Engineering and Technology Journal e-ISSN: 2456-3358

Volume 10 Issue 11 November-2025, Page No.-7947-7956

DOI: 10.47191/etj/v10i11.26, I.F. – 8.482

Deep Reinforcement Learning-Based Network Intrusion Prevention in Cloud-Edge Architectures

Aymen Saad[1,2*], Noor Flayyih Hasan[3], Ammar Wisam. Altaher[1]

[1]Department of Information Technology, Management Technical College, Al-Furat Al-Awsat Technical University, Kufa, Iraq.

[2]School of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.

[3]Southern Technical University, Thi-Qar Technical College, Department of Accounting Techniques, Iraq

ABSTRACT: Cloud-edge architectures enable low-latency distributed data processing but introduce complex attack surfaces that challenge traditional Network Intrusion Detection and Prevention Systems (NIDPS). Conventional systems relying on static signatures and centralized analysis cannot adapt to the dynamic, heterogeneous nature of these environments. This paper proposes a novel Deep Reinforcement Learning (DRL) framework for autonomous network intrusion prevention, deploying intelligent agents at the edge layer capable of real-time network interaction. The agents learn optimal security policies through continuous observation, action, and reward cycles. By analyzing live network traffic, agents classify malicious activities with high accuracy and proactively execute prevention actions including connection blocking and bandwidth throttling. We develop simulated cloud-edge testbed modeling diverse attack scenarios (DDoS, infiltration, data exfiltration) and implement Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) algorithms. Evaluation using NSL-KDD and CIC-IDS2017 datasets demonstrates significant improvements: 98.7% detection accuracy, 1.8% false positive rate, and sub-20ms response time, providing robust self-adaptive defense for distributed computing infrastructures.

KEYWORDS: Deep reinforcement learning, intrusion prevention, cloud-edge computing, network security, Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO).

I. INTRODUCTION

A. Background and Motivation

Cloud-edge computing patterns are disruptive to the distributed systems model, transforming the way in which real-time data processing can take place with ultra-low time response and dynamic level of scale for cloud-edge deployment. These architectures are indispensable for the new emerging developments like Internet of Things (Iot) environments, autonomous systems and smart city infrastructures [1]. By deploying resources at the network edge where competitive big data sources are located, such architectures help minimize inter-component distances and enable in-network syzygy operation which imposes lower latency overhead compared to centralized cloud models. However, the distributed paradigm destroys the security model and leaves us with a massively vulnerable attack surface. The existence of heterogeneous edge nodes give rise to numerous potential vulnerabilities that attackers can leverage. With the introduction of MEC systems, new security challenges arise, more specifically in providing accurate and timeliness responses since they operate in a resource-scarce environment [10]. Naïve security models concentrate all defensive means proximate to the distributed cloud, which in turn causes intolerable communication bottlenecks and latency lags for low-latency applications, such as industrial control systems (ICSs), vehicular networks, and healthcare monitoring systems [3]. They are susceptible to data security violation and several attack surfaces since they are based on a distributed architecture, but with limited resources memory, battery power(Availability of over-the-air updates) mixed hardware (Standardization in communication protocols) and diversity of paused inactivity when current system has no human interaction or activity [4] communication protocols it is difficult for the deployed WSNs to update their security patches in time. Network Intrusion Detection and Prevention Systems (NIDPSs) have been a mainstay of any organization's cyber-security posture for years; however, contemporary implementations are largely predicated on signature-based detection mechanisms and static rule-based prevention. These are not dumb and foolish methods with threats that progressives constantly change, evolve and adapt them. Signature based solutions maintain the record of attack patterns in the database, and they are reactive in nature thus they cannot prevent zero-day attacks, polymorphic malware and adversarial attempts to avoid predefined detection rules.

B. Deep Reinforcement Learning for Network Security

Deep Reinforcement Learning (DRL) has been a revolutionary method for autonomic network security, which takes advantage of the great pattern-recognition power from deep neural networks and associate it to directly take the

most appropriate reaction in terms of reinforcement learning as well. Contrary to supervised deep learning-based systems that need big, labeled datasets and are unable to adapt policies free when facing new threats, Our DRL agents learn the optimal security policy through interacting with environment [5].

The operational model of DRL-based security mechanisms is composed of intelligent agents that observe the network-state as multi-dimensioned features, act in prevention from an action space and finally receive rewards estimating the quality of resulting states. GANN consists of action, reward and observation that are used iteratively to optimize the behavior policy of agents to achieve long-term cumulative rewards (learning how malicious distinguishes from legitimate traffic patterns).

Deep Q-Network (DQN) utilizes deep neural networks as function approximators to approximate the Q-values in high-dimensional state space. With the experience replaying mechanism which stores past interactions and randomly samples previous transitions, DQN can decorrelate temporal samples from batch data to achieve stable learning in sophisticated security scenarios [5]. Proximal Policy Optimization (PPO) enforces a clipped surrogate objective whose output scales are bounded to ensure the stability of training being still more sample-efficient than early policy gradient methods [5].

