https://doi.org/10.31436/iiumej.v27i2.4240

Stability-Aware Evaluation of a CNN–LSTM–DQN Intrusion Detection System for Zero-Day and Drifted Network Traffic

RUSHENDRA, KALAMULLAH RAMLI*, PRIMA DEWI PURNAMASARI

Department of Electrical Engineering, Universitas Indonesia, Depok, Indonesia

**Corresponding author: kalamullah.ramli@ui.ac.id*

(Received: 26 February 2026; Accepted: 30 April 2026; Published online: 10 May 2026)

ABSTRACT: Intrusion Detection Systems (IDS) deployed in real-world environments must operate under severe class imbalance, evolving attack strategies, and non-stationary traffic distributions. Conventional supervised and deep learning–based IDS rely on fixed decision functions, limiting their adaptability to zero-day attacks and concept drift. This paper proposes a hybrid CNN–LSTM–DQN framework combined with a stability-aware evaluation methodology. The CNN–LSTM backbone extracts spatio-temporal representations, while a Deep Q-Network (DQN) learns adaptive detection policies using an ARMF-aware reward formulation. The framework is evaluated on eleven experimental stages (E1-11), including supervised baselines, reinforcement learning optimization, zero-day generalization (LOAO), and drift scenarios. Experimental results show that supervised models have high recall (up to 98.39%) but generate too many alerts (ARMF up to 44,845). The reinforcement learning model with prioritized experience replay (E7) achieves a more balanced performance with a recall of 91.40% and an ARMF of 1,031. The proposed PER-based approach significantly improves detection performance while maintaining low alert rates, achieving a recall of 42.47% compared to naive reinforcement learning (E5). Further evaluation in real drifting conditions showed robust recall (89-92%) with a tolerable number of alerts (ARMF ≈ 1,189). These results indicate that adaptive policy learning enables a more effective trade-off between detection performance and operational cost, while ARMF-based evaluation provides a practical complement to accuracy metrics for real-world IDS deployment.

ABSTRAK: Sistem Pengesanan Pencerobohan (IDS) dalam dunia nyata perlu beroperasi di bawah ketidakseimbangan kelas yang teruk, strategi serangan berevolusi, dan taburan trafik tidak pegun. IDS berasaskan pembelajaran penyeliaan dan pembelajaran mendalam konvensional bergantung pada fungsi keputusan tetap, mengehadkan kebolehsuaian terhadap serangan sifar hari dan hanyutan konsep. Kajian ini mencadangkan gabungan rangka kerja hibrid CNN–LSTM–DQN dan metodologi penilaian yang mementingkan kestabilan. CNN–LSTM mengekstrak representasi ruang-masa, manakala Rangkaian Q Mendalam (DQN) mempelajari dasar pengesanan adaptif menggunakan formulasi ganjaran ARMF. Rangka kerja ini dinilai dengan sebelas peringkat eksperimen (E1-11) termasuk garis dasar penyeliaan, pengoptimuman pembelajaran pengukuhan, generalisasi sifar hari (LOAO) dan senario hanyutan. Dapatan eksperimen menunjukkan bahawa model penyeliaan mempunyai kadar ingat semula (recall) yang tinggi (sehingga 98.39%) tetapi menjana terlalu banyak amaran (ARMF sehingga 44,845). Model pembelajaran pengukuhan dengan ulangan pengalaman berprioriti (E7) mencapai prestasi lebih seimbang dengan kadar ingat semula 91.40% dan ARMF sebanyak 1,031. Pendekatan berasaskan PER yang dicadangkan meningkatkan keupayaan pengesanan dengan ketara sambil mengekalkan kadar amaran yang rendah berbanding pembelajaran pengukuhan naif (E5) dengan kadar ingat semula 42.47%. Kajian selanjutnya dalam keadaan hanyutan sebenar menunjukkan kadar ingat semula yang teguh (89-92%) dengan bilangan amaran yang boleh diterima (ARMF ≈ 1,189). Dapatan ini

menunjukkan bahawa pembelajaran dasar adaptif membolehkan imbangan yang lebih berkesan antara prestasi pengesanan dan kos operasi, manakala penilaian berasaskan ARMF menyediakan pelengkap praktikal pada metrik ketepatan bagi penempatan IDS dunia nyata.

KEYWORDS: *Intrusion Detection System, Deep Reinforcement Learning, Deep Q-Network, Zero-Day Attack, Concept Drift, Stability Evaluation.*

1. INTRODUCTION

Modern intrusion detection systems (IDS) must operate under highly challenging conditions characterized by severe class imbalance, evolving attack strategies, and non-stationary traffic distributions. In such environments, attack events are extremely rare compared to benign traffic, while adversarial behaviors continuously evolve, resulting in concept drift and the emergence of zero-day attacks. These conditions expose the inherent limitations of traditional IDS approaches.

Conventional signature-based systems can achieve high-precision detection for known threats but cannot generalize to new attacks. Machine learning and deep learning–based IDS improve detection capability through data-driven modeling; however, they typically rely on supervised training and fixed decision functions. As a result, their performance degrades when deployed in dynamic environments where traffic distributions shift and new attack patterns emerge. Several recent studies [1-4] have investigated reinforcement learning (RL) to enable adaptive decision-making in IDS [5-7]. Although RL-based methods improve generalization for zero-day attacks, they typically rely on shallow feature representations and are evaluated only with accuracy-based metrics. Conversely, hybrid deep learning architectures such as CNN–LSTM provide strong spatio-temporal feature extraction but lack adaptive decision mechanisms. This creates a methodological gap between representation learning and adaptive policy optimization.

In addition, existing evaluation methodologies remain largely accuracy-centric and do not reflect operational constraints in real-world deployment. In very imbalanced scenarios, models can have high accuracy but produce too many false alarms, resulting in an unmanageable number of alerts. This demonstrates the need for evaluation metrics that explicitly account for the trade-off between detection performance and alert burden.

To address such problems, we develop a hybrid CNN–LSTM–DQN architecture [8,9] with prioritized experience replay (PER) [10] and a stability-aware evaluation method. The CNN–LSTM backbone learns spatio-temporal representations of network traffic, whereas the DQN agent learns adaptive detection policies through reward-driven optimization [8, 11, 12]. The reward formulation incorporates Alerts per Million Flows (ARMF) to explicitly regulate alert volume during learning.

The proposed framework is evaluated using a structured experimental protocol (E1–E11) that covers supervised baselines, reinforcement learning optimization, zero-day generalization using Leave-One-Attack-Out (LOAO), and drift scenarios. The results indicate that the supervised models can achieve high recall (up to 98.39%) but produce excessive alert volumes (ARMF up to 44,845). In contrast, the proposed PER-based reinforcement learning approach achieves a more balanced performance, maintaining a recall of 91.40% while reducing ARMF to 1,031, demonstrating a significant improvement in the trade-off between detection capability and alert volume.

The main contributions of this study are as follows:

1. A unified CNN–LSTM–DQN–PER framework that integrates spatio-temporal feature learning with adaptive policy optimization under severe class imbalance.
2. A stability-aware evaluation methodology that uses ARMF, FP-FN trade-off analysis, and window-based metrics for IDS performance evaluation beyond normal accuracy.
3. A structured experimental protocol (E1–E11) that systematically evaluates supervised baselines, reinforcement learning refinement, zero-day generalization, and robustness under distributional drift.
4. An empirical analysis of detection–alert trade-offs, showing that the proposed approach reduces alert volume from tens of thousands (ARMF up to 44,845) to approximately 1,031 while maintaining high recall (91.40%).

2. Related Works

2.1. Traditional and Machine Learning–based IDS

Traditional intrusion detection systems are mostly signature-based, which can accurately identify known attack patterns, but cannot detect new attacks. To overcome this limitation, machine-learning-based IDS has been proposed, including anomaly-detection and supervised-classification approaches. Classical algorithms such as Support Vector Machines (SVM) [13], Random Forests, and k-Nearest Neighbors (k-NN) [14] can learn network traffic patterns to enhance detection performance. However, these methods rely on hand-crafted features and operate under relatively stationary data distributions, which limits their generalizability under dynamic attack conditions. Consequently, their detection performance is generally degraded in zero-day attack and Advanced Persistent Threat (APT) scenarios [7, 15].

Table 1. Summary of IDS approaches, strengths & limitations across methodology [16], [17].

Approach Category	Method	Strengths	Limitations	Key References
Traditional IDS	Signature-based	High precision for known attacks	Cannot detect zero-day attacks	Signature-based Intrusion Detection
	Anomaly-based	Detects unknown attacks	High FPR; sensitive to profile definition	Anomaly-based Intrusion Detection
Machine Learning–based IDS	Classical ML (SVM, RF, DT)	Effective pattern recognition	Feature engineering struggles with drift	Machine Learning
Deep Learning–based IDS	CNN, LSTM, DNN	Auto feature extraction captures spatial/temporal patterns	Static decision boundary; high computational cost	Deep Learning
Reinforcement Learning–based IDS	DQN, DRL	Adaptive decision-making; dynamic environments	Weak feature extraction if not hybridized	Reinforcement Learning;
Hybrid DL + RL IDS	CNN–LSTM + DQN	Combines feature learning and adaptive policy optimization	Limited exploration of stability (e.g., replay strategies); evaluation is often static	Hybrid Deep Learning Reinforcement Learning
Advanced RL Enhancements	Prioritized Experience Replay (PER)	Focuses on important transitions	Operational trade-offs not well studied	Prioritized Experience Replay
Evaluation Approaches	Static Metrics (Accuracy, F1-score)	Standardized comparison; easy to interpret	Ignores alert volume and temporal stability	F1 Score

https://doi.org/10.31436/iiumej.v27i2.4240

2.2. Deep Learning–based IDS and Hybrid Architectures

Deep learning–based IDS learns representations directly from network traffic data, reducing reliance on handcrafted features used in classical approaches [5-7]. Convolutional Neural Networks (CNN) capture spatial correlations among flow features, while Long Short-Term Memory (LSTM) networks model temporal dependencies in sequential traffic data. Hybrid CNN–LSTM [18] architectures leverage both properties and have been reported to perform effectively in detecting complex, multi-stage attack patterns, such as Advanced Persistent Threats (APTs) [1-3]. Despite these advantages, most deep learning–based IDSs are trained under supervised settings using fixed decision functions learned during training. When attack conditions change due to novel attack types or distributional shifts, detection performance degrades, as these models lack mechanisms to adapt after deployment.

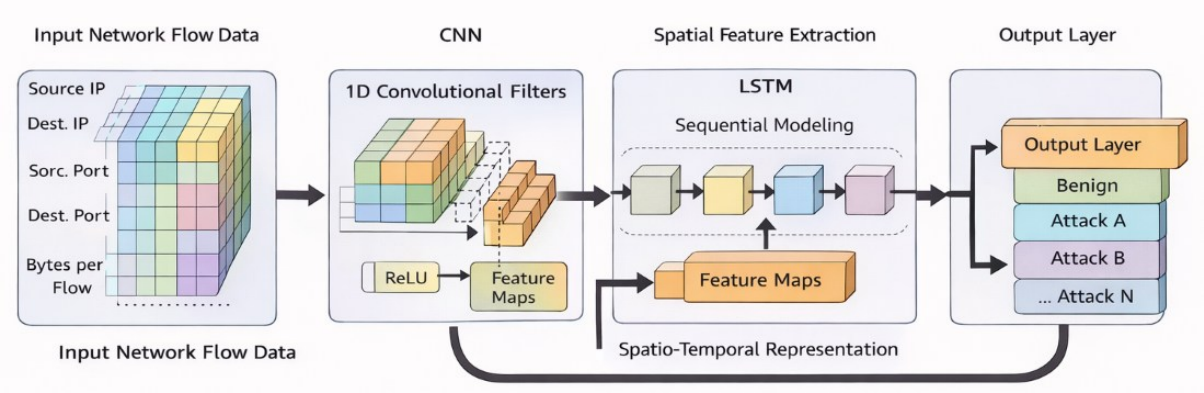

Figure 1. Illustration of CNN–LSTM architecture for spatio-temporal feature learning in IDS

2.3. Reinforcement Learning for Adaptive Intrusion Detection

Reinforcement learning (RL) provides an adaptive alternative to detection based on supervised learning, in which models can adapt their policies to environmental feedback rather than to fixed training distributions. Deep reinforcement learning (DRL) based IDS, especially those using Deep Q-Networks (DQN), have been reported to offer increased adaptability in dynamic network environments [8], [11], [12]. However, most RL-based IDS rely on relatively shallow feature representations and do not fully capture the spatio-temporal structure of network traffic. Moreover, their performance is usually evaluated with standard classification metrics that do not capture operational constraints such as alert volume and temporal stability.

