Indonesian Journal of Electrical Engineering and Computer Science
Vol. 43, No. 1, July 2026, pp. 299-313
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v43.i1.pp299-313

Deep Q learning algorithm for detecting DDoS attacks on IoT devices

Lana Kamla Ahmed[1], Kayhan Zrar Ghafoor[2]
[1]Department of Computer Science and Information Technology, College of Science, Salahaddin University-Erbil, Erbil, Iraq
[2]Department of Information and Communication Technology Engineering, Erbil Polytechnic University, Erbil, Iraq

Article Info

Article history:

Received Feb 21, 2026
Revised Jun 10, 2026
Accepted Jun 27, 2026

Keywords:

Deep Q-learning
Distributed denial of service
Internet of things security
Q learning
Reinforcement learning

ABSTRACT

The rapid expansion of internet of things (IoT) networks has heightened security risks, particularly regarding distributed denial of service (DDoS) attacks against devices with limited computing capacity. High detection accuracy is crucial for these resource-constrained environments, where false positives can disrupt legitimate traffic and false negatives allow attacks to persist. However, modern reinforcement learning (RL) and machine learning (ML) intrusion detection solutions often exhibit poor generalization due to static state representations. To address this, this paper proposes a deep Q-learning (DQL) framework that integrates K-means clustering directly into the RL action space. Unlike prior RL-based IDS models, our approach dynamically integrates clustering into the learning process, enabling adaptive state representation and improved generalization to unseen traffic patterns. The system is formulated as a Markov decision process where the agent optimizes a composite reward function based on accuracy, precision, recall, and F1-score. Evaluated on the N-BaIoT dataset using 10-fold cross-validation, the proposed method achieves a classification accuracy of 98.95% and a weighted F1-score of 98.73%, significantly outperforming traditional ML and RL baselines. These results demonstrate the framework's effectiveness as a scalable, adaptive solution for intelligent IoT DDoS detection.

Corresponding Author:

Lana Kamla Ahmed
Department of Computer Scinece and Information Technology, College of Science
Salahaddin University-Erbil
Kerkuk Road, Erbil, Iraq
Email: klanamanatik@gmail.com

1. INTRODUCTION

The rapid evolution of the internet of things (IoT) has transformed how devices interact while exposing systems to advanced security threats. Such attacks, including SYN floods and UDP amplification, are known as distributed denial-of-service (DDoS) attacks that exploit the limited defensive capabilities of IoT hardware, given its resource-constrained nature (2024 statistics show that over 21.3 million attacks are registered) [1]. High detection rates mean everything to such resource-constrained devices; false positives can disrupt legitimate traffic, but false negatives can enable adversarial survival. Conventional security frameworks that center on identity control, access control, and authorization often cannot handle large-scale, dynamic cyberattacks that have become a dominant feature of the IoT [2], [3]. Besides, high rates of malware, packet-sniffing, and spoofing require detection models that can adapt in real time to changing traffic patterns [4], [5].

Although primitive machine-learning (ML) and deep-learning (DL) models, like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders, routinely monitor common metrics of the network, such as the source IP addresses, they tend not to adapt to varying traffic patterns [6]-[8]. Keyboards thermal and energy analysis with exterior mesh screens of office spaces under diverse climatic conditions. But existing reinforcement-based IDS rely on fixed state modelling and lack adaptive feature abstraction [9], [10]. Even though the literature indicates that predictive accuracy has improved only marginally, significant challenges remain. The classical ML pipelines (hybrid K-means combined with artificial neural networks (ANNs) [11], support-vector machines [12], or random-forest ensembles [13]) are often linked to either high computational cost or poor generalization. Similarly, the accuracy of deep-learning models such as DEEP Shield [14], [15], XGBoost [16], and BiLSTM-CNN [17] is also quite reasonable, but they pose significant implementation challenges in resource-limited contexts [18], [19]. At the same time, RL-based models like MADDPG [20] and deep Q-learning (DQL)-IDS [21] are adaptive but are generally tested in synthetic simulations and have not been properly assessed on unseen traffic samples [22]-[25].

This Study fills this gap by making adaptive clustering an integral part of the learning process. Unlike previous models, we also introduce clustering into the agent's decision space, thereby developing an adaptive state representation that generalizes to new traffic topologies. In particular, the K-means algorithm is added to the DQL agent's action repertoire to enable it to identify the optimal number of clusters (K) in real time. As an extension of the non-adaptive preprocessing, we use a composite reward scheme that integrates accuracy, precision, recall, and F1-score to jointly optimize clustering decisions and intrusion-detection policies. This paradigm enables the agent to obtain a higher-level state abstraction and stabilize Q-value convergence, thereby improving the effectiveness of DDoS detection.

The main contributions of the work are summarized as follows: the creation of a DQL architecture that allows creating adaptive representations based on the action space, the design of a new composite mechanism of rewards to optimize them simultaneously, and an empirical validation is conducted with a significant amount of rigor using ten-fold cross-validation on the N-BaIoT dataset to prove the superiority over traditional baselines. The rest of the manuscript will be structured and presented as follows: section 2 will describe the materials and methodology, section 3 will present the empirical findings, and section 4 will present the conclusion and recommendations for future research.

2. METHOD

To maintain transparency, reproducibility, and scientific rigor, the section describes the instruments, materials, and experimental methods applied when carrying out the current study. Figure 1 illustrates the experimental pipeline used in the present research. It starts with the absorption of the N-BaIoT dataset, continues with data preprocessing and feature engineering, and ends with the creation and testing of two reinforcement learning models, i.e., Q-learning and DQL. The two models combine the K-means clustering method and are trained and evaluated using K-fold cross-validation. The pipeline concludes with a comparative performance analysis using predefined evaluation metrics.

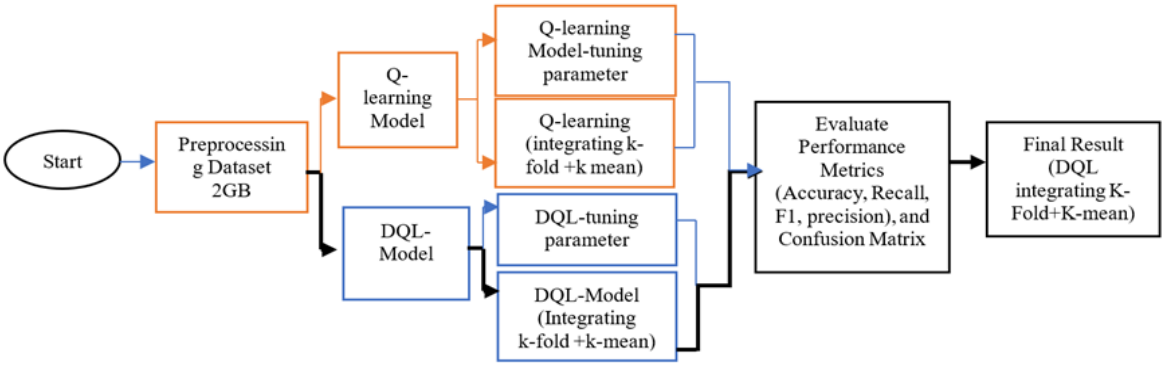

Figure 1. Experimental pipeline integrating adaptive clustering with DQL

2.1. Dataset description

The dataset used to train the proposed model is the NBaIoT dataset, publicly available on Kaggle [26]. This dataset comprises network traffic from nine real-world IoT devices, such as webcams, thermostats,

and baby monitors, captured during regular operation and under a variety of attack conditions, including SYN flood, UDP flood, and data exfiltration. Although the dataset is realistic in terms of the dynamics of a typical IoT device, it has a disproportionate class distribution, with the attack traffic sample significantly larger than that of normal traffic. This imbalance may have an impact on the model training and outputs. Table 1 will be the summary of the key points related to the dataset, such as sample sizes and dataset and traffic feature sizes [26].

