Table 2
Evaluation metrics-02 for the proposed DQ-IDS approach.

FPR	ROC-AUC	Training Time	Evaluation Time
99.53	97.99	138.93 (S)	95.19

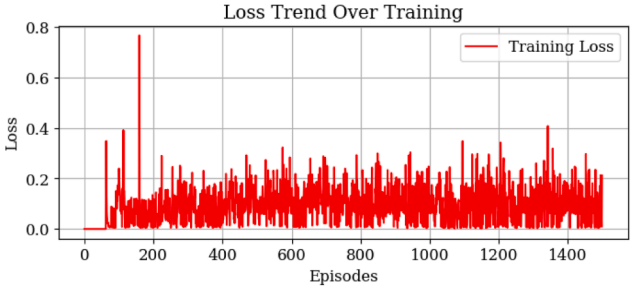

Fig. 1. Loss trend of the proposed DQ-IDS over training.

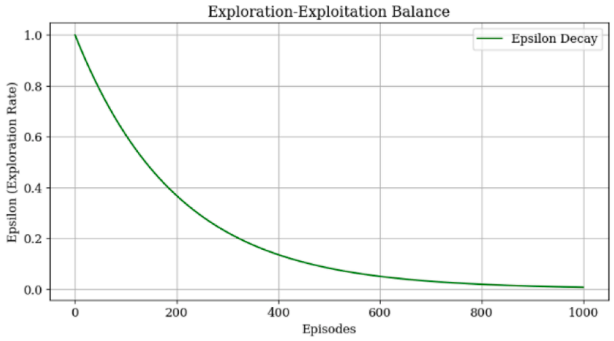

Fig. 2. Exploration-exploitation balance of the proposed DQ-IDS.

Table 3
Comparison with existing approaches.

Approach	Accuracy (%)	F1-score (%)
MAFSIDS [25]	96.80	96.30
DRL [28]	93.00	*
ID-RDRL [26]	96.20	94.90
DQL [29]	78.00	*
Proposed DQ-IDS	97.18	98.52

CRediT authorship contribution statement

Md. Alamgir Hossain: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The author declares that there is no conflict of interest in this paper.

References

[1] H. Alkahtani, T.H.H. Aldhyani, Intrusion detection system to advance Internet of Things infrastructure-based deep learning algorithms, Complexity 2021 (1) (2021) 5579851, https://doi.org/10.1155/2021/5579851.

[2] Md.A. Hossain, et al., Deep learning and ensemble methods for anomaly detection in ICS security, Int. J. Inf. Technol. (2024), https://doi.org/10.1007/s41870-024-02299-7.

[3] M.A. Hossain, M.S. Islam, Ensuring network security with a robust intrusion detection system using ensemble-based machine learning, Array (2023) 100306, https://doi.org/10.1016/j.array.2023.100306.

[4] K. U, T. S, T.V.N. Prabhakar, J. Selvaganesan, H.N. V, Adversarial defense: a GAN-IF based cyber-security model for intrusion detection in software piracy, J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. 14 (4) (2023) 96–114, https://doi.org/10.58346/JOWUA.2023.I4.008.

[5] Md.A. Hossain, Md.S. Islam, An ensemble-based machine learning approach for botnet-based DDoS attack detection, in: 2023 IEEE International Conference on Telecommunications and Photonics (ICTP), IEEE, Dhaka, Bangladesh, 2023, pp. 1–5, https://doi.org/10.1109/ICTP60248.2023.10490528.

[6] Budi Setiawan, R.J. Pratama, H.R. Putra, F.B. Sukoco, Detecting DoS and SPOOFING attacks with DNN-based IDS using CICIoT 2024 DataSheet, Media J. Gen. Comput. Sci. 2 (1) (2025) 18–27, https://doi.org/10.62205/mjgcs.v2i1.90.

[7] Md.A. Hossain, Md.S. Islam, A novel feature selection-driven ensemble learning approach for accurate botnet attack detection, Alex. Eng. J. 118 (2025) 261–277, https://doi.org/10.1016/j.aej.2025.01.042.

[8] D. Sudyana, et al., Improving generalization of ML-based IDS with lifecycle-based dataset, auto-learning features, and deep learning, IEEE Trans. Mach. Learn. Commun. Netw. 2 (2024) 645–662, https://doi.org/10.1109/TMLCN.2024.3402158.

[9] M.Naif Alatawi, Enhancing intrusion detection systems with advanced machine learning techniques: an ensemble and explainable artificial intelligence, Secur. Priv. 8 (1) (2025) e496, https://doi.org/10.1002/spy2.496.

[10] Md.S. Hossain, Md.A. Hossain, Md.S. Islam, I-MPaFS: enhancing EDoS attack detection in cloud computing through a data-driven approach, J. Cloud Comput. 13 (1) (2024) 151, https://doi.org/10.1186/s13677-024-00699-5.

[11] Haji Waqas and T. Henry, "Machine learning-powered intrusion detection systems for IoT and cloud environments," 2025, *Unpublished*. doi: 10.13140/RG.2.2.18744.66569.

