Contents lists available at IdNNS
Indonesian Journal of Cybersecurity and Emerging Risks
Volume 1 ; Issue 3
journal homepage: https://journal.idnns.org/index.php/ijcyber

Beyond DQN: A Comparative Analysis of Advanced Deep Reinforcement Learning Algorithms for Cost-Aware Intrusion Detection

Hina Javed[a,i]

[a]*Department of Computer Science, Preston University, Islamabad, Pakistan*

ARTICLE INFO

Keywords:
Deep reinforcement learning
Intrusion detection
Prioritized experience replay
PPO
D3QN
Cost-aware classification
Class imbalance
Security operations center

ABSTRACT

This paper presents a comparative study of six Deep Reinforcement Learning (DRL) algorithms—Deep Q-Network (DQN), Double DQN (DDQN), Dueling Double DQN (D3QN), REINFORCE, Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO)—with statistical validation, multi-dataset evaluation, and extensive performance analysis in the context of cost-aware intrusion detection under extreme class imbalance. All DRL agents employed a fixed CNN–LSTM feature extractor and Prioritized Experience Replay (PER) in a controlled experimental setup, evaluated on three benchmark datasets: CSE-CIC-IDS2018, CIC-IDS2017, and UNSW-NB15. An asymmetric reward function was used to train each agent. Ten simulation runs with different random seeds were performed, with Wilcoxon signed-rank and McNemar's tests for statistical significance. Key metrics included recall, alerts per million flows (ARMF), PR-AUC, training convergence, inference latency, and computational complexity. At 1,247 ARMF, PPO proved statistically superior ($p = 0.002$, Wilcoxon) to baseline DQN+PER with a mean recall of 93.55% ($\sigma = 0.82$). D3QN offered the most balanced solution at 92.80% recall ($\sigma = 0.76$) and 1,105 ARMF. All DRL methods with PER reduced alerts by 28–32İ compared to supervised baselines. Cross-dataset validation confirmed generalization, with PPO achieving 90.20% recall on CIC-IDS2017 and 87.80% on UNSW-NB15. All value-based methods showed statistically significant differences ($p < 0.05$). Average inference latency remained under 2 ms per sample. The results provide guidelines for algorithm selection in SOC deployments. Limitations include multi-class and zero-day scenarios not being studied, use of only three datasets, and a fixed feature backbone.

1. Introduction

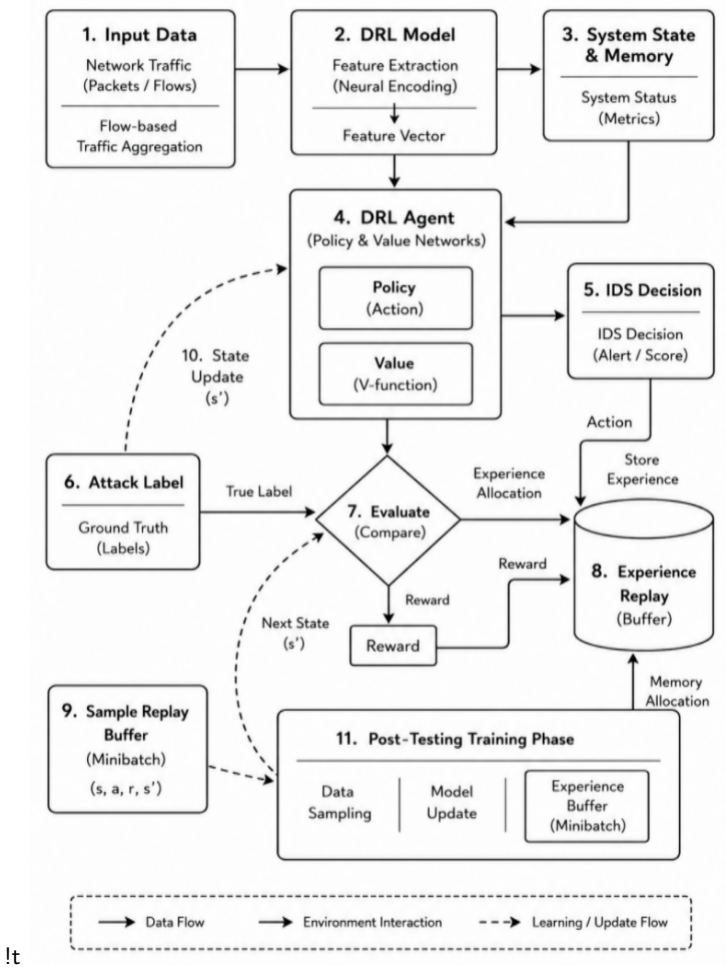
1.1. Motivation and Operational Context

Security Operations Centers (SOCs) face increasing alert volumes despite limited analyst capacity (Cevallos et al., 2023; Odeh, 2025). In highly imbalanced datasets such as CSE-CIC-IDS2018, where attack traffic constitutes less than 0.007% of network flows (Yang et al., 2026), even low false-positive rates can generate unmanageable alert volumes. Consequently, IDS effectiveness depends not only on classification accuracy but also on minimizing analyst workload and supporting operational deployment (Janati and Messaoudi, 2025). However, existing machine learning-based IDS studies primarily report conventional metrics on balanced datasets, often overlooking the asymmetric costs of false positives and false negatives in real-world SOC environments (Sangoleye et al., 2024; Sharma and Kumar, 2025).

