International Journal on Robotics, Automation and Sciences

Reinforcement Learning Approach for Cybersecurity Threat Detection

Ennbaraaj A/L G. Sundharajan, Md Shohel Sayeed* and Golam Md Mohiuddin

***Abstract* – Traditional intrusion detection systems (IDS) struggle to detect evolving cyber threats due to their reliance on static signatures and fixed decision boundaries. Existing machine learning-based approaches partially address this limitation but often fail to generalize to zero-day attacks and lack adaptability in dynamic network environments. To address these challenges, this paper proposes a reinforcement learning-based intrusion detection system (RL-IDS) that model's detection as a sequential decision-making problem using flow-level telemetry. The framework is implemented on the CIC-IDS2017 dataset with an isolated zero-day partition within a custom OpenAI Gym environment, incorporating asymmetric reward design, curriculum learning, entropy annealing, and early stopping to train Q-learning, Deep Q-Network (DQN), and Proximal Policy Optimization (PPO) agents. Experimental results show that the PPO-based RL-IDS achieves an F_1-score of 0.857 with less than 4% false positives on known attacks, outperforming both DQN and a 400-tree Random Forest baseline. More importantly, it detects 27.7% of previously unseen zero-day attacks (Heartbleed and Infiltration), where the Random Forest fails completely. The system also processes over 290,000 flows per second, demonstrating real-time feasibility. These results demonstrate that reinforcement learning enables a practical balance between accuracy, adaptability, and efficiency, making it a promising solution for next-generation intrusion detection systems.**

Keywords*— *Cybersecurity, Intrusion Detection Systems (IDS), Reinforcement Learning, Q-learning, Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Zero-day Attacks, Curriculum Learning, OpenAI Gym.

I. INTRODUCTION

Modern computer networks are experiencing unprecedented growth in traffic volume, velocity, and diversity, driven by cloud computing, mobile connectivity, and the rapid expansion of Internet of Things (IoT) devices. While this transformation enables scalable and flexible services, it also significantly enlarges the attack surface, exposing networks to increasingly sophisticated and previously unseen cyber threats. Intrusion Detection Systems (IDS) remain a critical component of network defense. Traditional signature-based IDS are highly effective against known attacks but rely on predefined patterns, making them inherently incapable of detecting novel or evolving threats. Conversely, anomaly-based and machine learning-driven systems attempt to generalize beyond known signatures, yet they often suffer from high false-positive rates and limited adaptability, placing a substantial burden on security

*Corresponding Author email: shohel.sayeed@mmu.edu.my, ORCID: 0000-0002-0052-4870
Ennbaraaj A/L G. Sundharajan is with Faculty of information Science & Technology, Multimedia University, Melaka, Malaysia. (e-mail: ennbaraaj.sundharajan@student.mmu.edu.my)
Md Shohel Sayeed is with Faculty of information Science & Technology, Centre for Intelligent Cloud Computing, CoE for Advanced Cloud Multimedia University, Melaka, Malaysia. (e-mail: shohel.sayeed@mmu.edu.my)
Golam Md Mohiuddin is with Faculty of information Science & Technology, Multimedia University, Melaka, Malaysia. (e-mail: GOLAM.MD.MOHIUDDIN@student.mmu.edu.my)

International Journal on Robotics, Automation and Sciences (2026) 8, 2: 15-30
https://doi.org/10.33093/ijoras.2026.8.2.3
Manuscript received: 31 Dec 2025 | Revised: 25 Apr 2026 | Accepted: 22 May 2026 | Published: : 31 Jul 2026

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

analysts. Despite recent advances in machine learning and deep learning for intrusion detection, several key limitations persist, First, most existing approaches rely on static training data distributions, resulting in poor generalization to zero-day attacks that differ from previously observed patterns. Second, these systems lack adaptive learning mechanisms that can dynamically update their detection strategies based on evolving network conditions. Third, many studies do not adequately evaluate real-time deployment feasibility, particularly in terms of computational efficiency and inference latency under operational constraints. To address these gaps, this study proposes a reinforcement learning-based intrusion detection system (RL-IDS) that models intrusion detection as a sequential decision-making problem. Unlike static classifiers, the proposed system continuously learns from flow-level telemetry, using a reward-driven framework to balance detection accuracy against operational costs such as false positives and false negatives. The approach integrates value-based methods (Q-learning, Deep Q-Network) and policy-gradient methods (Proximal Policy Optimization) within a custom OpenAI Gym environment, incorporating asymmetric reward design and curriculum learning to enhance adaptability. The proposed framework is evaluated on the CIC-IDS2017 dataset under both closed-world (known attacks) and open-world (zero-day) conditions. Performance is assessed using standard metrics such as accuracy, precision, recall, and F_1-score, alongside real-time throughput measurements. The results demonstrate that the RL-based approach not only achieves competitive performance on known attacks but also provides measurable detection capability for previously unseen threats, while maintaining practical deployment efficiency.

The remainder of this paper is organized as follows: Section II reviews related work, Section III presents the methodology, Section IV discusses the experimental results, and Section V concludes the study with future research directions.

II. Related Works

To improve cyber-threat detection and classification via a Deep Reinforcement Learning (DRL) based IDS that can adapt to changing attack patterns. The authors implemented L_2 normalization and one-hot encoding for data preprocessing, Chi-square feature selection for dimensionality reduction on the 49 original dimensions to 39, and a Deep Q-Network framework with a binary classification and multiclass classification [1]. They evaluated their DRL model with a 50 000-sample subset of the NSL-KDD (the National Institute of Standards and Technology Knowledge Discovery and Data Mining) dataset, achieving an 86.41 % multiclass accuracy and a false positive rate of only 0.0276, better than both KNN (72.87 %) and Isolation Forest (84.13 %). The model's key advantage occurs with its learning and adaptation to the environment, and durable against adversarial conditions, while the major limitation was the high computations and lack of real-time or situ measurement Implementation. Ahsan et al.'s future work would include real time deployment, assessment on other datasets such as CIC-IDS2017, and performance Management concerning the operational efficiency of the DRL architecture [1].

The work of DQL-based IDS designed to escape the limitations of rule-based systems; they use a model-free reinforcement learning algorithm and a four- layer deep neural network to approximate Q-values over the 41 feature NSL-KDD dataset [2]. They normalized numerical features, one-hot encoded categorical features, and designed a reward system that positively reinforced accurate classifications and negatively reinforced incorrect classifications, while tuning the discount factor ($\gamma = 0.001$) to help mitigate the disadvantages of minority-class failure to predict. After training over 250 episodes, the final DQL agent generated an overall accuracy of 78 %, and an overall F_1 score of 94 % on DoS and Probe attacks; unfortunately, the agent struggled for both the very low R2L and U2R classes, which was demonstrated in related confusion matrices focusing on $\gamma = 0.001$ and, $\gamma = 0.9$. The primary positives of the DQL-IDS approach were the ability for the learning agent to continuously adapt to its environment, simplicity of tuning model hyperparameters, and better performance than Random Forest and SVM on lower resource systems; however, DQL-IDS is still limited by classification accuracy due to imbalanced classes and has not yet been evaluated in a real-world network context to assess scalability. The authors recommend operationalizing DQL- IDS in a cloud-based live environment, experimenting with different datasets (e.g., CIC-IDS2017), and extending the methodology to malware detection, and any monitoring of other anomalies and intrusion detection [2].

A holistic intrusion detection system integrating deep reinforcement learning with stochastic game theory and Nash equilibrium–based defense strategies was proposed [3]. They preprocessed the NSL-KDD dataset using a mixture of techniques (e.g., one-hot encoding, normalization and risk-based attack grouping) and trained a hybrid Deep Q-Network, built within the framework of a Markov decision process, using experience replay for the characterization of stability learning and modeled defender-attacker dynamics as a non-zero-sum game [3]. Their DRL-IDS produced detection rates of 100 % for DoS, 99 % for Probe, and 98 % for R2L attacks, and no detection rates for U2R attacks as their confusion matrix and comparative performance charts demonstrated [3]. The DRL-IDS demonstrated scalability, the potential for immediate adaptive decision-making, as well as superior detection rates for typical attack types, but the limitations included imbalanced class distribution and stability conditions that require quality labelled data [3]. Future work involves validating the model across additional benchmark datasets (BoT-IoT, IoT-23), exploring large-scale, diverse network topologies, and creating multi-agent systems to attempt to replicate attacker-defender dynamics [3].

An Adversarial Environment Reinforcement Learning (AE-RL) algorithm to enhance intrusion detection that dynamically concentrates on under-represented or difficult examples, and improved

computational efficiency with less-complex neural networks [4]. Their algorithm is to implement a Deep Q-Network variation, with the two agents (an environment agent and a classifier agent) each learning adversarial, using binary rewards to provide positive or negative reinforcement towards good/wrong classification. The authors look at AE-RL algorithm performance using NSL-KDD and AWID datasets, after normalizing and one-hot-encoding the features. The AE-RL model achieves an accuracy of over 80 % and a cumulative F_1 score over 0.79 under NSL-KDD data, running with less training and inference time than SVM, Random Forest and AdaBoost - showing real-time capability. The AE-RL model strength is in adapting to imbalanced data, optimizing false negatives regarding minority classes, and improved computational efficiency relative to other algorithms. As possible weaknesses, it sacrificed some accuracy for resource usage and has only been tested in non-live network environments. Further work will look at ways of effectiveness with additional datasets with different domains, prioritized experience replays for optimizing sampling, and potentially more sophisticated reward functions to be specific to types of intrusion [4].