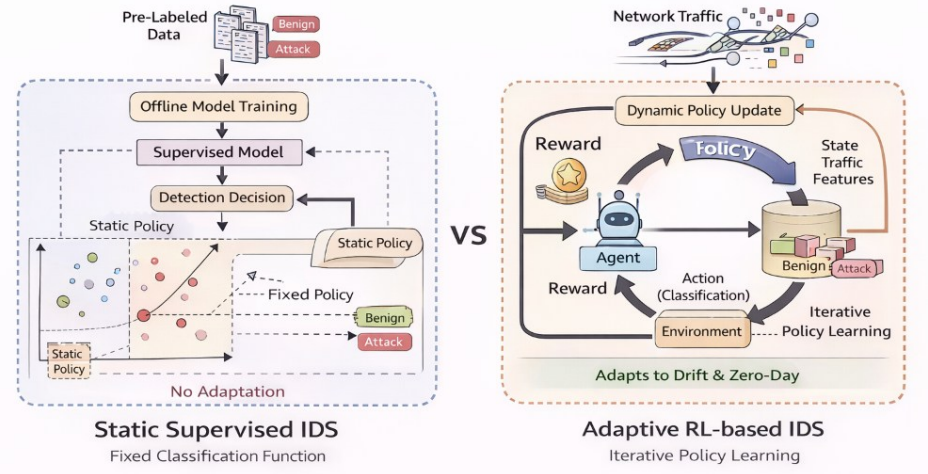

Figure 2. Comparison between static supervised IDS and adaptive RL-based IDS decision mechanisms

https://doi.org/10.31436/iiumej.v27i2.4240

2.4. Prioritized Experience Replay and Learning under Imbalance

Severe class imbalance is a fundamental challenge in intrusion detection, where attack events are orders of magnitude rarer than benign traffic. This problem is tackled by Prioritized Experience Replay (PER) [11], which assigns higher sampling probabilities to transitions with high temporal-difference (TD) errors, emphasizing the importance of informative experiences during training [10, 19, 20]. This is particularly relevant for intrusion detection, where attack samples are rare and uniform replay strategies tend to under-represent critical events. PER has primarily been explored as a training optimization technique, although it is effective at improving learning efficiency. Its effects on operational detection behaviors, e.g., false positive–false negative (FP–FN) trade-offs and temporal stability under imbalanced traffic conditions, are still largely unknown.

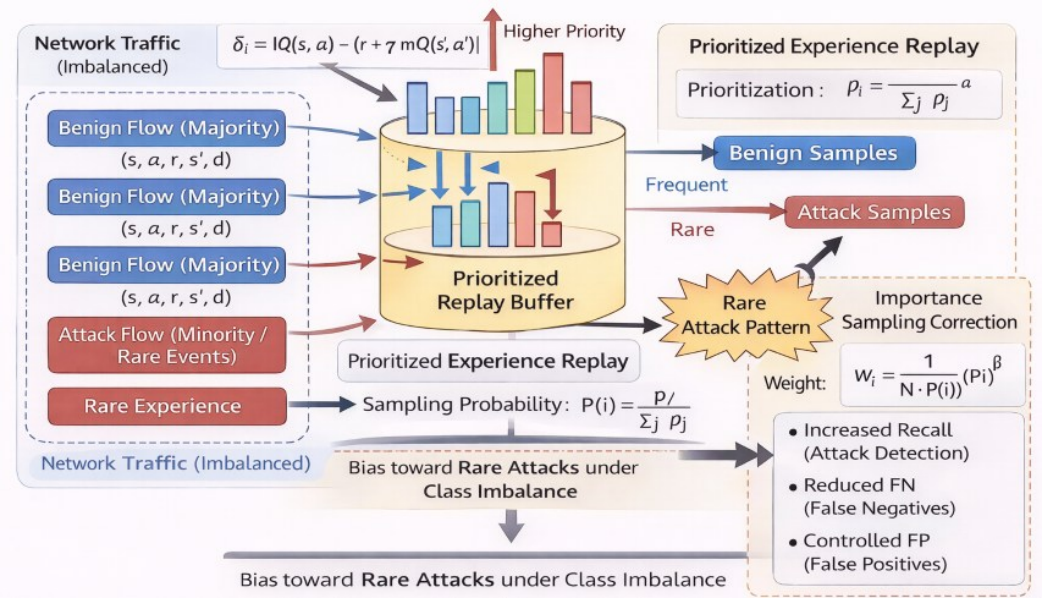

Figure 3. Prioritized Experience Replay mechanism highlighting sampling bias toward rare attack events.

2.5. Zero-Day and APT Detection: Fragmented Research Directions

Research on zero-day detection and APT detection has been conducted separately. Zero-day detection primarily aims to generalize to unseen attack types using reinforcement learning agents or anomaly-based techniques [8, 9], whereas APT detection research models multi-stage intrusion behavior as sequences [5, 6, 15]. This separation constrains the development of integrated IDS frameworks capable of handling both novel attack patterns and multi-stage attack dynamics. A further methodological limitation is that many existing studies rely on random train–test splits, which do not adequately reflect realistic deployment conditions where traffic distributions evolve over time. Table 2 presents a structured comparison of current IDS approaches.

Table 2. Comparative analysis of IDS approaches across zero-day detection, APT modeling, and adaptivity dimensions.

Approach Category	Core Technique	Zero-Day Detection Capability	APT Detection Capability	Adaptivity Level	Alert Control (ARMF Awareness)	Key Strengths	Key Limitations
Signature-Based IDS [21]	Rule-based detection	≈ 0% (no generalization)	≈ 0%	None	No	High precision for known attacks	Cannot detect unseen attacks
Hybrid Deep Learning (CNN–LSTM) [22], [23], [24], [25], [26]	Spatio-temporal feature learning	Low (typically <50% unseen recall)*	High (sequence modeling)	Static (retraining required)	No	Strong feature extraction	No adaptive decision mechanism
APT-Focused Deep Learning [7], [27], [28]	CNN–LSTM / SAE–LSTM	Low–Moderate	High (multi-stage modeling)	Static	No	Effective for known APT patterns	Limited generalization to new attacks
RL-based IDS (Basic DQN) [29], [30]	DQN + MLP	Moderate–High (family-level generalization)	Low–Moderate	Adaptive (policy learning)	Partial	Adaptive detection policy	Weak feature representation
RL + Temporal Modeling [20], [31]	DQN + LSTM	High	Moderate	Adaptive	Partial	Temporal context improves RL decisions	No explicit imbalance handling
RL for Active Learning [32]	DQN + BiLSTM	High (label-efficient)	Low–Moderate	Adaptive	Partial	Effective under limited labels	Not end-to-end IDS
Advanced DRL (Rainbow DQN) [10]	DQN + PER + advanced RL	High (imbalance-aware)	Moderate	Highly Adaptive	Partial	Stable RL training	No CNN–LSTM integration
Hybrid DL + RL (Partial) [33]	CNN–LSTM + Q-learning	Moderate–High	Moderate	Partially Adaptive	Limited	Combined representation + RL	No unified optimization
Proposed CNN–LSTM–DQN–PER	Hybrid DL + DRL + PER	**High (~91% zero-day recall in LOAO)**	**High**	**Fully Adaptive (online policy)**	**Yes (ARMF-aware, 1,031 vs 44,845 baseline)**	Unified feature + policy + imbalance handling	Higher computational cost

https://doi.org/10.31436/iiumej.v27i2.4240

2.6. Research Gap and Positioning of This Study

The literature review identifies four interconnected gaps. First, deep learning-based IDS offers strong spatio-temporal feature representations but lacks adaptive decision mechanisms. Second, RL-based IDS can be adaptive but relies on shallow feature encoders that do not capture the full temporal structure of traffic. Third, the role of Prioritized Experience Replay (PER) has been primarily studied in terms of training efficiency, with little analysis of its impact on detection stability and FP-FN operational trade-offs in IDS frameworks. Fourth, Zero-day and APT detection are usually treated as separate research topics, and evaluation methodologies are primarily based on accuracy metrics, without considering operational factors such as alert volume, temporal stability, or distributional shifts. This work closes these gaps by proposing a unified CNN-LSTM-DQN-PER framework and evaluating it using a stability-aware protocol alongside traditional accuracy-based metrics.

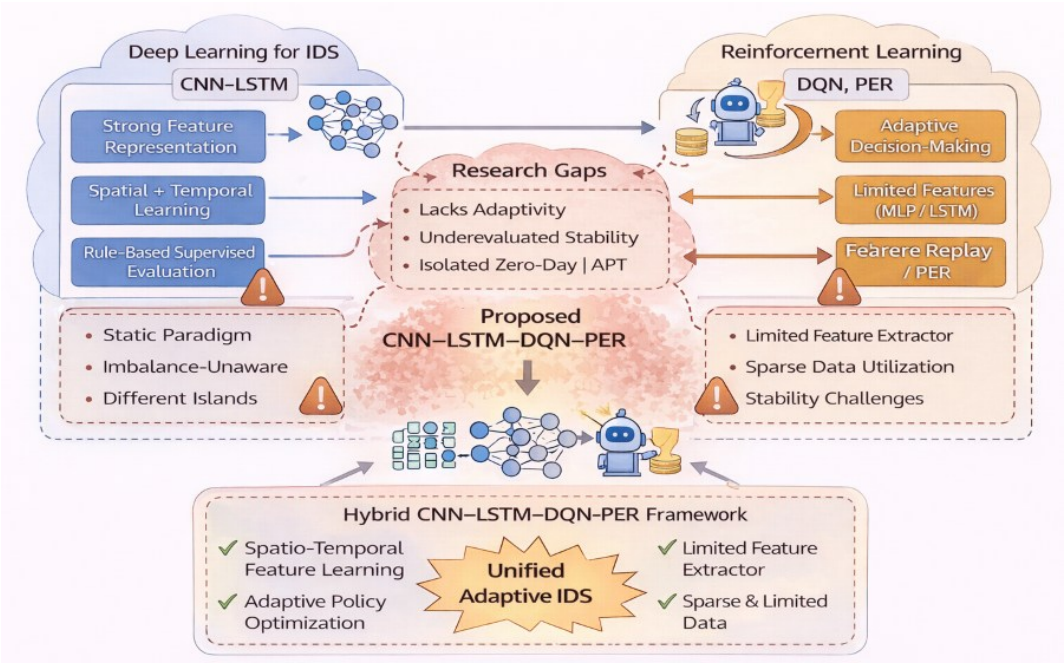

Figure 4. Research gap and positioning of the proposed CNN–LSTM–DQN–PER framework.

3. METHODOLOGY

3.1. Overall Framework

The proposed framework aims to address three major limitations: poor handling of zero-day attacks, vulnerability to distributional shifts, and poor detection performance under severe class imbalance. Instead of using the fixed decision function learned during training, the framework enables adaptive, stability-aware decision-making by integrating deep feature learning and reinforcement learning–based policy optimization.

The architecture learns spatio-temporal representations of network traffic by combining spatial features extracted with a CNN and temporal features modeled with an LSTM. These representations are used as state inputs to a DQN agent, which learns an adaptive detection policy through reward-driven learning. Prioritized Experience Replay focuses learning on rare but informative transitions, which is particularly relevant under imbalanced traffic conditions. Figure 5 illustrates the overall research framework.

https://doi.org/10.31436/iiumej.v27i2.4240

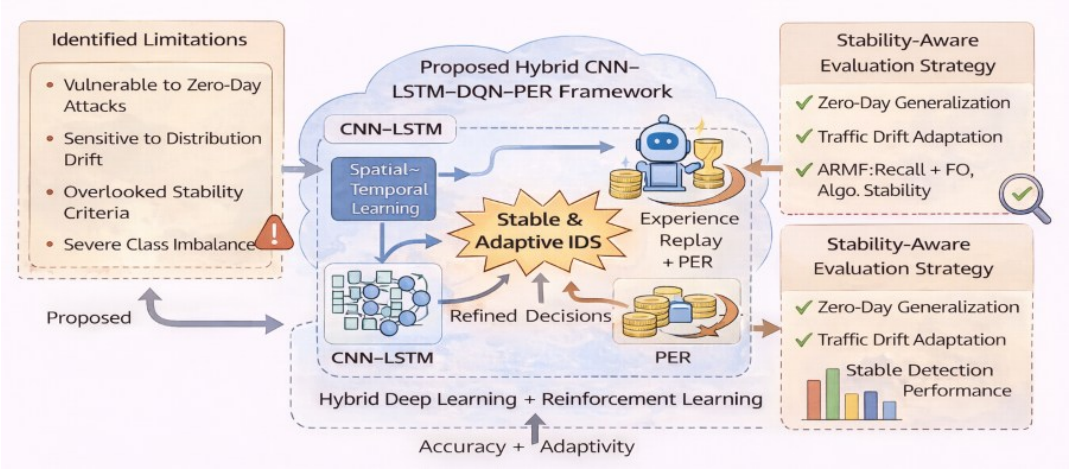

Figure 5. Overall Research Framework Diagram

3.2. Hybrid Deep Learning Architecture

The CNN module learns hierarchical spatial features from the normalized network flow attributes by using multi-scale one-dimensional convolution kernels, batch normalization, and ReLU activation. The feature maps generated by the CNN module are then fed into a bidirectional LSTM network, which can capture temporal dependencies in sequential traffic data, an important feature for identifying multi-stage, persistent attack behaviors such as Advanced Persistent Threats (APTs). Bidirectional processing can incorporate contextual information from both the past and future of the sequence, enabling more comprehensive modeling of temporal patterns. The CNN-LSTM backbone is a feature encoder that converts network flow data into a compact latent representation, which is then used as the state representation for the downstream reinforcement learning agent.