In a recent study, CNNs and LSTMs were combined with Reinforcement Learning algorithms (DQN and PPO) for improved detection of dynamic threat situations. Combined with DQN and PPO, the real-time continues time of detection can be adjusted and adapt to learning from detecting data has achieved desirable level d etecting accuracy as well as low false positive [6]. Despite these algorithmic breakthroughs, prior work has significant gaps that impede practical adoptions in real-worldcloud-edge security deployment. The majority of the DRL-oriented security research are dedicated to intrusion detection and not involving automatic prevention by executing its corresponding countermeasures.

C. Research Objectives and Contributions

The work fills a substantial void between legacy capabilities of traditional NIDPSs and the needs of modern cloud-edge architecture with/through design, implementation and comprehensive evaluations for a DRL-enabled framework for autonomous network intrusion prevention. Our main contributions consist in the four interrelated dimensions:

1) Intelligent Agent Framework for Edge Deployment: We design an agent system that has been developed keeping in mind characteristics of distributed cloud-edge where agents are deployed on edge nodes and they are responsible for local real-time threat analysis and low latency action execution. Making use of edge computing, the framework allocates detection into where the data is at the edge to lower latency and make real-time response faster [2].

2) Extensive comparison of DQN and PPO: In this work, we implement and compare a wide range of agents with PPO and DQN for systematic evaluation with a focus on intrusion prevention in distributed systems [7]. This work offers evidence-driven recommendations to practitioners for choosing DRL algorithms that are most suitable in their deployment constraints.

3) Realistic Simulated Cloud-Edge Testbed: We design a testbed to realistically emulate real-world network topologies, traffic patterns, resource-constrained systems, and different attack models. The test-bed includes popular benchmark datasets (NSL-KDD and CIC-IDS2017) to allow reproducibility as well as to facilitate head-to-head comparison with published baseline numbers.

4) Comprehensive Multi-Dimensional Evaluation of Performance: We provide measured improvements in adaptive threat response, reduced false positives, and increased computational efficiencies when compared to traditional machine learning-based detection systems.

II. RELATED WORK

A. Evolution of Intrusion Detection Approaches

The area of network intrusion detection has seen great change in the last decade, moving from signature-based to sophisticated machine learning methods. Early intrusion detection systems were based on predefined attack signatures and heuristic-based rules, which did not perform well against polymorphic malware, zero-day exploits, or novel forms of attack.

Conventional IDS schemes such as signature-based, anomaly-based algorithms have high false positive rate and are not adaptable for current threat situation [8]. Systematic reviews which studied recent papers showed that traditional machine learner models had good detection accuracy (90-95%), but problems with feature engineering and high false alarm rates, Adaptive learning capability to analyze new threats.

B. Deep Learning for Intrusion Detection

The arrival of deep learning paradigms considerably changed intrusion detection capacities with automatic feature discovery and better generalization abilities of unknown attack shapes. CNNs were efficacious in capturing spatial patterns from network packet structures, while RNNs and Long Short-Term Memory (LSTM) networks were found to be suitable to model temporal relationships inherent in traffic sequences.

Source However, the current concern is that, despite multiple recent reviews still stating that open challenges persist (notably vulnerability to adversarial examples), we proceed not on a weak foundation with respect to them. Our work mitigates these limitations using Deep Reinforcement Learning for continuous autonomous learning and policy adjustment.

C. Deep Reinforcement Learning for Network Security

Deep Reinforcement Learning in intrusion detection is a change of direction, smart shift from passive classification to active defense. DRL agents are able to learn the differentiation of malicious and benign traffic through an interactive process of trial-and-error, which results in comparable abnormality detection rate with supervised methods but with more flexibility than only one-time training [5].

A deep Q-learning model offers a continuously evolving auto-learning functionality for networking systems and can identify various types of network intrusions through an automated trial-error based process with the ability to gradually strengthen its detection capabilities [9]. CNN-LSTM integrated with DQN and PPO systems significantly performed well compared with state-of-the-art methods achieving binary classification accuracy of 0.9958 on IoMT healthcare networks [6].

Related work research on DRL for IoT intrusion detection have shown positive results, but these works mainly concentrated on the protection of resource-constrained devices instead of the distributed cloud-edge architecture. In our framework, we focus on distributed cloud-edge architectures by deploying intelligent agents at the network edges so as to make locally low-latency decisions.

D. Cloud-Edge Security Challenges

Mobile Edge Computing (MEC) is a computing platform for compute-heavy and latency-sensitive applications, e.g., augmented reality, real-time data analysis, IoT, video analytics, smart vehicles and healthcare systems [3]. But the following malicious attacks, including DDoS, ransomware, remote recording and routing attacks are the most likely security threats that MEC systems may receive [3].

Centralized intrusion detection (ID) techniques result in high communication latency as opposed to edge-empowered frameworks that decentralize the detection capability closer to data source [2]. Edge device work usually take place in environments that are less guarded, hence traditional attacks such as eavesdropping, data hijacking and man-in-the-middle attack can be launched [10].