Table 1. Characteristics and sample structure (N-BaIoT dataset)

Attribute	Description
Number of samples	7,000,000+
Number of features	115
Traffic types	Benign, Mirai-based, BASHLITE-based, TCP, UDP floods, scans, combos
Type of botnet	Mirai and BASHLITE (a.k.a. Gafgyt)
Classification framing	Binary (benign vs attack) and multiclass (11 total classes)
Key feature domains	Protocol stats, packet rate, ICMP frequency, stream anomalies
Class distribution	Imbalanced - benign dominates
IoT devices types	Collected from different types of IoT real device including Thermostat, Philips Baby monitor, provision security camera, samsung smart TV, Simple Home Cameras, and WeMo Smart Plug, Damini Doorbell, Eco Bee To.

2.2. Data preprocessing

The most important part in the preparation of the datasets was preprocessing. The initial phase of the process was to fill gaps in values with the help of the np. nan to num function, which substitutes missing values with numerical estimates. The additional dimensionality reduction was done through the removal of redundant low-variance and DDoS-detection-irrelevant features, which helped to filter the number of factors that may not contribute to the training of the models. Subsequent normalization of continuous features was performed using Z-score standardization with StandardScaler, which stabilizes the data distribution and improves learning [27]. Another change is that the scikit-learn KBinsDiscretizer was used to generate statistically equalized bin boundaries, thereby providing a deeper state representation for the DQL agent. Specifically, the N-BaIoT dataset is not categorical; however, since the raw input was discretized, the resulting bins were categorical and converted into an enumerated, one-hot format to make them compatible with the underlying learning model.

2.3. Model architecture

The current research outlines the framework for a DQL system implemented as a fully connected feed-forward neural network, as shown in Figure 2. The network has a network input layer, two hidden layers, and an output layer. Nonlinear patterns in the input data are captured by the hidden layers using the rectified linear unit (ReLU) activation function, which has been proven effective in deep learning architectures [28]. Furthermore, to address class imbalance and ensure high detection quality [29], we designed a novel composite reward function. Our reward signal, unlike standard models, combines Accuracy, Precision, Recall, and F1-score ($R = 0.25 * Acc + 0.25 * Prec + 0.25 * Rec + 0.25 * F1$). Although the reward function is computed from the ground-truth labels of the N-BaIoT dataset, this is a tactical design decision for the offline training stage. This enables the agent to learn an optimal classification policy that can ultimately be applied in an autonomous, label-free environment. A multidimensional feedback allows reducing the rate of false positives and false negatives at the same time, which is necessary to guarantee the availability of services in resource-limited IoT settings. The output layer uses a linear activation to estimated Q-values of every possible action available, therefore allowing the agent to categorize the network traffic as benign or malicious according to dynamically estimated cluster assignments.

In order to make the response to changing traffic configurations of the IoT networks more flexible, K-means clustering is used to complement the DQL framework. The process of integration allows states of the networks to be aggregated in a structured form to give state information of the high-dimensional states of the system. Tuples of (state, action, reward, next state) transitions of state are fed into a replay buffer in training to regularize the training process and to reduce correlations amongst related samples. It applies ε-greedy algorithm in order to make a trade-off with exploration and exploitation with the aim of maximizing long-term cumulative returns. Overall, this architecture will allow DQL agent to be trained to adopt useful policies to detect DDoS in IoT environments. The architecture provides a state change sequence, and also uses adaptive cluster, and this makes it possible to learn and detect patterns in high-dimensional network traffic.

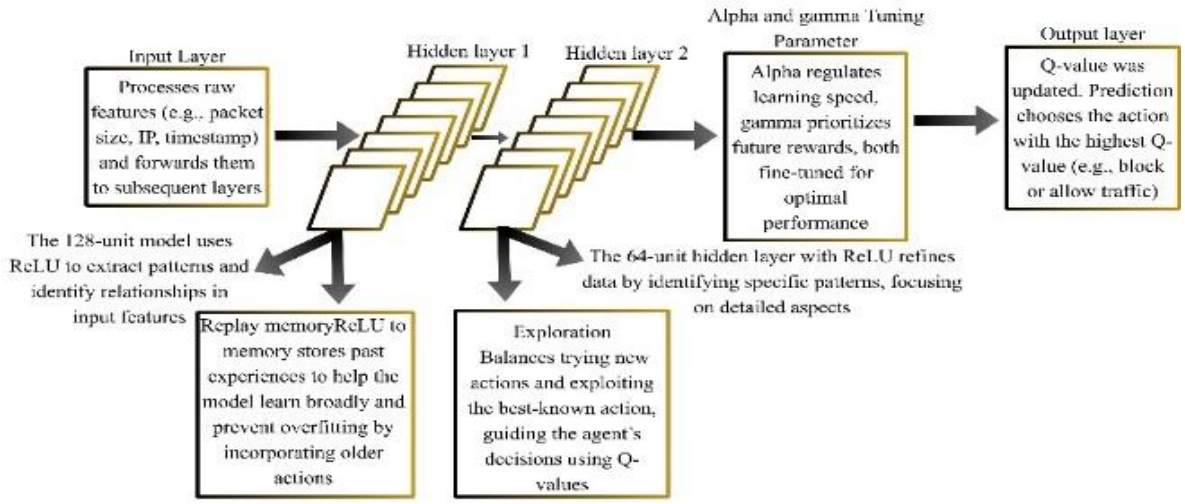

Figure 2. Proposed DQL architecture for dynamic Q-value approximation

2.4. Proposed enhanced DQL detection model

To select and evaluate a research model of DQL, the present paper followed a systematic approach and began by establishing the main hyperparameters, which defined the background of the experiment protocol, i.e., the dataset, learning rate, discount factor, memory size and the number of K-folds. The N-BaIoT was in the first steps preprocessed with the normalization and feature encoding to enhance the quality of data and correct the partial missingness. Following this, the K-fold cross-validation divided the data to form training and cross-validation folds, thus allowing intensive assessment of several folds [30]. This workflow maintains parameter settings in a process as shown in Figure 3. It helps in the ultimate evaluation of aggregate performance measures and, hence, good monitoring of DDoS attacks.

On top of this workflow, the system directly includes a K-means clustering module, the DQL agent, a meta-learner [31]. The agent, as opposed to using the static method of clustering, figures out the optimal number of clusters (K) on the fly, and this is considered to be the action performed by the K-means module. The enabling adaptive approach enables noisy features to be clustered automatically and boosts the separation of classes in complicated traffic. The replay memory (D) is employed to stabilize the network by keeping previous experiences, and an epsilon-greedy policy is employed to make the program balanced in exploring new attack patterns in addition to the previous ones. The whole rationale of the working mechanism of the system, the particular parameter setting, and the design elements are summarized in Table 1 [32] and formalized in Algorithm 1.