3.3. Reinforcement Learning for Adaptive Decision-Making

The proposed framework exploits a Deep Q-Network (DQN) to overcome the limitations of static classification models. The intrusion detection task is formulated as a sequential decision-making problem, in which an agent interacts with a network traffic environment and learns to maximize the expected cumulative discounted reward over time. The CNN–LSTM latent representation is used as the state representation for the DQN agent, which selects actions corresponding to binary classification decisions (benign or attack). The reward function provides positive rewards for correct detections, while negative rewards for false positives and missed attacks. This formulation encourages the agent to learn a detection policy that accounts for the infrequency of attack events while still exhibiting efficient detection behavior.

3.4. Prioritized Experience Replay for Imbalanced Learning

In our case, it is the severe class imbalance, where benign traffic outnumbers attacks by orders of magnitude, and standard uniform experience replay samples from the majority-class (benign) transitions. This problem has been alleviated by Prioritized Experience Replay (PER) [13], which samples transitions from memory with replacement, with a sampling probability proportional to the TD error, favoring samples with greater learning signals. This mechanism promotes better learning on rare attack instances while maintaining stable training dynamics.

https://doi.org/10.31436/iiumej.v27i2.4240

3.5 Two-Stage Learning Strategy

In this paper, a two-stage training method is proposed for symptom checkers: supervised pre-training and reinforcement learning-based optimization. The CNN–LSTM backbone is first trained in a supervised manner on labeled traffic data to learn discriminative feature representations. This warm-start phase provides the RL agent with an informative state representation, alleviating instability caused by random initialization during early exploration.

The second stage freezes the pre-trained backbone and attaches it to the DQN agent. Then the detection policy is optimized by the DQN agent with a reward-driven approach. Two-stage design that can utilize the expressive power of supervised deep learning as well as the adaptivity of reinforcement learning with appropriate action space shaping or conditioning. The complete experimental pipeline from our results is presented in Figure 6.

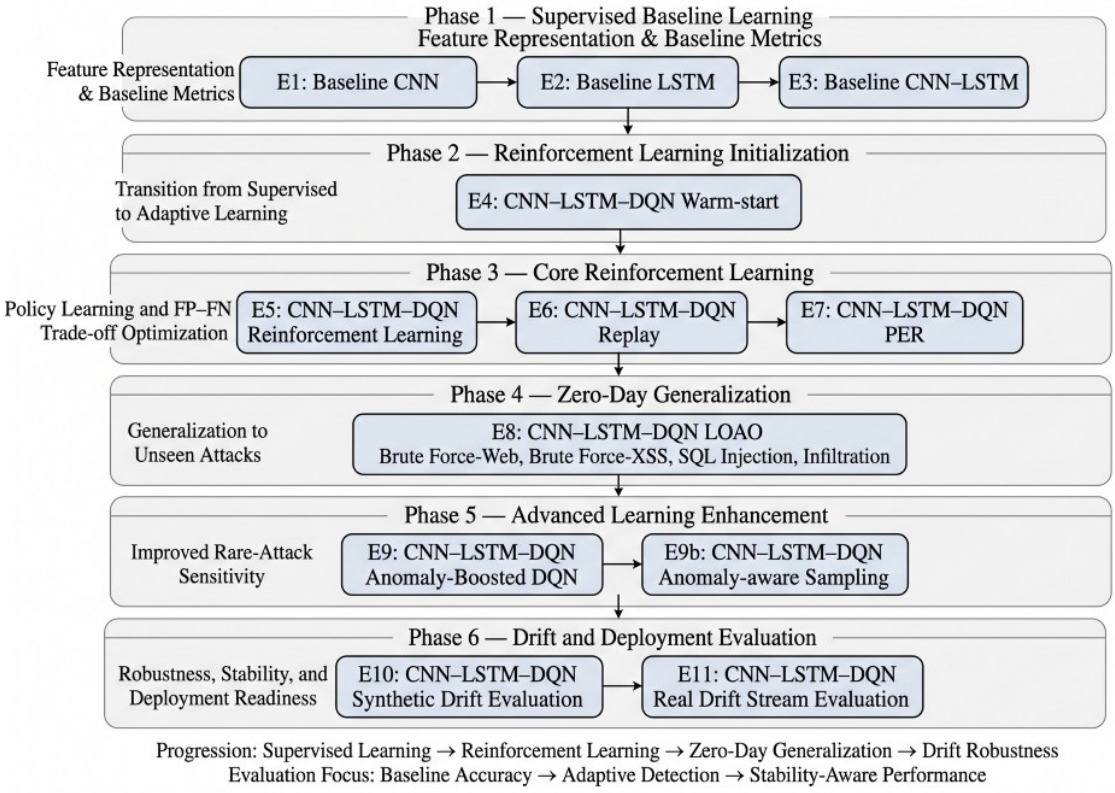

Figure 6. Experimental Pipeline of CNN-LSTM-DQN IDS (E1 - E11)

3.6. Dataset and Problem Formulation

We conduct experiments on a large-scale publicly available benchmark dataset, CSE-CIC-IDS2018 [34], characterized by high-dimensional network flow features, extreme class imbalance with benign traffic vastly outnumbering attack traffic, and various attack families, such as web-based and infiltrative. Let $D = \{(x_i, y_i)\}_{i=1}^{N}$. Denote the dataset, where $xi \in R^d$ represents a network flow feature vector and $y_i \in \{0,1\}$ denotes the binary label (benign or attack).

https://doi.org/10.31436/iiumej.v27i2.4240

Table 3. Dataset statistics and class distribution

Split / Scenario	Total Flows	Benign Flows	Attack Flows	Attack Ratio	Imbalance Ratio (Benign : Attack)
Training (Global)	10,788,508	10,787,766	742	0.0069%	14,538 : 1
Testing (Global)	2,697,128	2,696,942	186	0.0069%	14,499 : 1
RL Subset (E4–E9)	500,742	500,000	742	0.1481%	673 : 1
Window 1 (E10/E11)	674,282	674,235	47	0.0070%	14,345 : 1
Window 2 (E11)	674,282	673,945	337	0.0500%	1,999 : 1
Window 3 (E11)	674,282	674,080	202	0.0300%	3,337 : 1
Window 4 (E11)	674,282	674,215	67	0.0099%	10,061 : 1

3.7. Learning Objective

The intrusion detection problem is formulated as a sequential decision-making task with three operational objectives: (i) maximize recall for attack detection, (ii) minimize false positives to control alert volume, and (iii) maintain consistent detection behavior under varying traffic conditions. Unlike conventional supervised classifiers that optimize fixed decision functions, the proposed approach learns an adaptive policy through interaction with the environment.

3.8. Supervised Feature Learning via CNN–LSTM Backbone (E1–E4)

3.8.1. CNN-based Feature Extraction

For an input flow x_i, the CNN provides a latent spatial representation where:

$$z_i^{\{\mathrm{CNN}\}} = f_{\{\mathrm{CNN}\}}(x_i) \tag{1}$$

This ensures that all input to a model is higher-level feature representations of the data, less sensitive to noise in the original features.

3.8.2. Temporal Modeling with LSTM

The output from CNN is passed into a bidirectional LSTM in order to capture the sequential dependencies in traffic behavior $z_i = f_{\{\mathrm{LSTM}\}}\left(z_i^{\{\mathrm{CNN}\}}\right)$. The embedding z_i is then used as the input to downstream reinforcement learning, where it plays a role analogous to state representation. All components are trained in a supervised fashion during the warm-start phase and fixed during reinforcement learning optimization afterward, providing a stable state representation for all downstream tasks.

https://doi.org/10.31436/iiumej.v27i2.4240

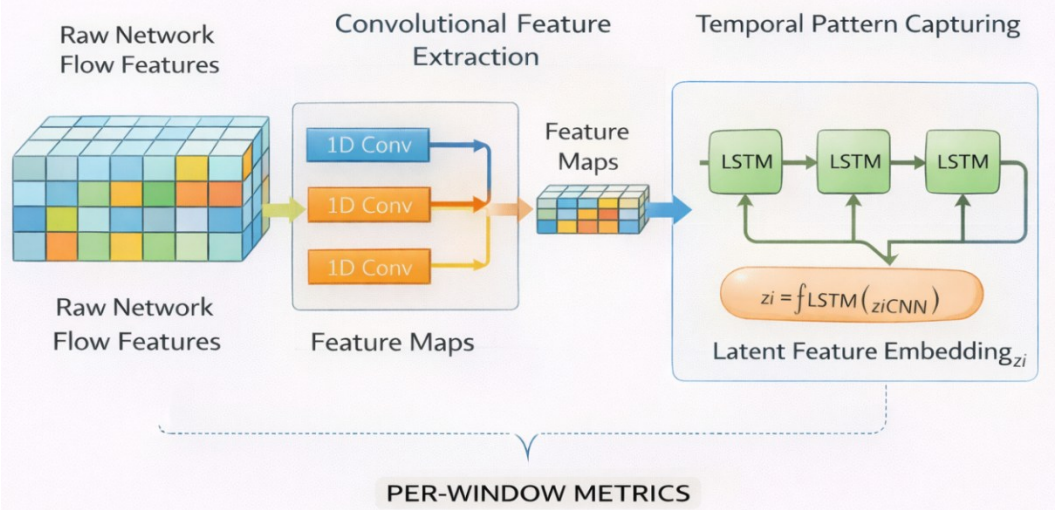

Figure 7. CNN–LSTM feature learning pipeline

3.8.3. Supervised Warm-Start Training (E4)

Before Reinforcement Learning training, the CNN–LSTM backbone is trained to minimize class-weighted cross-entropy loss: $\mathcal{L}_C = -\sum_i w_{y_i} \log p(y_i \mid x_i)$, where$(w_{y_i})$ denotes class-specific weights used to address class imbalance. This warm-start phase provides the model with informative feature representations, which support more stable exploration during the early stages of reinforcement learning.

3.9. Reinforcement Learning Formulation

3.9.1 Markov Decision Process (MDP)

The detection task is modeled as an MDP defined by: State$(s_i = [z_i, \psi_i])$, where (ψ_i) is the anomaly score included only in anomaly-boosted variants (E9/E9b); Action $(a_i \in 0,1)$ (benign or attack); Reward (r_i)as defined in Section 3.8.2; Transition $(s_i \rightarrow s_{i+1})$ following sequential flow processing.

3.9.2. Reward Design

The reward function is designed to trade off detection performance and operational cost. The base classification reward is denoted as:

$$r_t^{\text{cls}} = \alpha_{TP} \cdot \mathbb{1}[TP] + \alpha_{TN} \cdot \mathbb{1}[TN] - \alpha_{FP} \cdot \mathbb{1}[FP] - \alpha_{FN} \cdot \mathbb{1}[FN] \quad (2)$$

where $\alpha_{TP}, \alpha_{TN}, \alpha_{FP}, \alpha_{FN} > 0$ are reward and penalty weights.

To incorporate alert-rate awareness, an additional penalty based on Alerts per Million Flows (ARMF) [35] is applied. For an evaluation window w with N_w samples, ARMF is defined as:

$$\text{ARMF}_w = \left(\frac{\sum_{t \in w} \mathbb{1}[a_t = 1]}{N_w}\right) \times 10^6 \quad (3)$$

The final reward is given by:

$$r_t = r_t^{\text{cls}} - \lambda \cdot \frac{\text{ARMF}_w}{10^6}, \quad (4)$$

where $\lambda \geq 0$ controls the trade-off between detection performance and alert volume.

https://doi.org/10.31436/iiumej.v27i2.4240

3.10. DQN-Based Policy Optimization (E5–E7)

3.10.1. Q-Network and Action Selection

The detection policy is parameterized by a Deep Q-Network (DQN):

$$Q_{\phi}(s,a) \approx E[\,R_t \mid s_t = s, a_t = a\,] \tag{5}$$

where ϕ denotes the network parameters. Actions are selected using an ε-greedy strategy, with ε progressively annealed during training.

3.10.2. Experience Replay and Target Network (E6)

Transition (s,a,r,s') are stored in a replay buffer $(\mathcal{B})$ and sampled in mini-batches during training. A separate target network (Q_{ϕ^-}) is periodically updated using the parameters of the current Q-network Qφ. This delayed update mechanism stabilizes temporal-difference learning and helps reduce instability during policy optimization under imbalanced data conditions.