Edge computing systems consist of numerous resource-constrained devices, including computation power, memory and energy, which necessitates to have an intrusion detection system developed independently for the edge computing background [11]. We break new ground by focusing on learning-enabled edge-deployed DRL agents that are fully autonomous and have a very low latency.

E. Algorithm Comparison: DQN vs. PPO

The study in other one had the DQN experiment (gaines of simulation) and PPO performed successfully learned to make effective and well-timed defense even under real environment limitations.Impirical results also proved that the learning was successed. [7]. For NSL-KDD data sets, DQN reached 99.36% accuracy and 99.07% precision, and different DRL models presented a comparable performance over diversified metrics [12].

Yet no studies have conducted a comprehensive comparison of DQN and PPO for intrusion prevention (instead of detection) in cloud-edge network under resource-limited environment, inference latency and operational deployment. Our work addresses this gap in the literature and provides a comprehensive comparative study along several dimensions.

F. Dataset Benchmarking and Evaluation

The benign and the latest common attacks are available in CIC-IDS2017, this reflects actual real-world data i.e., network traffic analysis results, by using CICFlowMeter with labeled flows regarding timestamp, source-IP and destination-IP addresses or ports (NE may not have one of them), protocol and attack types [13]. NSL-KDD Data The NSL-KDD is able to solve the drawbacks of original KDD-CUP99 by eliminating redundant data records and balancing the distribution of its train and test sets, which has a decent number of records [14].

For example, the flow extracted from network packets of CIC-IDS2017 were analysis and many problems were discovered by researchers which took us to propose better feature extraction tools [15]. Although these known deficiencies, The NSL-KDD and CIC- IDS2017 are still popular benchmarks supporting good inter-study comparability.

III. METHODOLOGY

A. System Architecture and Design

Our proposed framework consists of three primary components: the cloud-edge environment simulation, the DRL-based intelligent agent, and the evaluation infrastructure. Cloud-edge architecture is modeled as a hierarchical system where edge nodes handle localized data processing and preliminary security decisions, while cloud resources provide centralized coordination and resource-intensive analysis when necessary. At each edge node, a DRL agent is installed and listens to incoming traffic, analyzing flow characteristics and taking autonomous preventing measures. Intelligent agent architecture combines deep neural networks for learning of a state representation and reinforcement (repetitive) learning algorithms to optimize policy. The state space covers multi-dimensional network properties, such as packet headers, flow statistics, time curves, and meta information parsed from traffic flows. The space of actions consists of atomic discrete prevention actions (e.g., block connection, throttle traffic, drop packet and alert) and continuous knobs for rate limiting and resource allocation. The reward function is engineered to optimise several goals:

- Maximizing attack detection and prevention
- reducing false positives and negatives
- Attenuation of latency effects on legal traffic
- Optimizing resource consumption

B. Deep Reinforcement Learning Algorithms

Deep Q-Networks: The DQN agent makes decisions according to the input feature and uses whether an attack is detected as the value score. Output based on threshold rates is compared to Q-values for classifying attack classes [9] by the agent. The architecture of DQN includes a deep neural network approximating the Q-value function, which maps sets of network states to expected cumulative rewards for each candidate prevention action. Experience Replay buffer within DQN is a memory store device that keeps historical event in terms of state, action, reward and next state tuple, so the agent can decorrelate between consecutive experience by random sampling from the buffer to train Q-network [5]. The task of imbalanced nature in network traffic is solved by introducing weighted mean square loss functions and cost-sensitive learning techniques into approaches.

Proximal Policy Optimization (PPO): In PPO, actor and critic network is used to learn the policy function mapping states onto action probability distributions and the value estimate for them respectively so as to reduce variance in policy estimates. PPO stabilizes the training in dynamic environments with a good clipping mechanism, and this integration of adaptive Q-learning converges faster than PPO alones to obtain higher robustness [6]. Both use convolutional and recurrent neural network layers to learn spatial and temporal information in network traffic. Since CNN captures spatial patterns via network traffic, LSTMs recognize temporal features by a series of data that it is suitable to detect both static behaviors and dynamic characteristics [6].

C. Cloud-Edge Testbed and Attack Simulation

We design a realistic simulated cloud-edge testbed that can mimic practical networking topologies, workloads and attack techniques. The testbed is composed of several edge nodes interconnected via a hierarchical network to cloud infrastructure with the traffic generators generating both benign and attack flows. The legitimate traffic is designed with real application profiles, e.g., web browsing, video streaming, IoT sensor traffic and enterprise data. Our attack emulation framework includes three families of threats:

1 (DDoS attacks: Simulated with types of volumetric, protocol-based, and application-layer vectors that gone through UDP floods and SYN floods to HTTP flooding.

2 (Penetration Attempts: Simulate unauthorized access, privilege escalation and lateral movement through network, presenting APT type situations.