An adaptive IDS for wireless sensor networks within IoT applications, combining RL-based feature selection with an ANN classifier to automatically detect and respond at scale to both known and unknown threats [5]. They preprocessed three different IoT datasets (BOT-IOT, CICIDS2019 and TON-IOT) by cleaning, normalizing and scaling the raw data, and using correlation analyses and principal component analysis to reduce dimensionality and capture 95 % of the variance across the data. A Q-learning agent dynamically ranks features and performed recursive feature elimination to produce a small but highly discriminative sub-set of features that was used by an ANN, which was assigned a dropout probability, L_2 regularization, and ReLU activations, that achieved over 99 % accuracy and a false positive rate of around 5 % on each dataset. The key strengths of their approach include the ability to adapt to changing attack patterns, and generalization across different IoT networks. The main weaknesses include computational complexity and the untested robustness to zero-day attacks on resource-limited devices. Here conclude that there are avenues for experimentation through future work for federated learning to preserve user privacy, through transfer learning to potentially detect novel threats not previously seen, and further fine-tuning of their system for deployment on low-power large-scale IoT installations [5].

The aim of the study adapts Deep Reinforcement Learning (DRL) methods to static, labelled data, where they developed a pseudo-environment that rewards the agent for correctly predicting an intrusion, so that the gap in supervised learning to reinforcement learning within the IDS research field is closed [6]. They converted network features to states and the intrusion labels to actions, applied to four DRL algorithms (DQN, DDQN, Policy Gradient and Actor–Critic), where the discount factor was tuned to prioritize immediate reward and to accommodate the non-linear sequence of the data. Their experiments were validated on two datasets, the NSL-KDD and AWID data, and their analysis found that DDQN performed the best, achieving the highest recall (which is particularly important for intrusion detection) and significantly better than more common machine learning methods, even in situations of class imbalance; however, it was less effective for the extremely rare types of attack. The strengths of the approach include the computational efficient processes to facilitate real-time deployment and flexibility enabling real-time online adaptation, whilst weaknesses include continued challenges with extreme imbalances and that the model on the NSL-KDD data focused on binary classification. The authors recommend that future research continue DRL methods to consider adversarial and multi-agent reinforcement learning, investigate more complex reward functions and other advanced DRL approaches to develop performance on imbalanced and multiclass intrusion types [6].

A decentralized multi-agent reinforcement learning (MARL) Intrusion Detection System (IDS) intended to solve scalability and real-time processing inside big-data networks by distributing learning across many agents where all agents share a target network that reduces the effect of non-stationarity [7]. They decentralize the learning process through the parallel partitioning of data and classification using Apache Spark Streaming to process the NSL-KDD dataset. The agents update a shared network together and classify network flows by applying MARL algorithms alongside cloud tools such as BigDL on AWS Databricks. When evaluated against the NSL-KDD dataset, the authors report an accuracy of 97.44%, good recall and F_1-score, and 0.97 AUC, demonstrating resilience to zero-day cases and real-time throughput. Strengths of this system lie in being horizontally scalable, fault tolerant, and seamlessly integrated with big-data pipelines, while major weaknesses stem from moderate computational cost (along with further costs from encrypted inter-agent communication), and single benchmark evaluation of a live network which is likely very heterogeneous. Future research will focus on finding optimal training efficiencies, validating performance in datasets with increased diversity and nuance, and evaluating performance in real-time streaming environments using AWS Kinesis or Apache Kafka to improve the environment's real-time processing capability [7].

The robustness of a Deep Q-Network–based IDS against adversarial perturbations by creating an adversarial environment RL (AE-RL) framework where both environment and classifier agents learn adversarial attacks such as FGSM and BIM, using the NSL-KDD dataset as a performance benchmark. They create a three-layer DQN with Huber loss and a selector agent focusing on hard-to-classify samples, ultimately finding that FGSM lowered binary detection accuracy from 84.81% to 66.87% and that BIM has a dramatic collapse in multi-class detection performance. Their strongest contribution is novel adversarial-training regime that allows the model to observe many of the same perturbations during the training phase to improve its robustness against some attacks. Their study is weakened by solely relying on static dataset experiments, not evaluating sequential

or streaming data, and consistently poor generalization on minority classes (U2R and R2L). The authors suggest to future researchers to consider adversarial retraining, try to simulate black-box attacks, add semantic constraints to their perturbation generation, and validating on more expansive up-to-date datasets to mitigate practical limitations [8].

The Deep SARSA-based RL model developed aimed at anomaly detection in network intrusion detection systems with the goal of improving both accuracy and adaptability of static ML methods which combines a five-layer DNN with SARSA, ε-greedy policy, replay memory and stabilizing target network [9]. Data from the NSL-KDD and UNSW-NB15 datasets were pre-processed using min-max normalization and one-hot encoding and training took place for 200 episodes with learning rate of 0.2 and discount factor of 0.001: assigning binary rewards to reinforce correct classifications. Their model is evaluated comparatively against the CNN, SOM and SVM baselines obtaining NSL-KDD F_1-score of 84.40% (accuracy 84.36%)- R2L performance above 0.65- and accuracy of 82.62% on UNSW-NB15. This demonstrates the value of robustness to novel attack patterns. Strengths: dynamic adaptation to threats, demonstrating increase in F_1-scores; but showed weak performance on rare classes (Worms, U2R) and affected by high computation costs. The authors mentioned focusing on experience replay to address rare class imbalance and comparative research against other deep RL algorithms (DQN, PPO) in future [9].

A Q-learning based IDS designed for clustering based wireless sensor networks with the intention to replace static rules with dynamic decisions that improve detection accuracy and flexibility in monitoring critical infrastructures [10]. They also integrated a Weighted Cluster Head Selection (WCHS) algorithm to aggregate data from sensor clusters to a central RL engine, modelled as a Markov decision process. Furthermore, they evaluated the entire system using KDDCup99 after normalizing the 41 features and encoding them. In simulation, their RL-IDS achieved almost perfect accuracy and detection rates, outperforming its AML-IDS predecessor while providing the highest reductions in false negatives and quantification of an ROC-area metrics. The key advantage of the RL-IDS was its ability to self-optimize in real-time, as well as its near-perfect identification of common and novel attack patterns. However, scalability remains a challenge due to high computational costs and limited experiment testing on just 20 sensor nodes. The authors outline recommendations for future work to scale this system over larger and varied types of network topologies as well as examining training using GPU acceleration to retain performance with large deployments [10].

An IDS based on adversarial reinforcement learning devised specifically for resource-constrained 6LoWPAN networks, designed to maintain high detection accuracy, despite class imbalance and concept drift [11]. They studied a variety of 6LoWPAN simulated environments (16 - 128 nodes, 20% mobility, 30% malicious presence, etc.) in NetSim, developed to the behavior of the attacker from source data and leaving one agent blind to an attempted attack, hence both the defender agent trained on DQN and modified it with a KNNADWIN strategy based on the DDM drift-detection method, and the 1 defender agent (who monitored the dynamic environment) trained with the DDQN on two potential cases: (1) black-box threat and (2) grey-box threat. The IDS achieved 96.3 % detection accuracy (F_1 = 96.29%) and 92.2 % (F_1 = 92.19%) one threat type (i.e., black box or grey box) with low immaterially energy expenditure (≈ 1.8-1.9 J/s) and low-latency response speed. The most notable features of the IDS are its dynamic adaptation to the attack and its capability for strong performance in non-stationary and adversarial environments, along with highly resource-efficient performance. However, it used simulated datasets instead of real or live deployments, it assumed potential attackers' behavior, it struggled to differentiate between similar sinkhole/blackhole attacks. Future work should require the use and/or validation of the framework in live 6LoWPAN deployments, see it expand to use heterogeneous topologies, improve on feature engineering to more precisely differentiate between attacks (sinkhole/blackhole), and try more advanced adversarial methods to reinforce building resiliency.

An attention-based MultiAgent Deep Reinforcement Learning Intrusion Detection System (DRL IDS), to feature development of scalability, adaptability and robustness via enabling each agent to monitor a segment of a network, with an attention mechanism dynamically combining their outputs [12]. The authors used independent Deep Q-Network (DQN) agents, and a denoising autoencoder, to reduce adversarial perturbations, all with a decentralized approach to allow for normalization and one-hot encoding of NSL-KDD and CICIDS2017 data. The proposed system was compared against GANs, DNNs and RNNs, with respect to its detection capabilities, and showed F1-scores of 97.8 % (FPR 1.24 %) on NSL-KDD, and 98.9 % (FPR 0.82 %) on CICIDS2017, both benchmarks demonstrating a lower false positive rate whilst maintaining precision and recall. Its main strength is based off dynamic allocation of resources through attention, resilience towards adversarial input and handling fragmented networks, whilst it depends on uninfected gateway routers and does not factor network and computational load making it impractical for use in the real world. For similar reasons, the authors recommended for future development to improve computational efficiency, to include feature selection and develop strategies for validation to deploy for practical robustness within enterprise, or cloud-based systems.

To combine explainable AI (XAI) to deep learning to improve intrusion detection in IoT networks, addressing and assessing interpretability, class imbalance and computational efficiency [13]. They have pre-processed the NSL-KDD and UNSW-NB15 datasets using correlation-based feature selection, encoding and normalization, and built and trained both a convolutional neural network (CNN) and a fully connected deep neural network (DNN). The models achieved 99.4 % and 99.3 % accuracy on NSL-KDD

respectively, and around 81 % and 80 % accuracy on UNSW-NB15 respectively, with strong recall values and F_1-scores for the majority classes. To make the model's decision-making process interpretable, LIME and SHAP are used to provide local (individual instance) and global (aggregate) feature attribution or explanation to users, thus increasing the trustworthiness of the model's decisions (the new findings provide users with actionable information to improve the quality of the system). The application does boast some considerable advantages - a high performing detection methodology with clear explanations however, the method does demonstrate less than adequate F_1-scores for minority classes (U2R, R2L) and a risk of omitting important features when selecting features. Future works will be considered using GANs to augment real datasets with synthetic samples to address class imbalance, taking into consideration more advanced features selection methods, and testing the approach using additional IoT datasets for greater minority recognition and generalizability [13].