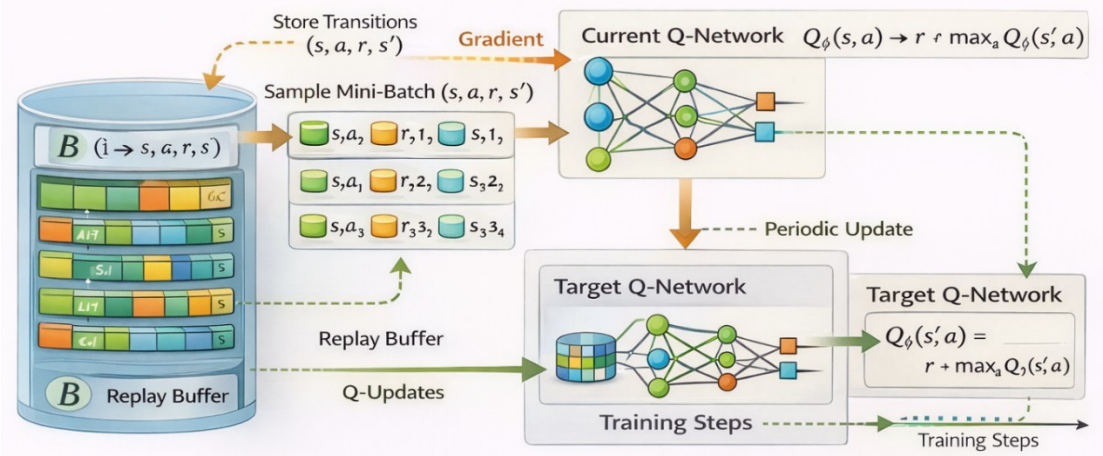

Figure 8. Replay buffer and target network mechanism

3.10.3. Prioritized Experience Replay (E7)

Prioritized Experience Replay (PER) samples transitions with a probability proportional to the absolute value of the temporal-difference (TD) error, thereby prioritizing better learning signal interruptions. This mechanism then biases the agent to remember the informative experiences during training, which is more democratic under severe class imbalance.

3.11. Anomaly-Boosted Reinforcement Learning (E9, E9b)

A shallow autoencoder trained exclusively on benign traffic is used to compute per-flow anomaly scores:

$$\psi_i = |x_i - \hat{x}_i|_2 \tag{6}$$

where $\hat{x}_i$ denotes the reconstructed input. These anomaly scores are concatenated with the CNN–LSTM latent representation to augment the RL state representation, providing the agent with an additional unsupervised signal during policy learning.

https://doi.org/10.31436/iiumej.v27i2.4240

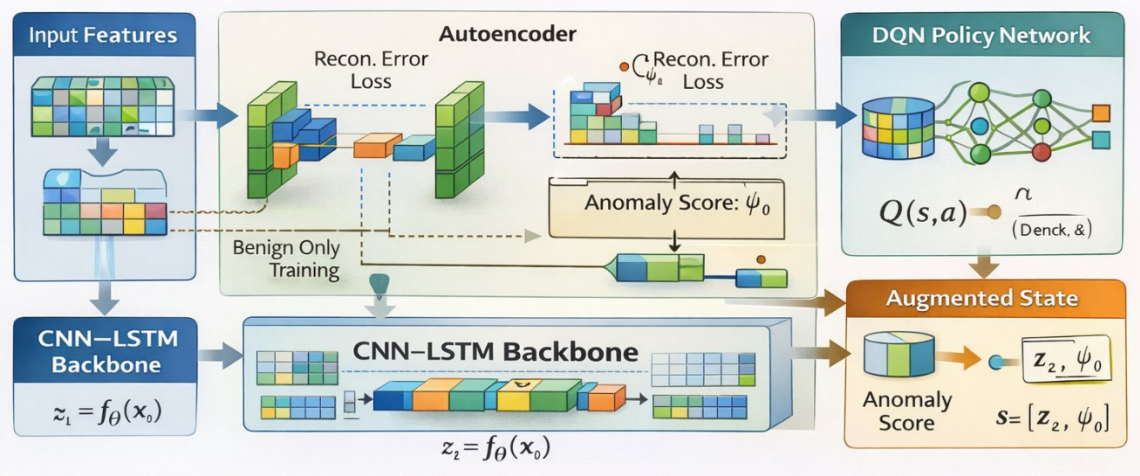

Figure 9. Autoencoder-assisted state augmentation

3.12. Evaluation Protocols (E1–E11)

3.12.1. Standard Imbalanced Test

All models are evaluated on a held-out imbalanced test set using confusion-matrix components (TP, TN, FP, FN) and derived metrics, including accuracy, precision, recall, F1-score, false-positive rate (FPR), false-negative rate (FNR), and Alerts per Million Flows (ARMF).

Table 4. Metrics for standard test evaluation

Exp	Model / Config.	TN	FP	FN	TP	Acc.	Precision (Attack)	Recall (Attack)	F1-score (Attack)	ARMF
E1	Linear SVM (baseline)	2,607,851	89,091	60	126	0.9669	0.0014	0.6774	0.0028	33,078
E1	CNN baseline	2,576,170	120,772	3	183	0.9552	0.0015	0.9839	0.0030	44,846
E2	LSTM baseline	2,270,467	426,475	110	76	0.8418	0.0002	0.4086	0.0004	158,150
E3	CNN–LSTM	2,612,399	84,543	5	181	0.9687	0.0021	0.9731	0.0043	31,413
E4	CNN–LSTM–DQN (warm-start)	2,608,503	88,439	3	183	0.9672	0.0021	0.9839	0.0041	32,858
E5	CNN–LSTM–DQN (RL)	2,695,987	955	7	79	0.9996	0.0764	0.4247	0.1295	383
E6	CNN–LSTM–DQN (Replay)	2,662,770	34,172	8	158	0.9873	0.0046	0.8495	0.0092	12,678
E7	CNN–LSTM–DQN (PER)	2,694,330	2,612	6	170	0.9990	0.0611	0.9140	0.1146	1,031
E9	AE + RL (Anomaly-boosted)	2,695,973	969	107	79	0.9996	0.0754	0.4247	0.1280	389
E9b	AE + RL (variant)	2,695,741	1,201	107	79	0.9995	0.0617	0.4247	0.1078	475

3.12.2 LOAO Zero-Day Evaluation (E8)

In the Leave-One-Attack-Out (LOAO) protocol, one attack family is excluded from the training set and used exclusively for testing. This setup provides a more realistic assessment of zero-day detection capability compared to a standard random train–test evaluation.

Table 5. LOAO zero-day evaluation setup

LOAO Split ID	Held-out Attack Type (Zero-day)	Training Data Composition	Test Data Composition	Purpose of Evaluation
LOAO-1	Brute Force – Web	All benign flows + all attack types **except** Brute Force – Web	Only Brute Force – Web attack flows	Evaluate generalization to unseen web-based brute-force patterns
LOAO-2	Brute Force – XSS	All benign flows + all attack types **except** Brute Force – XSS	Only Brute Force – XSS attack flows	Assess robustness against unseen XSS-based brute-force attacks
LOAO-3	SQL Injection	All benign flows + all attack types **except** SQL Injection	Only SQL Injection attack flows	Measure zero-day detection capability for injection-style attacks
LOAO-4	Infiltration	All benign flows + all attack types **except** Infiltration	Only the infiltration attack flows	Test detection of stealthy lateral movement and infiltration behaviors

3.12.3. Drift Stream Evaluation (E10–E11)

Model behavior is evaluated across sequential test windows with varying proportions of attack traffic, simulating distributional shifts over time. Temporal stability is quantified using the mean, variance, and standard deviation of ARMF and recall across evaluation windows.

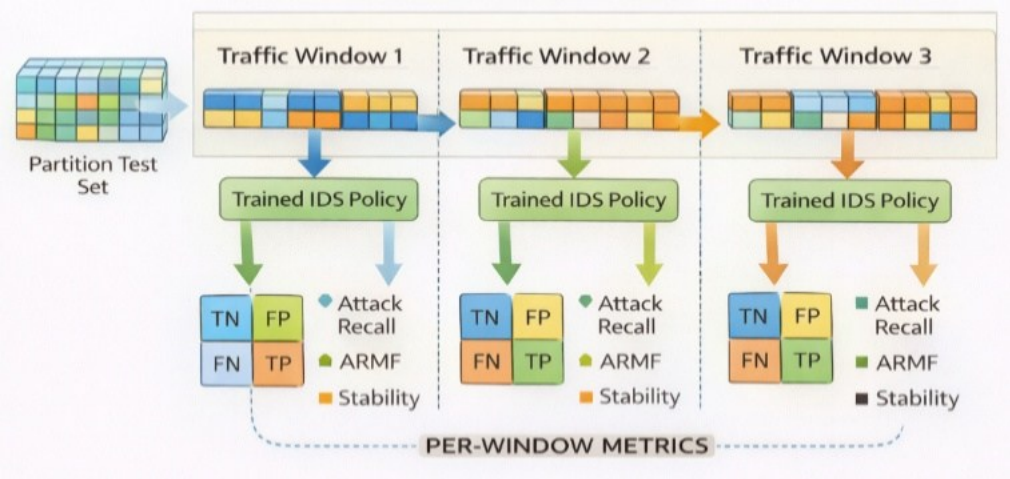

Figure 10. Drift stream evaluation workflow

3.13. Stability-Aware Evaluation Metrics

Beyond conventional accuracy-based metrics, this study reports Alerts per Million Flows (ARMF), FP–FN trade-off curves, and window-based stability statistics [31, 35-40]. Table 6 summarizes all evaluation metrics used throughout the study.

Table 6. Evaluation Metrics Definition for Stability-Aware Intrusion Detection System

Metric	Formula / Definition	Description	Relevance to This Study
Accuracy (Acc)	$\frac{TP+TN}{TP+TN+FP+FN}$	Measures the overall correctness of predictions.	Provides baseline performance, but is insufficient under class imbalance.
Precision (Prec)	$\frac{TP}{TP+FP}$	Proportion of predicted attacks that are truly attacks.	Critical to reduce false alarms in IDS environments.
Recall (Detection Rate)	$\frac{TP}{TP+FN}$	Proportion of actual attacks correctly detected.	Essential for detecting zero-day and rare attacks.

F1-Score	$\frac{Precision \cdot Recall}{Precision + Recall}$	Harmonic mean of precision and recall.	Balances detection capability and false alarm control.
False Positive Rate (FPR)	$\frac{FP}{FP + TN}$	Rate of benign flows incorrectly classified as attacks.	Impacts alert fatigue and operational overhead.
False Negative Rate (FNR)	$\frac{FN}{FN + TP}$	Rate of attacks missed by the system.	Represents critical undetected threat risk.
Alert Rate per Million Flows (ARMF)	$\frac{FP}{Total\ flows} \times 10^6$	Number of false alerts per million flows.	Reflects real-world IDS scalability and cost efficiency.
FP–FN Trade-off Curve	Relationship between FPR and FNR	Trade-off between false alarms and missed detections.	Enables risk-sensitive threshold tuning.
Zero-Day Detection Rate (LOAO)	Recall on unseen attack classes	Measures the detection of unknown attacks.	Evaluates generalization capability to zero-day threats.
Drift Stability (Window-based)	Mean, Variance, Std over time windows	Measures consistency across data distribution shifts.	Evaluates robustness under evolving attack patterns.

3.14. Implementation and Experimental Configuration

The CNN–LSTM–DQN framework is implemented in PyTorch, which provides a dynamic computation graph suitable for both supervised and reinforcement learning components. Classical machine learning baselines are implemented using scikit-learn. All experiments are conducted in the same Python environment, with GPU acceleration enabled when available. The random seeds for Python, NumPy, and PyTorch are fixed to improve reproducibility.

3.14.1. Supervised Training Configuration (E1–E4)

Supervised training uses class-weighted cross-entropy loss. The Adam optimizer is employed with a learning rate of 1×10^{-3} and a batch size of 256. All models are trained for five epochs to ensure consistent training conditions across methods.

3.14.2. Reinforcement Learning Configuration (E5–E9)

The DQN is trained with a discount factor γ=0.99. Experience replay is implemented using a replay buffer with mini-batch updates of size 64. An ε-greedy exploration strategy is applied, with ε gradually reduced during training. Target network synchronization is introduced in E6 to stabilize Q-value updates. From E7 onward, prioritized experience replay (PER) with β-annealing is applied. All reinforcement learning stages use the CNN–LSTM backbone trained in E4 as a fixed feature extractor.