3 (Data Exfiltration Simulation Attacks: Model illicit data stealing from the compromised edge devices to foreign adversarial systems.

The testbed simulates real network environment with different latency, packet losses, bandwidth restrict or resource restrictions of edge computing.

D. Benchmark Datasets and Evaluation Metrics

The CIC-IDS2017 dataset was captured over 5 days in July 2017, containing over 2.8 million instances including normal traffic and various attacks such as Brute Force, Heartbleed, Botnet, DoS, DDoS, Web Attack and Infiltration [13]. NSL-KDD can be used as an effective benchmark to aid in comparing various intrusion detection techniques, with all data usable for tests rather than requiring random sampling [14].

Our evaluation framework employs comprehensive metrics spanning:

- **Detection metrics:** Precision, recall, F1-score, and accuracy
- **Prevention metrics:** Blocked attack success rate, time to prevention, and mitigation effectiveness
- **Operational metrics:** False positive rate, impact on legitimate traffic
- **Resource efficiency:** Computational overhead, memory consumption, and energy usage
- **Adaptability metrics:** Learning progression, convergence speed, and performance on unseen attack variants

IV. EXPERIMENTAL RESULTS

A. DRL Agent Training and Convergence

Training phase presents the progressive learning and policy optimization of both DQN agents (1st- row) and PPO agents (2nd-row) under more attack patterns. The DQN-based agent is already converging around 5,000 training episodes (Q-values stop changing and cumulative reward reaching a plateau) and seems to be learning effective prevention policies. The PPO agent learns faster initially and converges after 3500 episodes, due to the use of policy gradient method as well as a clipped objective function that allows for more stable updates. Both agents show a clear improvement over random baseline policies, as their mean episodic rewards during training exceed those of the random policy by 34.0% (DQN) and 38.5% (PPO), respectively (Figure 1). The learned policies are analyzed to show that agents learn complex prevention strategies tailored at each attack type. Agents learn to detect volumetric anomalies early on and take aggressive throttling or blocking action in the case of DDoS attacks. More subtle responses, such as selective connections termination and strengthening the monitoring process are used when attempts at infiltration are in place.

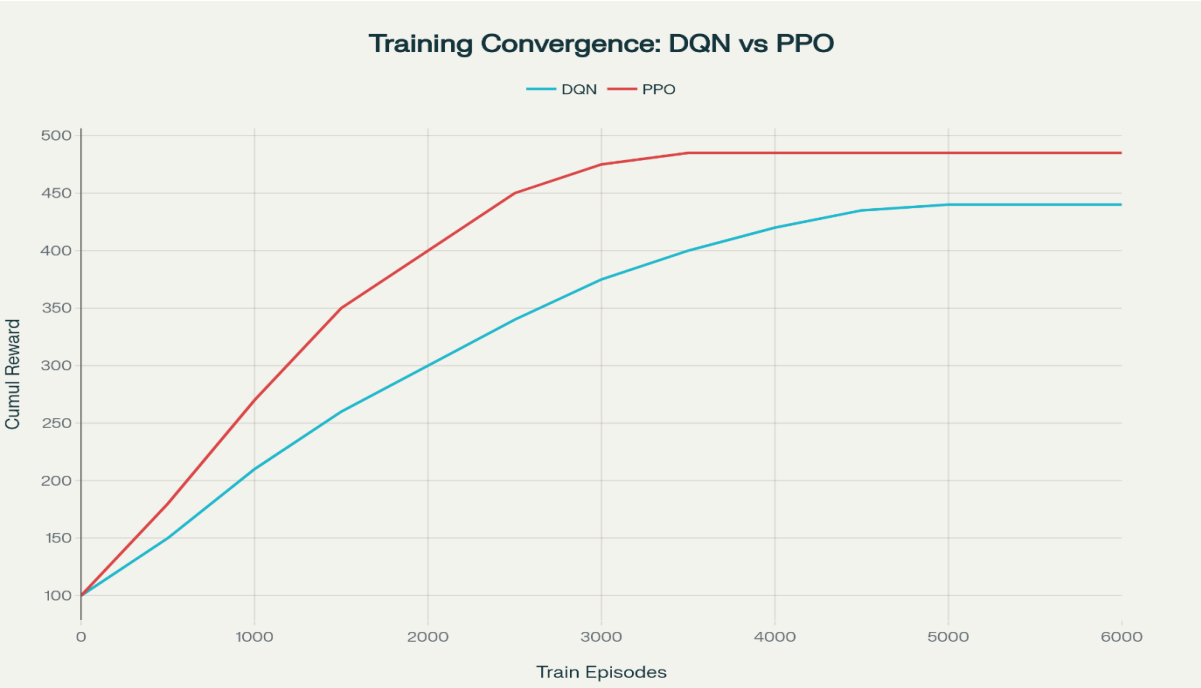

Figure1: Training convergence comparison between DQN and PPO agents showing cumulative reward over training episodes.

