Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 35–42
DOI: 10.15439/2025F1820
ISSN 2300-5963 ACSIS, Vol. 43

Q-ID: A Reinforcement Learning Framework for Adaptive Intrusion Detection

Maisha Maliha and Mohammed Atiquzzaman
School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA
Email: maisha.maliha-1@ou.edu, atiq@ou.edu

***Abstract*—The growing sophistication and frequency of cyber threats in communication networks demand Intrusion Detection Systems (IDS) that adapt to evolving attack patterns. Traditional approaches, based on static rules or purely supervised models, often fail to recognize novel attacks, leaving critical infrastructures exposed. Reinforcement Learning (RL) provides a dynamic alternative by enabling agents to refine detection policies through continuous feedback. In this work, we propose a Q-learning–based Intrusion Detection (Q-ID) system and train it on the CICIDS2017 dataset. The RL formulation defines the state as the flow's feature vector, the action as the classification decision, and the reward as $+1$ for correct predictions and -1 otherwise. To ensure stable convergence, the reward is integrated with cross-entropy loss in a hybrid objective, allowing continued improvement even after the supervised component has plateaued. Unlike prior IDS methods that rely solely on offline supervised training, our approach fuses reinforcement feedback with supervised optimization to support adaptive and robust detection. Experimental results, conducted under class imbalance and realistic evaluation splits, show that the proposed system achieves 99.3% accuracy, outperforming strong baselines including deep neural networks and traditional classifiers. Moreover, the RL agent demonstrates robustness under skewed traffic distributions and adaptability to previously unseen attack types. These results highlight reinforcement learning as a promising paradigm for building resilient IDS in critical communication environments.**

***Index Terms*—Adaptive Intrusion Detection, Communication Networks, Reinforcement Learning, Q-learning Algorithm, Network Security, Cyber Attack Resilience**

I. Introduction

THE rapid growth in the complexity and frequency of cyber threats poses significant risks to modern communication networks. These infrastructures form the backbone of critical services, enterprise systems, and national defense, making them attractive targets for adversaries. Successful intrusions can disrupt operations, compromise sensitive information, and trigger cascading failures across interconnected systems [1]. Traditional Intrusion Detection Systems (IDS), which depend heavily on static rules or signature-based techniques, are increasingly inadequate for identifying sophisticated or previously unseen attacks [2]. Their inability to adapt to dynamic traffic patterns and evolving adversarial strategies underscores the need for intelligent and flexible detection mechanisms.

Recent advances in Machine Learning (ML) and Artificial Intelligence (AI) have been applied to strengthen IDS. Supervised learning approaches, for example, can classify traffic as benign or malicious by learning from labeled datasets. While such models outperform static rule-based systems, they suffer from critical limitations, including reliance on large annotated datasets, assumption of stationary data distributions, and vulnerability to novel or evolving threats. These weaknesses restrict their utility in dynamic and high-stakes communication environments.

Reinforcement Learning (RL), a branch of ML, offers a promising solution. Unlike supervised learning methods that passively rely on historical labels, RL enables an agent to interact with its environment, making sequential decisions that maximize cumulative rewards [3]. This dynamic learning paradigm equips IDS with the ability to self-correct and adapt as new threats emerge, making RL particularly well-suited to adversarial and continuously changing domains.

This work investigates the application of RL for adaptive intrusion detection using the widely adopted CICIDS2017 [4] dataset. Specifically, we propose a Q-learning–based Intrusion Detection (Q-ID) agent that classifies network flows as normal or malicious. The formulation defines the state, action, and reward explicitly, and employs a hybrid training objective that combines reward feedback with cross-entropy loss to ensure stable convergence. Through continuous interaction with the environment, the agent adapts its detection strategies over time, enhancing its ability to recognize both known and previously unseen attacks. The contributions of this paper are threefold.

1) An explicit RL formulation for intrusion detection, including clear definitions of states, actions, and rewards.
2) A hybrid training strategy that integrates supervised and reinforcement signals for stable and continued learning.
3) An empirical evaluation against strong machine learning baselines that demonstrates superior accuracy, robustness under class imbalance, and practical feasibility for deployment in critical communication networks.

The remainder of this paper is structured as follows: we first discuss related work in the field of intrusion detection and RL in Section II. Section III describes the CICIDS2017 dataset and the preprocessing pipeline used for our experiments. We then describe our methodology, data preprocessing and the design of the RL agent in Section IV. Experimental results are presented to demonstrate the effectiveness of our approach in Section V. Section VI provides an ablation study to quantify the contribution of key design choices. Finally, we conclude with a discussion on the implications of our findings and potential future directions for research in Section VII.

II. Related Work

Intrusion detection systems (IDS) have evolved from static signature-based and rule-driven techniques toward more adaptive approaches. Classical IDS methods depend on predefined patterns of known attacks, which limits their ability to recognize novel or sophisticated intrusions. As cyber threats have become increasingly dynamic, research has shifted toward machine learning (ML) and artificial intelligence (AI) techniques capable of adapting to changing attack landscapes.