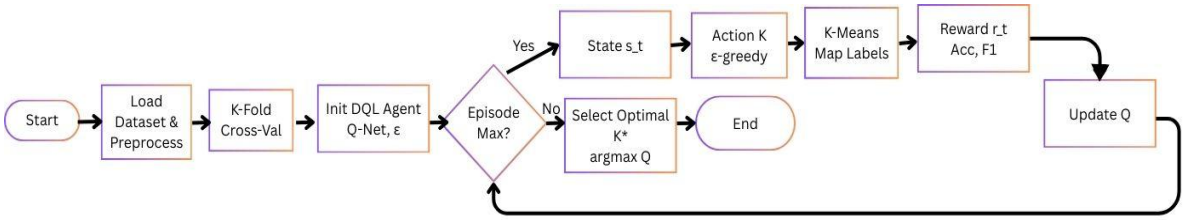

Figure 3. System workflow for agent-based optimal cluster selection

Table 2. DQL framework parameters and design elements summary

Parameter	Definition
State(S)	Discrete network traffic characteristics of the dataset
Action(A)	Choosing the best number of clusters (K) of the K-means module.
Reward (R)	A composite signal: 0.25 *(Accuracy + Precision + Recall + F1-score)
Policy	An epsilon-greedy strategy (starting at 1.0, decaying to 0.01)
Q (st, at)	Current Q-value state and action
α (Learning rate)	Regulating the weight of new information
γ (Discount factor)	Determining the importance of future rewards

Algorithm 1. Enhanced DQL for IoT DDoS detection

```
input: dataset_folder, alpha (α), gamma (γ), ϵ_start, ϵ_decay, ϵ_min,num_episodes,
replaymemory d, clustersize (possible-ks),k_fold
       output: performance metrics, confusion matrix
    1. Initialize replay memory
    2. initialize q-network q(s, a; θ) with random weights
    3. set ϵ = ϵ_start
    4. for each file in dataset_folder do
          4.1 Preprocess x (normalize features, handle missing values)
          4.2 Perform k-fold cross-validation:
              for each fold in k_fold do
                    Split data into xtrain, ytrain, xtest, ytest
                 for episode = 1 to num_episodes do
            4.2.1 set state = get_features(xtrain)
            4.2.2 while training_state exists do
                              if (random(0,1) < ϵ) then
                                      set action = randomaction()
                                else
                                       set action = argmax(q(state))
                             end if
             4.2.2.1 Perform k-means clustering using k = action
             4.2.2.2 Map clusters to labels using majority voting
             4.2.2.3 set acc = calc_accuracy()
                                     set prec = calc_precision()
                                    set rec = calc_recall()
                                     set f1 = calc_f1_score()
             4.2.2.4 set reward = (0.25 * acc) + (0.25 * prec) + (0.25 * rec) + (0.25 * f1)
             4.2.2.5 store (state, action, reward, next_state) in d
             4.2.2.6 update q(state, action) using bellman:
                                        set q = q + α * (reward + γ * max_q - q)
             4.2.2.7 set ϵ = max(ϵ * ϵ_decay, ϵ_min)
                           end while
                      end for
           4.3 set optimal_action = argmax(q(test_state))
           4.4 Evaluate xtest using k-means with k = optimal_action
           4.5 Calculate final metrics and confusion matrix
                 end for
       END FOR
    5. AGGREGATE metrics_across_folds.
    6. RETURN overall metrics,  Confusion Matrix.
```

The DQL model hyperparameters were chosen to guarantee stable convergence and a strong ability to detect different patterns of attacks. The learning rate (0.001) was selected to have a balance between the speed at which the weights are updated; the higher the values, the faster the network would get away, and the lower the values, the sooner the network would learn. The discount factor (γ) was determined to be 0.95 so that the agent could model the future network states, and this is necessary in order to model the time dependencies of a multi-stage DDoS attack. The epsilon-decay schedule begins with 1.0 and decays to 0.01 throughout the first 100 episodes, which allows the agent to explore the high-dimensional features of N-BaIoT broadly at the start before changing on to a stable exploitation policy. The selected values correspond to the best validation results obtained during the experimental evaluation. Specifically, α controls the magnitude of weight updates in Q-network optimization, thereby ensuring pruned learning of knowledge. γ, on the other hand, highlights the future benefits, allowing the agent to take into consideration long-term benefits in the combination of cumulative rewards. The exploration coefficient (ϵ) is the tension between the desires to explore and to exploit in favor of an action selection process that navigates the agent between policy refinement and successive iteration. This is stored in a replay buffer (D) to reduce training instability. Episodic experiences, including actions, states, and rewards, are temporarily stored here, allowing minibatches to be stochastically extracted and reducing correlations inherent in temporally ordered data. The size of the chosen batch determines how many experiences are added to the buffer each time an optimization step is performed, which, in turn, influences the convergence rate and overall learning stability. Besides, the ε -decay schedule and minimal exploration threshold (θ) define the temporal decadence of ε and place an absolute lower limit, ensuring that occasional exploratory behavior is maintained during the training history. It uses the Q-network, which is referred to as Q (s, a, θ), to approximate Q-values of action-state pairs, and θ is the trainable parameter of the neural network. The number of training episodes (Num ep episodes) tells the amount of training. The phase of estimation of the K-means clustering is contingent on the variety of possible candidate cluster values (KS). It gives the agent guidelines on the way to determine the best number of clusters. In order to give strong and objective estimates of the performance, K-fold cross-validation is used on a series of data splits. The set of features (X) and labels of the same (y) are divided into a training (Xtrain, ytrain) and test (Xtest, ytest) set.

Epsilon-greedy strategy is taken to pick the best action and target Q-values are calculated in Bellman equation. The evaluation metric consists of accuracy and F1-score, and, temporarily, is implemented by generating the feedback signal r t of the agent using ground-truth labels. The legality of the approach is the lack of control over the decisions made by the agent in his operation, where learning is promoted only by labelled information. Once the training ends, a confusion table will be created to determine the level of efficiency of the system in separating normal and malicious network traffic. The adaptive clustering is in turn integrated so that it can deal with such a harsh class imbalance of the N-BaIoT data set as the dynamic K would allow the agent to detect minor patterns of attack that could be overlooked by global classification. The batch normalization and dropout experimental tests were done, but neither of the two facilitated convergence speed or stability significantly. They were omitted to make the architecture simple and computationally efficient to the edge devices of the IoT without sacrificing the detection performance.

The proposed algorithm was coded in Python with the help of the TensorFlow framework, and the key components of the simple reinforcement learning of experience replay, reward selection, action selection policy, and successive updates of Q-networks were utilized. The input file was N-BaIoT data, and the K-means provisioning model that was utilized was employed to draw a relation between the actions of the agent. Training was carried out on 100 episodes, and performance was evaluated by using the 10-fold cross-validation in order to add strength and stability to it. Figure 4 also demonstrates the overall workflow and loop of interaction between the agent and environment, including the choice of action, observation of the situation, and Q-value update through the neural network.