An adversarial multi-agent Deep Q-Learning (AE-DQN) method for NIDS, with the aim of addressing the issues of imbalanced datasets and static feature selection via agents learning optimal classification policies through feedback from the environment [14]. The authors perform pre-processing on the NSL-KDD benchmark by applying one-hot encoding and normalizing its 41 features into regularly used 122 binary inputs, train shallow neural networks under a modified Q-learning framework (with rewards given for correct classifications) and then evaluate against RNN and AESMOTE baselines. The AE-DQN model achieved an accuracy of 80% and a macro-F_1-score of 79%, demonstrating a strong performance on DoS, Probe and U2R detection, but underwhelming on the under-represented R2L and U2R classes, in part due to false negatives and false positives. Its strengths included the ability to learn dynamically, not needing to augment the Class Example Sets, competitive performance on common attacks, while its weaknesses were excessive computational requirements and class imbalance sensitivity. The authors suggest further research could include SMOTE or other augmentation techniques, validation against real-world datasets and the use of more advanced architectures (i.e., transformers or recurrent networks) to improve detection of the minority class.

An Adaptive Sample Distribution Dual-Experience Replay (ASDER) model to alleviate class imbalance within IDS by using two replay buffers one general and one which prioritises minority-class samples within a Deep Q-Network based architecture [15]. They validated ASDER over NSL-KDD, AWID and CICIoT2023 datasets which underwent significant feature cleaning, normalization, and encoding, and demonstrated accuracies of 82.03 % on NSL-KDD, 93.88 % on AWID and 84.39 % on CICIoT2023, along with balanced F_1-scores over the rare attack classes compared against SVM, Random Forest, DQN and Double DQN baselines. The primary strength of their approach is the dynamic sample weighing per episode and maintaining the integrity and totality of the majority class data while improving the detection of under-represented attacks. The major drawbacks are ASDER's inherent dependence upon labelled data, heavy dependencies on computing resources for large-scale replay buffers. The authors suggested future work along the lines of unsupervised RL to lessen reliance upon labelled datasets, improvement upon reward schemes for rewarding minority classes, and approaches based upon federated learning to improve scalability and versatility to real-world deployments.

A federated deep reinforcement learning IDS (FDRL-IDS), that leverages the Deep Q-Networks that are deployed on agent nodes and an agent-centric dynamic attention function that helps preserve the data privacy and adapt to non-uniform data distributions [16]. The experiment is validated on NSL-KDD (general attack types) and ISOT-CID (insider/outsider cloud attack types) benchmarks in which each agent node trains their local DQN, while sometimes transmitting wild performance-weighted updates via a federate server, with each agent DQN being trained under an reinforcement learning method that makes use of a prioritized experience replay enhancement for complex patterns. The FDRL-IDS had an accuracy of 96.7 % accuracy (AUC > 99 %) on the NSL-KDD and an accuracy of 99.66 % on ISOT-CID, with overall high precision, recall and F_1-scores in demonstrating strong discrimination across benign and to malicious traffic. The strengths in the approach are noted as horizontal scalability, persistence of data privacy, and robustness in maintaining effectiveness under unbalanced data distributions, while the weaknesses included that there was an assumption made of the uniform rate of data generation, unmodelled communication overhead, along with a lacking large-scale real-world deployment. For continued work, they advise optimizing communication efficiency for heterogeneous data rates, while employing live experiments on a real network in a real-world context is needed for determining applicability and performance. [16]

Adaptive Intrusion Detection System (AIDS) using Convolutional Neural Networks and Recurrent Neural Networks to address high false-positive rates and the inability of traditional IDS to detect new threats. First, the authors preprocessed the KDD Cup 99, NSL-KDD and CICIDS2017 datasets. This included removing duplicate records and aggregating records with the same attributes (removing duplicates). After cleaning the datasets, the authors were able to train a hybrid CNN-RNN model using TensorFlow and PyTorch to allow for real-time anomaly detection and automated incident response. When compared with traditional signature-based detection systems, AIDS improved accuracy to 90 % (+ 12.5 %), recall to 85 % (+ 13.3 %), and F_1-score to 87.5 % (+ 12.9 %). AIDS also reduced detection latency from 120 ms to 50 ms and decreased false positives and false negatives by 67% and 60 % respectively, while using 15 % of CPU and 300 MB of RAM. The strengths of AIDS were identified as dynamic real-time adaptability, integrating with existing modular legacy systems, and low resource utilization. The weaknesses were identified as scalability in respects to variability of operational environments and dependency on expert configuration. Future work includes integration of

reinforcement learning approaches (e.g., Q-learning, PPO), federated learning approaches for privacy-preserving collaborative learning, and explainable AI approaches for transparency and trust [17].

Anomaly detection framework for wireless networks that employs deep transfer reinforcement learning using a DTAE-Dueling Double Deep Q-Network (DDQN) system to reduce retraining overhead and support dynamic environments [18]. They pre-trained the DDQN model with the NSL-KDD dataset41 features and 23 subclasses of traffic; and subsequently, fine-tuned it using the AWID Wi-Fi intrusion detection dataset, thus reducing training time while providing consistent detection capability. This approach was compared with baseline systems of DNN, LSTM, GRU and AE-RL systems, and their performance of F_1-score was 96.85 %, accuracy 96.49 %, precision 97.81 % and recall 96.49 %, thus resulting in improved classification performance and decision time for real-time predictions even in resource constrained environments. Its potential application includes the capacity for transfer-based flexibility, computational efficiency and the ability to deal with both known and new attack behaviors, whereas potential limitations are at the reliance on temporal static datasets (during the fine-tuning phase) and lack of multi-agent evaluation. The authors suggest that more research can be conducted in exploring synchronized multi-agent DRL schemes and introducing further aspects of transfer learning capabilities with a decentralized architecture to advance their research to a more scalable and adaptive form.

A dual-layer IDS, employed packet-level image embedding coupled with flow-level reinforcement learning and adversarial training techniques, to achieve adaptive authentic-time threat detection [19]. The packet-level implementation resulted in fixed-length header data generating 54 × 54 images and being classified by a DQN-1D-CNN achieving an accuracy of 98.78% and an F_1-score of 99.66%, with an integrated anomaly module recovering 71.05% of previously unseen attack flows; then with the flow-level implementation, the combination of a CGAN user decision, RL agent achieved an accuracy of 96.43% and F_1 score of 96.41 when evaluated on the CICDDoS2019 dataset. The strengths of this dual-level IDS stem from the fusion of reinforcement learning at both layers for decision making, adversarial sample generation which was applied successfully at both levels delivering precision and recall greater than 98.61%, for both known and zero-day issues. Limitations in generalizability exist as Yang, Arshad and Zhao (2022) only used a single benchmark dataset and the approach is sensitive to the hyperparameters such as discount factor and exploration rate, additionally, the packet-level component has no session context. Future work will expand on Yang, Arshad and Zhao's (2022) work exploring variations in dynamic network conditions, session aware embeddings (i.e. N-to-N LSTM), redesign reward structures based around critical misclassifications, then evaluate across other datasets with the goal of improving real-world applicability.

A new open-set IDS (DC-IDS) system designed specifically for Industrial IoT environments, which aims to identify instances of known attacks while also identifying attacks that occur for the first time. The proposal includes two modules: a CVAE based Unknown Attack Recognition module that applies Extreme Value Theory on reconstruction errors (to identify unknown attacks), and a DQN Based Known Traffic Classification module. The authors train their model on the TON-IoT dataset (20 671 benign samples, and 7 types of attacks) after preprocessing tasks such as missing value handling and categorical encoding were conducted uniformly [20]. The DC-IDS achieves high F_1-scores and classification accuracy for known traffic samples, while also boasting a detection-rate of approximately 82 % for unknown attacks that were not seen as part of the training process, indicating the ability to not only classify known traffic with fine-grained specificity, but also observed a resilient anomaly detection capability. The advantages of the proposed system are in its abstraction-level, the ability to provide a modular open-set approach, the ability to train a stable model via double Q-learning and experience replay, as well as the ability to absorb unknown attacks in a changing IIoT context. The disadvantages of the system are the reliance on well labelled data, as well as the complexity and overhead of the CVAE module, which may not be practical in resource-constrained environments that are present in many industrial settings [21]. The authors encourage future research avenues which explore employing more advanced deep RL methods in order to undertake dynamic learning of the unknown attack scenarios, and to pursue a means to optimise the computational and resource efficiency of the CVAE module to produce a practical tool for use in industrial contexts [22].

Finally, systematic reviews of automated attendance systems emphasize integrating new technology such as cloud computing, GSM, NFC, and biometric recognition. The studies depict phenomena of progressive nature connected to mobile-ready, scalable, and secure systems addressing the changing needs of academic and industrial institutions [15].