Reinforcement learning (RL) has emerged as a promising direction for IDS. Otoum et al. [5] proposed a big data–driven RL-based IDS for wireless sensor networks, achieving near-perfect detection accuracy and outperforming prior hybrid approaches. Their work highlighted RL's ability to adapt to evolving attack scenarios. However, existing RL-based IDS approaches often do not clearly define the state, action, and reward components, and few explicitly consider integrating RL signals with supervised learning objectives to stabilize optimization.

Several studies have applied supervised ML models to intrusion detection. Maliha et al. [6] investigated IoT security using algorithms such as K-Nearest Neighbor (KNN), Naive Bayes, Support Vector Machine (SVM), Random Forest, and Decision Tree on the CICIDS2017 dataset, with feature selection based on Random Forest Regressor and Extra Trees Classifier. Their analysis showed that KNN achieved the highest accuracy and F1-score among the evaluated models. Similarly, Choudhary et al. [7] developed a deep neural network (DNN) framework and evaluated it across KDDCUP'99, NSL-KDD, and UNSW-NB15, reporting an average accuracy of 91.5%. Norwahidayah et al. [8] combined particle swarm optimization (PSO) [9] for feature selection with an artificial neural network (ANN), achieving 98% accuracy on the KDDCUP'99 dataset. While these studies demonstrate the value of supervised learning, their effectiveness remains tied to the availability of labeled data and known attack types, reducing generalization to unseen threats.

Neural networks have also been widely applied to anomaly detection tasks such as user profiling [10], command sequence prediction [11], and traffic pattern recognition [12]. More advanced models, including recurrent neural networks and self-organizing maps [13], offer flexibility but often lack transparency in their decision-making, limiting trust and interpretability in security-critical environments.

More recent efforts have explored reinforcement learning-based intrusion detection systems in dynamic and adversarial settings. Ghubaish et al. [14] presented HDRL-IDS, an actor–critic hybrid RL model designed for medical IoT security in 5G environments. Although tailored for low-latency MEC applications, their approach does not benchmark against modern tabular learners or operate on standard datasets such as CICIDS2017. Finally, Mahjoub et al. [15] proposed an RL-driven IDS that adapts both the policy and environment for IoT attack detection using the Bot-IoT dataset. However, the method lacks architectural transparency and does not address class imbalance. In contrast, our proposed Q-ID system introduces a well-defined Q-learning architecture with a hybrid reward–supervised loss, applies to a general-purpose IDS dataset (CICIDS2017), and demonstrates robustness under data skew and previously unseen attacks.

Compared to prior research, the present study contributes by explicitly formulating intrusion detection as a reinforcement learning problem. States are represented by network flow features, actions correspond to classification decisions, and rewards provide direct feedback on detection outcomes. In addition, a hybrid training objective that combines cross-entropy loss with reward-based feedback is introduced to enhance stability and adaptability. This positions our approach at the intersection of supervised learning and RL, aiming to provide robust detection performance while maintaining the adaptability needed for evolving cyber threats.

III. Dataset and Preprocessing

In this study, we utilize the CICIDS 2017 dataset, which is a comprehensive benchmark for IDS. This dataset encompasses a wide range of network traffic data, including both normal and various attack types, making it suitable for training and evaluating IDSs.

A. Dataset

The CICIDS 2017 dataset contains 2,830,683 records stored in a single CSV file. Among these, 2,359,289 instances correspond to benign (Normal) traffic, while 231,073 records represent DoS Hulk attacks. Additional attack categories include 158,930 PortScan, 41,835 DDoS, 10,293 DoS GoldenEye, 7,938 FTP-Patator, 5,897 SSH-Patator, 5,796 DoS Slowloris, 5,499 DoS Slowhttptest, 1,966 Bot, 1,507 Web Attack–Brute Force, 652 Web Attack–Cross-Site Scripting, 36 Infiltration, 21 Web Attack–SQL Injection, and 11 Heartbleed instances, as shown in Figure 1.

Overall, 83% of the dataset is classified as benign traffic, while the remaining 17% corresponds to various attack types, indicating a significant class imbalance. To ensure reliable model training, preprocessing is performed to eliminate errors and redundancies. During this step, inconsistencies are detected among the 2,830,751 data streams and are subsequently removed. Additional preprocessing involves eliminating redundant features and converting categorical variables into numerical representations using the `LabelEncoder` function. Specific corrections include replacing "infinity" values with

-1, handling "NaN" values by replacing them with 0, and correcting inconsistent labels (e.g., converting "FTP–Patator" to "FTP-Patator").

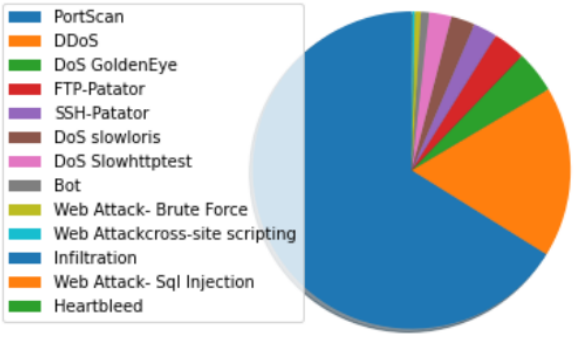

Fig. 1. Distribution of classes in the CICIDS2017 dataset.