The model can communicate with the Internet-of-things environment by monitoring the current situation of the network traffic (S) and making an action choice (A), which, according to an e-greedy exploration policy, evaluates a cluster size of KK. To reward the agent, a reward (R) is obtained when the chosen action is implemented based on the metrics of classification performance, e.g., accuracy and F1-score. Each experience is added to a replay buffer, which is subsequently utilized to update the weights of the Q-network and enhance the learning process by making it more stable and convergent. The way the agent provides feedback on the reward, re-updates Q-values, behaves in the environment, and selects the next action is first described in detail in Figure 4, and illustrates the DQL training loop. This training process involves experience replay and tries to balance between exploration and exploitation, so that it can allow optimal learning of the attack patterns and responsiveness to novel attack patterns.

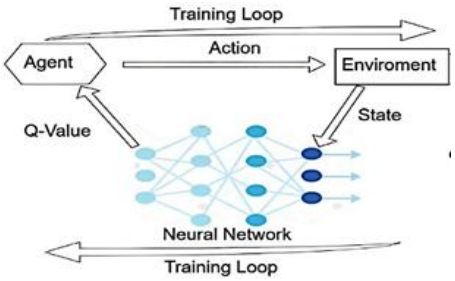

Figure 4. Iterative training loop using composite reward feedback

2.5. Detection and flow of IoT traffic

The proposed DDoS attack-detecting system is deployed as a DQL-based traffic classifier that has the potential to sort the normal and malicious traffic. The overall process of the model is depicted in Figure 5.

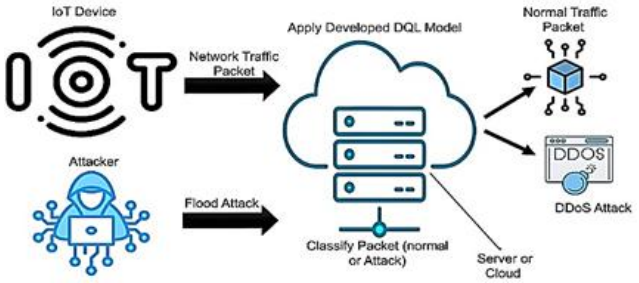

Figure 5. DQL-based model for DDoS detection in IoT traffic

2.6. Experimental equipment

Python 3.12.3 was used as the implementation language for the proposed model, enabling scientific computing and machine-learning-based research in a modern environment. TensorFlow 2.17.0 was used to develop and train the DQL model, as it is mature and compatible with TensorFlow Lite, enabling deployment on resource-limited IoT edge devices. Scikit-learn was used to preprocess and cluster data with K-means, while NumPy and Pandas assisted with data manipulation, and Matplotlib was used for performance visualization. All calculations were performed on a multi-core Intel Core i7-10750H CPU at 2.60 GHz with 16GB of RAM and a 512 GB SSD, simulating a realistic high-end IoT edge gateway. The experimental process involved processing 91 distinct data files from the N-BaIoT dataset. The mean cross-validation time of 65.6 seconds, or 65.6 seconds per fold of cross-validation, implied a total training and cross-validation time of about 3.9 hours in the full course of the 10-fold cross validation. The highest memory consumption was maintained at 500 MB, and the average inference time was 0.05 ms per instance. These values verify that the model is computationally efficient and can be used in low-resource and low-memory IoT applications.

2.7. Performance metrics

Four popular metrics, namely, accuracy, precision, recall, and F1-score, were used to assess the classification performance of the proposed DQL model [33]. Such measures can be used to determine the effectiveness of the model in deciding the presence of malignant and legitimate network traffic in the detection of IoT-based DDoS.

Accuracy is a measure of the fraction of true samples in a set of samples:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

The accuracy is a measure of the number of positives that are predicted correctly out of all the predicted positives:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall: Recall is a ratio that is used to measure how many positive cases are correctly identified out of all the true positive cases:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1-score is the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

All of these metrics represent a strong system that can be used to evaluate the detection potential of the model in particular cases of imbalanced data, when the presence of attack traffic samples may overwhelm the normal data traffic. These metrics are applied in order to compare the proposed DQL-K-means model with multiple baselines, such as Random Forest, CNN, and Q-Learning, in order to guarantee methodological rigor. Also, an ablation study is carried out with the use of the same metrics to check the performance of the model when the adaptive clustering and the elements of composite reward are absent. All results are given as the mean of the 10-fold cross-validation folds in order to consider statistical variability and obtain significance, along with the standard deviation (±σ).

3. RESULTS AND DISCUSSION

In this section, the results of the experiment of the proposed DQL-based framework are provided. This evaluation has been performed based on the N -BaIoT data set and the common performance measures of the N-BaIoT: accuracy, F1-score, true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The convergence visualizations and the temporal progression of the model describe the learning process of the model in a series of episodes and display stability and stable convergence behaviour. Also, measures of computational efficiency were made in terms of time (in seconds) taken to train an example, per-sample inference latency, and memory consumption. All these findings are a comprehensive review of the effectiveness and efficiency of the suggested DQL model.

3.1. Classification performance results

Accuracy, precision, recall, and F1-score are four standard measures that were employed to test the proposed DQL model. The values under these metrics have been calculated through the elements of the confusion matrices because the elements have been summarized in Table 3 [33]. The proposed model has demonstrated a high level of detection and an acceptable proportion of precision and recall, as presented in the results of the experiment. These values of F1-score are also an excellent indication that the model can distinguish between the malicious traffic and the real IoT network traffic. Figure 6 shows the distribution of realized and estimated classes of an average fold of the cross-validation procedure of 10 folds. Figure 6 indicates that false positives were low (21,635) in the model. This is essential to IoT environment since high false positive will result in blocking of legitimate traffic, like smart sensors updates or user commands, to bring down the service. On the other hand, the False Negative (9,896) is low, which means that the majority of malicious DDoS traffic will be applied prior to damaging the equipment. Examining these overlooked instances of attacks has shown that the model sometimes has difficulties stealthy UDP scanning, which eschews the high-frequency of the heartbeat signals of malicious IoT devices. The adaptive quality of the DQL agent however assists in reducing this by changing the cluster representation as the behavior of the attack adapts.

Table 3. Binary classification performance of the confusion matrix

	Predicted positive	Predicted negative
Actual positive	4,500,000	9,896
Actual negative	21,635	2,000,000

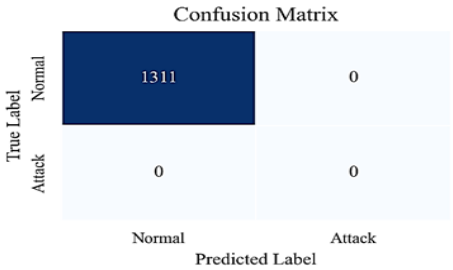

Figure 6. Confusion matrix of the proposed model, which has high values of classification and low values in errors

3.2. Analysis of training evolution and convergence

On the stability of the proposed DQL model, we present measures of performance through the course of all episodes and folds of a ten-fold cross-validation strategy. Also, in training and testing, accuracy and F1-score were employed to test the dynamics of learning in the course of time. Figure 7 shows the training development, in which the accuracy and F1-score level off early during the development. Such a quick convergence is explained by the composite reward function that gives rich feedback to the agent so that it can rapidly differentiate between benign traffic and different DDoS attack vectors in the N-BaIoT data set. These measurements tend to be extremely elevated in majority of the episodes and greatly vary because of personal discrepancy in the experimental functioning of the DQL algorithm. Successful learning can be attributed to this trend, and it shows that the model is convergent in nature.