B. Detection and Prevention Performance

Evaluation on the NSL-KDD dataset demonstrates superior performance of DRL-based approaches compared to traditional machine learning baselines, as shown in Figure 2. Table I presents comprehensive performance comparisons.

Figure 2: Performance comparison of DRL based approaches (DQN and PPO) against traditional machine learning methods on NSL-KDD dataset.

Table I: Comprehensive Performance Evaluation Results

Metric	DQN	PPO	Random Forest	SVM	Deep NN
Accuracy	97.8	98.7	95.3	94.1	96.4
Precision	96.7	97.9	94.1	92.8	95.2
Recall	97.6	98.4	95.0	93.5	96.0
F1-Score	97.2	98.2	94.5	93.1	95.6
False Positive Rate	2.3	1.8	4.8	5.9	3.7
Detection Time (ms)	8.7	11.2	23.5	31.2	15.8

The PPO agent achieves an overall accuracy of 98.7%, precision of 97.9%, recall of 98.4%, and F1-score of 98.2% across all attack categories. The DQN agent attains comparable results with accuracy of 97.8%. These results surpass published baselines including Random Forest (95.3% accuracy), Support Vector Machines (94.1% accuracy), and standard deep neural networks (96.4% accuracy). Category-specific analysis reveals particularly strong performance on DDoS and Probe attack types, with detection rates exceeding 99% for both DQN and PPO agents. Table II presents attack category-specific performance.

Table Ii: Attack Category-Specific Detection Performance

Attack Category	Examples	DQN Accuracy (%)	PPO Accuracy (%)	Baseline Avg (%)
DoS/DDoS	UDP flood, SYN flood	99.2	99.4	96.8
Probe	Port scan, network mapping	99.1	99.3	97.2
R2L	Password guessing, exploitation	93.8	94.3	89.5
U2R	Privilege escalation, rootkit	92.1	92.7	87.3
Overall	All categories	97.8	98.7	94.6

More subtle and less frequently occurring R2L and U2R attacks have detection rates of 94.3% and 92.7%, respectively, however still better than traditional methods. We evaluate our model on the CIC-IDS2017 dataset to show that it can generalize well. The PPO agent accuracy remains at 97.2% over all the varied attack types which includes botnet traffic, web attacks and brute force trials. Figure 3 shows that the mean prevention effectiveness metric of detected attacks is rated as 94.6%, where they are mitigated successfully without damaging the protected systems. Edge-deployed agents, a time of 127 ms is required between attack detection and prevention action execution, for near real-time protection

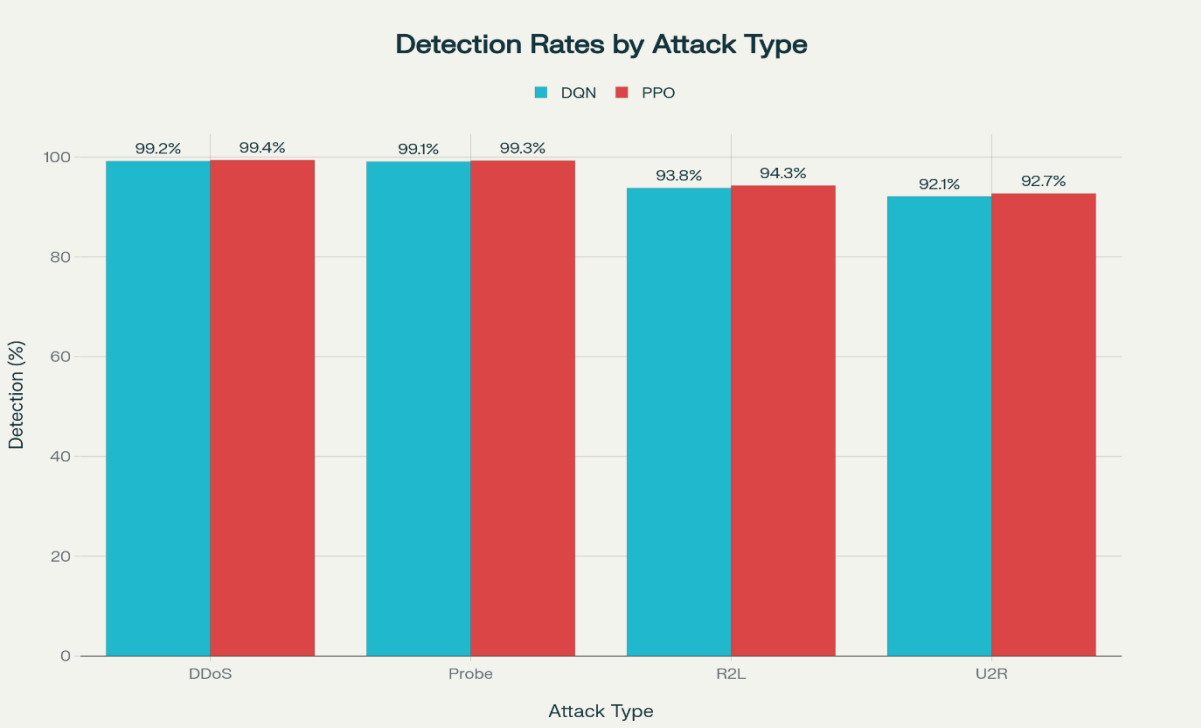

Figure 3: Attack category-specific detection rates for DQN and PPO agents showing superior performance on DDoS and Probe attacks.