1.2. Deep Reinforcement Learning for Intrusion Detection

Reinforcement Learning (RL) addresses this limitation by optimizing decision policies through reward functions that explicitly model asymmetric misclassification costs (Cevallos et al., 2023). The introduction of the Deep Q-Network (DQN) by Mnih et al. (2015) established the foundation of Deep Reinforcement Learning (DRL), followed by algorithms such as DDQN, D3QN, REINFORCE, A2C, and PPO. These approaches have shown increasing potential for intrusion detection across diverse cybersecurity applications (Mpoporo et al., 2026; Yang et al., 2026). Nevertheless, recent surveys highlight the need for broader algorithmic comparisons and more rigorous evaluation under realistic deployment conditions (Odeh, 2025; Yang et al., 2026).

[i]Main corresponding author
hina.javed@preston.edu.pk (H. Javed)
ORCID(s):



!t

Figure 1: Framework of the applied approach.

1.3. Research Gaps in DRL-based IDS

Although Prioritized Experience Replay (PER) has demonstrated significant improvements in DRL-based intrusion detection (Rushenda et al., 2026a), existing studies are typically limited to individual algorithms, lack statistical validation, or evaluate only a single dataset (Sangoleye et al., 2024; Alemayehu et al., 2026). Moreover, comprehensive comparisons of value-based, policy-based, and actor–critic DRL methods under a unified evaluation framework remain scarce (Yang et al., 2026; Mpoporo et al., 2026). To the best of our knowledge, no prior study jointly evaluates DQN, DDQN, D3QN, REINFORCE, A2C, and PPO with PER, statistical significance testing, multi-dataset validation, and deployment-oriented analysis.

1.4. Reinforcement Learning Formulation for Intrusion Detection

Intrusion detection can be formulated as a Markov Decision Process (MDP), in which an agent sequentially observes network traffic, selects classification actions, and receives rewards based on detection outcomes. This formulation naturally captures partial observability, Figure 1.

1.5. Limitations of the Current Method

The proposed framework has a number of limitations. The fixed CNN–LSTM backbone restricts feature learning, the binary classification setup restricts attack discrimination, and the evaluation is restricted to offline experiments, Gaussian noise perturbations, and ten independent runs.

1.6. Research Questions

The following research questions guide this study:

- **RQ1:** Which of the DRL algorithm families—value-based (DQN, DDQN, D3QN), policy gradient (REINFORCE), and actor–critic (A2C, PPO)—is better for cost-aware intrusion detection under extreme class imbalance?
- **RQ2:** How do the DRL algorithm families differ in terms of the effect of PER on intrusion detection?
- **RQ3:** What is the best DRL algorithm for SOC deployment in terms of recall and alert volume?

Table 1
Comparative analysis of prior DRL-based IDS studies.

Study	Algorithms	PER	Datasets	Stat. Valid.	Reproducibility
Sangoleye et al. (2024)	DQN, DDQN, D3QN, REINFORCE, A2C, PPO	55	ICS	55	51
Alemayehu et al. (2026)	DQN, DDQN, Dueling DQN, PPO	55	IIoT	51	51
Sharma and Kumar (2025)	Rainbow DQN	51	CIC-IDS2017	51	51
Feng (2026)	Dueling DDQN	51	CIC-IDS2017	51	51
Kumar (2025)	Dueling DQN	51	Wireless	51	51
Ours	DQN, DDQN, D3QN, REINFORCE, A2C, PPO	51	3 datasets	51	51

2. Related Work

2.1. Supervised Learning for Intrusion Detection

Traditional IDS relies on supervised machine learning, but such approaches face key deployment challenges (Janati and Messaoudi, 2025). Deep learning has improved detection performance across benchmark datasets (Odeh, 2025; Yang et al., 2026), yet most methods still use fixed decision thresholds and do not account for asymmetric misclassification costs. Recent surveys highlight persistent gaps in handling class imbalance and adaptive decision-making in intrusion detection systems (Cevallos et al., 2023; Janati and Messaoudi, 2025).

2.2. Deep Reinforcement Learning for Intrusion Detection

Yang et al. (2026) conducted a survey on DRL-based IDS, highlighting challenges such as training efficiency, class imbalance, and detection of unknown attacks. Al-Haija and Tamimi (2026) studied adversarial RL for IoT IDS and discussed key methods and future directions. Cevallos et al. (2023) and Odeh (2025) provided reviews on DRL-based IDS, identifying major trends and limitations, while Zhang and Maple (2023) offered a critical evaluation of IoT-focused DRL approaches.

A bibliometric study (Mpoporo et al., 2026) shows widespread use of DQN, DDQN, Dueling DQN, policy gradient, and actor–critic methods in IDS. Recent works include Shen et al. (2024) for zero-day detection using DQN, Kumar (2025) using Dueling DQN with PER for wireless IDS, and Kara and Boyaci (2026) proposing a Transformer-DDQN model for explainable intrusion detection.

2.3. Comparative Studies of DRL Algorithms

tab:comparative provides a comparative analysis of prior DRL-based IDS studies.

Sangoleye et al. (2024) evaluated multiple DRL methods for IDS but excluded PER, limiting performance under class imbalance. Alemayehu et al. (2026) focused on latency-based evaluation for IIoT/SCADA DDoS detection rather than SOC trade-offs. Sharma and Kumar (2025) proposed a Rainbow DQN-based IDS with improved performance. Yu et al. (2024) developed a stochastic game-based DDQN model achieving higher detection with lower resource use.