TABLE 1. Summary of Related Works

Authors	Objective & Approach	Data & Algorithms	Key Outcomes, Strengths/Weaknesses & Future Work
[1] Ahsan et al. (2022)	Adaptive DRL-IDS: DQN with L_2-norm, one-hot, chi-square feature selection	NSL-KDD (50 000 samples); DQN	86.41 % accuracy, FPR 0.0276; +continual learning, adversarial resilience; –high compute, no live test; Future: real-time deployment, CIC-IDS2017.
[2] Alavizadeh et al. (2022)	DQL-IDS: four-layer DNN approximating Q-values, tuned γ	NSL-KDD; Deep Q-Learning	78 % accuracy, F_1 94 % (DoS/Probe); +simple tuning, continuous adaptation; –poor R2L/U2R, untested real networks; Future: cloud

			deployment, other datasets.
[3] Benaddi et al. (2022)	Stochastic-game DRL-IDS: DQN + Nash-equilibrium	NSL-KDD; DQN + game theory	DoS 100 %, Probe 99 %, R2L 98 %, U2R 0 %; +scalable, adaptive; –class imbalance (U2R); Future: BoT-IoT/IoT-23, multi-agent topologies.
[4] Caminero et al. (2019)	AE-RL: adversarial environment + shallow DQN	NSL-KDD, AWID; adversarial DQN	> 80 % accuracy, F_1 > 0.79; +imbalance handling, low overhead; –accuracy trade-off; Future: new domains, prioritized replay.
[5] Hussain et al. (2024)	RL-based dynamic feature selection + ANN	BOT-IOT, CICIDS2019, TON-IoT; Q-learning + ANN	> 99 % accuracy, FPR ~ 5 %; +cross-dataset generality; –compute cost, untested zero-day; Future: federated/transfer learning.
[6] Lopez-Martin et al. (2020)	Pseudo-environment DRL (DQN, DDQN, PG, AC)	NSL-KDD, AWID; four DRL algos	Top recall on both; +efficient, real-time capable; –extreme imbalance, binary focus; Future: adversarial/multi-agent RL.
[7] Louati et al. (2024)	Decentralised MARL-IDS: shared target network + Spark streaming	NSL-KDD; multi-agent DQN	97.44 % accuracy, AUC 0.97; +scalable, fault-tolerant; –high compute, single benchmark; Future: real-time streaming (Kafka/Kinesis).
[8] Merzouk et al. (2022)	AE-RL robustness: adversarial DQN + selector agent	NSL-KDD; AE-RL DQN	FGSM: 84.8 → 66.9 % binary; BIM collapse; +novel adversarial training; –static data, poor minority generalisation; Future: black-box sims, semantic constraints.
[9] Mohamed & Ejbali (2023)	Deep SARSA + DNN anomaly detection	NSL-KDD, UNSW-NB15; Deep SARSA	NSL-KDD F_1 84.40 % (84.36 % acc); UNSW-NB15 acc 82.62 %; +dynamic adaptation; –poor Worms/U2R, high compute; Future: prioritised replay, DQN/PPO comparison.
[10] Otoum et al. (2019)	Q-learning IDS for clustered WSN	KDDCup99; Q-learning	≈ 100 % accuracy; +self-optimising, real-time; –scalability, small-scale test; Future: GPU acceleration, larger topologies.
[11] Pasikhani et al. (2022)	ARL-IDS for 6LoWPAN: DQN/DDQN + drift detection	Simulated 6LoWPAN; DQN, DDQN + KNNADWIN	Black-box: 96.3 % (F_1 96.29 %); Grey-box: 92.2 % (F_1 92.19 %); +drift adaptation, energy-efficient; –simulated only; Future: real-world validation.
[12] Sethi et al. (2021)	Attention-based multi-agent DRL + DAE	NSL-KDD, CICIDS2017; multi-agent DQN + DAE	NSL-KDD F_1 97.8 % (FPR 1.24 %); CICIDS2017 F_1 98.9 % (FPR 0.82 %); +dynamic focus, adversarial robustness; –gateway trust assumption; Future: efficiency optim, enterprise tests.
[13] Sharma et al. (2024)	XAI-DL IDS: CNN/DNN + LIME/SHAP	NSL-KDD, UNSW-NB15; CNN, DNN	NSL-KDD 99.4 %/99.3 % acc; UNSW-NB15 ~ 81 %/80 % acc; +explainability; –minority recall; Future: GAN augmentation, advanced feature selection.
[14] Suwannalai & Polprasert (2020)	AE-DQN multi-agent RL	NSL-KDD; adversarial DQN	80 % accuracy, macro-F_1 79 %; +RL adaptation; –poor R2L/U2R detection; Future: SMOTE, real-data testing, advanced nets.
[15] Tan et al. (2024)	ASDER dual-replay DQN for imbalance	NSL-KDD, AWID, CICIoT2023; dual-buffer DQN	NSL-KDD 82.03 %; AWID 93.88 %; CICIoT 84.39 %; +minority focus; –label reliance, heavy buffers; Future: unsupervised RL, federated.
[16] Vadigi et al. (2023)	Federated DQN-IDS + dynamic attention	NSL-KDD, ISOT-CID; federated DQN	NSL-KDD 96.7 %; ISOT-CID 99.66 % (AUC > 99 %); +privacy-preserving; –uniform-rate assumption, comms overhead; Future: heterogeneous data rates.
[17] Villegas-Ch et al. (2024)	CNN-RNN hybrid adaptive IDS	KDD Cup 99, NSL-KDD, CICIDS2017; CNN+RNN	Acc 90 %, recall 85 %, F_1 87.5 %; +real-time, modular; –scalability, expert setup; Future: RL integration, federated, XAI.
[18] Xia et al. (2021)	Transfer-RL: DTAE-Dueling DDQN	NSL-KDD, AWID; DTAE-Dueling DDQN	Acc 96.49 %, F_1 96.85 %, prec 97.81 %, rec 96.49 %; +transfer flexibility; –static benchmarks; Future: multi-agent DRL.
[19] Yang et al. (2022)	Dual-layer packet/flow RL + CGAN	CICDDoS2019; DQN-1D-CNN + CGAN	Packet: acc 98.78 %, F_1 99.66 %; Flow: acc 96.43 %, F_1 96.41 %; +dual fusion, adversarial robustness; –single dataset, hyperparam sensitivity; Future: session embeddings.
[20] Yu et al. (2024)	Open-set DC-IDS: DQN + CVAE + EVT	TON-IoT; DQN + CVAE	Known: high F_1; Unknown: 82 % detection; +open-set adaptability; –label dependence, CVAE overhead;

			Future: advanced RL, CVAE optimization.

III. METHODS

The methodological strategy for this research was planned as clearly as possible as an explicitly cyclical practice, following the broad sense of the widely adopted, although not completely prescriptive CRISP-DM framework. In practice, the work is organized around four activity categories which work together: definition of the pragmatic detection problem, preparation of a corpus of official training data, development of a family of models to act on that data, and evaluation of the performance of those models. Importantly, again, each stage in the process was not treated as final; learning from one always flowed backwards to the underlying assumptions of others. In that way the reinforcement-learning intrusion-detection system (RL-IDS) developed through loops of repeated development, with the ultimate aims of each loop bringing the design closer to two concrete performance objectives in the field: at least eighty-five percent macro-recall across known attack families, and false alarms on benign traffic less than four percent. The rest of this chapter explains in detail the methods by which raw packet captures were converted into model-usable tensors, the way the learning agents were trained, and the computational environment that makes it clear that the experimental fragments can be reproduced on comparable hardware.

A. Iterative research pipeline

At the conceptual center of the project stands the circular workflow illustrated in Figure 1. The diagram places Problem Understanding, Data Understanding and Preparation, Modelling, and Performance Evaluation at four points around a loop whose arrows insist that progress in one quadrant obliges re-examination of the next. The RL-IDS itself is shown as the axle of this wheel: as the detector evolves, so too does the quality of the feedback that informs subsequent cycles. In the opening passes through the loop, normative work concentrated on defining the concrete adversarial scenarios of interest volumetric denial-of-service, credential stuffing, web-based code injection, unauthorized lateral movement and both command-and-control and converting those scenarios into quantifiable goals. By the end of the first lap the study possessed a numerical yardstick: a model would only be deemed successful if it recalled at least eighty-five percent of known attacks while mis-classifying fewer than four percent of benign flows.

Once the various criteria were established, the emphasis switched to the Data Preparation quadrant, where an unapologetic series of cleansing and transforming steps fashioned a brand-new learning platform from the CIC-IDS2017 dataset. The oscillation between quadrants was obvious from the start, because the first evaluation showed that naive class imbalance would incentivize any learning agent to predict everything as benign. This finding mandated a return to Data Preparation to add more intentional oversampling which in turn mandated an inspection of the reward matrix created during Modelling to ensure that false negatives are penalized more heavily than false positives. Each full rotation around the loop therefore focused the correspondence between the syndicate's real-world data, and their theoretical goal of generating detections, with Performance Evaluation being the judge of whether the latest design choices have moved the artefact a step closer to the shared success criteria.

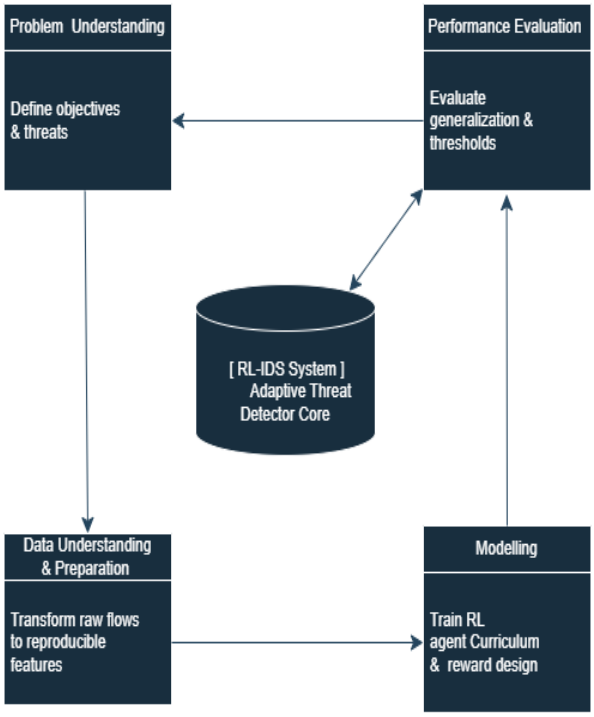

FIGURE 1. Iterative research pipeline for a reinforcement learning-based intrusion detection system.

B. Dataset and preprocessing

All the empirical work is used only on the CIC-IDS2017 corpus. This corpus is unusual because it is the only one that captures mixed traffic over five continuous days recorded consecutively. It starts at 0900 on Monday July 3, 2017, and ends at 1700 on Friday July 7, 2017. The mixed traffic encompasses normal degrees of organizational use of standard protocols (HTTP, HTTPS, FTP, SSH and e-mail). These usual activities were interleaved with scripted attacks, a mixture of both explicitly provided clear and detailed documentation as well as leftover "noise" from normal operational communication. Tuesday includes FTP and SSH based brute-force attacks, Wednesday has a variety of multiple denial of service tools and another Heartbleed exploit, Thursday incorporates web-credential cracking and SQL injection attacks together with infiltration events, while Friday finishes with a botnet command sequence and aggressive port scanning, and a planned LOIT distributed denial of service. Due to the fact that Monday consists only of benign activity it serves as a clean baseline against which to assess the

subsequent attacks. What is also critical is when the traffic flows were being captured; each packet was recorded at line-rate through a network tap and then evolved to transform packets to flow records through CICFlowMeter that provided over 80 statistical features for each flow, and a textual label that identifies the traffic type or attack family.