3.14.3 Architecture Details

The CNN module is built using multiple 1D convolutional layers, batch normalization, and ReLU activations. The LSTM module is multi-layered, bidirectional, and its output is fed into a fully connected layer. The DQN agent is implemented as a fully connected Q-network with NoisyLinear layers to facilitate exploration. The replay buffer supports both uniform sampling (E5–E6) and prioritized sampling (E7 onwards).

The experimental pipeline is structured as follows: E1–E3 establish supervised baselines; E4 performs warm-start training for CNN–LSTM; E5–E7 introduce and refine reinforcement

learning optimization; E8 evaluates zero-day generalization; E9–E9b apply anomaly-augmented reinforcement learning; and E10–E11 assess robustness under distributional drift.

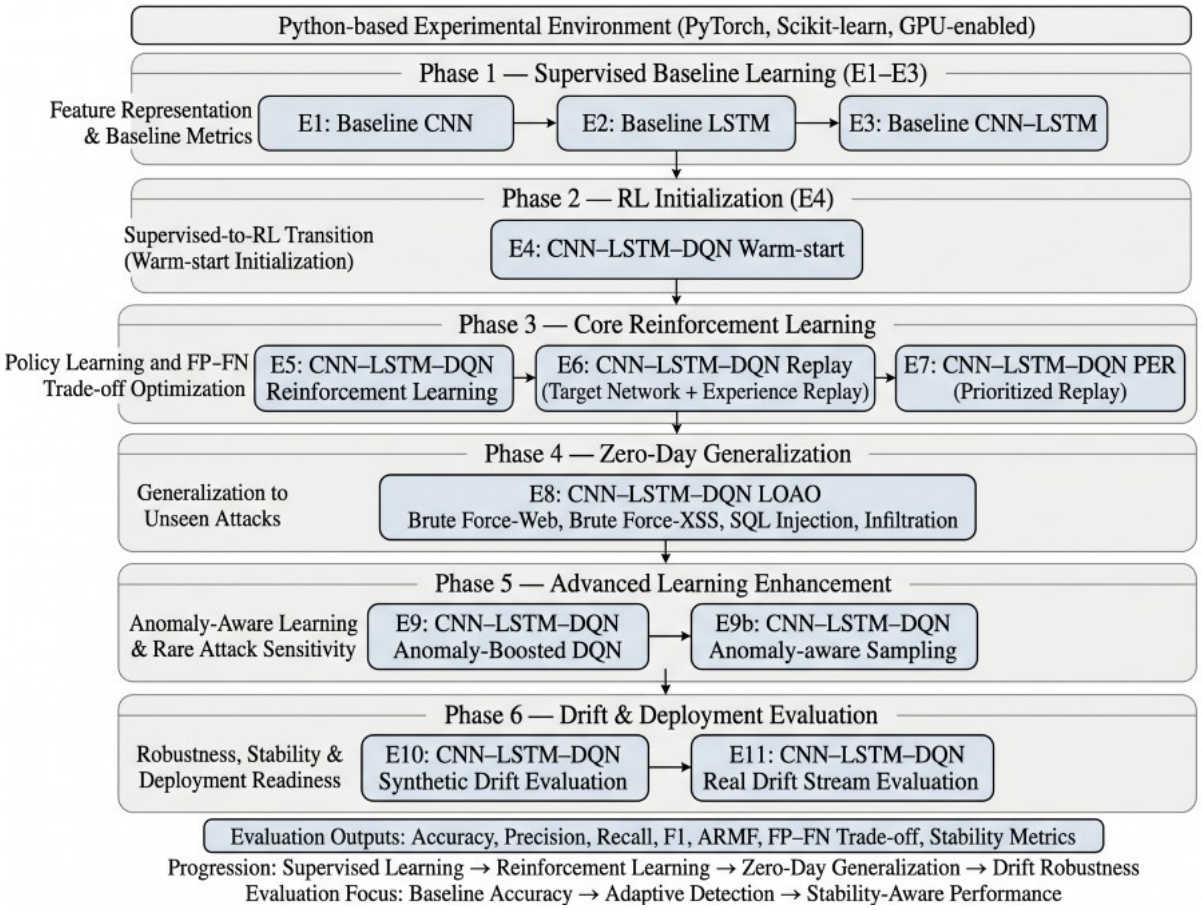

Figure 11. Implementation and experimental configuration of the proposed CNN-LSTM-DQN framework

4. Experimental Results and Analysis

4.1. Experimental Overview

In this section, a systematic assessment using the experimental protocol provided in Section 3 is conducted to review the proposed CNN–LSTM–DQN–PER framework. We proceed from supervised baselines (E1–E3) to CNN–LSTM warm-start initialization (E4), and then optimization through reinforcement learning (E5–E7). The LOAO protocol (E8) is used to evaluate zero-day generalization, while tests with anomaly-augmented reinforcement learning variants are shown in E9–E9b. Finally, robustness is evaluated under synthetic and real-world traffic-drift scenarios (E10–E11). The overall experimental pipeline is shown in Figure 12.

https://doi.org/10.31436/iiumej.v27i2.4240

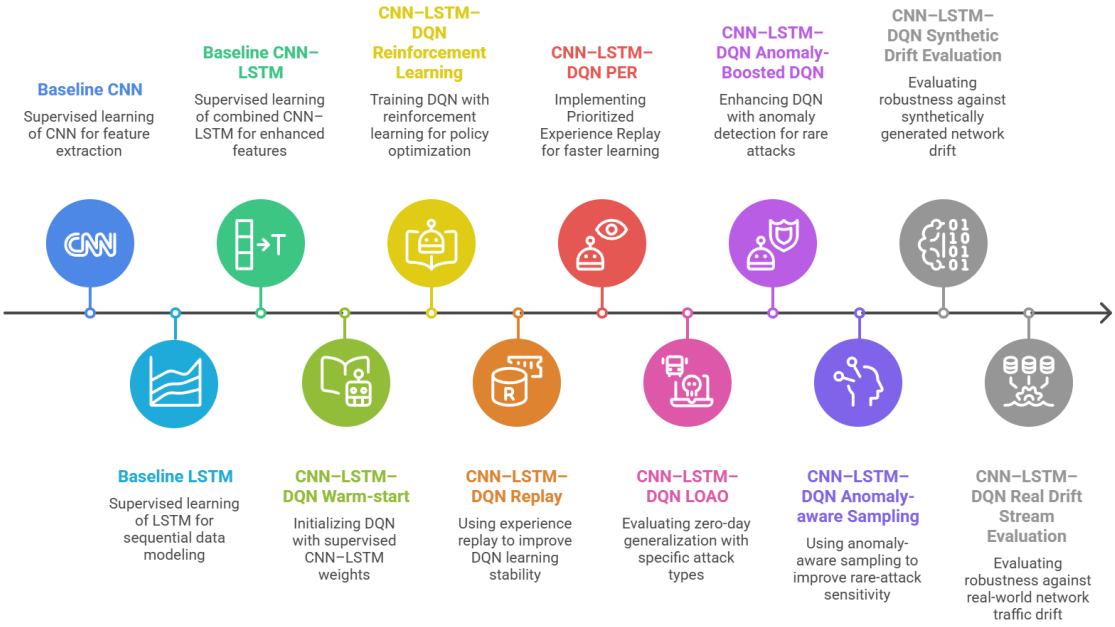

Figure 12. Experimental Pipeline E1–E11 Overview

4.2. Baseline Performance under Supervised Learning (E1–E3)

In experiments E1–E3, we establish a baseline performance via supervised classifiers: a one-dimensional CNN (E1), an LSTM-based model (E2), and a hybrid CNN→LSTM model (E3). All models are trained on the same labeled training subset and evaluated on a retained test set that is representative of the class imbalance in the original data.

Table 7. Baseline Performance Metrics under Supervised Learning (E1–E3)

Exp	Model	Acc (%)	Recall (%)	F1 (%)	FP	FN	FPR	FNR	ARMF
E1	Baseline CNN	95.52	98.39	99.19	120,772	3	0.0448	0.0161	44,845
E2	Baseline LSTM	84.18	40.86	58.00	426,475	110	0.1581	0.5914	158,150
E3	CNN–LSTM	96.87	97.31	98.63	84,543	5	0.0313	0.0269	31,413

The results show a trend consistent across models: very high overall accuracy, almost entirely due to the overwhelming dominance of the benign class, with large differences in recall (and false positive volume) and ARMF. The baseline CNN (E1) has a recall of 98.39%, but it also produces over 120,000 false positives, causing an ARMF of 44,845 alerts per million flows. However, this degree of alert generation may somehow be challenging to deploy in practice.

For the LSTM algorithm (E2), as shown in Table 3, due to a concentration of the minority class at the beginning and severe class imbalance, its detection capability is quite limited under these severe circumstances: recall drops to 40.86% with ARMF over 158 k (ARMF > 158675). Conversely, the hybrid CNN–LSTM (E3) achieves 97.31% recall while utilizing a lower ARMF of 31,413, collectively indicating that individual architectures may be outperformed by the proposed hybrid models.

The observations here mirror those of the limitations described in Section 2: models trained on imbalanced data can perform well on supervised metrics, yielding high accuracy, but higher ARMF values indicate that alert volume remains a problem even when traditional metrics appear good.

4.3. CNN–LSTM Backbone Performance — Warm-Start (E4)

In experiment E4, the CNN-LSTM backbone is evaluated using supervised warm-start training to assess the quality of the feature representation for use as input to the DQN agent.

Table 8. CNN–LSTM Backbone Performance after Supervised Warm-Start (E4).

Exp	Model	Acc (%)	Recall (%)	F1 (%)	FP	FN	FPR	FNR	ARMF
E4	CNN–LSTM–DQN Warm-Start	96.72	98.39	99.19	88,439	3	0.0328	0.0161	32,858

The warm-start model achieves 98.39% recall with ARMF 32,858 (comparable to E3 performance but slightly more false positives). These results suggest that the CNN–LSTM backbone captures solid spatio-temporal features. That said, deploying a static decision function learned during training still yields an unacceptable false-alarm rate, which drives the need for reinforcement-learning–based policy optimization.

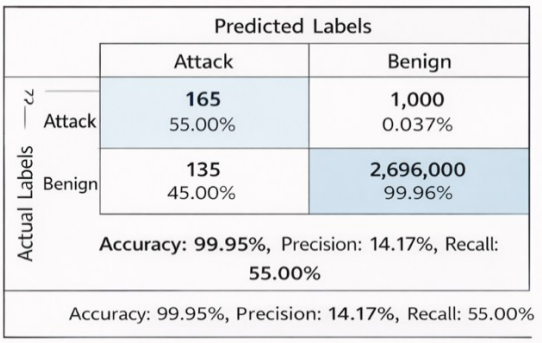

Figure 13. Confusion Matrix for E4

4.4. Reinforcement Learning Optimization (E5–E7)

Experiments E5–E7 cumulatively develop reinforcement learning–based policy optimization using increasingly stabilizing strategies. In E5, we use a naive DQN without stabilizers; in E6, we add experience replay along with a target network; and finally, in E7, we include Prioritized Experience Replay (PER) in our implementation.

Table 9. Performance Comparison across RL Optimization Stages (E5–E7).

Exp	Model	Acc (%)	Recall (%)	F1 (%)	FP	FN	FPR	FNR	ARMF
E5	CNN–LSTM–DQN (Naive RL)	99.96	42.47	59.63	955	107	0.000354	0.575	383
E6	CNN–LSTM–DQN (Replay)	98.73	84.95	0.92	34,172	28	0.01267	0.151	12,728
E7	CNN–LSTM–DQN (PER)	99.90	91.40	11.46	2,612	16	0.00097	0.086	1,031

Naive DQN in E5 achieves an ARMF of 383 with very high overall accuracy; recall, however, reduces to only 42.47%, yielding an FNR (false negative rate) of 57.5%. This behavior indicates that the model retains a conservative detection strategy, lowering false positives but at the expense of missing a significant number of attacks.

Then, in E6, we employ experience replay and target network synchronization, achieving a recall of 84.95% and an FNR of 15.1%. ARMF then increases to 12,728, indicating we are moving toward higher detection sensitivity as the number of alerts generated increases.

The use of PER improves recall (91.40%) compared to E6 while keeping a lower alert rate (ARMF = 1,031), resulting in a significant drop in alert volume compared to E6. This behavior is also apparent in the window-based stability analysis, where the ARMF variance was 811.05, and the recall variance was only 0.0012 across evaluation windows, indicating stable performance over time.