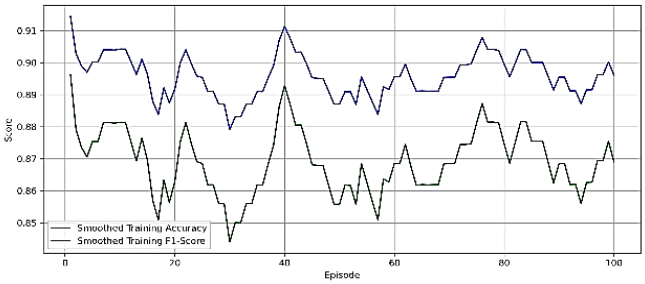

Figure 7. DQL training for accuracy and F1 score

Following the training evaluation, performance measures of testing are illustrated in Figure 8, where accuracy and F1-score are plotted with the various folds of cross-validation of the testing. Figure 8 represents the testing performance of the 10-fold cross-validation. The fact that the variance among the folds is minimal (it is observed in the tight clustering of the data points) proves the fact that the model has good generalization power. This will make the system effective when new, unknown IoT devices or different conditions of network traffic are involved, which were not experienced during the training fold.

The analysis of the progression of the maximum Q-values of the agent over episodes was also monitored in the experiment to examine convergence. Convergence of the learning process is assumed to have been successful in the event of a constant of stable trend in the Q-values; the trend in this case is presented in Figure 9.

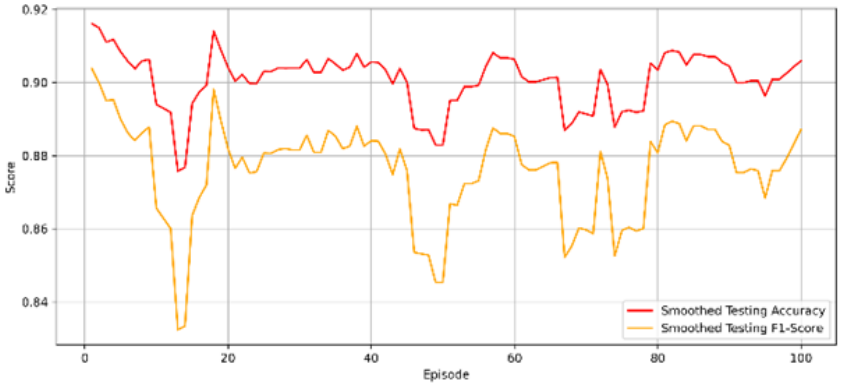

Figure 8. Accuracy and F1-score folds for testing

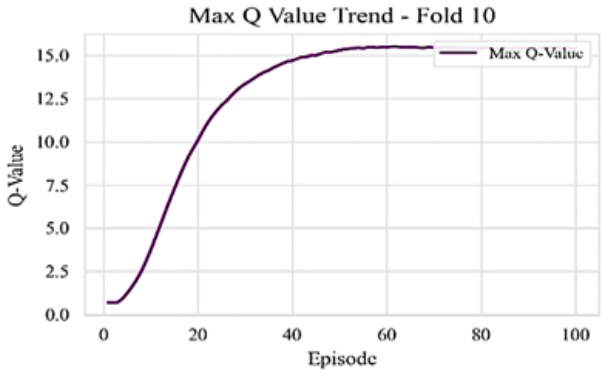

Figure 9. Maximum Predicted Q-value Per Training Episode

Time dynamics of the maximum Q-values in sequential training sessions were seen to test the stability of the learning process. Figure 9 shows the maximum predicted Q-values are varying intra-episode as opposed to a straight line. This is a good sign that the epsilon-decay mechanism is behaving properly, the agent keeps searching a high-dimensional space of IoT traffic without falling into the local optima of the strategy, so that the resulting policy will be resistant to stealthy, adaptable attack executions. Nonetheless, the accuracy and F1-score curves in Figure 9 do not differ between the cross-validation folds, which indicates the presence of good results in terms of learning. The cross-validation fold required approximately 65.6 seconds of wall-clock time; the mean inference time was 0.05ms per instance, and the model occupancy was around 500 MB, indicating the computational efficiency of the model and making it a useful instance to be used in low-memory IoT applications.

3.3. Q-learning results

The Q-learning model proposed has reached the accuracy of the classification of 92.31, precision of 92.31, recall of 92.13, and F1-score of 92.40 by adjusting the learning rate (0.05) and discount factor (0.95) over the number of iterations. To find the best-known parameters, the values of the hyperparameters α and γ were obtained through the intuitive (non-computational) search of a selected hyperparameter space, by

assessing the validation performance. Substantial enhancement of the untuned baseline was obtained by this tuning process. In addition, 10-fold cross-validation (K = 10) using the optimized hyperparameters resulted in an accuracy of 95.24, precision of 95.02, recall of 95.24, and an F1-score of 95.13. Table 4 also gives a more specific comparison of evaluation measures of the proposed Q-learning model as is and after the application of the K-fold and K-means.

Table 4. Q-learning models result

Performance metrics	Q-learning	Q-learning (K-fold and K-mean)
Accuracy	92.31%	95.24%
Precision	92.31%	95.02%
Recall	92.13%	95.24%
F1-Score	92.40%	95.13%

It is shown in the table above that the performance measures of the Q-learning model and the modified model are compared, including K-fold cross-validation and K-means clustering. The cross-validation plus the K-means is fairly successful in enhancing all assessment measures, especially the recall and accuracy. The results of the comparisons are represented in Figure 10; therefore, the focus is put on the better sensitivity of the modified model to identify the individual.

Figure 10 demonstrates the influence of hyperparameter optimization on the performance of a Q-learning model to identify DDoS attacks stating that it generally increases the ability to detect these attacks. Some extra performance would be obtained with K-fold cross-validation and K-Means clustering that would reduce overfitting and provide greater regularity across divisions of data. Nevertheless, their specific deficiencies remain the limits in the precision and their ability to detect attacks, which leads to the necessity to introduce more sophisticated approaches that will be able to work on the new dynamically evolving attack patterns on IoT networks.

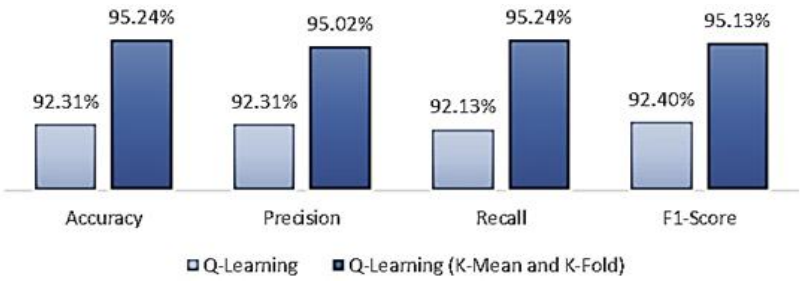

Figure 10. Q-learning performance and integrating (K-Fold and K-Mean)

3.4. DQL results

A learning rate (0.001) was used so that the weight changed without acceleration; a higher value led to a globbing, and a lower value led to non-learning. We have fixed the discount factor (0.95) to enable the agent to capture future network states, which is appropriate to capture the temporal dependencies of multi-stage DDoS attacks. The epsilon-decay schedule begins with an epsilon of 1.0 and decreases to 0.01 in the first 100 episodes, giving a preliminary broad search of the high-dimensional N-BaIoT properties after which the agent is considered to switch to an equilibrium exploitation policy. The result of these optimizations was a baseline performance for the model of 95.83% accuracy, 95.83% recall, 93.75% precision, and an F1-score of 94.44%. Later optimizations included adding K-means clustering to the DQL training process and using K-fold cross-validation. K-means clustering was used to reduce noise and make classes more separable, thereby making the data more amenable to DQL training.