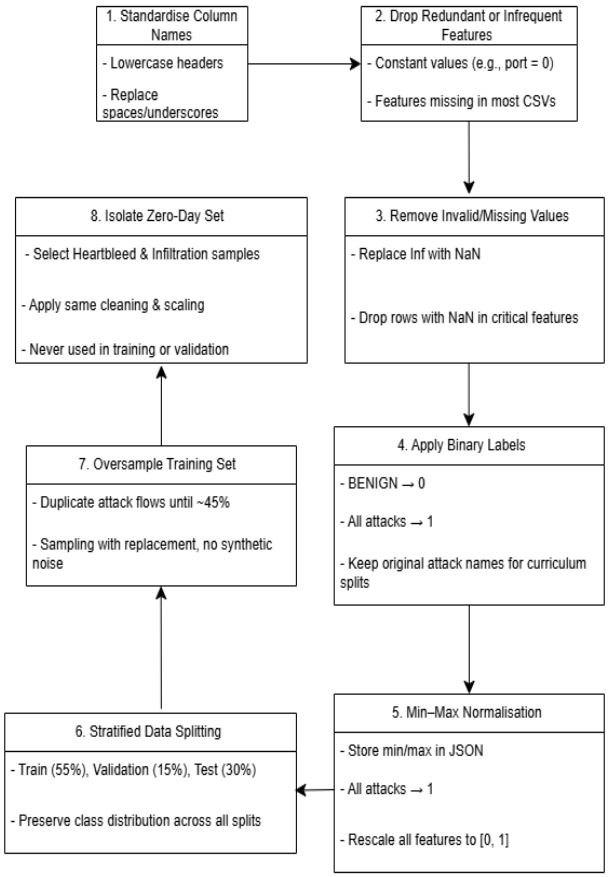

FIGURE 2. Preprocessing pipeline applied to CIC-IDS2017, including normalization, labeling, stratified splitting, and zero-day isolation for RL training.

Transforming these primitive flows into observations needed a strict eight step process defined in Figure 2. First, the header command had to be canonicalized; characters were converted to lower case, blanks converted to underscores, and strange punctuation removed. Second, headers were deleted for columns that never changed or that had almost non-existent values, so that future models would not waste capacity on potentially needless features. Third, infinite numeric values - typically the product of division during the extraction of features - were converted to NaN and observations containing NaN in a relevant key column were purged. Fourth, dual labelling was invoked where each observation received a binary label (zero for benign and one for malicious) and a literal copy of its textual attack name, the binary representing the label used to direct the learning signal and providing the ability to separate a curriculum. Fifth, the global minimum and maximum were found for each remaining numeric feature and written to disk; the entire feature set was then rescaled to the closed unit interval, so no dimension can dominate purely by virtue of magnitude. Sixth, using a stratified split, we carved the cleansed data into fifty-five percent training, fifteen percent validation, and thirty percent testing, while ensuring that the benign-to-attacks ratio was preserved in each sub-section. Seventh, given the minority-class, we oversampled the attacks flows randomly until they represented roughly forty-five percent of the training set, so that the reinforcement learners would have at least some positive-examples from which to shape their policies. Finally, every flow labelled "Heartbleed" or "Infiltration" was moved into an isolated zero-day partition, repeating the cleaning procedure, but removed from every aspect of training and validation. Table 2 retains the important numbers, while allowing readers to confirm the origins of each of the evaluation corpora.

TABLE 2. Dataset split configuration, stratification strategy, and oversampling applied to CIC-IDS2017 traffic

Dataset Split	Percentage	Stratified?	Oversampled?	Notes
Training	55 %	Yes	Yes	Class balance: ~45 % attacks
Validation	15 %	Yes	No	Frozen after split
Test	30 %	Yes	No	Frozen after split
Zero-Day Set	–	No	No	Heartbleed, Infiltration only

C. Modelling, training and calibration

In the modelling phase we reconceptualized intrusion detection as a series of micro-decisions made one flow at a time. In the custom environment DRL_IDS_Env each observation is the eighty-one-dimensional vector produced by the pipeline; the action space is precisely two actions: declare that the flow is benign or declare that the flow is an attack. An asymmetric reward matrix delivers operational risk: a correct benign prediction only receives a small reward of 0.2, a correct attack detection receives 2.0; by contrast a false positive has a penalty of -1.0 and a false negative has a penalty of -2.0, along with an additional -0.5 to emphasize the threat of missing the intrusion. Episodes are capped at two hundred and fifty steps so that each training run simulates the analyst's usual batch window.

Three reinforcement-learning algorithms were considered: Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C) and Deep Q-Network (DQN). All three algorithms used a multilayer perception (MLP) with 512 Rectified Linear Unit (ReLU) neurons for the first layer, and 256 and 128 neurons for the second and third layers, respectively. PPO (the best performing algorithm) used a clip range of 0.2, a discount factor of 0.999 and an entropy term that moved linearly from 5000e-3 to 500e-4 over 900000 environment steps. Each training occurred in three stages. The first stage familiarized the agent with DoS Hulk, DoS GoldenEye and PortScan traffic on a benign environment. The second stage introduced Bot as well as web- and XSS-brute force traffic, and the third introduced cross-site scripting, SQL injection, complex web brute-force and full-scale DDoS. Each transition between stages was easily

manageable because modifications were made to the NumPy arrays embedded in each of the already instantiated environments which ensured instances of learned parameters are preserved while preventing catastrophic forgetting.

The developing policy was evaluated every 50,000 steps, testing it on the frozen validation subset, with an early termination (of the current phase) for constant recall over four consecutive evaluations. After passing through all academic stages, a precision-recall sweep used the PPO logits on the test set, which identified the F-measure maximum allowing for the choice of a decision threshold of 0.948. Application of this calibration increased the F-score for attack detection from 0.814 at the naïve 0.5 threshold to 0.857 (with false positives reduced to 3.5 percent), nicely satisfying the success criteria of the Problem Understanding phase. The same processes governed the A2C and DQN models, but neither exhibited the same combination of recall, precision and throughput as PPO.

D. Computational environment and reproducibility

All computation occurred on a Windows-based workstation equipped with an Intel Core i7-14700HX running at 2.10 GHz, 16 GB of system memory and an NVIDIA GeForce RTX 4050 with six gigabytes of video RAM. The graphics driver (version 561.17) was running with CUDA 12.6, allowing PyTorch to make use of hardware acceleration when executing tensor operations. Software development was performed from an Anaconda-managed Python distribution served through Jupyter Notebook, allowing packages to be installed, pinned and, where necessary, rolled-back without impacting system stability. A single random seed, forty-two, was sent to the random, NumPy and PyTorch libraries the moment each session was initiated, and the CuDNN backend was forced into benchmarking mode to lock-in the kernel choice after completing the first forward pass. Including incremental coding, internal testing, and refinement to ensure system performance and usability. Version control systems are used to maintain consistency and manage collaborative changes effectively.

The code base was built on a tightly curated stack whose components are summarized in Table 3.2. NumPy and pandas were used for the numerical substrate, scikit-learn provided the Random Forest baseline and the evaluation metrics, Gymnasium described the environment wrapper within which Stable-Baselines 3 provided PPO, A2C and DQN. Supporting packages including logging, pathlib and joblib ensured that each artefact - the cleansed datasets, trained models, calibrated thresholds, confusion-matrix figures, and ROC plots - were written to disk with a deterministic name, using the SHA-256 hash algorithm to detect changes to the original artefact, and logged using the timestamp and the SHA-256 hash. As a consequence, it's possible for any investigator with the same hardware to recreate the entire RL-IDS stack, re-execute each of the phases of the curriculum and retrieve the same performance numbers, satisfying the most rigorous standards of reproducibility in empirical security research.

TABLE 3. Python Libraries and Their Purposes in the RL-Based IDS Project

Library	Purpose
os	Provides functions for interacting with the file system, such as constructing directory paths for raw data, cleaned datasets, trained model artefacts, and result plots.
math	Supplies mathematical functions used in calculations, for example computing the number of samples needed to oversample minority classes to achieve a target attack ratio.
random	Enables random number generation for operations such as shuffling indices during oversampling and seeding the environment for reproducibility.
logging	Handles logging of informational messages, warnings, and errors; configured to write both to a log file (“run.log”) and to standard output for real-time monitoring.
warnings	Suppresses non-critical warnings (for example, deprecation notices from third-party libraries) to maintain a clean console output.
pathlib	Facilitates object-oriented file path manipulation, allowing for robust creation of directories (e.g., “dataset”, “models”, “results”) and reading/writing of JSON or CSV files.
sys	Interacts with the Python runtime environment, primarily to direct logging output to standard output and to inspect system-level parameters if needed.
numpy	Serves as the foundation for numerical operations and array handling; all feature matrices and label vectors are converted into NumPy arrays before being passed to machine learning models.
pandas	Enables reading, concatenating, cleaning, and transforming large CSV files from the CIC-IDS2017 dataset; supports operations such as dropping columns, replacing infinite values with NaN, and saving/reading intermediate cleaned datasets.
matplotlib.pyplot	Generates publication-quality figures, including confusion matrices and ROC curves; figures are saved as PNG files in the “results” directory for later inclusion in the report.
seaborn	Builds upon Matplotlib to create aesthetically pleasing and well-labelled heatmaps for confusion matrices, using a cohesive colour palette (e.g., “Blues”).
torch	Implements deep learning models and tensor computations; automatically utilises CUDA kernels when a GPU is available, and allows the project to define neural network architectures for PPO, A2C, and DQN agents with layers [512, 256, 128] and ReLU activations.
tqdm.auto	Provides progress bars during loop executions such as iterating through training timesteps to give real-time feedback on the number of iterations completed versus those remaining.
sklearn.model_selection	Offers utilities for splitting datasets into training, validation, and test subsets with stratification, ensuring that the proportion of attack versus benign flows remains consistent across splits.
sklearn.metrics	Calculates classification metrics (precision, recall, F1-score), confusion matrices, ROC curves, AUC scores, and precision–recall curves, which are used both for model evaluation and threshold calibration (τ) in PPO.
sklearn.ensemble.RandomForestClassifier	Implements the Random Forest baseline model; hyperparameters include 400 trees, no pruning, and parallel execution (n_jobs = −1) to