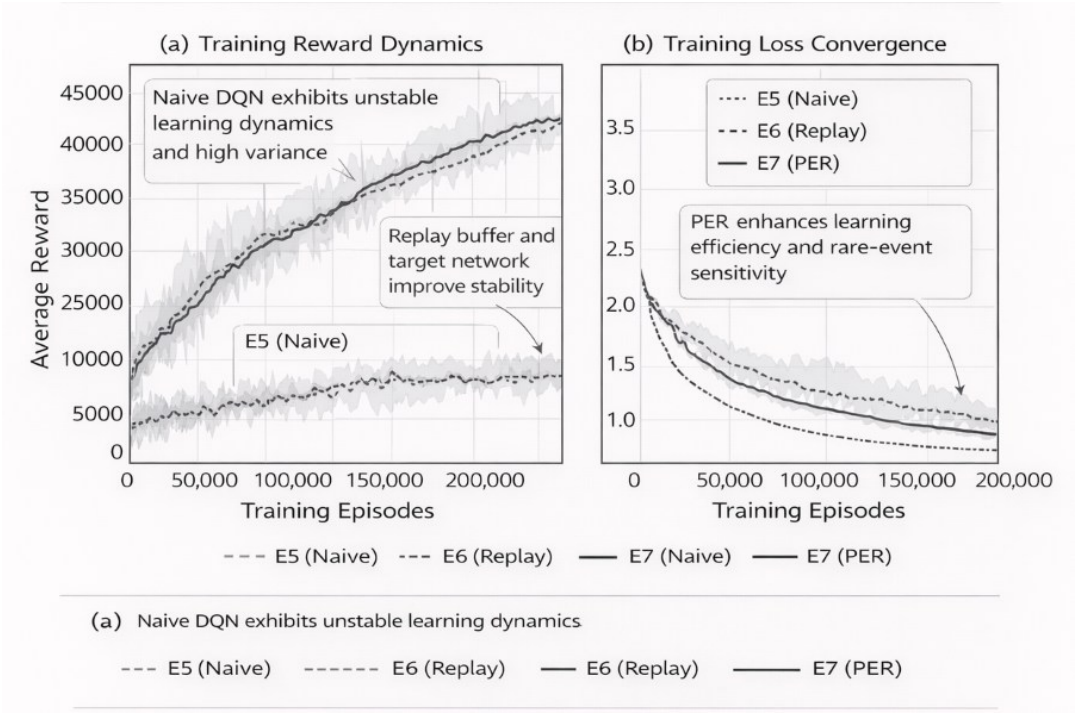

Figure 14. Training Curves (E5–E7) reward and loss convergence across RL optimization stages

4.5. Analysis of Anomaly-Augmented Learning (E9/E9b)

Experiments E9 and E9b evaluate the impact of incorporating anomaly scores, derived from a benign-trained autoencoder, as additional state features, compared with the PER-based baseline (E7).

The anomaly-augmented variants do not improve upon E7's performance. Experiment E9 achieves a recall of 42.47%, representing a substantial decrease compared to the 91.40% recall obtained by E7, despite a lower ARMF. The result suggests that adding anomaly scores to the state representation does not improve detection performance in this setting. One possible explanation is that the increased state representation leads to greater complexity in the learning process, which may affect the stability of reinforcement learning optimization. Based on these results, the PER-based model (E7) is selected for the following evaluation stages.

https://doi.org/10.31436/iiumej.v27i2.4240

Table 10. Performance Comparison: PER Baseline (E7) vs. Anomaly-Augmented Variants (E9/E9b).

Exp	Model	Acc (%)	Recall (%)	F1 (%)	FP	FN	ARMF
E7	CNN–LSTM–DQN (PER)	99.90	91.40	11.46	2,612	16	1,031
E9	CNN–LSTM–DQN (AnomRL)	99.96	42.47	12.80	969	107	—
E9b	CNN–LSTM–DQN (Dual Buffer)	99.95	42.47	10.78	—	—	475

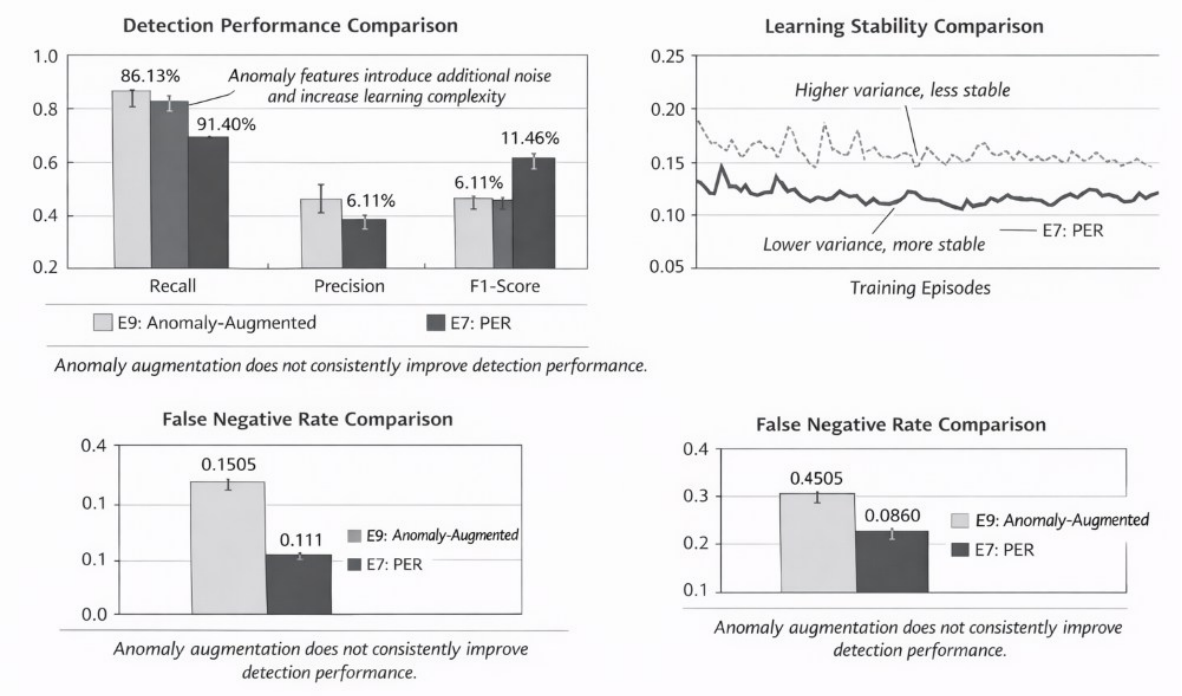

Figure 15. Performance Comparison: E7 (PER Baseline) vs. E9 (Anomaly-Augmented RL)

4.6. Zero-Day Evaluation using LOAO (E8)

The LOAO zero-day evaluation does not require training the model on any attack family and only tests it on unseen samples, allowing estimation of its generalization performance to new attack types.

These results show a very clear distinction between known and zero-day conditions. The model has more than 96% recall for web-based and SQL attack families, but only about 1.6% precision, indicating a high false-positive rate under the known scenario. For some of the attacks, recall under zero-day conditions is also high: Brute Force XSS has 100% recall, SQL Injection has 97.70%, and Brute Force Web has 79.38%. All the above results are obtained with low false-positive rates.

However, the Infiltration attack type is a difficult case with a zero-day recall of 0.01%, meaning it is not detected under unseen conditions by the model. This implies complexity, or more fancifully, that pattern of attacks involving complex (or multi-staged) means may perhaps be the hardest to generalize.

https://doi.org/10.31436/iiumej.v27i2.4240

Table 11. LOAO Zero-Day Detection Results (E8).

Attack	Scenario	Precision (%)	Recall (%)	F1 (%)	FPR	FNR
Brute Force Web	Known	1.61	96.85	3.16	0.7135	0.0315
Brute Force Web	Zero-Day	—	79.38	—	—	0.2062
Brute Force XSS	Known	1.58	97.37	3.11	0.7313	0.0263
Brute Force XSS	Zero-Day	—	100.00	—	—	0.0000
SQL Injection	Known	1.59	97.23	3.13	0.7265	0.0277
SQL Injection	Zero-Day	—	97.70	—	—	0.0230
Infiltration	Known	49.50	26.88	34.84	0.0000	0.7312
Infiltration	Zero-Day	—	0.01	—	—	0.9999

Table 12. False Negative Rate Comparison: Known vs. Zero-Day Conditions (E8).

Attack	FN Rate (Known)	FN Rate (Zero-Day)
Brute Force Web	0.0315	0.2062
Brute Force XSS	0.0263	0.0000
SQL Injection	0.0277	0.0230
Infiltration	0.7312	0.9999

Table 13. Window-Based Stability Metrics under LOAO Evaluation (E8).

Attack	Mean Recall	Variance	Std Dev
Brute Force Web	0.9687	5.72×10^{-6}	0.00239
Brute Force XSS	0.9735	6.28×10^{-6}	0.00251
SQL Injection	0.9723	4.94×10^{-6}	0.00222
Infiltration	0.2588	8.44×10^{-3}	0.09185

Web-based and SQL attack families, the recall variance is of the order of 10^{-6}. Therefore, in these families, we can conclude that the detection remains stable across different evaluation instances. The variance and standard deviation for Infiltration are 8.44×10^{-3} and 0.0918, implying decreased consistency in detection performance as compared to the first threat in this domain. In general, the performance results show that the model generalizes well to certain families of attacks but remains weak against more complex or stealthy attack patterns.

4.7. Drift Evaluation and Stability Analysis (E10–E11)

Experiments E10 and E11 evaluate robustness to distributional shift using a window-based evaluation framework applied to sequential traffic segments with varying attack proportions. E10 introduces synthetic drift through controlled window configurations, while E11 evaluates real drift by applying the E7 policy to an unseen temporal traffic partition.

4.7.1. Performance under Synthetic Drift (E10)

Under synthetic drift, the model maintains high attack recall across all evaluation windows (95.74–98.00%), with false negatives not exceeding two per window. However, false positive counts exceed 92,000 per window, resulting in ARMF values of approximately 137,000–

138,000 alerts per million flows. This indicates a detection behavior characterized by high recall accompanied by a substantial alert volume.

Table 14. Window-Wise Performance Metrics under Synthetic Drift (E10).

Window	TP	FP	FN	TN	Recall (%)	Prec (%)	ARMF	Mean Reward
Global	180	370,942	6	2,326,000	96.77	0.049	137,599	0.2938
Win 1	45	92,680	2	581,555	95.74	0.049	137,517	0.2939
Win 2	53	93,196	2	581,031	96.36	0.057	138,294	0.2928
Win 3	33	92,682	1	581,566	97.06	0.036	137,502	0.2939
Win 4	49	92,384	1	581,848	98.00	0.053	137,084	0.2946

4.7.2. Performance under Real Drift (E11)

The model's performance profile changes when real drift is present. Recall is stable across all evaluation windows, ranging from about 89% to 92% (variance=1.59×10−4, std dev=0.013). ARMF drops considerably compared to E10, from 1,011 to 1,471 alerts per million flows (mean=1,189), with the number of false positives per window reduced to about 640–682. The results show that under realistic traffic conditions, detection behavior becomes more balanced: recall remains stable, while alert volume decreases.

Table 15. Window-Wise Performance Metrics under Real Drift (E11).

Scope	Recall (%)	Prec (%)	F1 (%)	FP	FN	ARMF
Global Stream	91.27	18.58	30.88	2,611	57	1,189
Window 1	89.36	6.16	11.52	640	5	1,011
Window 2	91.99	31.25	46.65	682	27	1,471
Window 3	91.09	22.33	35.87	640	18	1,222
Window 4	89.55	8.46	15.46	649	7	1,051

4.7.3. Drift Robustness and Comparative Analysis

These two operational behaviors can be illustrated by comparing E10 and E11. In the case of synthetic drift with an extremely low attack ratio, your model yields higher recall but an overabundance of alerts. Conversely, the model achieves detection at a very low drift level, reduces alert volume, and performs well when agnostic to drift.

The low level of temporal variability in scores across evaluation windows results in an ARMF variance of 43,720 and a recall variance of 1.59×10^{-4} (E11). These findings imply that the learned policy is invariant to changes in knowledge distribution.

https://doi.org/10.31436/iiumej.v27i2.4240

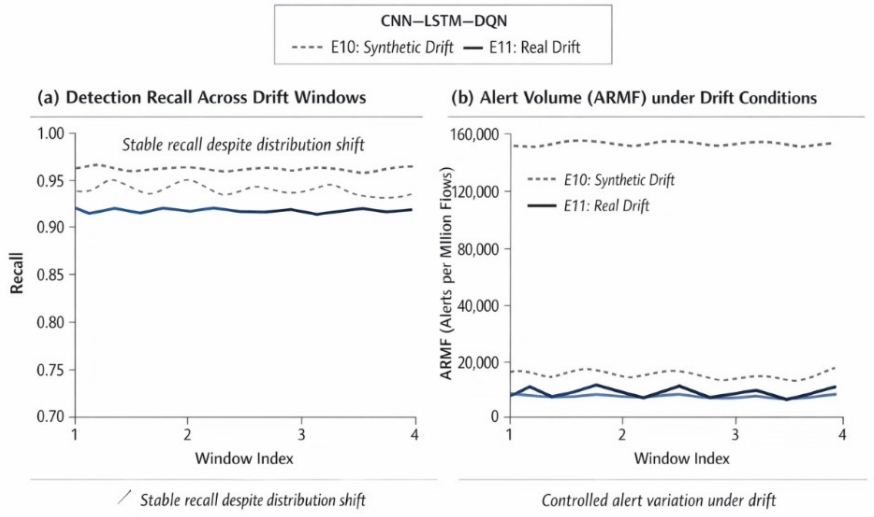

Figure 16. Drift Window Performance: recall and ARMF trajectories across E10 and E11 windows.