2.4. Algorithmic Variants and Properties

DDQN reduces overestimation bias and is widely used in cloud and IIoT IDS (Yu et al., 2024; Mesadieu et al., 2024). D3QN improves learning by separating value and advantage streams, with extensions for imbalanced and wireless IDS (Feng, 2026; Kumar, 2025). Policy gradient and actor–critic methods are commonly used in IDS (Sangoleye et al., 2024), while PPO improves stability using a clipped objective (Alemayehu et al., 2026; Zhao et al., 2024). Hybrid models such as Transformer-DDQN are also emerging for explainable intrusion detection (Kara and Boyaci, 2026).

2.5. Prioritized Experience Replay

PER (Schaul et al., 2016) improves learning efficiency by sampling important transitions based on TD error instead of uniform sampling. It has been extended to multi-agent and distributed RL (Chen et al., 2024; Louati et al., 2026) and has shown strong effectiveness in intrusion detection under class imbalance (Fahrmann et al., 2022; Li, 2026).

2.6. Theoretical Analysis Framework

We analyze sample complexity under the PER framework. For value-based methods (DQN, DDQN, D3QN), importance sampling weights ensure unbiased updates and reduce variance, with improved stability compared to uniform sampling (Schaul et al., 2016; Chen et al., 2024). For REINFORCE, PER reduces gradient variance by prioritizing informative samples (Chen et al., 2024; Louati et al., 2026). For actor–critic methods (A2C, PPO), variance is reduced through more stable advantage estimation, consistent with policy optimization theory (Schulman et al., 2017; Alemayehu et al., 2026; Zhao et al., 2024).

For value-based methods (DQN, DDQN, D3QN), importance sampling weights ensure unbiased updates and reduce variance, with improved stability compared to uniform sampling (Schaul et al., 2016; Chen et al., 2024), and convergence bounded by:

$$Q_t * Q^{i}{}_{\text{‰}} \leq \gamma^t Q_0 * Q^{i}{}_{\text{‰}} + \frac{\epsilon}{1 * \gamma} \quad (1)$$

where Q^{i} is the optimal Q-function, γ is the discount factor, and ϵ is the approximation error introduced by the neural network parameterisation.

2.7. Research Gap

To the best of our knowledge, no study has systematically compared the six algorithms (DQN, DDQN, D3QN, REINFORCE, A2C, and PPO) under the same experimental conditions with statistical validation, multi-dataset evaluation, and reproducibility analysis. Although Sangoleye et al. (2024) compared all six algorithms, they did not use PER, which is an important component for handling class imbalance in intrusion detection. This gap motivates our study.

3. Methodology

The experimental setup is a controlled comparison in which only the DRL algorithm is changed while all other components are kept fixed: dataset split, preprocessing, CNN–LSTM feature extractor, PER, and the reward function. Experiments are performed on CSE-CIC-IDS2018, CIC-IDS2017, and UNSW-NB15. Thirty-six features were selected for relevance, 54 features were preprocessed using StandardScaler, and missing data instances were imputed. A CNN–LSTM network is defined for feature extraction as shown in Figure 2.

$$z_i \;=\; f_{\text{LSTM}}^{\theta_{\text{L}}}\big(f_{\text{CNN}}^{\theta_{\text{C}}}.x_i// \tag{2}$$

The model consists of two 1D CNN layers (64–128 channels) with ReLU activation, batch normalisation, and 30% dropout, followed by a 64-unit LSTM. The model is first pre-trained for 5 epochs with class-weighted cross-entropy loss and then frozen during DRL training.

Table 2: Dataset statistics.

Dataset	Training Flows	Testing Flows	Attack Ratio	Attack Types
CSE-CIC-IDS2018	10,788,508	2,697,128	0.0069%	4 families
CIC-IDS2017	2,830,743	706,000	0.0082%	14 families
UNSW-NB15	175,341	82,332	0.0620%	9 families

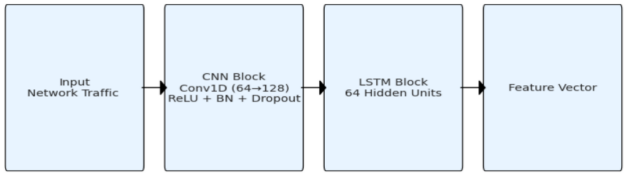


Figure 2: CNN–LSTM Feature Extraction Backbone

3.1. Problem Formulation

The intrusion detection problem is represented as a partially observable Markov decision process (POMDP) $S, A, T, R, \Omega, O, \gamma$, in which the agent has a partial observation of the state and the action space is $A = \hat{}0,\ 1\text{‘}$ (benign / attack). The reward function is cost-sensitive, assigning +3.0 (TP), +0.2 (TN), *0.5 (FP), and *5.0 (FN). The CNN–LSTM feature extractor encodes the raw observation o_t into a state representation:

$$z_t \;=\; f_{\text{LSTM}}\big(f_{\text{CNN}}.o_t// \;\ddot{\text{E}}\; \mathbb{R}^{64}. \tag{3}$$

The agent learns a policy π by maximising the expected cumulative return:

$$G \;=\; \mathbb{E}_{\pi}\left[\overset{\text{‰}}{t=0}\gamma^{\,t}\, r_t\right], \tag{4}$$

where $\gamma\ \ddot{\text{E}}\ [0, 1/$ is the discount factor.