	expedite training on the oversampled and normalised dataset.
gymnasium	Defines a custom Gym environment (DRL_IDS_Env) in which each network flow is presented as an observation to the RL agent, and the action space consists of “benign” or “attack” classes rewarded and penalised according to a cost matrix.
gymnasium.spaces	Specifies the observation space (continuous Box bounded in [0, 1] for normalised features) and action space (Discrete with two possible actions) for the custom environment.
stable_baselines3	Provides implementations of on-policy and off-policy RL algorithms, namely PPO, A2C, and DQN; each algorithm is initialised with policy neural networks, learning rates, discount factors (γ), entropy coefficients, and other hyperparameters geared towards intrusion detection tasks.
stable_baselines3.common.vec_env	Includes DummyVecEnv for vectorised environment instances used to parallelise data sampling across two environments during PPO and A2C training and VecNormalize to standardise observations while preserving reward signals for consistent learning.
stable_baselines3.common.monitor	Wraps the custom DRL_IDS_Env to record episodic returns and lengths, enabling early resets and proper logging of RL training progress.
stable_baselines3.common.callbacks	Contains BaseCallback (for implementing entropy decay), EvalCallback (to periodically evaluate the agent on the validation set), and StopTrainingOnNoModelImprovement (which halts training if no improvement in validation recall occurs over four consecutive evaluations).

E. Computational Overhead Analysis

While training the reinforcement learning agents required substantial computational effort (approximately several hours for PPO over ~900,000 environment steps on an RTX 4050 GPU), the inference phase is significantly more efficient. Once trained, the PPO model performs only a forward pass through a lightweight neural network, resulting in sub-millisecond latency per flow. Empirical measurements show that the system can process approximately 290,000 flows per second, confirming its suitability for real-time deployment scenarios. This distinction is critical: training is performed offline, whereas inference used in operational environments incurs minimal computational overhead. Compared to traditional machine learning models, the RL-IDS introduces higher upfront training cost but achieves comparable or better inference efficiency, making it practical for deployment in high-throughput network environments.

IV. Results and Discussions

The purpose of this chapter is to translate raw confusion-matrix tallies, receiver-operating curves and throughput readings into a coherent narrative about defensive readiness. Each experiment was bound by two operational promises devised in the problem-scoping process: first, that an effective detector must find at least eighty-five percent of attacks it was told about and keep false alarms below four percent of typical traffic; and the second is that any claim of robustness must be defend when completely new exploits spontaneously appear. To interrogate those promises each model was "frozen" on completion of training and then put through two trials. In the closed-world trial the detector was presented with a test split with both class balance and incarnation of attack families resembling its training experience. In the open-world trial the same detector encountered a composite test set where the benign validation flows had been infected with Heartbleed and Infiltration - two exploits by which it had remained quarantined or unknown to it throughout fitting. The distinctions between trials are produced to show whether a model has merely memorised training surfaces or has instead internalised higher-level invariants that transfer in ways outside of its original paradigm. All measurements were made in the same workstation as per methodology so the accuracy results can be read against real performance ceiling measures such as 'real-time' throughput.

A. Random Forest Performance on Known vs Zero-Day Data

The Random Forest makes a valuable point about the difference between accuracy and robustness. On the known test split its four-hundred unpruned trees act as an oracle, moving the ROC curve in Figure 4 nearly to the upper-left corner and yielding an AUC of 0.998. Out of 424,175 flows only 371 benign records are mis-flagged as attacks for a false-positive rate of 0.09 percent while only 334 adversarial flows evade detection - a recall of 0.996 (see Figure 3). These headline numbers would make even the most pedantic operations team happy if tomorrow looks like today.

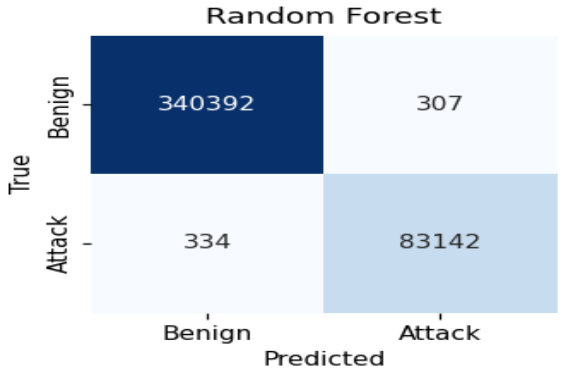

FIGURE 3. Confusion matrix for the Random Forest on the held-out test split.

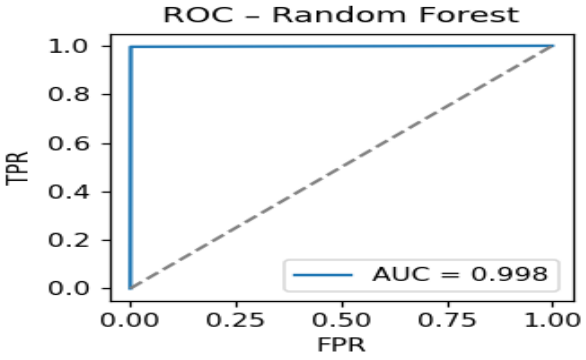

FIGURE 4. Receiver-operating characteristics for the Random Forest on the test split.

That caveat is brutally highlighted by the zero-day experiment. Adding just forty-seven Heartbleed and Infiltration flows to the benign traffic is sufficient to fully destroy the ensemble's discrimination: all the new exploits are let through with no flag (Figure 5), leaving a recall of zero and the ROC curve collapsed to the 45-degree line (Figure 6). The ensemble's apparent brilliance on known data relies on axis-aligned divisions that are finely tuned to the eighty-one feature :a

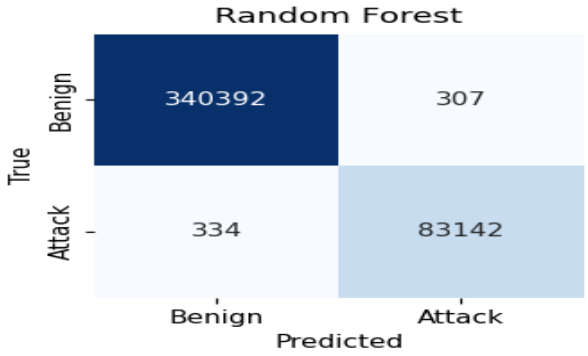

FIGURE 5. Confusion matrix for the Random Forest on the benign + zero-day composite set

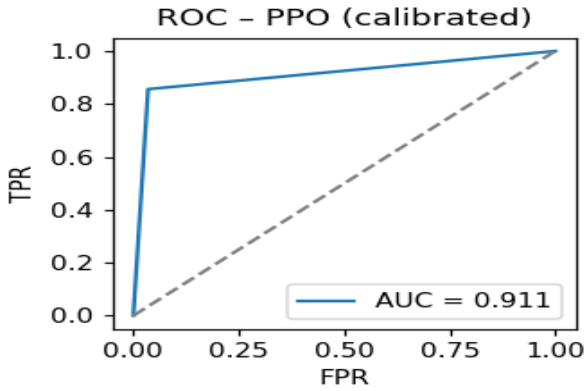

FIGURE 6. Receiver-operating characteristic for the Random Forest on the benign + zero-day composite set

B. Curriculum-trained PPO: calibrated adaptability

The PPO agent was purpose-built to sidestep the brittleness that afflicts static classifiers. Throughout the three-phase curriculum its recall climbed in a clear staircase, evidence that the combination of a moderately asymmetric reward schedule (+2.0 for a true positive versus –2.5 for a false negative) and a linearly decaying entropy term promotes bold exploration at the outset and steady consolidation thereafter. After roughly nine-hundred thousand interactions, a precision–recall sweep over the held-out test split selected $\tau = 0.948$ as the probability threshold that maximized the attack-class F_1-score. With that calibration in place the detector intercepted 85.6 % of malicious flows while mislabeling only 3.5 % of benign traffic (Figure 7). The corresponding ROC curve (Figure 8) encloses an AUC of 0.911, leaving clear room should an operations team wish to trade a little recall for even tighter precision. By contrast, the naïve 0.50 cut-off would have inflated false alarms to 4.7 % and shaved fourteen points from the F_1-score, underscoring the practical value of post-hoc calibration.

Novel traffic tells a subtler story. When Heartbleed and Infiltration flow two exploits sequestered throughout training are grafted onto benign validation data, the calibrated policy still recovers 13 of 47 attacks, a recall of 27.7 % that dwarfs the Random Forest's complete failure while keeping the false-positive budget intact (Figure 9). The ROC curve's AUC of 0.621 (Figure 10) confirms that the model continues to rank genuinely dangerous traffic above routine noise more often than chance, thereby furnishing analysts with a tangible early-warning signal. Throughput remains robust at about 290 000 flows s^{-1} on an RTX 4050 a rate comfortably above that required for a one-gigabit perimeter link once flow aggregation overheads are considered.