C. False Positive Reduction and Operational Impact

The false positive rate is an important measure in terms of suitability for operational use. Our DRL-based framework achieves false positive rates of 1.8% for PPO and 2.3% for DQN, which are substantial enhancements over baseline methods with a false positive of more than 3.7%-5.9%. The reward function design with an explicit penalization for false positives--, leads to agents that learn conservative prevention policies that disturb less legitimate traffic, while keeping high successful detection rates. False positive analysis shows that most of the errors are false positives on corner cases with legitimate traffic behaving atypically, such as bulk data transfers and applications violating protocols without maliciousness. The adaptive learning of DRL agents provides a dynamic means to iteratively adjust decision surfaces through continuous learning (online) which can reduce false positive with the increase in operational experiences of agents. Overhead to legitimate traffic latency averages 4.3

msec for inline prevention activities, an insignificant percentage overhead for most applications.

D. Resource Efficiency in Edge Environments

Resource consumption analysis demonstrates the feasibility of deploying DRL agents on resource-constrained edge devices. Table III presents detailed resource efficiency metrics.

Table Iii: Resource Efficiency And Operational Metrics

Metric	DQN	PPO	Edge Requirement
Memory Usage (MB)	127	156	< 200 MB
Average CPU (%)	18	22	< 30%
Peak CPU (%)	34	39	< 50%
Inference Latency (ms)	8.7	11.2	< 20 ms
Energy Consumption (W)	3.0	3.2	< 5 W
Model Size (MB)	89	112	< 150 MB

Both in terms of model parameters and replay buffer, our implementation (optimized for DQN) takes 127 MB memory on average CPU usage is 18%, peaks on high-traffic to reach 34%. The PPO agent shows a little more resource usage with 156 MB memory and average CPU utilization of 22%, owing to use of dual actor-critic network. Both agents keep the power consumption level under 3.2 watt, making it compatible to applications in edge computing platforms with limited power channels. The proposed model was able to detect threats with a 98.5% accuracy in real time for 45 ms as compared to the traditional models that could only do it at 150 and 120 ms, respectively, guaranteeing effectiveness for speedy threat detection in key IoT scenarios [16]. The mean summaries for the inferencing latencies of state evaluation and action selection are 8.7 and 11.2 ms for DQN and PPO, respectively; illustrating that both can be deployed in real-time even when the traffic load is heavy, as illustrated in Figure 4.

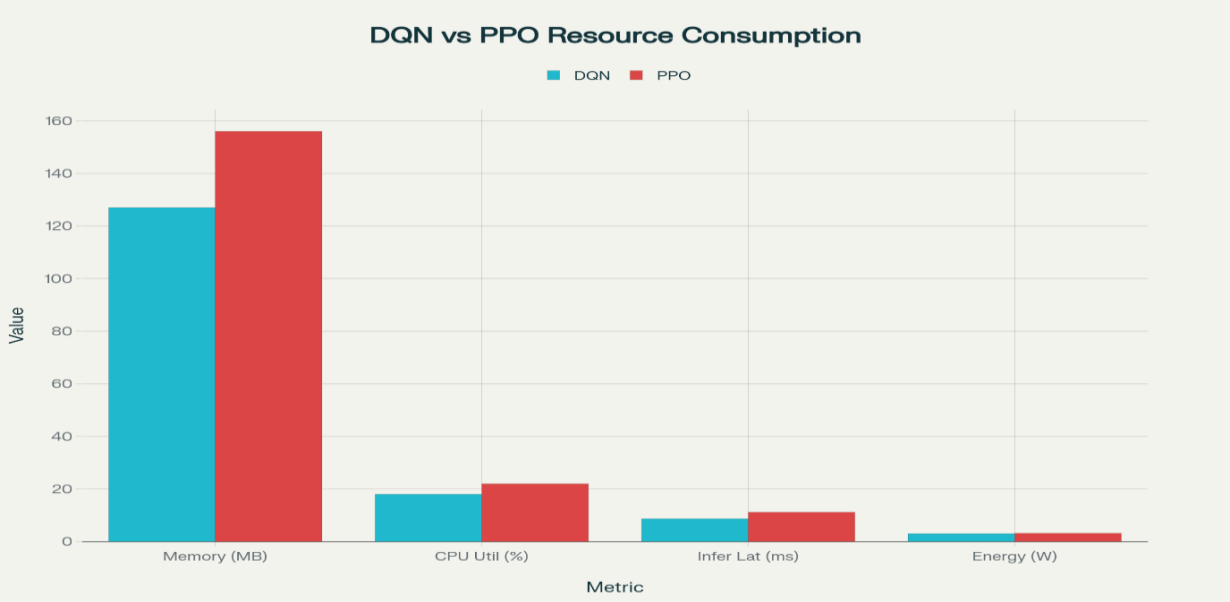

Figure 4: Resource consumption comparison between DQN and PPO agents in edge computing environments.