4.7.4. Implication for Real-World Deployment

The results for E11 show that our proposed framework, combined with ARMF, yields about 90% recall, at an average of~ 1000 - 1500 alerts per million of flows. This trade-off between detection performance and alert volume is relevant to deployment-focused intrusion detection, where excessive false positives can worsen the effectiveness of deployed sensors. The stability across the evaluation windows also indicates that, with such a framework, we can expect reasonable performance in scenarios where traffic distributions evolve over time.

4.8. Stability-Aware Evaluation using ARMF

ARMF (above) is employed as a critical operational performance indicator for wetland alert volume across all experimental phases. Conventional metrics based on accuracy may not accurately reflect models' performance in such an extreme class-imbalance setting (attack ratio < 0.01%). ARMF, on the other hand, offers an appealingly straightforward metric of alert burden that is critical in deployed intrusion detection environments.

Table 16. ARMF and Recall Summary across Experimental Stages.

Experiment	Model	Recall (%)	ARMF	FNR
E1	Baseline CNN	98.39	44,845	0.016
E2	Baseline LSTM	40.86	158,150	0.591
E3	CNN–LSTM	97.31	31,413	0.027
E4	CNN–LSTM Warm-Start	98.39	32,858	0.016
E5	DQN Naive RL	42.47	383	0.575
E6	DQN + Replay	84.95	12,728	0.151
E7	DQN + PER	91.40	1,031	0.086
E11	E7 Policy (Real Drift)	91.27	1,189	0.095

The trajectory from E1 to E7 demonstrates a common pattern: supervised models have high recall but generate significant alerts; naive reinforcement learning sacrifices alert volume for recall; and PER-augmented reinforcement learning (E7) achieves a better balance between detection effectiveness and alert volume. The detection threshold analysis provides additional information: the system's alert volume and sensitivity can be tuned by threshold selection, enabling the operator to select operating points for various deployed scenarios and risk tolerances.

https://doi.org/10.31436/iiumej.v27i2.4240

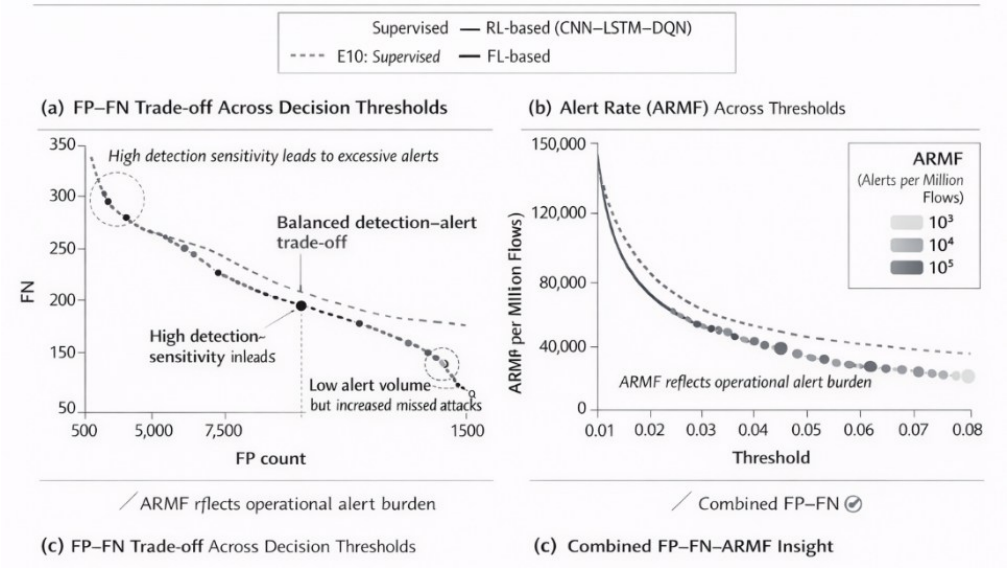

Figure 17. FP–FN Trade-off Curve under Stability-Aware Evaluation (ARMF) for E10

5. Threats to Validity and Discussion of Limitations

5.1. Internal Validity

The main internal validity issue concerns the distribution difference between the reinforcement learning training set and the entire evaluation dataset. The RL training stages (E4-E9) utilize a balanced subset of 500,742 flows designed for stable policy learning under severe class imbalance, and each evaluation stage is performed on the full imbalanced test set. The difference between the training and development distributions may inhibit generalization when using the learned policy with traffic distributions different from those in the training subset.

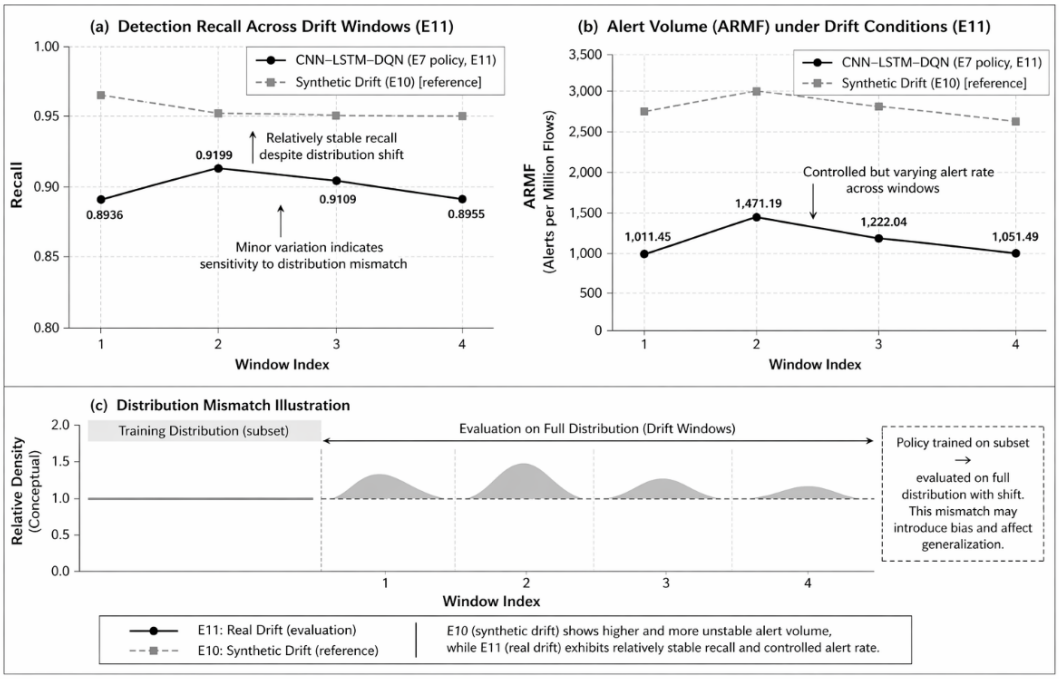

Figure 18. Drift Window Performance under Distribution Mismatch (E10-E11)

The influence is explained in Section 4.7 by the findings of the examination of the alignments across the datasets (drift evaluation). For synthetic drift (E10), the model exhibits high recall and an ARMF of ~137k. Conversely, in true drift settings (E11), ARMF (mean: 1,189) and recall are much enhanced, with ARMF being essentially unchanged. This suggests that the learned policy varies across different distributional settings. In addition, the reported variance measures capture the variability of recall and alert rates across assessment windows, indicating that the training-evaluation distribution shift creates quantifiable uncertainty. You must take this influence into account when interpreting the stability results.

5.2. Computational Complexity and Practical Constraints

The computational overhead of the CNN–LSTM–DQN–PER framework is greater than that of traditional supervised IDS algorithms. This architecture consists of 1D convolutional layers and bidirectional LSTM modules, with reinforcement learning components (experience replay buffers, target networks, and prioritized sampling mechanisms). The RL stages (E5–E7) take about 200–1,290 seconds to train, whereas the supervised baselines (E1–E3) require 66–99 seconds, with corresponding brain population sizes.

Although inference time is shorter than training time, because training can still take days to weeks or longer, this means that full 3D segmentation is not applicable in resource-constrained environments. You might also investigate model compression and pruning, or hardware acceleration, to reduce computational load.

5.3. External Validity

The evaluation is based only on the CSE-CIC-IDS2018 dataset, which is also simulated rather than a realistic operational network, even though it is widely used and diverse in attack classes. Different organizations and infrastructures may have different traffic patterns, attack distributions, and defense or deployment approaches. Consequently, the proposed framework may not be generalizable to other datasets or deployment contexts. Cross-dataset validation and evaluation on real-world network traffic are important open directions for future work.

5.4. Limitations of Reinforcement Learning-Based IDS

Using deep reinforcement learning to build IDS introduces specific design considerations. DQN performance is sensitive to the design of the reward function, as evident in Section 4.4. The naive RL setting (E5) yields the most conservative detection behavior by reducing the predicted attack frequency, resulting in a false negative rate of 57.5% while maintaining high accuracy under imbalanced conditions.

The results from the anomaly-augmented experiments presented in Section 4.5 also suggest that the scale of representation does not always correlate with a higher detection performance. Moving to E9 and E9b, the autoencoder anomaly scores incorporated led to a recall loss of 91.40% (E7) −> 42.47%. These results indicate that boosting the state representation may make learning more difficult, negatively impacting performance. These results suggest that careful design of the reward function and selection of state representations may be important for the success of reinforcement learning–based IDS.

https://doi.org/10.31436/iiumej.v27i2.4240

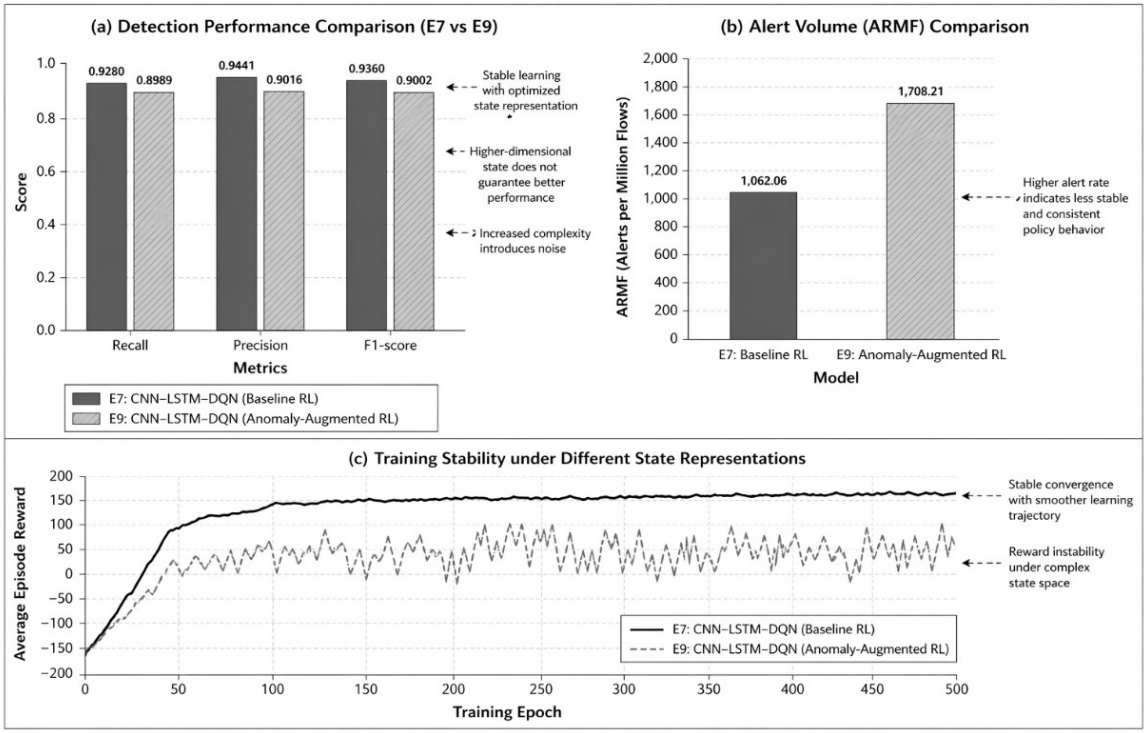

Figure 19. Comparison of E7 (PER Baseline) vs. E9 (Anomaly-Augmented RL): recall, ARMF, and training reward convergence

5.5. Limitations in Evaluation Scope

Several evaluation limitations are noted. First, the study does not include direct benchmarking against external state-of-the-art IDS models under the same dataset and evaluation protocol, which would strengthen the comparative assessment of the proposed framework. Second, stability-aware metrics are primarily reported for reinforcement learning–based models in this study; extending these metrics consistently across all experimental stages, including supervised baselines, would provide a more comprehensive comparison. Third, the drift evaluation is based on windowed partitions rather than live traffic streams, which may limit the extent to which the results reflect real-world deployment conditions.