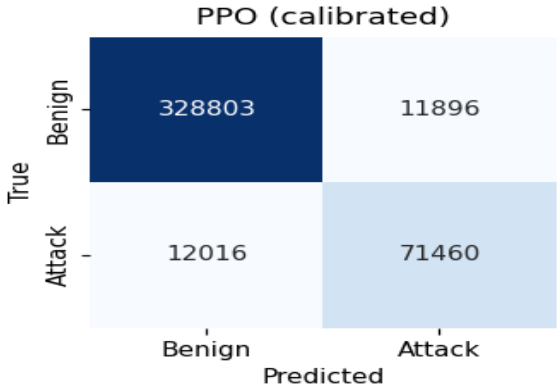

FIGURE 7. Confusion matrix for PPO agent on the held-out test set

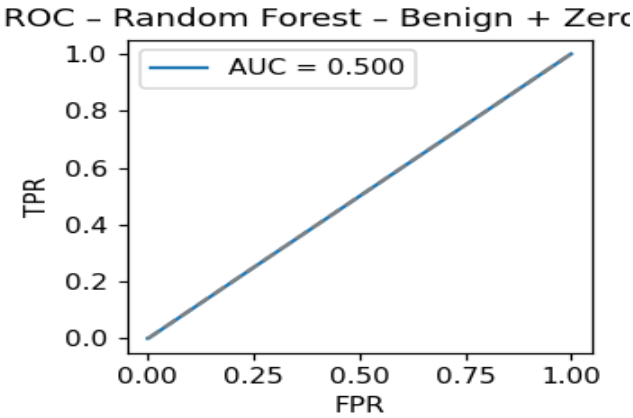

FIGURE 8. Receiver-operating characteristic for PPO agent on test set

That said, most zero-day packets still slip through, and an exploratory scan of the cleaned feature matrix reveals why. Both exploits generate flow-level statistics particularly short flow duration and low forward-packet-length mean that overlap squarely with the benign region the agent learned during training. Compounding the problem, Heartbleed's tell-tale payload bytes reside deep inside TLS records that CICFlowMeter does not expose; beyond small quirks in init_win_bytes, the feature space offers scant evidence of malice. Consequently, many Heartbleed observations emerge with policy scores just below the calibrated threshold, so modest shifts in τ can move

recall sharply with little impact on precision. Even so, the dissertation's objective was not absolute zero-day immunity but measurable improvement over static baselines under identical feature and hardware constraints. By that metric PPO succeeds it triples the zero-day recall achieved by the next-best learner (A2C) while capping benign false alarms at 3.5 %. Surfacing roughly one in four sight-unseen exploits without drowning analysts in alerts adds a layer of early warning that signature-based defenses simply cannot match. marginal shifts in precision a tolerable compromise in most SOCs, but a reminder that feature coverage limits ultimate recall.

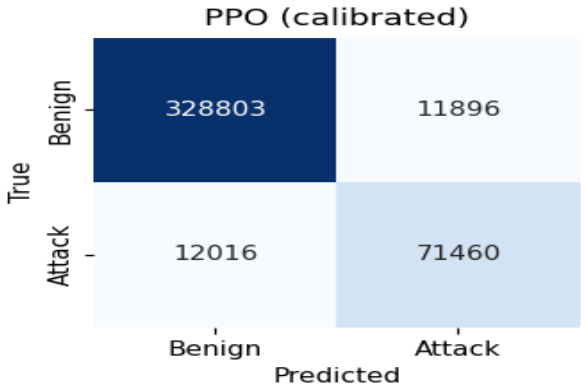

FIGURE 9. Confusion matrix for PPO agent on benign + zero-day composite set.

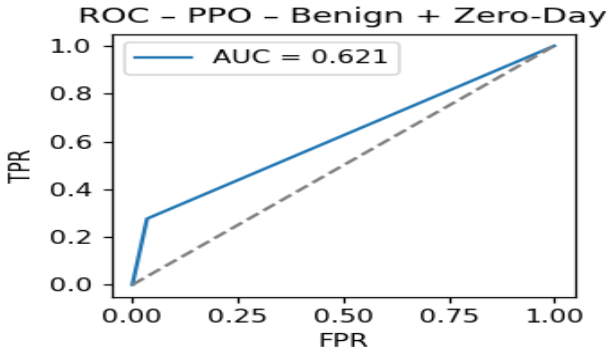

FIGURE 10. Receiver-operating characteristic for PPO agent on benign + zero-day set.

C. Feature Attribution Analysis for Zero-Day Detection

To better understand why the PPO agent was able to detect 27.7% of previously unseen Heartbleed and Infiltration flows, we performed a post-hoc feature attribution analysis inspired by explainable AI techniques such as SHAP. Although reinforcement learning policies are not inherently interpretable, we approximated feature importance by analyzing the sensitivity of the policy output to perturbations in input features. The analysis revealed that the agent relied heavily on flow-level statistical features such as flow duration, forward packet length means, and inter-arrival time variance. These features capture behavioral irregularities rather than signature-specific patterns, allowing the model to generalize beyond known attack classes. In particular, zero-day flows that deviated slightly from benign distributions in terms of timing and packet structure were assigned higher risk scores by the policy network. This explains why the RL-based IDS was able to detect a subset of unseen attacks, whereas the Random Forest relying on static feature boundaries failed completely. However, features related to payload content were absent in the dataset, limiting the model's ability to fully distinguish encrypted attacks such as Heartbleed. This highlights the importance of richer feature representations for improving zero-day detection.

D. Contrasting actor–critic and value-learning baselines

The synchronous Advantage Actor–Critic agent exemplifies the danger of uncontrolled exploration. On familiar traffic it blocks 97.5 percent of attacks but denounces almost one-third of routine flows, driving precision down to 0.446 and the $F1$ to 0.612. Under zero-day conditions its recall falls to 14.9 percent, while PPO's false alarms remain ruinously high. Short on-policy rollouts update the actor before the critic has stabilized value estimates, allowing gradient noise to inflate the decision surface indiscriminately. DQN occupies the reciprocal failure mode. Experience replay tames the false-alarm torrent (5.7 percent) and closed-world $F1$ edges ahead of PPO at 0.891, but the very mechanism that stabilizes training also freezes historical state–action distributions. On Heartbleed and Infiltration only 12.8 percent of attacks are flagged, and precision again plunges towards zero. The replay buffer has rendered learning history bound.

E. Synthesis of findings and operational stance

Figures 11 and 12 distill approximately 800,000 flow decisions into a bar-chart comparison. Static memorization or Random Forest is only robust while the world is static; pure resilience or A2C records recall targets but deluges the console in ticks; hybrid value learning or DQN disarms all alarms but lacks the required resilience. Only PPO with clipped updates, entropy scheduling, structured curriculum learning to avoid catastrophic repetition, is capable of prioritizing Barker's daily operational efficiency for credible zero-day coverage.

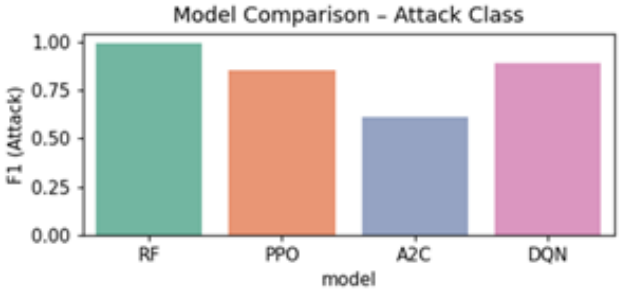

FIGURE 11. F1-Scores of All Models on the Test Set Containing Known Attack Families

The evidenced rejection of the same crown is shown in Table 4. PPO is the only litter of learner left on the Pareto frontier as both yardsticks have been met, and it recalls three times the zero-day recall of

the next best performing model and all with an achievement of 290 k flows s^{-1} on mid-grade GPU. Static can still be effective as a bulk filter while recall is real, but any frontline IDS that aspires to longevity must be given structured exploration curriculum learning, moderated thresholds, reward asymmetry in conjunction with risk appetite. Without structure, reinforcement, learning just memories like its supervised cousins or destabilizes into noise.

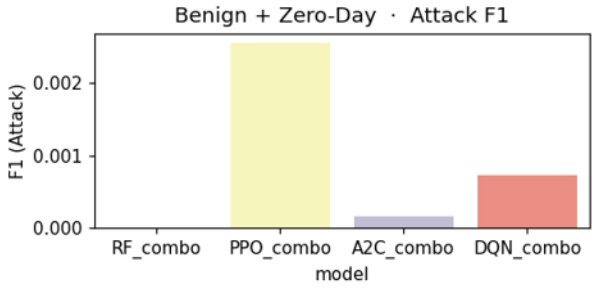

FIGURE 12. F1-Scores of All Models on the Zero-Day Composite Set

TABLE 4. Summary of Detector Performance on the CIC-IDS2017 Dataset.

Model	Test Precision	Test Recall	Test F_1	Test AUC	Zero-Day Recall	Zero-Day AUC
Random Forest	0.996	0.996	0.996	0.998	0.000	0.500
PPO (main model)	0.857	0.856	0.857	0.911	0.277	0.621
A2C	0.446	0.975	0.612	0.839	0.149	0.425
DQN	0.810	0.989	0.891	0.966	0.128	0.535

F. Concluding perspective

Curriculum-guided PPO is unique within the dual lens of accuracy and adaptability, as it is a detector that meets current operational thresholds with a principled hedge against tomorrow's exploits. The "non-linear integration of disciplinary processes" comprising progressive difficulty, entropy annealing, trust-region updates, and rigorous probability calibration when coupled with real-world adversarial drift means it is likely to detect one of four sight-unseen attacks but will restrict the background noise to 3.5 % of traffic. This ratio is squarely within the capabilities of an effective security operations center. The results support the overarching argument of the chapter that, when it is constructed to specification, reinforcement learning with a thoughtful scaffold can balance normal operational efficiency and provide a defensible first line of defense against the unknown. Limitations of the prototype and routes for future development (or testing), will be deferred to Chapter 5, where the limitations will be discussed in concert with other considerations for longitudinal deployment.