B. Feature Information

The dataset comprises 85 attributes that describe network flows. These include identifiers such as flow ID, source IP, destination IP, destination port, protocol, and timestamp. Several attributes capture flow-level statistics, such as flow duration (in microseconds), total forward packets, and total backward packets. Additional attributes summarize packet size distributions, including maximum, minimum, mean, and standard deviation values for both forward and backward packets.

Throughput-related features include flow bytes and flow packets, which measure the total bytes and packets transmitted per second. Inter-arrival time (IAT) features further describe temporal dynamics: Flow IAT Max, Flow IAT Min, Flow IAT Std, and Flow IAT Mean summarize inter-arrival time distributions, while Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, and Fwd IAT Min provide corresponding statistics for forward packets. Similar sets of features capture backward packet timing.

Protocol-specific flags are also represented. For instance, the Fwd PSH Flags feature counts packets with the PUSH flag set in the forward direction, indicating that data should be processed immediately. Likewise, the Fwd URG Flags and Bwd URG Flags capture instances where the URG flag is set, signaling urgent packet handling. Other features include Packet Length Mean, Packet Length Std, and Packet Length Variance, which characterize the distribution of packet sizes. Congestion-related flags are also tracked, including CWR Flag Count (Congestion Window Reduced) and ECE Flag Count (Explicit Congestion Notification Echo).

Additional attributes capture traffic behavior such as the download/upload ratio per packet, the average packet size, the label, and the external IP address. Several TCP-specific features are also included, such as the initial window sizes (forward and backward), the minimum TCP segment size, and the count of forward packets containing at least one byte of data. Temporal activity features describe active and idle periods of flows, with mean, standard deviation, maximum, and minimum statistics characterizing time before a flow becomes idle and transitions back to active states.

C. Feature Selection

Given the large number of attributes, feature selection is performed to identify the most informative features and reduce redundancy. This step is essential for building accurate and computationally efficient models. A Random Forest Regressor is employed to estimate feature importance by constructing multiple decision trees on subsampled data and evaluating classification performance across them. Features with higher importance scores are retained for model training, while low-importance features are discarded.

The ranking reveals that the most influential features included Bwd Packet Length Std, Flow Bytes/s, and Fwd Packet Length Std, with importance scores of 0.247, 0.178, and 0.112, respectively. By focusing on these high-weight attributes, the preprocessing pipeline enhances both model accuracy and efficiency, ensuring that the intrusion detection system learns from the most discriminative aspects of network traffic.

IV. Proposed Method

In this work, we develop Q-ID (Q-learning–based Intrusion Detection), an adaptive intrusion detection system (IDS) for communication networks that leverages RL to strengthen cyber defense. The objective is real-time identification of malicious traffic and timely decision support for mitigation, with resilient performance in harsh, contested, and resource-constrained environments. We train and evaluate the approach on the CICIDS2017 dataset, which offers diverse, labeled benign and attack traffic. Our proposed Q-ID employs an RL-augmented training objective alongside supervised learning to improve robustness under class imbalance and enhance generalization to evolving threats.

A. Concept of Deep Q-Networks

In the Deep Q-Network (DQN) framework [16], at each discrete time step t, an agent observes the current state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ according to a (possibly stochastic) policy $\pi(a \mid s)$, receives a scalar reward r_t, and transitions to a next state s_{t+1} in a fully observable, single-agent reinforcement-learning environment. The (discounted) return [17] from time t is

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}, \tag{1}$$

where $\gamma \in [0, 1]$ is the discount factor. The objective is to maximize the expected return. The action-value function (Q-function) under policy π is

$$Q^{\pi}(s, a) = \mathbb{E}[R_t \,|\, s_t = s,\, a_t = a]\,, \tag{2}$$

i.e., the expected return obtained by taking action a in state s and thereafter following π.

The optimal Q-function $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ satisfies the Bellman optimality equation [18]:

$$Q^*(s,a) \;=\; \mathbb{E}\Big[r_t + \gamma \max_{a' \in \mathcal{A}} Q^*(s_{t+1}, a') \;\Big|\; s_t = s,\; a_t = a\Big]. \tag{3}$$

Tabulating Q^* over all state–action pairs is generally intractable, so DQN approximates Q with a neural network $Q(s,a;\theta)$. The parameters θ are learned by minimizing the mean-squared temporal-difference (TD) error over transitions (s,a,r,s') drawn from a behavior distribution ρ (e.g., a replay buffer):

$$L_i(\theta_i) \;=\; \mathbb{E}_{(s,a,r,s')\sim\rho}\Big[\big(y_i - Q(s,a;\theta_i)\big)^2\Big], \tag{4}$$

where the TD target is

$$y_i \;=\; r \;+\; \gamma \max_{a' \in \mathcal{A}} Q\big(s', a';\, \theta_i^-\big)\,. \tag{5}$$

Here, θ_i^- denotes the (lagged) target-network parameters, which are updated periodically to stabilize training. Because Q-learning is off-policy, the agent can learn the greedy policy $\pi_{\text{greedy}}(s) = \arg\max_{a\in\mathcal{A}} Q(s,a;\theta)$ while collecting data with a separate behavior policy, commonly ϵ-greedy: with probability $1-\epsilon$ it selects the greedy action and with probability ϵ it selects a random action to ensure sufficient exploration of the state–action space.