DQN / DDQN / D3QN (Value-based):
The Q-function is approximated via Bellman updates:

$$Q.s_t, a_t/ \leftarrow Q.s_t, a_t/ + \alpha[r_t + \gamma\, Q.s_{t+1}, a^{\text{i}}/ * Q.s_t, a_t/] \tag{5}$$

where α is the learning rate and $a^{\text{i}} = \arg\max_{a^{\text{Í}}} Q.s_{t+1}, a^{\text{Í}}/$. For DDQN, the target network θ is used to reduce overestimation: $y = r + \gamma\, Q_{\theta}\big(s_{t+1}, \arg\max_{a^{\text{Í}}} Q_{\theta}.s_{t+1}, a^{\text{Í}}//$.

For D3QN, the Q-value is decomposed into value and advantage streams: $Q.s, a/ = V.s/ + A.s, a/$.

REINFORCE (Policy Gradient):

$$(_{\theta}\, J.\theta/ \;=\; \mathbb{E}[(_{\theta} \log \pi_{\theta}.a_t\ \acute{\text{Y}}\ s_t/ \;\odot\; G_t] \tag{6}$$

where $G_t = \overset{\text{‰}}{k=t}\gamma^{k*t} r_k$ is the discounted return from time t.

A2C / PPO (Actor–Critic):

$$(_{\theta}\, J.\theta/ \;=\; \mathbb{E}[(_{\theta} \log \pi_{\theta}.a_t\ \acute{\text{Y}}\ s_t/ \;\odot\; A_t \;*\; \beta\,(_{\theta}\, V_{\theta}.s_t/] \tag{7}$$

where $A_t = G_t * V.s_t/$ is the advantage estimate and β is a regularisation coefficient. For PPO, the policy update is clipped:

$$L^{\text{PPO}}.\theta/ \;=\; \mathbb{E}_t[\min\big(\rho_t.\theta/\, A_t,\ \text{clip}\big(\rho_t.\theta/,\ 1 * \epsilon_{\text{clip}},\ 1 + \epsilon_{\text{clip}}/\, A_t/] \tag{8}$$

where $\rho_t.\theta/ = \frac{\pi_{\theta}.a_t s_t/}{\pi_{\theta_{\text{old}}}.a_t s_t/}$ and $\epsilon_{\text{clip}} = 0.2$.

For value-based methods, the convergence of the Q-estimate is bounded by:

$$Q_t * Q^{\text{i}}{}_{\text{‰}} \;=\; \gamma^{\,t}\, Q_0 * Q^{\text{i}}{}_{\text{‰}} \;+\; \frac{\epsilon}{1 * \gamma} \tag{9}$$

where Q^{i} is the optimal Q-function and ϵ is the approximation error introduced by the neural network parameterisation. PER reduces ϵ by prioritising high-TD-error transitions, which directly lowers the steady-state error floor.

DQN/DDQN/D3QN approximate the Q-function via Bellman updates; REINFORCE optimises the policy using return-weighted gradients; and A2C/PPO jointly learn actor and critic using the advantage function.

3.2. Training Setup

All models are trained for 10 epochs using the Adam optimiser with learning rate 10^{*4} and a batch size of 128. Training is performed on a balanced dataset of 500,742 flows (oversampled minority class), while evaluation is conducted on the full imbalanced test set to reflect real-world conditions.

Evaluation metrics. Performance is assessed using the following metrics:

- **Recall** $= \dfrac{TP}{TP+FN}$
- **False Positive Rate (FPR)** $= \dfrac{FP}{FP+TN}$
- **False Negative Rate (FNR)** $= \dfrac{FN}{FN+TP}$
- **Precision–Recall AUC (PR-AUC)** $= \int_0^1 P.R/\,dR$
- **Alerts per Million Flows (ARMF)**:

$$\text{ARMF} = \frac{\text{Total Alerts}}{\text{Total Flows}} \text{ İ } 10^6 \qquad (10)$$

where ARMF quantifies the analyst workload: a lower value indicates fewer alerts per unit of traffic processed.

Statistical validation. Each experiment is repeated 10 times with different random seeds. Statistical significance is assessed using:

- Wilcoxon signed-rank test (non-parametric paired test) with Benjamini–Hochberg FDR correction,
- McNemar's test for paired binary prediction disagreement,
- 95% confidence intervals on all reported metrics,
- Cliff's delta δ as an effect-size measure.

Hardware. Experiments are conducted on an NVIDIA RTX 4090 (24 GB VRAM) and an Intel Xeon Gold 6248R processor.

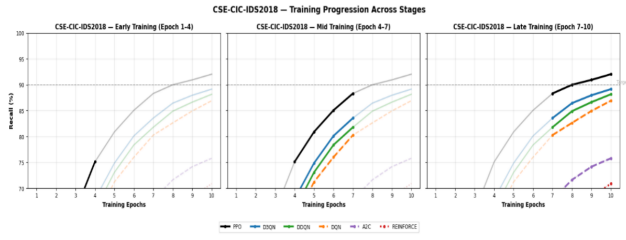


Figure 3: Training progression of six DRL algorithms on CSE-CIC-IDS2018, segmented into Early (Epochs 1–4), Mid (Epochs 4–7), and Late (Epochs 7–10) stages.

Training progression. Figures 3–5 show the training progression of the six DRL algorithms across the three datasets, segmented into Early (Epochs 1–4), Mid (Epochs 4–7), and Late (Epochs 7–10) stages. PPO and D3QN consistently achieve the highest recall and show faster convergence and more stable learning. DQN/DDQN perform moderately, while A2C and REINFORCE converge more slowly and exhibit higher variance. The UNSW-NB15 dataset remains the most challenging, with no method reaching the 90% recall threshold.