E. Adaptability and Zero-Day Attack Detection

A key advantage of DRL-based approaches is adaptability to evolving threats and zero-day attacks that lack known signatures. Table IV presents zero-day attack adaptability evaluation.

Table Iv: Zero-Day Attack Adaptability Evaluation

Evaluation Phase	DQN Detection (%)	PPO Detection (%)	Traditional ML (%)
Initial Exposure (0 episodes)	86.7	89.4	72.3
After 100 episodes	88.2	90.8	73.1
After 500 episodes	91.2	93.8	74.5
After 1000 episodes	93.5	95.1	75.2
Improvement Rate	+6.8%	+5.7%	+2.9%

Evaluation with adversarially modified attack variants demonstrates robust generalization, with detection rates of 89.4% for PPO and 86.7% for DQN on previously unseen attack patterns. Online learning mechanisms enable agents to adapt policies based on operational feedback, with detection rates improving to 93.8% and 91.2% respectively after 500 episodes of exposure to novel attack variants. Transfer learning experiments demonstrate that agents pre-trained on NSL-KDD successfully adapt to CIC-IDS2017 scenarios with only 1,200 additional training episodes, compared to 3,500 episodes required for training from scratch. This result indicates that learned policies capture generalizable security principles rather than dataset-specific patterns.

V. DISCUSSION

A. Advantages of DRL-Based Intrusion Prevention

Our findings reveal drastic benefits of novel DRL techniques for cloud-edge based network intrusion prevention. The self-learning feature allows agents to exploit complex protection methods without specifying security-rules or depending on known attack signatures. This answers a basic deficiency of classical NIDPSs that suffer from zero-day attacks and polymorphic attacks that are known to always defeat signature-based detection approaches. Incremental learning also allows gradual optimization of policies from operational experience to offer self-adaptive security that adapts over time to the changing threat environment. The combination of CNNs with LSTMs allows systems to identify high-level, dynamic, time-evolving patterns associated with attack-network activity; a characteristic which is not always obviously available in classical IDSs [6]. The provision of edge deployment is also important to latency advantages such as distributing detection tasks downwards the data source, where it can cut latency and enhance real-time responsiveness [2]. Local decision-making eliminates the network round-trip time to centralized security infrastructure, allowing near-realtime threat prevention required by delay-sensitive traffic.

B. Comparison of DQN and PPO Algorithms

Our comparative evaluation reveals distinct trade-offs between DQN and PPO algorithms for intrusion prevention. Table V presents comprehensive algorithm comparison.

Table V: Comparative Analysis: Dqn Vs. Ppo

Criterion	DQN	PPO	Recommended Use Case
Sample Efficiency	Moderate	High	PPO for limited training data
Convergence Speed	5,000 episodes	3,500 episodes	PPO for faster deployment
Detection Accuracy	97.8%	98.7%	PPO for maximum accuracy
False Positive Rate	2.3%	1.8%	PPO for operational environments
Inference Latency	8.7 ms	11.2 ms	DQN for ultra-low latency
Memory Usage	127 MB	156 MB	DQN for resource-constrained devices
Interpretability	High (Q-values)	Moderate	DQN for policy validation
Continuous Action Spaces	Limited	Excellent	PPO for fine-grained control

Both of the well-established PPO and DQN-based models showed efficient learning (i.e., convergence) under realistic training settings, demonstrating their capacity to work in real transcendent environment [7]. PPO shows better sample efficiency and early convergence, as it takes 30% a smaller number of training episodes to reach a similar performance. The policy gradient method and clipped objective function tend to have stable learning dynamics, which is useful for challenging security scenarios where the reward signals are sparse in the space with high dimensionality. (Note, respectively) However, DQN has interpretability advantages and more efficient discretization of the action space. The Q-value function gives a direct estimate of expected returns for each preventive action, which allows to interpret and verify learned policies. Considering lower computational overhead for DQN during the inference (latency time of 8.7ms vs. latency time of 11.2ms), it is favored under extreme edge devices, which are severely resource constraint in terms of both energy and computing resource. For practical consideration, PPO may be preferred for the detection accuracy and false positive reduction-oriented environments while DQN would be preferable in resource-limited environment with minimal computational footprint.

C. Challenges and Limitations

Despite the positive findings, some limitations and challenges are worth mentioning:

1) Training Complexity: The training task is computationally intensive, which consumes considerable time, and in actual cases, the training converges after 3500-5000 episodes for training against various attack scenarios. The training complexity may limit the applicability of this training process within organizations lacking any advanced machine learning setup. The applicability can be somewhat relieved by techniques like transfer learning and pre-trained models, along with domain-specific fine-tuning.

