JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Adaptive Cyber Defense using Advanced Deep Reinforcement Learning Algorithms: A Real-Time Comparative Analysis

Atheer Alaa Hammad [1,*] and Firas Tarik Jasim [2]

[1] Ministry of Education, Anbar Education Directorate, Al Anbar, Iraq; e-mail : atheer.alaa@ec.edu.iq
[2] Department of Medical Instrumentation Technique, Northern Technical University, Al-Dour Technical Institute, Iraq; e-mail : firas.tj@ntu.edu.iq
* Corresponding Author : Atheer Alaa Hammad

Abstract: Cybersecurity is continuously challenged by increasingly sophisticated and dynamic cyber-attacks, necessitating advanced adaptive defense mechanisms. Deep Reinforcement Learning (DRL) has emerged as a promising approach, offering significant advantages over traditional intrusion detection methods through real-time adaptability and self-learning capabilities. This paper presents an advanced adaptive cybersecurity framework utilizing five prominent DRL algorithms: Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Twin Delayed DDPG (TD3), Soft Actor-Critic (SAC), and Asynchronous Advantage Actor-Critic (A3C). The effectiveness of these algorithms is evaluated within complex, realistic simulation environments using live-streaming data, emphasizing key metrics such as accuracy (AUC-ROC), response latency, and network throughput. Experimental results demonstrate that the SAC algorithm consistently achieves superior detection accuracy (95% AUC-ROC) and minimal disruption to network performance compared to other approaches. Additionally, A3C provides the fastest response times suitable for real-time defense scenarios. This comprehensive comparative analysis addresses critical research gaps by integrating traditional and novel DRL techniques and substantially validates their potential to improve cybersecurity defense strategies in realistic operational settings.

Keywords: A3C; Adaptive Defense; Cybersecurity; Deep Reinforcement Learning; Intrusion Detection Systems; Network Security; SAC.

1. Introduction

The field of cybersecurity is witnessing a continuous increase in the complexity and sophistication of digital attacks, necessitating highly adaptive defensive approaches capable of responding to new threats in real-time. To address this, machine learning (ML) [1]–[3] and deep learning (DL)[4], [5] approaches have been widely applied in intrusion detection. While these models perform well on known attacks, they often struggle to generalize to new or evolving threats and typically require retraining with updated data.

Deep reinforcement learning (DRL) has emerged as one of the most promising directions for building cybersecurity systems capable of self-learning and making optimal decisions during attacks[6]. Reinforcement learning techniques are distinguished by their ability to allow intelligent agents to interact directly with the network environment, enabling the discovery of new defense strategies that outperform fixed rules[7]–[10] or in this case traditional intrusion detection models. For instance, recent studies have shown that traditional ML models trained on historical data suffer from performance degradation when facing unfamiliar attacks. In contrast, reinforcement learning agents improve through direct interaction with dynamic environments, giving them better generalization and adaptability.

Despite the initial successes of applying DRL in cybersecurity, there remains a gap in leveraging the latest algorithms in this field[11]. Recent literature reviews have indicated that many advanced DRL techniques have not yet been fully utilized in modern intrusion detec-

Received: March, 25th 2025
Revised: April, 9th 2025
Accepted: April, 15th 2025
Published: April, 23rd 2025

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) licenses (https://creativecommons.org/licenses/by/4.0/)

tion systems. Although, most prior attempts relied on well-known algorithms such as Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Twin Delayed DDPG (TD3), which have shown success in classifying and responding to cyber threats[6]. In contrast, newer algorithms such as Soft Actor-Critic (SAC) and Asynchronous Advantage Actor-Critic (A3C) that combine entropy regularization mechanisms and asynchronous parallel training can improve exploration capabilities and reduce convergence problems, resulting in more robust performance in complex environments[12], [13]. Therefore, there is a need to study the integration of these advanced algorithms in adaptive cyber defense contexts, particularly after identifying each approach's relative strengths and weaknesses through performance analysis, to understand their potential improvements over traditional methods better. Research [14] focuses on applying the DQN algorithm for real-time cyber threat detection, demonstrating its effectiveness in reducing false positives compared to traditional methods. Research [11] presents a comparative analysis of DRL algorithms in intrusion detection systems, recommending SAC and PPO for their balanced accuracy and response time performance.

This study aims to conduct a comprehensive comparative evaluation of adaptive cybersecurity defense mechanisms based on DRL through the following:

- Implementing and comparing the latest reinforcement learning algorithms (e.g., SAC and A3C) alongside previously used algorithms (DQN, PPO, and TD3) within a unified experimental setup, enabling a comprehensive performance comparison across diverse cyber threat scenarios;
- Expanding the experimental setup to include more complex network simulation environments that closely resemble real-world conditions, using live streaming network data to test each algorithm's real-time adaptability; and
- Evaluation metrics such as the Area Under the ROC Curve (AUC-ROC) are adopted to measure classification accuracy and latency, assess reaction speed and throughput, and evaluate defence actions' impact on network performance and data flow.

The remainder of this paper is organized as follows: Section 2 reviews recent DRL approaches in cybersecurity and related work; Section 3 describes the methodology, including DRL algorithm design and simulation environment; Section 4 presents the experimental results and discusses findings and implications; and Section 6 concludes the study and outlines directions for future research.

2. Literature Review

In recent years, several research efforts have been aimed at utilizing DRL in various network security domains, including intrusion detection systems (IDS) and proactive cyber defense. Below, we highlight the most notable trends from 2020 to 2024 and how they inform our work.