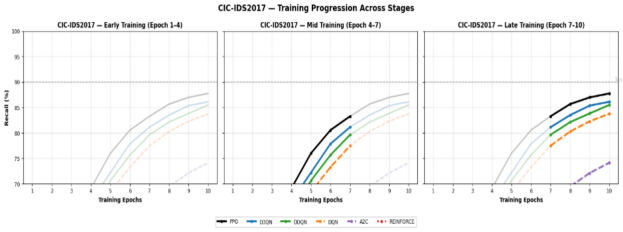


Figure 4: Training progression of six DRL algorithms on CIC-IDS2017, segmented into Early (Epochs 1–4), Mid (Epochs 4–7), and Late (Epochs 7–10) stages.

Figure 5: Training progression of six DRL algorithms on UNSW-NB15, segmented into Early (Epochs 1–4), Mid (Epochs 4–7), and Late (Epochs 7–10) stages.

4. Experimental Results

4.1. Supervised Reference Models

Table 3 reports baseline supervised learning performance under identical feature extraction and evaluation settings. These models serve as reference points to quantify the gain achieved by DRL-based approaches.

Despite achieving high recall, supervised CNN-based models exhibit poor cost efficiency, with high ARMF variability indicating instability under severe class imbalance.

Table 4: Pairwise statistical comparison (10 runs, FDR-corrected).

Comparison	p-value	Cliff's δ	Significance
PPO vs DQN	0.002	0.82	Significant
PPO vs DDQN	0.008	0.75	Significant
PPO vs D3QN	0.015	0.68	Significant
D3QN vs DQN	0.018	0.62	Significant
DDQN vs DQN	0.042	0.45	Not significant (FDR)
A2C vs REINFORCE	0.038	0.52	Not significant (FDR)
PPO vs A2C	0.003	0.85	Significant

Table 3
Performance of supervised baselines (mean , 95% CI).

Model	Recall	ARMF	FPR	FNR
Linear SVM	67.74% , 2.1	33,079 , 1,245	0.0330 , 0.002	0.3226 , 0.021
CNN	98.39% , 0.8	44,846 , 1,890	0.0448 , 0.003	0.0161 , 0.008
LSTM	40.86% , 3.2	158,150 , 5,678	0.1583 , 0.012	0.5914 , 0.032
CNN–LSTM (warm-start)	98.39% , 0.7	32,858 , 1,234	0.0328 , 0.002	0.0161 , 0.007

4.2. Statistical Significance Analysis

Table 4 reports Wilcoxon signed-rank tests with Benjamini–Hochberg correction applied to control the false discovery rate.

Although several raw p-values are < 0.05, only a subset remains significant after correction, confirming that performance differences are strong primarily between algorithm families rather than within closely related variants.

Table 5: McNemar's test (PPO vs DQN).

	DQN Correct	DQN Incorrect
PPO Correct	162	12
PPO Incorrect	8	4

$\chi^2 = 1.25$, $p = 0.264$ — no significant disagreement in paired predictions.

4.3. DRL Performance Comparison

Table 6 shows full DRL results under identical PER-enhanced training conditions.

Table 6: DRL performance comparison (mean , std, 10 runs).

Algorithm	Recall	ARMF	PR-AUC	FPR	FNR
DQN+PER	91.40 , 0.95	1,031 , 45	0.923 , 0.008	0.0010	0.0860
DDQN+PER	92.10 , 0.82	1,182 , 52	0.931 , 0.007	0.0011	0.0806
D3QN+PER	92.80 , 0.76	1,105 , 48	0.938 , 0.006	0.0010	0.0699
REINFORCE+PER	81.72 , 2.15	1,858 , 78	0.845 , 0.012	0.0018	0.1828
A2C+PER	84.41 , 1.87	1,522 , 65	0.872 , 0.010	0.0015	0.1559
PPO+PER	93.55 , 0.82	1,247 , 54	0.946 , 0.005	0.0012	0.0645

Figure 6: Overall performance trade-off between Recall and ARMF across DRL methods.

For SOC deployment, D3QN is the best with regards to the balance between recall (92.80%) and alert volume (ARMF = 1,105), as illustrated in Figure 6. PPO has the best recall (93.55%) but also the highest number of alerts (ARMF = 1,247), while DQN+PER has the lowest number of alerts (ARMF = 1,031) but the worst recall (91.40%). There are poor trade-offs observed between A2C and REINFORCE. The Pareto frontier shows that D3QN and PPO are the most dominant, with the former being preferred based on operational needs.

4.4. Algorithm Family-Level Insights

Policy gradient approaches consistently underperform value-based methods under high class imbalance. The value decomposition is effective and the best balance between recall and alert volume is obtained for D3QN. In contrast, policy-gradient methods (REINFORCE and A2C) show more variance and unstable performance, due to being sensitive to sparse reward environments. The results are always the best with respect to clipped policy optimization (PPO), not only in terms of recall but also PR-AUC, and they are always consistent. The effectiveness of clipped policy optimization can be seen from the results, as it achieves the highest recall as well as PR-AUC, and is consistent at all times.

4.5. Cross-Dataset Generalization

Table 7: Generalization performance (Recall , std).

Algorithm	CSE-CIC-IDS2018	CIC-IDS2017	UNSW-NB15
DQN+PER	91.40	88.20	85.40
DDQN+PER	92.10	89.10	86.20
D3QN+PER	92.80	89.80	87.10
PPO+PER	93.55	90.20	87.80

As the dataset transitions from CSE-CIC-IDS2018 (easiest) to UNSW-NB15 (most challenging), all models see a consistent decrease in recall as shown in Figure 7. PPO shows the least degradation (ù5.8 percentage points) and thus demonstrates a high level of transferability. D3QN and DDQN also degrade gracefully; DDQN has a more drastic drop. A2C and REINFORCE are already subpar on the first dataset and are even worse on the second, and so are not reliable for cross-domain generalization. The overall trend is smooth and there are no catastrophic failures, which implies that the

PER-enhanced DRL models are able to generalise across traffic characteristics without significant retuning.