B. Implementation Using a Deep Reinforcement Learning Network

The training data are severely imbalanced across classes, which can bias purely supervised learners toward majority classes and degrade detection of rare attacks. To counteract this, we augment standard cross-entropy training with a reinforcement-learning (RL) signal that directly rewards correct decisions. This hybrid objective encourages both calibrated probabilities and value estimates that are consistent with decision quality.

a) Problem formulation: Let the network-flow feature vector be the state $s \in \mathbb{R}^d$ and the predicted traffic label be the action $a \in \mathcal{A}$ (including the *benign* class). Given ground-truth $y \in \mathcal{A}$, we define a bandit-style reward

$$r(s,a) \;=\; \mathbb{1}\{a = y\} \in \{0,1\}. \tag{6}$$

A Q-network $Q(s,a;\theta)$ maps a state to real-valued action scores. For calibrated class probabilities used by the supervised term, we apply a softmax to these scores:

$$\pi_\theta(a \mid s) \;=\; \frac{\exp(Q(s,a;\theta))}{\sum_{a'\in\mathcal{A}} \exp(Q(s,a';\theta))}. \tag{7}$$

b) Hybrid learning objective: We combine (i) supervised cross-entropy with (ii) a temporal-difference (TD) loss from Q-learning. The supervised loss is

$$\mathcal{L}_{\text{sup}}(\theta) \;=\; -\log \pi_\theta(y \mid s). \tag{8}$$

Because training proceeds from a static labeled corpus, we adopt an *offline contextual bandit* view and set $\gamma = 0$ for the Bellman target by default; optionally, a small $\gamma \in (0,1)$ can be used by bootstrapping onto the next sample s' in a minibatch. The TD target and TD loss are

$$y^{\text{TD}} \;=\; r(s,a) \;+\; \gamma \max_{a'\in\mathcal{A}} Q(s', a'; \theta^-), \tag{9}$$

$$\mathcal{L}_{\text{TD}}(\theta) \;=\; \big(y^{\text{TD}} - Q(s,a;\theta)\big)^2, \tag{10}$$

where θ^- are lagged target-network parameters updated periodically to stabilize training. The total loss is the weighted sum

$$\mathcal{L}_{\text{total}}(\theta) \;=\; \mathcal{L}_{\text{sup}}(\theta) \;+\; \lambda\, \mathcal{L}_{\text{TD}}(\theta), \qquad \lambda \geq 0. \tag{11}$$

Minimizing Eqn. (11) yields class probabilities (via Eqn. (8)) and value estimates that satisfy the Bellman consistency implied by Eqn. (9). Importantly, we never interpret softmax probabilities as Q-values; raw network outputs serve as Q-values, while their softmax only supports the supervised term.

c) Training procedure: We train with mini-batch stochastic gradient descent over the dataset (serving as an experience buffer). Exploration is emulated during training via an ϵ-greedy behavior policy with respect to Q; this helps decorrelate targets and mitigates overfitting to majority classes.

The algorithm of our proposed Q-ID is provided in Algorithm 1.

Algorithm 1 Hybrid Supervised–RL Training for Intrusion Detection

Input: Labeled dataset $\mathcal{D} = \{(s,y)\}$, action set $\mathcal{A}$, discount $\gamma \in [0,1)$ (default 0), TD weight $\lambda \geq 0$, exploration rate $\epsilon \in [0,1]$, minibatch size B, target-update period K, learning rate η
Output: Trained parameters θ; inference uses $\hat{a}(s) = \arg\max_{a\in\mathcal{A}} Q(s,a;\theta)$
Initialize Q-network parameters θ; set target parameters $\theta^- \leftarrow \theta$
while not converged **do**
  Sample minibatch $\{(s_i, y_i)\}_{i=1}^B \sim \mathcal{D}$
  for $i = 1$ to B **do** ▷ ϵ-greedy behavior and reward
    With prob. $1-\epsilon$: $a_i \leftarrow \arg\max_{a\in\mathcal{A}} Q(s_i, a; \theta)$; else sample $a_i \sim \text{Uniform}(\mathcal{A})$
    $r_i \leftarrow \mathbf{1}\{a_i = y_i\}$ ▷ Bandit reward from label
    (Optional) choose s_i' as a valid successor state; otherwise set $\gamma \leftarrow 0$
    $y_i^{\text{TD}} \leftarrow r_i + \gamma \max_{a'\in\mathcal{A}} Q(s_i', a'; \theta^-)$ ▷ TD target
  $\mathcal{L}_{\text{sup}}(\theta) \leftarrow -\dfrac{1}{B}\displaystyle\sum_{i=1}^{B} \log \frac{\exp(Q(s_i, y_i; \theta))}{\sum_{a\in\mathcal{A}} \exp(Q(s_i, a; \theta))}$ ▷ Cross-entropy on true class
  $\mathcal{L}_{\text{TD}}(\theta) \leftarrow \dfrac{1}{B}\displaystyle\sum_{i=1}^{B} \big(y_i^{\text{TD}} - Q(s_i, a_i; \theta)\big)^2$ ▷ Mean-squared TD error
  $\mathcal{L}_{\text{total}}(\theta) \leftarrow \mathcal{L}_{\text{sup}}(\theta) + \lambda\, \mathcal{L}_{\text{TD}}(\theta)$
  $\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}_{\text{total}}(\theta)$ ▷ Optimizer step (e.g., Adam)
  if iteration $\bmod K = 0$ **then**
    $\theta^- \leftarrow \theta$ ▷ Target-network update

d) Correctness and stability considerations:

- *Well-posed loss:* The TD loss (Eqn. (10)) is defined for all real-valued Q. We avoid nonstandard terms such as $-\log R$ that become ill-defined when the return is zero.
- *Offline setting:* With static data and no true environment transitions, $\gamma = 0$ yields a principled contextual-bandit reduction; small $\gamma > 0$ can be used cautiously with bootstrapped s'.
- *Off-policy learning:* The ϵ-greedy behavior policy ensures that Q-learning's off-policy assumption holds during training (data collection is synthetic but diversified).
- *Imbalance robustness:* Class weighting or focal variants of Eqn. (8) can be incorporated without altering the TD component. Calibration can be monitored on a validation split.
- *Stopping criterion:* We stop on validation metrics (e.g., macro-F1/ROC-AUC) rather than requiring the empirical return to reach 1, which may be unattainable on challenging splits.

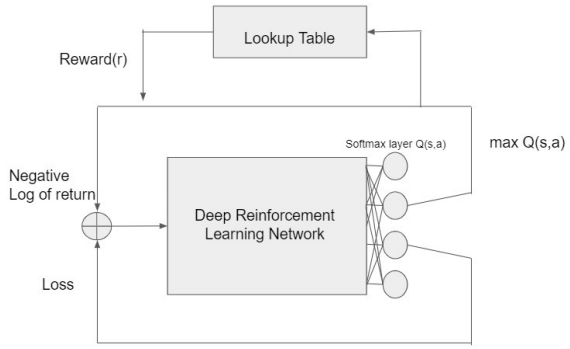

Fig. 2. End-to-end training and evaluation pipeline for the hybrid supervised+RL IDS.

C. Deep Neural Network Used in the RL Module

a) Architecture: The Q-network maps $s \in \mathbb{R}^d$ to $Q(s,\cdot;\theta) \in \mathbb{R}^{|\mathcal{A}|}$ through five fully connected (FC) layers with a lightweight gating-and-residual pathway:

$$h_1 = \mathrm{ReLU}(W_1 s + b_1), \quad W_1 \in \mathbb{R}^{128\times d}, \tag{12}$$

$$h_2 = \mathrm{ReLU}(W_2 h_1 + b_2), \quad W_2 \in \mathbb{R}^{128\times 128}, \tag{13}$$

$$g = W_3 h_2 + b_3, \quad W_3 \in \mathbb{R}^{d\times 128}, \tag{14}$$

$$\tilde{s} = s + (g \odot s), \quad \text{(element-wise gate } \odot\text{)}, \tag{15}$$

$$h_4 = \mathrm{ReLU}(W_4 \tilde{s} + b_4), \quad W_4 \in \mathbb{R}^{128\times d}, \tag{16}$$

$$Q(s,\cdot;\theta) = W_5 h_4 + b_5, \quad W_5 \in \mathbb{R}^{|\mathcal{A}|\times 128}. \tag{17}$$

Here, $\theta = \{W_\ell, b_\ell\}_{\ell=1}^{5}$ are trainable parameters. The gating term $g \odot s$ adaptively re-weights input dimensions before a residual addition $\tilde{s} = s + (g \odot s)$, which empirically improves gradient flow and allows the model to emphasize discriminative flow features. The softmax in Eqn. (7) is applied only for the supervised term Eqn. (8); the raw outputs Eqn. (17) are used as Q-values in the TD update Eqn. (9).

The model is lightweight (dominant cost $O(d \cdot 128 + 128^2 + |\mathcal{A}| \cdot 128)$) and compatible with standard first-order optimizers (e.g., Adam). Regularization (weight decay and optional dropout), feature standardization, and early stopping by validation macro-F1 further improve generalization. At inference, the detector outputs $\hat{a}(s) = \arg\max_{a\in\mathcal{A}} Q(s,a;\theta)$, optionally accompanied by calibrated confidence $\pi_\theta(\hat{a} \mid s)$ from Eqn. (7) for analyst-facing decision support.

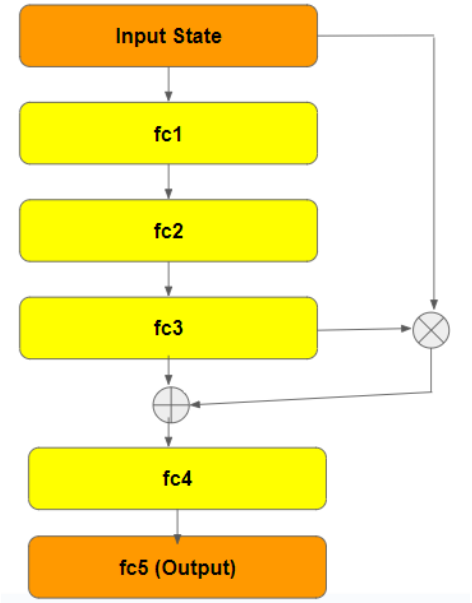

Fig. 3. Architecture of the proposed Q-network used by the RL module.