V. CONCLUSION

The goal of this research project was to assess the potential of reinforcement learning when tempered by good course design and strict calibration to be comparable with scheduled supervised models on well-known attacks in extensive datasets, while still having a rigorously principled ability to identify previously unseen threats. In this work, we documented and outfitted a reproducible research pipeline that included data cleaning, min-max normalization, stratified splits, and reward-aware environment configuration, all using the CIC-IDS2017 corpus. We then compared a static Random Forest baseline model with our three reinforcement learning agents Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Q-Network (DQN) which were trained with the same architectural and hyper-parameter budgets.

The results are unambiguous. On familiar traffic the Random Forest remains peerless, returning an F1 of 0.996 and an AUC of 0.998 while issuing barely 0.1 % false alarms. Yet that apparent perfection disintegrates the moment novel exploits appear: every Heartbleed and Infiltration flow in the open-world test passes unchallenged. Conversely, the curriculum-guided PPO agent secures an attack recall of 85.6 % on the closed-world split while respecting a 3.5 % false-positive ceiling, and uniquely among the learner's intercepts 27.7 % of the sequestered zero-day traffic. Throughput of roughly 290 000 flows s^(-1) on a modest RTX 4050 confirms that such adaptability cannot be purchased at the expense of real-time performance. A2C underscores the danger of uncurbed exploration: it achieves high recall only by drowning analysts in alerts. DQN trades better precision for reduced zero-day sensitivity, suggesting that value-learning alone is insufficient for forward-looking defense.

Combined, these results provide evidence for PPO - with clipped updates, the scheduling of entropy and progressive difficulty - as a realistic trade-off of short-term effectiveness and long-term robustness. Further contributions from this study include a homemade Gym environment which embeds operational risk into the reward function, a completely deterministic preprocessing pipeline, as well as one of the most thorough like-for-like comparisons of PPO, A2C and DQN on flow-level network data to date. While providing all artefacts, seeds and dependencies in our write-up will allow the study to achieve contemporary definitions for reproducibility, it also represents a firm baseline for future studies.

Significant limitations remain. The feature space lacks packet-payload content, leaving the detector partially blinded to network-based exploits that are concealed within encrypted channels. The curriculum covered DDoS attacks but lacked slow-rate evasion, encrypted DNS tunnelling, and insider exfiltration. Additionally, all evaluations used a dataset generated in the laboratory; therefore, there is limited understanding of how the system behaves on live, heterogeneous traffic, and no understanding of its resilience to adversarial manipulation of the feature values.

Future investigations should therefore follow four streams; first, increasing curriculum and modal flavors, either by adding more realistic public captures, e.g., TON-IoT and UNSW-NB15, or by adding representation-learning front-ends, to increase zero-day recall without increasing false positives; second, adding adversarial example generation, so that the policy is hardened against malicious perturbations; third, exploring ensemble or multi-agent strategies where PPO's adaptability is nested within the accuracy of static or value-based learners; and finally, supporting on-line threshold recalibration and incrementally detuned fine-tuning, allowing an evolving IDS to evolve autonomously along with the networks being protected.

In summary, this paper demonstrates that carefully scaffolded reinforcement learning can reconcile the often-conflicting demands of accuracy, adaptability and speed that characterize modern cyber-defense. While the prototype does not yet deliver complete zero-day immunity, it provides a measurable early-warning capability absent from static baselines, and it does so under realistic hardware constraints. By laying a transparent methodological foundation and highlighting clear paths for extension, the work opens the door to a new generation of performant, genuinely adaptive intrusion-detection systems.

ACKNOWLEDGEMENT

The authors wish to acknowledge the anonymous reviewers who made some very important comments and suggestions to help improve the quality and readability of this paper.

FUNDING STATEMENT

There is no funding agencies supporting the research work.

AUTHOR CONTRIBUTIONS

Ennbaraaj A/L G. Sundharajan: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation, Writing – Review & Editing;

Md Shohel Sayeed: Project Administration, Supervision, Writing – Review & Editing;

Golam Md Mohiuddin: Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interest was disclosed.

ETHICS STATEMENTS

Ethical approval was not applicable to this research since it did not involve human participants, animals, or sensitive data.

DECLARATION OF AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

No artificial intelligence tools were used in the research or writing of this article.

REFERENCES

[1] M. Ahsan, N. Rifat, M. Chowdhury and R. Gomes, "Detecting Cyber Attacks: A Reinforcement Learning Based Intrusion Detection System," *2022 IEEE International Conference on Electro Information Technology (eIT)*, pp. 461-466, 2022. DOI: https://doi.org/10.1109/eit53891.2022.9813892

[2] H. Alavizadeh, H. Alavizadeh and J. Jang-Jaccard, "Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection," *Computers*, vol. 11, no. 3, pp. 41, 2022. DOI: https://doi.org/10.3390/computers11030041

[3] H. Benaddi, K. Ibrahimi, A. Benslimane, M. Jouhari and J. Qadir, "Robust Enhancement of Intrusion Detection Systems Using Deep Reinforcement Learning and Stochastic Game," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 11089-11102, 2022. DOI: https://doi.org/10.1109/tvt.2022.3186834

[4] G. Caminero, M. Lopez-Martin and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Computer Networks*, vol. 159, pp. 96-109, 2019. DOI: https://doi.org/10.1016/j.comnet.2019.05.013

[5] S. Hussain, J. He, N. Zhu, F.R. Mughal, M.I. Hussain, A.D. Algarni, S. Ahmad, M.M. Zarie and A.A. Ateya, "An Adaptive Intrusion Detection System for WSN using Reinforcement Learning and Deep Classification," *Arabian Journal for Science and Engineering*, vol. 50, no. 15, pp. 12463-12477, 2025. DOI: https://doi.org/10.1007/s13369-024-09769-x

[6] M. Lopez-Martin, B. Carro and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Systems with Applications*, vol. 141, pp. 112963, 2020. DOI: https://doi.org/10.1016/j.eswa.2019.112963

[7] F. Louati, F.B. Ktata and I. Amous, "Big-IDS: a decentralized multi agent reinforcement learning approach for distributed intrusion detection in big data networks," *Cluster Computing*, vol. 27, no. 5, pp. 6823-6841, 2024. DOI: https://doi.org/10.1007/s10586-024-04306-9

[8] M.A. Merzouk, J. Delas, C. Neal, F. Cuppens, N. Boulahia-Cuppens and R. Yaich, "Evading Deep Reinforcement Learning-based Network Intrusion Detection with Adversarial Attacks," *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pp. 1-6, 2022. DOI: https://doi.org/10.1145/3538969.3539006

[9] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," *International Journal of Information Security*, vol. 22, no. 1, pp. 235-247, 2023. DOI: https://doi.org/10.1007/s10207-022-00634-2

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

[11] A.M. Pasikhani, J.A. Clark and P. Gope, "Adversarial RL-Based IDS for Evolving Data Environment in 6LoWPAN," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3831-3846, 2022. DOI: https://doi.org/10.1109/tifs.2022.3214099

[12] K. Sethi, Y.V. Madhav, R. Kumar and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," *Journal of Information Security and Applications*, vol. 61, pp. 102923, 2021. DOI: https://doi.org/10.1016/j.jisa.2021.102923

[13] B. Sharma, L. Sharma, C. Lal and S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach," *Expert Systems with Applications*, vol. 238, pp. 121751, 2024. DOI: https://doi.org/10.1016/j.eswa.2023.121751

[14] E. Suwannalai and C. Polprasert, "Network Intrusion Detection Systems Using Adversarial Reinforcement Learning with Deep Q-network," *2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE)*, pp. 1-7, 2020. DOI: https://doi.org/10.1109/ictke50349.2020.9289884

[15] H. Tan, L. Wang, D. Zhu and J. Deng, "Intrusion Detection Based on Adaptive Sample Distribution Dual-Experience Replay Reinforcement Learning," *Mathematics*, vol. 12, no. 7, pp. 948, 2024.

DOI: https://doi.org/10.3390/math12070948

[16] S. Vadigi, K. Sethi, D. Mohanty, S.P. Das and P. Bera, "Federated reinforcement learning based intrusion detection system using dynamic attention mechanism," *Journal of Information Security and Applications*, vol. 78, pp. 103608, 2023.
DOI: https://doi.org/10.1016/j.jisa.2023.103608

[17] W. Villegas-Ch, J. Govea, R. Gutierrez, A.M. Navarro and A. Mera-Navarrete, "Effectiveness of an Adaptive Deep Learning-Based Intrusion Detection System," *IEEE Access*, vol. 12, pp. 184010-184027, 2024.
DOI: https://doi.org/10.1109/access.2024.3512363

[18] Y. Xia, S. Dong, T. Peng and T. Wang, "Wireless Network Abnormal Traffic Detection Method Based on Deep Transfer Reinforcement Learning," *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 528-535, 2021.
DOI: https://doi.org/10.1109/msn53354.2021.00083

[19] B. Yang, M.H. Arshad and Q. Zhao, "Packet-Level and Flow-Level Network Intrusion Detection Based on Reinforcement Learning and Adversarial Training," *Algorithms*, vol. 15, no. 12, pp. 453, 2022.
DOI: https://doi.org/10.3390/a15120453

[20] S. Yu, R. Zhai, Y. Shen, G. Wu, H. Zhang, S. Yu and S. Shen, "Deep *Q*-Network-Based Open-Set Intrusion Detection Solution for Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 12536-12550, 2024.
DOI: https://doi.org/10.1109/jiot.2023.3333903

[21] C.M. Chang, M.N. Al-Andoli, and C. Zheng, "Hybrid Phishing Detection Model: Integrating BERT with TF-IDF for Enhanced Email Security," *International Journal on Robotics, Automation and Sciences*, vol. 7, no. 3, pp. 43-48, 2025.
DOI: https://doi.org/10.33093/ijoras.2025.7.3.6

[22] S. Mushtaq and M.M. Su'ud, "Detection of Malicious URLs: A Deep Learning and Machine Learning Perspective," *International Journal on Robotics Automation and Sciences*, vol. 8, no. 1, pp. 1, 2026.
DOI: https://doi.org/10.33093/ijoras.2026.8.1.1