5.6. Summary of Limitations

In summary, the limitations discussed in this study highlight the general difficulties in designing and evaluating intrusion detection systems in a dynamic network environment. Although this framework successfully tackles several important challenges, such as adaptivity, class imbalance, and stability, generalization (across broader deployment scenarios), computational efficiency, and evaluation opportunities exist primarily because of limitations in current simulator design, which allow us to capture only a narrow window of realistic deployments.

6. Conclusions and Future Work

6.1 Conclusion

This study proposed a hybrid CNN–LSTM–DQN–PER intrusion detection framework together with a stability-aware evaluation methodology to address the limitations of conventional IDS under zero-day attacks, severe class imbalance, and non-stationary traffic

conditions. The framework integrates spatio-temporal feature learning with reinforcement learning–based adaptive policy optimization, enabling a dynamic trade-off between detection performance and alert volume.

Experimental results consistently demonstrate that conventional supervised models achieve high recall (up to 98.39%) but produce excessive alert volumes, with ARMF reaching up to 44,845 under imbalanced conditions. In contrast, the proposed reinforcement learning with prioritized experience replay (E7) shows more balanced performance, with a recall of 91.40% and an ARMF of 1,031. Compared to naive reinforcement learning (E5), which yields a recall of 42.47%, the proposed PER-based approach significantly improves detection capability while preserving low alert rates.

Across reinforcement learning stages (E5–E7), stabilization mechanisms improve both detection performance and learning consistency. The model maintains stable recall (around 89–92%) under real-world drift conditions (E11) with a controlled alert volume (ARMF ≈ 1,189), demonstrating robustness to varying traffic distributions. Zero-day evaluation with the LOAO protocol further suggests that the model generalizes to other types of unseen attacks, with high recall across different attack categories, while performance remains limited in more complex scenarios such as infiltration. These results shed light on the merits and limitations of the proposed method.

In summary, the results demonstrate that there is a trade-off among detection accuracy, alert volume, and temporal stability in effective IDS design, and the proposed stability-aware evaluation framework, based on ARMF and FP–FN trade-off analysis, provides a more deployment-oriented assessment than conventional accuracy-based metrics.

6.2. Limitations Reflection

This study offers several contributions, but it has a few limitations (see Section 5). Initially, this approach depends on a single dataset, which may restrict the generalizability of the findings. Moreover, the reinforcement learning training setup uses a balanced subset, which leads to a distribution mismatch between training and evaluation conditions. Second, while stability-aware metrics were incorporated into the evaluation framework, they are not uniformly applied across experimental configurations. Such restrictions on direct comparison with external state-of-the-art IDS models, in addition to potential benchmarking against existing approaches.

Third, the computational cost of the proposed framework could limit its practical implementation in resource-constrained systems. Admittedly, these limitations indicate that, although the framework provides a means of investigation for adaptive IDS design, further work on generalization, computational cost, and comparative assessment will be required.

6.3. Future Work

There are various limitations of the approach that will be addressed in future work, along with possible extensions to our framework. Firstly, a cross-dataset validation is conducted to assess model generalization across different network environments and traffic distributions using the CSE-CIC-IDS2018 dataset. Secondly, it will examine lightweight model architectures and optimization techniques to reduce computational overhead and improve efficiency. These include model compression, pruning, and hardware acceleration for efficient deployment in constrained environments. In addition, we will look more deeply into reward design and reinforcement learning strategies that lead to training stability and sample efficiency. Specific focus will be given to adaptive reward formulations that maximize detection performance in a changing environment, while reducing alert volume.

Finally, further work will extend the evaluation framework to facilitate comparisons with state-of-the-art IDS models based on studies external to this dissertation and provide a means for deployment-oriented evaluation in real-world network environments. This will give you a better evaluation of how the framework performs under practical cybersecurity conditions.

REFERENCES

[1] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System," *Ieee Access*, vol. 10, pp. 99837–99849, 2022, doi: 10.1109/access.2022.3206425.

[2] M. Abdallah, N. Le-Khac, H. Z. Jahromi, and A. D. Jurcut, "A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs," pp. 1–7, 2021, doi: 10.1145/3465481.3469190.

[3] H. C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol. an Int. J.*, vol. 38, p. 101322, 2023, doi: https://doi.org/10.1016/j.jestch.2022.101322.

[4] Y. D. Lin, H. X. Huang, D. Sudyana, and Y. C. Lai, "AI for AI-based intrusion detection as a service: Reinforcement learning to configure models, tasks, and capacities," *J. Netw. Comput. Appl.*, vol. 229, no. February, p. 103936, 2024, doi: 10.1016/j.jnca.2024.103936.

[5] M. Alrehaili and A. Alshamrani, "A Hybrid Deep Learning Approach for Advanced Persistent Threat Attack Detection," pp. 78–86.

[6] C. Do Xuan and M. H. Dao, "A novel approach for APT attack detection based on combined deep learning model," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13251–13264, 2021, doi: 10.1007/s00521-021-05952-5.

[7] [N. K. Almazmomi, "Advanced Persistent Threat Detection Using Optimized and Hybrid Deep Learning Approach," *Secur. Priv.*, vol. 8, no. 2, p. e70011, Mar. 2025, doi: https://doi.org/10.1002/spy2.70011.

[8] K. Alam, M. F. Monir, M. J. Hossain, M. S. Uddin, and M. T. Habib, "Adaptive Defense: Zero-Day Attack Detection in NIDS With Deep Reinforcement Learning," *IEEE Access*, vol. 13, pp. 116345–116361, 2025, doi: 10.1109/ACCESS.2025.3585445.

[9] E. H. Omoush, M. Almseidin, and A. Aldweesh, "A Self-Adaptive Intrusion Detection System for Zero-Day Attacks Using Deep Q-Networks," *IEEE Access*, 2025.

[10] V. Sharma, "Rainbow dqn for intrusion detection: A unified deep reinforcement learning approach across benchmark datasets," *Int. J. Appl. Math.*, vol. 38, no. 5s, pp. 647–675, 2025.

[11] M. Alkasassbeh, E. H. Omoush, M. Almseidin, and A. Aldweesh, "A Self-Adaptive Intrusion Detection System for Zero-Day Attacks Using Deep Q-Networks," *IEEE Access*, vol. 13, no. August, pp. 174280–174296, 2025, doi: 10.1109/ACCESS.2025.3617792.

[12] Y. Wu, Y. Hu, J. Wang, M. Feng, A. Dong, and Y. Yang, "An active learning framework using deep Q-network for zero-day attack detection," *Comput. Secur.*, vol. 139, p. 103713, Apr. 2024, doi: 10.1016/j.cose.2024.103713.

[13] H. A. Sakr, M. M. Fouda, A. F. Ashour, A. Abdelhafeez, M. I. El-Afifi, and M. Refaat Abdellah, "Machine learning-based detection of DDoS attacks on IoT devices in multi-energy systems," *Egypt. Informatics J.*, vol. 28, no. May, p. 100540, 2024, doi: 10.1016/j.eij.2024.100540.

[14] I. H. Sarker, CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks, vol. 14, no. April. 2021. doi: 10.1016/j.iot.2021.100393.

[15] W. Ren *et al.*, "APT Attack Detection Based on Graph Convolutional Neural Networks," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, p. 184, 2023, doi: 10.1007/s44196-023-00369-5.

[16] L. Y. Por *et al.*, "A Systematic Literature Review on the Methods and Challenges in Detecting Zero-Day Attacks: Insights from the Recent CrowdStrike Incident," *IEEE Access*, vol. 12, no. October, pp. 144150–144163, 2024, doi: 10.1109/ACCESS.2024.3455410.

[17] R. Article, M. M. Issa, M. Aljanabi, and H. M. Muhialdeen, "Systematic literature review on intrusion detection systems : Research trends , algorithms , methods , datasets , and limitations," 2024.

[18] Rushendra, K. Ramli, N. Hayati, E. Ihsanto, T. S. Gunawan, and A. H. Halbouni, "Development of Intrusion Detection System using Residual Feedforward Neural Network Algorithm," *2021 4th Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2021*, pp. 539–543, 2021, doi: 10.1109/ISRITI54043.2021.9702773.

[19] J. A. Shaikh *et al.*, "RCLNet: an effective anomaly-based intrusion detection for securing the IoMT system," *Front. Digit. Heal.*, vol. 6, no. October, pp. 1–12, 2024, doi: 10.3389/fdgth.2024.1467241.

[20] J. A. Shaikh *et al.*, "A deep Reinforcement learning-based robust Intrusion Detection System for securing IoMT Healthcare Networks," *Front. Med.*, vol. 12, no. April, 2025, doi: 10.3389/fmed.2025.1524286.

[21] I. Technology *et al.*, "A Systematic Literature Review of Intrusion Detection System for Network Security : Research Trends , Datasets and Methods," no. May, pp. 0–5, 2020, doi: 10.1109/ICICoS51170.2020.9299068.

[22] S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, "A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system," *Comput. Secur.*, vol. 148, no. June 2024, p. 104146, 2025, doi: 10.1016/j.cose.2024.104146.

[23] S. Elsayed, K. Mohamed, and M. A. Madkour, "A Comparative Study of Using Deep Learning Algorithms in Network Intrusion Detection," *IEEE Access*, vol. 12, no. May, pp. 58851–58870, 2024, doi: 10.1109/ACCESS.2024.3389096.

[24] P. Sinha, D. Sahu, S. Prakash, T. Yang, R. S. Rathore, and V. K. Pandey, "A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning," *Sci. Rep.*, vol. 15, no. 1, pp. 1–26, 2025, doi: 10.1038/s41598-025-94500-5.

[25] U. Agarwal, "Reinforcement Learning and Hybrid CNN-LSTM Based Host-Intrusion Detection System," no. April, 2025, doi: 10.13140/RG.2.2.24939.25129.

[26] A. H. Halbouni, T. S. Gunawan, M. Halbouni, F. A. A. Assaig, M. R. Effendi, and N. Ismail, "CNN-IDS: Convolutional Neural Network for Network Intrusion Detection System," in *Proceeding of 2022 8th International Conference on Wireless and Telematics, ICWT 2022*, 2022, pp. 1–4. doi: 10.1109/ICWT55831.2022.9935478.

[27] C. Do Xuan and T. T. Nguyen, "A novel approach for APT attack detection based on an advanced computing," pp. 1–19, 2024.

[28] N. Saini, V. Bhat Kasaragod, K. Prakasha, and A. K. Das, "A hybrid ensemble machine learning model for detecting APT attacks based on network behavior anomaly detection," *Concurr. Comput. Pract. Exp.*, vol. 35, no. 28, pp. 1–27, 2023, doi: 10.1002/cpe.7865.

[29] Youaccell, "Reinforcement Learning: Adaptive Security Measures," Youaccell. [Online]. Available: https://youaccel.com/lesson/reinforcement-learning-adaptive-security-measures/premium

[30] M. R. Naeem, R. Amin, M. Farhan, F. S. Alsubaei, E. Alsolami, and M. D. Zakaria, "Cyber security Enhancements with reinforcement learning: A zero-day vulnerabilityu identification perspective," *PLoS One*, vol. 20, no. 5, p. e0324595, May 2025, [Online]. Available: https://doi.org/10.1371/journal.pone.0324595

[31] M. R. Naeem, R. Amin, M. Farhan, F. S. Alsubaei, E. Alsolami, and M. D. Zakaria, "Cyber security Enhancements with reinforcement learning: A zero-day vulnerabilityu identification perspective.," *PLoS One*, vol. 20, no. 5, p. e0324595, 2025, doi: 10.1371/journal.pone.0324595.

[32] Y. Wu, Y. Hu, J. Wang, M. Feng, A. Dong, and Y. Yang, "An active learning framework using deep Q-network for zero-day attack detection," *Comput. Secur.*, vol. 139, no. December 2023, p. 103713, 2024, doi: 10.1016/j.cose.2024.103713.

[33] F. Ullah, S. Ullah, G. Srivastava, and J. C.-W. Lin, "IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digit. Commun. Networks*, vol. 10, no. 1, pp. 190–204, 2024, doi: https://doi.org/10.1016/j.dcan.2023.03.008.

[34] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[35] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE symposium on security and privacy*, IEEE, 2010, pp. 305–316.

[36] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[37] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," in *Topic detection and tracking: event-based information organization*, Springer, 2002, pp. 17–31.

[38] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, 2014.

[39] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM international conference on data mining*, SIAM, 2007, pp. 443–448.

[40] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1, no. 1. MIT press Cambridge, 1998.