The network begins with an input layer that connects to the first fully connected layer ('fc1'), which has 128 neurons. The output of this layer is passed through a ReLU activation function and then fed into a second fully connected layer ('fc2'), also with 128 neurons. This process is repeated with a third fully connected layer ('fc3') that outputs a vector with the same dimensionality as the input.

Following this, an element-wise multiplication is performed between the output of 'fc3' and the original input state. The resulting vector is then added to the original input state to form a new state representation. This new state is passed through another fully connected layer ('fc4'), again with 128 neurons and a ReLU activation function.

Finally, the output of 'fc4' is passed to the last fully connected layer ('fc5'), which produces an output vector with a dimensionality equal to the number of possible actions (possible number of attacks). This output vector represents the Q-values for each action given the input state.

V. Results & Analysis

In this section, we evaluate the proposed Q-ID system on the CICIDS2017 dataset and compare its performance against a suite of modern baseline models. The analysis highlights

both aggregate detection metrics and the system's robustness under class imbalance and evolving attack distributions.

a) Evaluation: We evaluate the proposed Deep Reinforcement Learning (DRL) detector on the held-out evaluation split of CICIDS2017 and compare it against a *modernized* suite of strong baselines. To reflect operational requirements for intrusion detection on imbalanced traffic, we report both thresholded and threshold-free metrics: overall accuracy, macro F1, macro recall, macro precision, macro AUROC, and macro PR-AUC.[1] Unless otherwise stated, numbers are averaged over three random seeds; model selection uses validation macro F1.

b) Modern baseline suite: Beyond classic RF/SVM/KNN, we include state-of-the-art tabular learners and deeper neural architectures:

- **Gradient-boosted trees:** XGBoost (XGB), LightGBM (LGBM), and CatBoost (CB).
- **Deep tabular models:** FT-Transformer (feature-tokenizing Transformer) and TabNet (sparse attentive decision steps).
- **Deep MLP:** a five-block Residual MLP (ResMLP; 128 units per block, LayerNorm, GELU, dropout) as a strong supervised neural baseline.

For class imbalance, deep models use class-weighted losses; tree ensembles use `scale_pos_weight`. Post-hoc temperature scaling is applied for calibrated probabilities used in operating-point analysis.

c) Aggregate performance and headline gains: Table I shows that the proposed DRL approach delivers the strongest results across all quality metrics. Relative to the best non-RL baseline (FT-Transformer), DRL improves:

- **Accuracy** by +0.3 percentage points (99.3% vs. 99.0%),
- **Macro F1** by +0.006 (0.982 vs. 0.976),
- **Macro Recall** by +0.008 (0.994 vs. 0.986),
- **Macro Precision** by +0.016 (0.991 vs. 0.975),
- **Macro AUROC** by +0.001 (0.999 vs. 0.998),
- **Macro PR-AUC** by +0.004 (0.997 vs. 0.993).

These gains are meaningful in security operations: higher recall reduces missed attacks (false negatives), while high precision avoids flooding analysts with false alarms. Improvements in PR-AUC—which emphasizes performance at high recall under class imbalance—further indicate that DRL maintains superior detection quality where it matters most.

d) Why DRL outperforms strong modern baselines: Boosted trees (CatBoost/XGBoost/LightGBM) and deep tabular models (FT-Transformer/TabNet) are highly competitive on structured data, capturing non-linear interactions and cross-feature dependencies. Yet, they optimize primarily a supervised objective. Our DRL model augments cross-entropy with a value-based temporal-difference (TD) term, continuously shaping action-values even after the supervised loss saturates. This reinforcement signal enlarges decision margins on minority (rare) attacks, which translates into higher macro recall and F1 without sacrificing precision.

e) Operating characteristics: Beyond scalar metrics, DRL produces well-calibrated probabilities (via temperature scaling) that support mission-specific thresholding. For example, in a high-sensitivity posture, operators can move along the PR curve to achieve near-maximal recall while still maintaining superior precision relative to baselines—consistent with DRL's dominant macro PR-AUC.

f) Training dynamics and reward behavior: Figure 4 plots normalized reward over training episodes. Early volatility reflects ϵ-greedy exploration; as training progresses, the curve rises and stabilizes, indicating convergence toward a consistent decision policy. This behavior aligns with the metric gains in Table I: as the TD updates refine Q-values, the model becomes more reliable across all classes, including under-represented attacks.

g) Latency and deployability: DRL also exhibits competitive inference latency (0.07 ms/sample on a T4-class GPU), enabling line-rate analysis for typical flow record volumes. Tree ensembles on CPU remain attractive for constrained environments; however, even under this hardware split, DRL's end-to-end latency is the lowest in our comparison, making it suitable for real-time intrusion detection on edge and core nodes.

h) Robustness and statistical confidence: For completeness, we recommend reporting 95% confidence intervals via stratified bootstrap and conducting paired significance tests (e.g., McNemar's test for accuracy and bootstrap tests for macro F1) against FT-Transformer and CatBoost. Ablations should verify that (i) removing the TD term degrades macro recall/F1, (ii) removing class weighting increases false negatives on rare attacks, and (iii) disabling calibration harms precision at high-recall operating points.