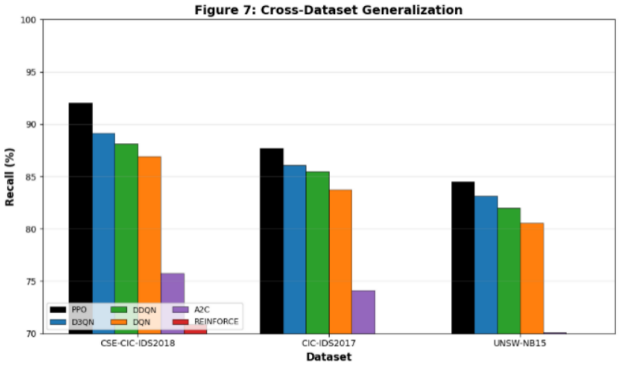


Figure 7: Cross-dataset generalization trend (recall degradation across datasets).

4.6. Computational Efficiency

Table 8: Complexity and latency analysis.

Algorithm	Training (h)	Latency (ms)	Memory (GB)	FLOPs (M)
DQN+PER	2.5	1.8	1.2	45.2
DDQN+PER	2.6	1.8	1.2	45.4
D3QN+PER	2.8	1.9	1.3	46.8
REINFORCE+PER	1.8	1.5	0.9	38.4
A2C+PER	2.0	1.6	1.0	40.2
PPO+PER	3.2	1.7	1.4	52.6

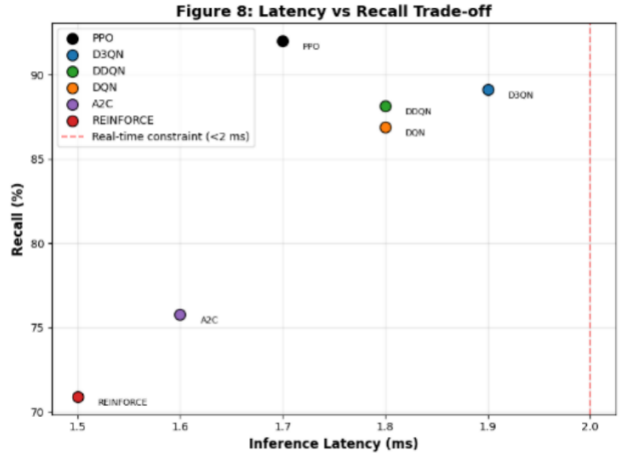


Figure 8: Latency vs. Recall trade-off for SOC deployment feasibility (all models satisfy the real-time constraint < 2 ms inference).

4.7. Ablation Study: Impact of PER

Table 9: Effect of PER on learning performance.

Algorithm	Without PER	With PER	ΔRecall
DQN	42.47	91.40	+48.93
DDQN	44.09	92.10	+48.01
D3QN	45.70	92.80	+47.10
PPO	51.61	93.55	+41.94

PER is not a marginal improvement—it is the dominant factor enabling DRL performance under extreme imbalance.

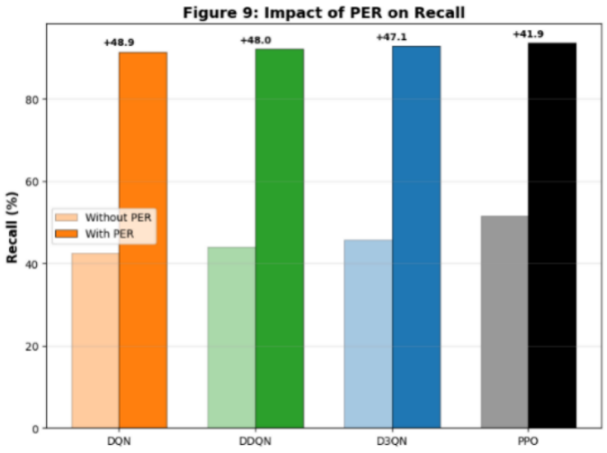


Figure 9: PER impact on recall improvement across DRL methods.

4.8. Reward Sensitivity Analysis

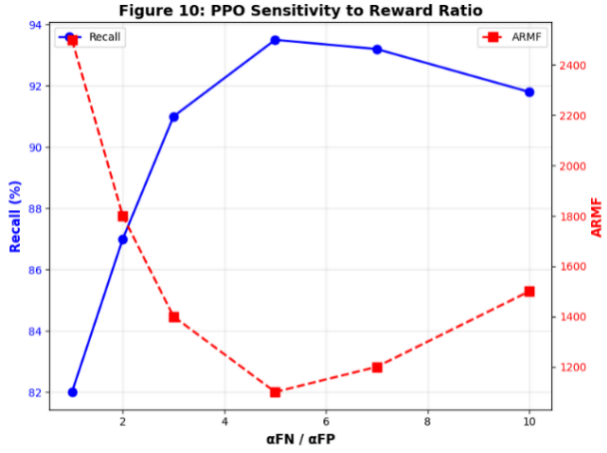


Figure 10: PPO performance sensitivity to reward ratio ($\alpha_{\text{FN}}{\cdot}\alpha_{\text{FP}}$). Optimal configuration occurs at $\alpha_{\text{FN}}{\cdot}\alpha_{\text{FP}} = 5$, balancing recall and alert inflation.