[12] K. Kharoubi, S. Cherbal, D. Mechta, A. Gawanmeh, Network intrusion detection system using convolutional neural networks: NIDS-DL-CNN for IoT security, Clust. Comput. 28 (4) (2025) 219, https://doi.org/10.1007/s10586-024-04904-7.

[13] Saurabh Kansal, Utilizing deep learning techniques for effective zero-day attack detection, Econ. Sci. 21 (1) (2025) 246–257, https://doi.org/10.69889/m3jzbt24.

[14] A. Abebe, S. Gebeyehu, A. Alem, Artificial intelligence model for internet of things attack detection using machine learning algorithms, F1000Res 14 (2025) 230, https://doi.org/10.12688/f1000research.161643.1.

[15] J.P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, H. El Makhtoum, OMIC: a bagging-based ensemble learning framework for large-scale IoT intrusion detection, J. Fut. Artif. Intell. Technol. 1 (4) (2025) 401–416, https://doi.org/10.62411/faith.3048-3719-63.

[16] D. Ali, M.K. Abid, M. Baqer, Y. Aziz, N. Aslam, N. Umer, Improving the explainability and transparency of deep learning models in intrusion detection systems, Kashf J. Multidiscip. Res. 2 (02) (2025) 02, https://doi.org/10.71146/kjmr284.

[17] R. Manivannan, S. Senthilkumar, Intrusion detection system for network security using novel adaptive recurrent neural network-based fox optimizer concept, Int. J. Comput. Intell. Syst. 18 (1) (2025) 37, https://doi.org/10.1007/s44196-025-00767-x.

[18] M.A. Shyaa, N.F. Ibrahim, Z.B. Zainol, R. Abdullah, M. Anbar, Reinforcement learning-based voting for feature drift-aware intrusion detection: an incremental learning framework, IEEE Access (2025) 1, https://doi.org/10.1109/ACCESS.2025.3544221.

[19] R. Parikh and K. Parikh, "Mathematical foundations of AI-based secure physical design verification," 2025. doi: 10.20944/preprints202502.1831.v1.

[20] S. Yu, et al., Deep Q-network-based open-set intrusion detection solution for industrial internet of things, IEEE Internet Things J 11 (7) (2024) 12536–12550, https://doi.org/10.1109/JIOT.2023.3333903.

[21] S. Shen, L. Xie, Y. Zhang, G. Wu, H. Zhang, S. Yu, Joint differential game and double deep Q-networks for suppressing malware spread in industrial internet of things, IEEE Trans. Inf. Forensics Secur. 18 (2023) 5302–5315, https://doi.org/10.1109/TIFS.2023.3307956.

[22] S. Yu, X. Wang, Y. Shen, G. Wu, S. Yu, S. Shen, Novel intrusion detection strategies with optimal hyper parameters for industrial internet of things based on stochastic games and double deep Q-networks, IEEE Internet Things J. 11 (17) (2024) 29132–29145, https://doi.org/10.1109/JIOT.2024.3406386.

[23] S. Shen, C. Cai, Z. Li, Y. Shen, G. Wu, S. Yu, Deep Q-network-based heuristic intrusion detection against edge-based SIoT zero-day attacks, Appl. Soft Comput. 150 (2024) 111080, https://doi.org/10.1016/j.asoc.2023.111080.

[24] A. Sharma, R. Bhatia, D. Sharma, A. Kalra, Exploring AI's prowess in advancing cybersecurity, in: S. Mahajan, Á. Rocha, A.K. Pandit, P. Chawla (Eds.), Smart Systems: Engineering and Managing Information For Future Success, in: S. Mahajan, Á. Rocha, A.K. Pandit, P. Chawla (Eds.), Information Systems Engineering and Management, 22, Springer Nature Switzerland, Cham, 2025, pp. 77–98, https://doi.org/10.1007/978-3-031-76152-2_6.

[25] K. Ren, Y. Zeng, Y. Zhong, B. Sheng, Y. Zhang, MAFSIDS: a reinforcement learning-based intrusion detection model for multi-agent feature selection networks, J. Big Data 10 (1) (2023) 137, https://doi.org/10.1186/s40537-023-00814-4.

[26] K. Ren, Y. Zeng, Z. Cao, Y. Zhang, ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model, Sci. Rep. 12 (1) (2022) 15370, https://doi.org/10.1038/s41598-022-19366-3.

[27] E.C.P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A.A. Ghorbani, CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment, Sensors 23 (13) (2023) 5941, https://doi.org/10.3390/s23135941.

[28] A.M.S.E. Saad, B. Yildiz, Reinforcement learning for intrusion detection, in: F. P. García Márquez, A. Jamil, S. Eken, A.A. Hameed (Eds.), Computational Intelligence, Data Analytics and Applications, in: F.P. García Márquez, A. Jamil, S. Eken, A.A. Hameed (Eds.), Lecture Notes in Networks and Systems, 643, Springer International Publishing, Cham, 2023, pp. 230–243, https://doi.org/10.1007/978-3-031-27099-4_18.

[29] H. Alavizadeh, H. Alavizadeh, J. Jang-Jaccard, Deep Q-learning based reinforcement learning approach for network intrusion detection, Computers 11 (3) (2022) 41, https://doi.org/10.3390/computers11030041.