The cross-validation process was performed using K different data partitions, and the model was validated, providing consistency and enabling generalization to new data. The most positive results were achieved with the integrated model, which combined optimized hyperparameters, K-means clustering, and K-fold cross-validation. This setup achieved 98.95% ± 0.17% accuracy, 98.72% ± 0.17% precision, 98.95% ± 0.17% recall, and an F1-score of 98.73% ± 0.17%, which was by far better than the baseline and the improved DQL models. A t-test comparing the F1 Scores from 10-fold cross-validation provided evidence of the statistical significance of this improvement. The p-value obtained from the test was 5.08×10^-5, indicating that

the improvement is statistically significant relative to the baseline Q-learning model. Though the analysis was limited to a single dataset due to computational constraints, the same trend of consistent performance improvement was observed when applied to additional datasets, which in turn indicated the stability of the method and the need for further statistical validation of the analysis. To evaluate the specific contribution of each component, an ablation study was performed as summarized in Table 5. This correlation removes the effects of the composite reward function and the dynamic integration of clustering. As indicated, the integrated final model is highly superior to the typical Q-learning and simple DQL baselines.

Table 5. Performance comparison of incremental model enhancements

Performance metrics measure	Q-learning	Q-learning (K-fold and K-mean)	DQL	DQL-integrating (K-fold and K-mean clustering)
Accuracy	92.31%	95.24%	95.83%	98.95%
Precision	92.31%	95.02%	93.75%	98.72%
Recall	92.13%	95.24%	95.83%	98.95%
F1-Score	92.40%	95.13%	94.44%	98.73%

Table 4 shows performance of the simple Q-learning with the suggested DQL-Integrating model. The integration of the proposed model shown is quite helpful in all measurements as it attained 98.95% accuracy, and 98.73% F1, and a standard deviation of 17.0. This shows K-fold and K-means improvements to be successful and reliable. The outcomes of Figure 11 demonstrate that the better DQL model is supported with the growth of precision, accuracy, F1-score, and recall, proving the efficiency of the K-fold and K-means enhancements.

As Figure 11 demonstrates, K-means clustering is highly effective in reducing the impact of untidy and skewed data, and K-fold cross-validation guarantees effective generalisation to unknown subsets. These findings reflect the need to elaborate on data preprocessing and stringent validation schemes to apply the DQL-based DDoS detection systems to the real IoT environment.

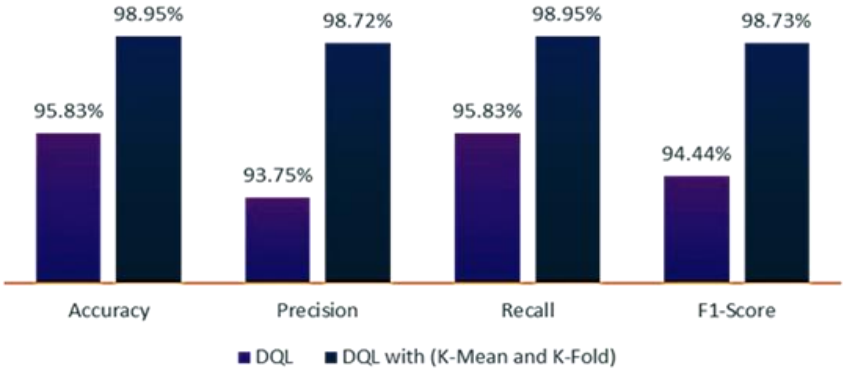

Figure 11. Performance comparison of DQL with and without K-fold and K-means

3.5. Comparison performance

The model, developed based on the K-means clustering algorithm and tested with 10-fold cross-validation, achieved a high, consistent 98.95% accuracy with a low standard deviation of 0.17% on the N-BaIoT data Table 5. The small standard deviation across folds indicates stability and flexibility of the model, suggesting strong generalization to a variety of IoT traffic and attack patterns without overfitting. To benchmark it, the proposed model was compared with recent state-of-the-art approaches evaluated on the same dataset. To illustrate, the optimized machine-learning approach in [34] achieved 99.93 per cent accuracy, whereas the optimal traditional ML model in [35] also reported high performance. Also, a semi-supervised federated learning method [36] achieved an accuracy of 97.30 ± 0.3% in a distributed setting. Although they are accurate in competition, they primarily rely on a fixed classification learned during training, they lack the dynamism to respond to evolving threats or to identify zero-day and unknown anomalies.

Conversely, the suggested model utilizes DQL to offer dynamic and adaptive response optimization, whereas the K-means clustering allows identifying any group of attacks that is not labeled. This can solve some of the major methodological shortfalls presented by purely supervised, non-adaptive systems and offer a more realistic and robust means of real-world environments of smart IoT.

Table 5. Comparison of DDoS detection methods for iot on the N-BaIoT dataset

Ref	Method Used	Accuracy	Dataset	Limitations
[34]	(DT, RF, NN)	99.93%	N-BaIoT	Used predefineddatasets, does not have an adaptable dynamism to modify IoT settings
[35]	(RF, SVM, KNN, DT, NB, LR)	99% (RF, best performer)	N-BaIoT	The traditional ML has the drawback of not utilizing raw and complex network data and high-dimensional feature space
[36]	semi-supervised federated learning	97.30% ,0.49% (in non-IID FL setting)	N-BaIoT	Weak in tackling zero-day threats and does not have dynamically adjustable protection systems
Proposed work	DQL + K-Means + K-Fold	98.95 %	N-BaIoT	Combines DQL and K-means into one strong and generalizable model that can detect DDoS attacks in dynamic traffic

3.6. Discussion

The research achieved significant gains, as uncertain to the suggested DQL-based methodology, as observed in the present research. In all the relevant metrics such as precision, accuracy, F1-score as well as recall, the model outperformed the base Q-learning paradigm, therefore, supporting its reliability and scalability with regards to identifying IoT ecosystem DDoS attacks. The K-mean clustering proved to be particularly important in removing noise and separating classes, particularly with the high-dimensional IoT data, and it achieved this by grouping of similar data points and prioritizing salient properties. This was done to increase the model capability of distinguishing between normal and malicious traffic with greater accuracy. The use of K-fold cross-validation was done to ensure that performance on different subsets of data was consistent, and thus, overfitting was avoided, and the scope of the model in generalization was larger. By including the preprocessing and validation tasks, the DQL model turns out to be a potent and versatile tool to defy complex cybersecurity challenges that typify dynamic IoT networks. Even so, this has a few negative points. Dependency on K - means clustering also carries with it an additional computing burden that may reduce the applicability of the model to real-time systems, especially the applications that concern resources in the IoT system. Moreover, there are false positives (FP) and false negatives (FN) in spite of high accuracy. FP can cause false alarms and can interfere with normal operation, whereas FN will not prevent traffic that is hostile, which presents a real security risk. In order to further prove how strong the model was, it was applied on a bigger scale of about 660 MB, and it still performed highly, 99% accuracy and 99.7% F1-score. Hence, it can be seen that the model can be applied without the fact that it will bring an undesirable effect on its performance to very large-scale datasets.