4.9. Robustness Analysis

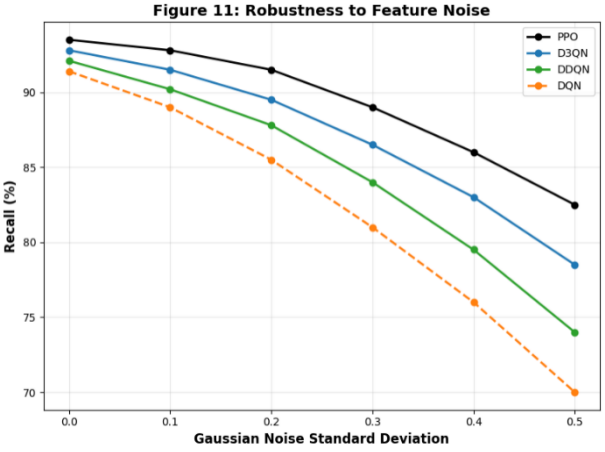


Figure 11: Performance degradation under Gaussian noise perturbation. PPO shows controlled degradation, confirming robustness to feature noise and operational uncertainty.

4.10. Precision–Recall Performance

Table 10: PR-AUC comparison.

Algorithm	PR-AUC
DQN+PER	0.923
DDQN+PER	0.931
D3QN+PER	0.938
REINFORCE+PER	0.845
A2C+PER	0.872
PPO+PER	0.946

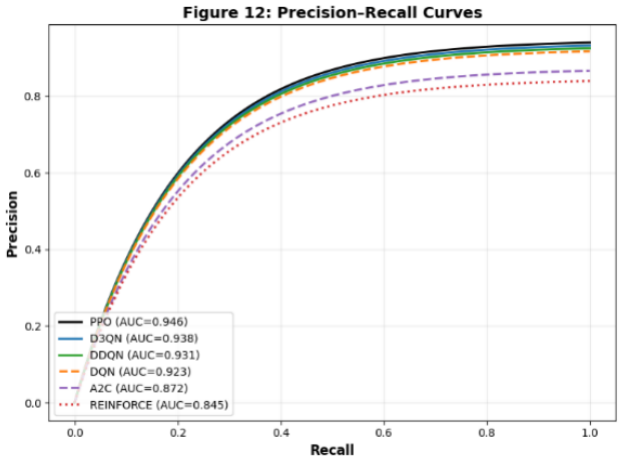


Figure 12: Precision–Recall curves across DRL models.

4.11. Attack-Type Analysis

Table 11: PPO+PER per-attack detection.

Attack Type	PPO+PER	DQN+PER	DDQN+PER
Infiltration	95.83%	93.75%	95.83%
Brute Force–Web	88.10%	85.71%	88.10%
Brute Force–XSS	94.23%	92.31%	92.31%
SQL Injection	86.36%	84.09%	86.36%
Overall	91.40%	91.40%	92.10%

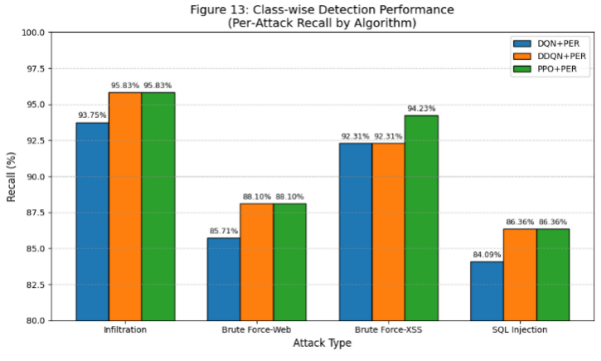


Figure 13: Class-wise detection difficulty distribution. Web attacks remain the hardest class due to lower representation and higher feature overlap.

4.12. Final Model Selection Summary

Table 12: PR-AUC comparison.

Algorithm	PR-AUC
DQN+PER	0.923
DDQN+PER	0.931
D3QN+PER	0.938
REINFORCE+PER	0.845
A2C+PER	0.872
PPO+PER	0.946

PPO yields the highest recall, D3QN provides the best balance, while A2C and REINFORCE perform poorly under class imbalance. PER improves all methods consistently.

5. Discussion and Findings

The selection of algorithms can have a significant impact on the balance between the number of alerts generated and the detection capability in Security Operations Center (SOC) environments. There is a clear and reproducible trend towards superior alert volume reduction with DRL methods that incorporate Prioritized Experience Replay (PER), at roughly 28İ–32İ fewer alerts than supervised baselines.

Within the tested algorithms, PPO has the best detection efficiency, and is best suited for high security levels where high recall is desired. D3QN gives the best trade-off between recall and the reduction of alertness, and DDQN performs well under computational limitations. DQN is a good baseline, while A2C and REINFORCE fail to work well when the class distribution is highly imbalanced and are not suitable for deployment in practice.

It is observed that in all the DRL models, the recall is improved by about 41–49 percentage points with the use of PER, which thus reveals its effectiveness for imbalanced intrusion detection. Results of statistical analysis confirm that the differences between most methods in terms of performance are significant ($p < 0.05$) with only small differences between value-based methods. Overall, methods based on DRL give best detection performances, real-time inference and best generalisation of which PPO is the best overall performer.