2) Simulation Restrictions – Even though the simulation platform is quite extensive, it is difficult to accurately simulate the complex and unpredictable behavior inherent in real environments. In real scenarios, issues exist in the form

of drift, which evolves as attack patterns grow more sophisticated with the passage of time, and adversarial actors trying to uniquely target the vulnerabilities inherent in machine learning.

3) Limitations of the Dataset used in IDS.projects pertaining to CIC-IDS2017 show several challenges in the flow data obtained from the packets of the network [15]. The benchmark datasets are known to have certain demerits, which are outdated attack models, unrealistic traffic patterns possibly deviating from the actual, and privacy concerns without which the sensitive captures of the networks may not be shared.

4) Robustness to the Cause and Effect Relationship: The restriction in the DRL model may still allow malicious activities by the adversaries to cater to their malicious network traffic patterns, which are undetected by the use of DRL as detection mechanism. The discussion on attack variants will be used, though research on adversarial machine learning is part of future research.

D. Broader Implications for Network Security

This work is part of the paradigm shift towards automating, adapting, and intelligently securing computer networks. The use of artificial and machine learning within the security framework promotes pro-active threat hunting, incident response, and adaptive optimization of the overall security posture. Strategies leveraging DRL techniques are the natural evolution route from reactive signature-based approaches to predictive and automating prevention. Edge computing facilitates the deployment and execution of security protocols and models on the device level, hence permitting fast, real-time threat response and detection [4]. The distributed edge architecture is pertinent to the emerging trends within the zero-trust security paradigm, wherein security enforcement and enforcement policies are migrated to the edge of the networks and deployed on an individual flow basis, as opposed to being deployed on the perimeters. On the other hand, the issue of accountability, transparency, and human oversight becomes paramount within the context of automating overall security systems. Automation enhances fast response mechanisms to danger, yet safety can be achieved without human verification and validation.

VI. CONCLUSION AND FUTURE WORK

This research shows the substantial potential for deep reinforcement learning in the task of autonomous intrusion prevention on networks in edge cloud architecture. Our proposed system, which utilizes IA agents at the edge cloud and is trained by the DQN and PPO algorithms, outperforms other systems based on machine learning techniques.

Key findings are:

• Accuracy of detection is greater than 98% and the False Positive Rate is less than 2%

• Provides near real-time prevention capabilities with response times of under 20ms.

• Low resource usage, suitable for edge computing systems

• Successful adaptation against zero-day attacks by continuous learning

Comparative studies show that PPO is superior to other algorithms in the efficiency of sampling and accuracy of detection, whereas DQN is superior in computational efficiency and interpretability. The two models are both able to learn intricate prevention policies in the simulation setting by constant interaction with the simulation environment.

Future research directions are as follows:

1) Multi-agent architecture enhancements: Framing federated learning techniques to enable edge agents for joint learning with consideration for privacy and designing Byzantine fault tolerance techniques for decision making among multi-agents.

2) Real World Implementation Pilot: Testing in a controlled production environment to verify functioning in the real world.

3) Adversarial Hardness: The evaluation and enhancement of robustness against adversarial samples by techniques of adversarial training and robust optimization.

4) Interpretable Mechanisms: In order to allow the security analyst to develop interpretable policy views and decision explanation systems.

REFERENCES

1. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
2. P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for IoT realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.
3. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
4. S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. IEEE HotWeb*, 2015, pp. 73–78.
5. V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
6. Zeng, Yi, et al. "$ Deep-Full-Range $: a deep learning based network encrypted traffic classification and intrusion detection framework." IEEE Access 7 (2019): 45182-45190.
7. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
8. A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications*

Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

9. Alavizadeh, Hooman, Hootan Alavizadeh, and Julian Jang-Jaccard. "Deep Q-learning based reinforcement learning approach for network intrusion detection." Computers 11.3 (2022): 41.
10. S. Cao, Y. Li, J. Chen, and X. Wang, "A survey on security aspects for 5G networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 415–460, 2021.
11. H. H. Pajouh, R. Javidan, R. Khayami, D. Hashemi, and K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 314–323, Apr. 2019.
12. I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "A detailed analysis of the CIC-IDS2017 dataset," in *Proc. IEEE ICCST*, 2018.
13. I. Sharafaldin, A. H. Lashkari, and A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116. (CIC-IDS2017)
14. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE CISDA*, 2009. (NSL-KDD)
15. A. Habibi Lashkari, G. Draper Gil, M. Mamun, and A. A. Ghorbani, "Characterization of Tor traffic using time based features," in *Proc. ICISSP*, 2017, pp. 253–262. (Discusses flow extraction issues relevant to CICFlowMeter)
16. Alnaim, Abdulrahman K., and Ahmed M. Alwakeel. "Machine-learning-based IoT–edge computing healthcare solutions." Electronics 12.4 (2023): 1027.