The weaknesses can be overcome in the future by optimizing the proposed structure to be deployed in real-time and investigating other options for preprocessing. The reduction of the computational load by means of the lightweight clustering or feature selection algorithms could be specifically a solution to the problem of some of these algorithms keeping the detection performance intact. In addition, online reinforcement learning can be used to improve adaptive learning processes because this model can change its behaviour in response to variations in network traffic patterns and attack behaviour. It would be positive to increase the scale of the experiment to cover more types of IoT devices and types of attacks more accurately to gauge the overall generalizability and strength of the model.

Although the current paper will not apply the model to a practical IoT deployment yet, the short inference time and minimal memory usage point to the fact that real-time detection will be a possibility even with relatively low-energy-consuming devices like the Raspberry Dot. The further research would include real-time implementation and testing within the real-life operations of IoT implementation that could involve pruning of the models and other optimization techniques to make it even more efficient. These advancements will render the suggested DQL framework a convenient, multifunctional, and solution-applicable to the IoT cybersecurity.

4. CONCLUSION

The study will need to introduce an advanced paradigm of how to identify DDoS attacks on IoT systems, so that the current practices will become more generalized, flexible, and efficient. The integration will also ensure that variations in the patterns of the IoT traffic will be easier to capture and at the same time ensuring robustness on the dissimilar data partitions. The proposed DQL algorithm is effective in solving critical problems, such as distinguishing classes and detecting more complex types of attacks, including UDP floods, SYN floods, and other volumetric attacks. The experimental findings indicate that the given model can demonstrate much higher performance in comparison with the traditional Q-learning models. Based on the accuracy of their detection as well as the resilience of their DQL model, it is quite high, but this is largely due to the fact that the neural network is in a position to handle even very complex traffic in the IoT. K-means clustering can be used to reduce noise and give a more representative assurance of the states, where

K-fold cross-validation can help to ensure the cross-validation and the generalization of the model related to heterogeneous IoT environments. This means that the suggested model has massive potential to assist in real-time, scalable, and agile DDoS detection options in dynamic IoT networks. The elasticity and robustness of DQL have been particularly in its excellence in solving emerging cybersecurity needs that necessitate counteractions to respond to them.

However, there are still some drawbacks. The additional computational overhead of K-means clustering may make it less usable in resource-constrained or time-sensitive IoT environments. Future research efforts ought to focus on refining preprocessing pipelines to reduce computational costs and on defining reinforcement-learning-based real-time defense mechanisms. In addition, the model will be justified through empirical assessments in situations of unfamiliar and heterogeneous threats in actual IoT implementations.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

This paper's conceptualization, methodology, software development, data curation, validation, formal analysis, investigation, visualisation, project administration, and original draft preparation have been done by Lana Kamal Ahmed. Dr. Kayhan Z. Ghafoor has done the supervision and final approval of this research work.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Lana Kamal	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Kayhan Zrar Ghafoor	✓									✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the Kaggle repository at https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset, reference number [26].

REFERENCES

[1] NETSCOUT, "2024 threat intelligence report: ddos trends in H2 2024," NETSCOUT. Accessed: Mar. 30, 2026. [Online]. Available: https://www.netscout.com/threatreport
[2] O. Yoachimik and J. Pacheco, "Record-breaking 5.6 Tbps DDoS attack and global DDoS trends for 2024 Q4," The Cloudflare Blog. Accessed: Mar. 30, 2026. [Online]. Available: https://blog.cloudflare.com/ddos-threat-report-for-2024-q4
[3] M. Abomhara and G. M. Køien, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, May 2015, doi: 10.13052/jcsm2245-1439.414.
[4] M. Kuzlu, C. Fair, and O. Guler, "Role of artificial intelligence in the internet of things (IoT) cybersecurity," *Discover Internet of Things*, vol. 1, no. 1, p. 7, Dec. 2021, doi: 10.1007/s43926-020-00001-4.
[5] P. S. Saini, S. Behal, and S. Bhatia, "Detection of DDoS attacks using machine learning algorithms," in *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, Mar. 2020, pp. 16–21. doi: 10.23919/INDIACom49435.2020.9083716.
[6] J. Pei, Y. Chen, and W. Ji, "A DDoS attack detection method based on machine learning," *Journal of Physics: Conference Series*, vol. 1237, no. 3, p. 032040, Jun. 2019, doi: 10.1088/1742-6596/1237/3/032040.
[7] A. Singh and B. B. Gupta, "Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web- enabled computing platforms: issues, challenges, and future research directions," *International Journal on Semantic Web and Information Systems*, vol. 18, no. 1, pp. 1–43, Apr. 2022, doi: 10.4018/IJSWIS.297143.
[8] S. Yeom and K. Kim, "Improving performance of collaborative source-side DDoS attack detection," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, IEEE, Sep. 2020, pp. 239–242. doi: 10.23919/APNOMS50412.2020.9237014.