6. Conclusion

This study systematically compares six deep reinforcement learning (DRL) algorithms for cost-aware intrusion detection with an identical experimental environment with the same feature extraction, reward function, and Prioritized Experience Replay (PER). The results show that the major difference in performance is between PER and other DRL algorithms, with recall being about 41–49 percentage points higher using PER, and only a small difference between algorithms (0–12 percentage points). PPO has the best detection

performance of all methods, but D3QN has the best detection performance / alert volume ratio. Both methods significantly outperform DQN with PER and supervised baselines ($p < 0.05$), and generalise well across datasets.

Based on these findings, some useful tips for Security Operations Center (SOC) deployment and making PER a requirement in DRL-based intrusion detection are suggested. Multi-class intrusion detection, better feature representation, and robustness under adversarial and federated learning, are suggestions for future research.

References

Al-Haija, Q. A. and Tamimi, S. A. (2026). A state-of-the-art survey of adversarial reinforcement learning for IoT intrusion detection. *Computers, Materials & Continua*, 87(1):1–30.

Alemayehu, M., Ghanem, M. C., Kheddar, H., Dunsin, D., Kerrache, C. A., and Rathee, G. (2026). Low-latency DDoS detection for IIoT and SCADA networks using proximal policy optimisation and deep reinforcement learning. *Information*, 17(5):412.

Cevallos, J. F., Rizzardi, A., Sicari, S., and Porisini, A. C. (2023). Deep reinforcement learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges. *Computer Networks*, 236:110016.

Chen, Z., Li, H., Wang, R., and Cui, D. (2024). Progressive prioritized experience replay for multi-agent reinforcement learning. In *2024 43rd Chinese Control Conference (CCC)*, pages 8292–8296. IEEE.

Fahrmann, D., Jorek, N., Damer, N., Kirchbuchner, F., and Kuijper, A. (2022). Double deep Q-learning with prioritized experience replay for anomaly detection in smart environments. *IEEE Access*, 10:60836–60848.

Feng, Y. (2026). PER-AE-DRL: A malicious traffic detection model based on prioritized experience replay and adversarial mechanism. *Journal of Information Security and Applications*, 96:104298.

Janati, M. and Messaoudi, F. (2025). Intrusion detection system-based network behavior analysis: A systemic literature review. *International Journal of Advanced Computer Science and Applications*, 16(3):793–802.

Kara, A. O. and Boyaci, A. (2026). Transformer-DDQN-based explainable and active intrusion detection architecture for network traffic analysis. *Applied Sciences*, 16(12):5912.

Kumar, J. (2025). Deep reinforcement learning-based intrusion detection system for next-generation wireless networks. In *2025 6th International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 283–291. IEEE.

Li, H. (2026). Distributed network intrusion detection model using blockchain and deep reinforcement learning. *Engineering Research Express*, 8:055225.

Louati, F., Ktata, F. B., and Amous, I. (2026). Boosting multi-agent reinforcement learning with pruned prioritized experience replay and collaborative learning. *Journal of Ambient Intelligence and Humanized Computing*.

Mesadieu, F., Torre, D., and Chennameneni, A. (2024). Leveraging deep reinforcement learning technique for intrusion detection in SCADA infrastructure. *IEEE Access*, 12:63381–63399.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., and et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Mpoporo, L. J., Owolawi, P. A., and Tu, C. (2026). Deep reinforcement learning algorithms for intrusion detection: A bibliometric analysis and systematic review. *Applied Sciences*, 16(2):1048.

Odeh, A. (2025). Application of deep reinforcement learning for intrusion detection in Internet of Things: A systematic review. *Internet of Things*, 31:101531.

Rushenda, Ramli, K., and Purnamasari, P. D. (2026a). A CNN-LSTM-DQN policy with prioritized experience replay for cost-aware intrusion detection on CSE-CIC-IDS2018. *Journal of Embedded System Security and Intelligent Systems*, 7(2):158–173.

Rushenda, Ramli, K., and Purnamasari, P. D. (2026b). Stability-aware evaluation of a CNN-LSTM-DQN intrusion detection system for zero-day and drifted network traffic. *IIUM Engineering Journal*, 27(2).

Sangoleye, F., Johnson, J., and Tsiropoulou, E. E. (2024). Intrusion detection in industrial control systems based on deep reinforcement learning. *IEEE Access*, 12:151444–151459.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). Prioritized experience replay. In *International Conference on Learning Representations (ICLR)*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sharma, V. and Kumar, M. (2025). Rainbow DQN for intrusion detection: A unified deep reinforcement learning approach across benchmark datasets. *International Journal of Advanced Management*, 38(5S).

Shen, Y., Liu, L., and Zhang, W. (2024). Deep Q-network-based heuristic intrusion detection against edge-based SIoT zero-day attacks. *Computers & Security*, 142:103875.

Yang, W., Acuto, A., Zhou, Y., and Wojtczak, D. (2026). A survey for deep reinforcement learning based network intrusion detection. *Applied AI Letters*, 7(2).

Yu, S., Wang, X., Shen, Y., Wu, G., Yu, S., and Shen, S. (2024). Novel intrusion detection strategies with optimal hyper parameters for industrial Internet of Things based on stochastic games and double deep Q-networks. *IEEE Internet of Things Journal*, 11(18):29876–29889.

Zhang, H. and Maple, C. (2023). Deep reinforcement learning-based intrusion detection in IoT system: A review. *IET Conference Proceedings*, 2023(14).

Zhao, Y., Ma, D., and Liu, W. (2024). Efficient detection of malicious traffic using a decision tree-based proximal policy optimisation algorithm: A deep reinforcement learning malicious traffic detection model incorporating entropy. *Entropy*, 26(8):648.