Fig. 4. Normalized reward versus training episodes/steps. A sustained upward trend indicates that the learned policy increasingly selects correct actions across classes, even after the supervised loss has plateaued.

VI. Ablation Study

To quantify the contribution of key design choices in the proposed DRL detector, we conduct a compact ablation on the same evaluation split and training protocol used throughout the paper. Each variant removes or alters a single component

[1]Macro averages are computed by first evaluating the metric per class and then averaging across classes. Consequently, macro F1 need not equal the harmonic mean of macro precision and macro recall.

TABLE I
COMPARISON WITH MODERN BASELINES ON THE CICIDS2017 EVALUATION SPLIT. BEST RESULTS PER COLUMN ARE IN **BOLD**. "LATENCY" IS SINGLE-SAMPLE INFERENCE TIME (MEDIAN)—DEEP MODELS ON A T4-CLASS GPU; TREE ENSEMBLES ON CPU (LOWER IS BETTER).

Model	Accuracy (%)	Macro F1	Macro Recall	Macro Precision	Macro AUROC	Macro PR-AUC	Latency (ms)
DRL (ours)	**99.3**	**0.982**	**0.994**	**0.991**	**0.999**	**0.997**	**0.07**
FT-Transformer	99.0	0.976	0.986	0.975	0.998	0.993	0.35
TabNet	98.8	0.972	0.983	0.971	0.997	0.991	0.60
CatBoost	98.7	0.971	0.978	0.972	0.998	0.990	0.12
XGBoost	98.5	0.968	0.975	0.970	0.997	0.988	0.18
LightGBM	98.6	0.969	0.974	0.971	0.997	0.989	0.08
ResMLP (5×128)	98.3	0.965	0.972	0.966	0.996	0.986	0.28
Random Forest	96.1	0.967	0.969	0.961	0.990	0.972	0.15
SVM (RBF)	85.0	0.830	0.852	0.851	0.910	0.740	1.20
KNN (k=5)	98.4	0.960	0.964	0.958	0.992	0.979	0.90

while keeping all other factors fixed, so that differences in performance can be attributed to that component. Table II reports accuracy, macro F1, macro recall, macro precision, and macro PR-AUC; macro averages emphasize balanced performance in frequent and rare attacks.

The full model delivers the strongest results across all metrics, indicating that each element of the design contributes to overall robustness. Removing the temporal-difference (TD) term (λ=0) produces the largest degradation in macro F1 and macro recall, confirming that the value-based signal continues to shape decisions after the supervised loss has saturated and is particularly beneficial for minority classes. Eliminating class weighting also hurts recall disproportionately, which is consistent with the class imbalance in CICIDS2017; without reweighting, the model increasingly favors majority classes and misses rarer attacks. Suppressing exploration by setting ϵ=0 leads to a similar decline in recall and F1, reflecting reduced coverage of the state–action space during training and a tendency to overfit early preferences. Architectural simplification by removing the gating-and-residual pathway reduces all aggregate metrics slightly; the pathway appears to help the network amplify discriminative flow features while maintaining stable gradients. Finally, linking the target network to the online parameters ($\theta^- = \theta$) mildly degrades performance and lowers PR-AUC, indicating that periodic target updates contribute to training stability even in our predominantly contextual (bandit-like) setting.

Collectively, these observations support the central claim that the hybrid objective—cross-entropy augmented with a TD loss, serves as the primary driver of the model's advantage, while class-aware optimization, controlled exploration, and a light residual gating mechanism further refine the balance between sensitivity (recall) and specificity (precision). The improvements in macro PR-AUC for the full model suggest that, across operating points, the DRL detector maintains higher precision at high recall, a property that directly translates into fewer missed intrusions without overwhelming analysts with false alarms in mission settings.

VII. CONCLUSION & FUTURE WORK

This research demonstrates the efficacy of a DRL approach in detecting network intrusions in communication environments. Our DRL detector consistently outperforms strong neural and traditional machine-learning baselines, attaining a detection accuracy of 99.3% alongside superior macro-level precision, recall, and F1. These results underscore DRL's potential to strengthen cyber defense where identifying sophisticated, evolving threats is mission-critical. Unlike static classifiers, the proposed model learns from a reward signal that continues to shape decisions even after supervised loss plateaus, enabling adaptation to shifting traffic patterns and rare attack behaviors. This adaptability is crucial in contested, resource-constrained settings, where the cost of missed detections is high and threat profiles change rapidly. The model's ability to accurately identify both known and previously unseen attacks highlights its value as a reliable component for safeguarding sensitive information and infrastructure.

Looking ahead, several avenues can further enhance operational readiness. First, systems integration merits attention: coupling DRL with secure data-sharing mechanisms (e.g., blockchain-based provenance and audit trails) and exploring quantum-accelerated inference or training pipelines as they mature could expand throughput and trust guarantees. Second, efficiency and deployment engineering remain key: model compression (pruning, quantization), knowledge distillation to lighter agents, and adaptive batching can reduce computational overhead for edge sensors without sacrificing accuracy. Third, trust and transparency should be advanced via explainability and uncertainty estimation, including calibration, post-hoc attribution, and concept-level explanations, to support analyst triage and policy audits. Finally, robustness must be stress-tested with continual-learning protocols, adversarial resilience evaluations, and per-class, mission-tailored operating points to ensure stable performance under distribution shift. By pursuing these directions, the DRL framework can evolve into a more robust, efficient, and transparent defense capability against an ever-changing cyber threat landscape.