[9] A. Cheema, M. Tariq, A. Hafiz, M. M. Khan, F. Ahmad, and M. Anwar, "prevention techniques against distributed denial of service attacks in heterogeneous networks: a systematic review," *Security and Communication Networks*, vol. 2022, pp. 1–15, May 2022, doi: 10.1155/2022/8379532.
[10] A. K. B. Arnob, M. F. Mridha, M. Safran, M. Amiruzzaman, and M. R. Islam, "An enhanced LSTM approach for detecting IoT-based DDoS Attacks using honeypot data," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, p. 19, Feb. 2025, doi: 10.1007/s44196-025-00741-7.
[11] P. Machaka, O. Ajayi, F. Kahenga, A. Bagula, and K. Kyamakya, "Modelling DDoS attacks in IoT networks using machine learning," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 503 LNICST, 2023, pp. 161–175. doi: 10.1007/978-3-031-35883-8_11.
[12] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106432, Aug. 2023, doi: 10.1016/j.engappai.2023.106432.
[13] O. Ebrahem, S. Dowaji, and S. Alhammoud, "Towards a minimum universal features set for IoT DDoS attack detection," *Journal of Big Data*, vol. 12, no. 1, p. 88, Apr. 2025, doi: 10.1186/s40537-025-01146-1.
[14] O. Kayode and A. S. Tosun, "Deep Q-Network for enhanced data privacy and security of IoT traffic," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, IEEE, Jun. 2020, pp. 1–6. doi: 10.1109/WF-IoT48130.2020.9221318.
[15] S. K. H. Hassan and M. A. Daneshwar, "Anomaly-based network intrusion detection system using deep intelligent technique," *Polytechnic Journal*, vol. 12, no. 2, pp. 100–113, Mar. 2022, doi: 10.25156/ptj.v12n2y2022.pp100-113.
[16] F. Laiq, F. Al-Obeidat, A. Amin, and F. Moreira, "DDoS attack detection in edge-IIoT using ensemble learning," in *2023 7th Cyber Security in Networking Conference (CSNet)*, IEEE, Oct. 2023, pp. 204–207. doi: 10.1109/CSNet59123.2023.10339784.
[17] M. Al-Khafajiy, G. Al-Tameemi, and T. Baker, "DDoS-FOCUS: a distributed DoS attacks mitigation using deep learning approach for a secure IoT network," in *2023 IEEE International Conference on Edge Computing and Communications (EDGE)*, IEEE, Jul. 2023, pp. 393–399. doi: 10.1109/EDGE60047.2023.00062.
[18] A. Berqia, H. Bouijij, A. Merimi, and A. Ouaggane, "Detecting DDoS attacks using machine learning in IoT environment," in *The Sixth edition of the International Conference on Intelligent Systems and Computer Vision (ISCV 2024)*, IEEE, May 2024, pp. 1–8. doi: 10.1109/ISCV60512.2024.10620122.
[19] R. Kale, Z. Lu, K. W. Fok, and V. L. L. Thing, "A hybrid deep learning anomaly detection framework for intrusion detection," in *2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, IEEE, May 2022, pp. 137–142. doi: 10.1109/BigDataSecurityHPSCIDS54978.2022.00034.
[20] D. K. Dake, J. D. Gadze, and G. S. Klogo, "DDoS and flash event detection in higher bandwidth SDN-IoT using multiagent reinforcement learning," in *2021 International Conference on Computing, Computational Modelling and Applications (ICCMA)*, IEEE, Jul. 2021, pp. 16–20. doi: 10.1109/ICCMA53594.2021.00011.
[21] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-learning based reinforcement learning approach for network intrusion detection," *Computers*, vol. 11, no. 3, p. 41, Mar. 2022, doi: 10.3390/computers11030041.
[22] M. Al-Fawa'Reh, J. Abu-Khalaf, P. Szewczyk, and J. J. Kang, "MalBoT-DRL: malware botnet detection using deep reinforcement learning in IoT networks," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 9610–9629, Mar. 2024, doi: 10.1109/JIOT.2023.3324053.
[23] D. Said, M. Bagaa, A. Oukaira, and A. Lakhssassi, "Quantum entropy and reinforcement learning for distributed denial of service attack detection in smart grid," *IEEE Access*, vol. 12, pp. 129858–129869, 2024, doi: 10.1109/ACCESS.2024.3441931.
[24] M. Khayat, E. Barka, M. A. Serhani, F. Sallabi, K. Shuaib, and H. M. Khater, "Reinforcement learning with deep features: a dynamic approach for intrusion detection in IoT networks," *IEEE Access*, vol. 13, pp. 92319–92337, 2025, doi: 10.1109/ACCESS.2025.3569312.
[25] K. Byeon, J. Lim, and J. T. Kwak, "DeLECA: Deblurring for Long and short Exposure images with a dual-branch multimodal cross attention mechanism," *ICT Express*, vol. 11, no. 4, pp. 611–617, Aug. 2025, doi: 10.1016/j.icte.2025.06.003.
[26] Y. Meidan *et al.*, "N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, Jul. 2018, doi: 10.1109/MPRV.2018.03367731.
[27] Y. K. Beshah, S. L. Abebe, and H. M. Melaku, "Drift adaptive online DDoS attack detection framework for IoT system," *Electronics*, vol. 13, no. 6, p. 1004, Mar. 2024, doi: 10.3390/electronics13061004.
[28] E. Suwannalai and C. Polprasert, "Network intrusion detection systems using adversarial reinforcement learning with deep Q-network," in *2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE)*, IEEE, Nov. 2020, pp. 1–7. doi: 10.1109/ICTKE50349.2020.9289884.
[29] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, IEEE, Nov. 2020, pp. 1–6. doi: 10.1109/INMIC50486.2020.9318216.
[30] F. Hussain *et al.*, "A two-fold machine learning approach to prevent and detect IoT botnet attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021, doi: 10.1109/ACCESS.2021.3131014.
[31] G. Chukwukelu, A. Essien, A. I. Salami, and E. Utuk, "Comparative analysis of machine learning techniques for DDoS intrusion detection in IoT environments," in *21st International Conference on Smart Business Technologies*, 2024, pp. 19–27. doi: 10.5220/0012765200003764.
[32] D. J. Richter, R. A. Calix, and K. Kim, "A review of reinforcement learning for fixed-wing aircraft control tasks," *IEEE Access*, vol. 12, pp. 103026–103048, 2024, doi: 10.1109/ACCESS.2024.3433540.
[33] S. Pokhrel, R. Abbas, and B. Aryal, "IoT security: botnet detection in IoT using machine learning." Apr. 06, 2021. [Online]. Available: http://arxiv.org/abs/2104.02231
[34] M. Al-Akhras, A. Alshunaybir, H. Omar, and S. Alhazmi, "Botnet attacks detection in IoT environment using machine learning techniques," *International Journal of Data and Network Science*, vol. 7, no. 4, pp. 1683–1706, 2023, doi: 10.5267/j.ijdns.2023.7.021.
[35] M. Hossain, R. Al Mamun Rudro, R. Razzaque, and K. Nur, "Machine Learning Approaches for Detecting IoT botnet attacks: a comparative study of N-BaIoT dataset," in *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, IEEE, Dec. 2024, pp. 1–7. doi: 10.1109/DASA63652.2024.10836622.
[36] V. T. Nguyen and R. Beuran, "FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection," *Computers and Security*, vol. 151, p. 104337, Apr. 2025, doi: 10.1016/j.cose.2025.104337.

BIOGRAPHIES OF AUTHORS

Lana Kamal Ahmed Biography is a student of MSc in Computer Science and IT at the College of Science, Salahaddin University. She had earlier graduated with a Bachelor of Computer Science from the same university. Lana's research field is the integration of machine learning and cybersecurity methods. Committed to expanding her knowledge and application of technology, Lana is engaged in community initiatives focused on enhancing tech education. She can be contacted at email: klanamanatik@gmail.com.

Kayhan Zrar Ghafoor Biography is Dean of collage of technology. Before that, he was a postdoctoral research fellow at Shanghai Jiao Tong University, where he contributed to two research projects funded by National Natural Science Foundation of China and National Key Research and Development Program. He is also served as a visiting researcher at University Technology Malaysia. He received the BSc degree in Electrical Engineering from Salahaddin University, the MSc degree in Remote Weather Monitoring from Koya University and the PhD degree in Wireless Networks from University Technology Malaysia in 2003, 2006, and 2011, respectively. He has published over 110 scientific/research papers in ISI/Scopus indexed international journals and conferences. Dr Kayhan authored three books named “Cognitive Networks: Applications and Deployments” and "Privacy and Cybersecurity in Smartcities". He is worked as the General Chair of a workshop named Smart Sensor Protocols and Algorithms under the 9th International Conference on Mobile Ad-hoc and Sensor Networks (MSN-2014) which is held in Hungary. He also served as an Associate Editor, Editorial Board Member and reviewer for numerous prestigious international journals, appeared as a workshop general chair for international workshops and conferences, and worked as a TPC member for more than 37 international conferences including GlobCom and InfoCom. He is the receipt of the UTM Chancellor Award at the 48th UTM convocation in 2012. He can be contacted at email: kayhan.ghafoor@epu.edu.iq.