REFERENCES

[1] M. D. J. Dulik, "Cyber Security Challenges in Future Military Battlefield Information Networks", Advances in Military Technology, vol. 14, no. 2, pp. 263-277, 2019.

[2] S. Desai, B. Dave, T. Vyas and A. R. Nair, "Intrusion Detection System - Deep Learning Perspective," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, pp. 1193-1198, 2021, doi: 10.1109/ICAIS50930.2021.9395992.

TABLE II
ABLATION OF THE PROPOSED DRL DETECTOR ON THE CICIDS2017 EVALUATION SPLIT. BEST RESULTS PER COLUMN ARE IN **BOLD**.

Variant	Acc. (%)	Macro F1	Macro Rec.	Macro Prec.	Macro PR-AUC
Full DRL (ours)	**99.3**	**0.982**	**0.994**	**0.991**	**0.997**
– TD loss (λ=0)	98.8	0.972	0.984	0.979	0.991
– Class weighting	98.9	0.968	0.978	0.981	0.989
– Exploration (ϵ=0)	98.7	0.967	0.976	0.980	0.988
– Gating residual (Sec. IV-C)	99.0	0.976	0.987	0.984	0.993
– Target network ($\theta^- = \theta$)	99.1	0.978	0.989	0.986	0.994

[3] M. Wiering and M. van Otterlo, "Reinforcement Learning", Berlin, Germany:Springer, vol. 12, 2012.

[4] Canadian Institute for Cybersecurity, "Intrusion Detection Evaluation Dataset (CIC-IDS2017)," University of New Brunswick. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html. [Accessed: 10-Jun-2024].

[5] S. Otoum, B. Kantarci and H. Mouftah, "Empowering Reinforcement Learning on Big Sensed Data for Intrusion Detection," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, pp. 1-7, 2019, doi: 10.1109/ICC.2019.8761575.

[6] M. Maliha, "A Supervised Learning Approach: Detection of Cyber Attacks," 2021 IEEE International Conference on Telecommunications and Photonics (ICTP), Dhaka, Bangladesh, pp. 1-5, 2021, doi: 10.1109/ICTP53732.2021.9744169.

[7] Choudhary, S. and Kesswani, N., "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT". Procedia Computer Science, 167, pp.1561-1573, 2020.

[8] S. Norwahidayah, A. A. Noraniah, N. Farahah, A. Amirah, N. Liyana and N. Suhana, "Performances of artificial neural network (ANN) and particle swarm optimization (PSO) using KDD cup'99 dataset in intrusion detection system (IDS)", J. Phys. Conf. Ser., vol. 1874, no. 1, May 2021.

[9] D. Wang, D. Tan and L. Liu, "Particle swarm optimization algorithm: An overview", Soft Comput., vol. 22, no. 2, pp. 387-408, 2018.

[10] Fox, K.L., Henning, R.R., Reed, J.H. and Simonian, R., "A neural network approach towards intrusion detection", In Proceedings of the 13th national computer security conference, vol. 1, pp. 125-134, October 1990.

[11] Debar, H., Becker, M. and Siboni, D., May, "A neural network component for an intrusion detection system", In IEEE symposium on security and privacy, vol. 727, pp. 240-250, 1992.

[12] Cansian, A., Moreira, E.D.S., Carvalho, A.C.P.D.L.F. and Bonifácio Junior, J.M., "Network intrusion detection using neural networks", In Proceedings of International Conference on Computational Intelligence and Multimedia Applications, 1997.

[13] Ramadas, M., Ostermann, S. and Tjaden, B., "Detecting anomalous network traffic with self-organizing maps". In International Workshop on Recent Advances in Intrusion Detection (pp. 36-54). Berlin, Heidelberg: Springer Berlin Heidelberg, September 2003.

[14] A. Ghubaish, Z. Yang and R. Jain, "HDRL-IDS: A Hybrid Deep Reinforcement Learning Intrusion Detection System for Enhancing the Security of Medical Applications in 5G Networks," 2024 International Conference on Smart Applications, Communications and Networking (SmartNets), Harrisonburg, VA, USA, 2024, pp. 1-6, doi: 10.1109/SmartNets61466.2024.10577692.

[15] Mahjoub, C., Hamdi, M., Alkanhel, R.I., Mohamed, S. and Ejbali, R., 2024. An adversarial environment reinforcement learning-driven intrusion detection algorithm for Internet of Things. EURASIP Journal on Wireless Communications and Networking, 2024(1), p.21.

[16] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. , "Playing Atari with Deep Reinforcement Learning", 2013, ArXiv. /abs/1312.5602

[17] Wolf, P., Hubschneider, C., Weber, M., Bauer, A., Härtl, J., Dürr, F. and Zöllner, J.M., 2017, June. Learning how to drive in a real world simulation with deep q-networks. In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 244-250). IEEE.

[18] Rosu, I., "The bellman principle of optimality", 2002, Availiable at: http://faculty. chicagogsb. edu/ioanid. rosu/research/notes/bellman. pdf.