Many recent studies have focused on employing reinforcement learning algorithms to build more adaptive IDS solutions. In this context, Yang et al. [15] presented a comprehensive survey of DRL applications in IDS. The review demonstrated that some DRL models have outperformed traditional deep learning methods on standard benchmark datasets. However, it also noted that many of the latest sample-efficient DRL algorithms remain underexplored in this domain.

One practical example is the study by Badr et al. [16], which demonstrated the feasibility of using the Dueling Double Deep Q-Network (D3QN) architecture to build a self-learning IDS. Their study compared the DRL approach against traditional supervised models using datasets like KDD Cup 99 and ISCX 2012 and found that traditional models struggled with generalization over time due to model drift. In contrast, the DRL model overcame this by continuously learning through interaction, reducing the need for frequent retraining and allowing better adaptation to new threats.

Additionally, some works highlighted the potential of reinforcement learning in improving feature selection for IDS rather than relying on all available features, which may include irrelevant data. For example, proposed a model where a DRL agent learns a policy to select an optimal subset of network features before passing them to a traditional classifier. This approach improved detection accuracy and reduced computational complexity, integrating algorithms like KNN and SVM alongside DRL in a unified framework. Overall, these

studies support the strong potential of DRL in enhancing IDS capabilities, particularly when agents learn progressively from successful and failed detection attempts.

Beyond intrusion detection, another line of research focuses on deploying DRL agents to take automated defensive actions upon detecting attacks in what is known as adaptive or moving target defense (MTD). The idea is that an intelligent agent continually changes system states or configurations to make it harder for attackers to succeed. For instance, Water et al. [17] used Microsoft's CyberBattleSim to train a defense agent that deploys honeypots as a form of dynamic deception. They showed that changing deception strategies reduced the effectiveness of automated attacks and diverted attackers to decoy environments. Similarly, Zennaro and Erdődi [18] proposed a DRL approach to simulate Capture-the-Flag attack scenarios, where the defense agent learned to randomly perform port hopping on a server to disrupt attacker strategies. Although these strategies added complexity to the training environment, they significantly increased the failure rate of attackers.

Another important development is using multi-agent game models, where both attacker and defender agents learn simultaneously. Xiong et al. [19] modeled the attacker-defender interaction as a multi-player Markov game, combining game theory (Stackelberg equilibrium) with multi-agent reinforcement learning (e.g., WoLF algorithm) to reach a strategic balance. Their experiments showed that the defender agent could arrive at a dynamic optimal strategy that outperformed traditional Markov models like Nash-Q in maximizing its protection reward.

Furthermore, some recent studies have included Quality of Service (QoS) factors in defense decision-making. Lei et al. [20] demonstrated that it is possible to design an MTD defense policy that balances security benefits (e.g., reducing damage and preventing intrusions) with network service quality (such as data rates and transmission delays). This inspired other researchers to propose evaluating defense agents not only by detection accuracy but also by real-time performance metrics like throughput and latency—an approach we also adopt in this research to ensure both security and operational efficiency.

The aforementioned research employed a range of DRL algorithms, emphasizing specific types. The DQN algorithm and its enhancements (Double-DQN, Dueling DQN) were widely used in environments with discrete defensive actions due to their ability to learn effective policies in large state spaces using neural networks[6]. Policy-based algorithms like PPO and A3C were notable for their training stability and consistent performance. For instance, Muhati and Rawat [21] applied A3C in cognitive network security, allowing agents to learn asynchronously across multiple sub-environments and accelerate training while maintaining stability.

Other researchers favored off-policy actor-critic algorithms for continuous action spaces and sample efficiency. Notable among these were DDPG and its improved TD3. For example, [22] developed a TD3-based model (TD3-AP) to prioritize IDS alerts and reduce false positives, enhancing analyst efficiency. Recently, SAC emerged as one of the most powerful off-policy algorithms due to its ability to balance exploration and exploitation via entropy maximization [11]. Research [23] also introduced the SAC-AP model for alert prioritization, achieving up to 30% reduction in defense losses compared to traditional DDPG. These results show that SAC improved the agent's ability to identify and respond to critical alerts, reducing the burden on human analysts and improving system readiness [24], [25].

Based on these studies, DRL has shown strong potential in cybersecurity. However, most works focus on evaluating individual algorithms and often rely on static datasets, making it difficult to assess performance in real-time settings under similar conditions [26], [27][. This study addresses these gaps by comparing five DRL algorithms under the same experimental setup: DQN, PPO, TD3, A3C, and SAC. The evaluation uses simulated and live-streaming network data, with performance assessed through detection accuracy, response time, and network impact.

3. Methodology

In this section, we describe the proposed approach for adaptive cybersecurity defense based on DRL, including the formulation of the learning problem, simulation environment, learning components states, actions, rewards), applied algorithms, training settings, and evaluation metrics.

3.1. Framing Adaptive Defense as a Reinforcement Learning Problem

The network security problem was formulated as a Markov Decision Process (MDP), in which a defense agent was designed to interact with the network environment at sequential time steps. At each step, the environment provided a state s_t, representing a snapshot of the network's security status, including the condition of devices and servers, e.g., compromised or safe), traffic levels, and alerts from intrusion detection systems.

A set of defensive actions $A(s_t)$ was associated with each state, from which the agent selected an action based on its current policy π. These actions included resetting firewall rules, isolating suspicious devices, deploying honeypots, updating access policies, or requesting manual inspection. The action space was designed to support discrete and continuous types of actions, enabling different DRL algorithms. For example, toggling firewall rules, e.g., turning ports and services on or off) was categorized as discrete, while adjusting IDS sensitivity or data transfer limits were treated as continuous.

Once an action was executed, the environment transitioned to a new state s_{t+1} nd returned a reward r_t, indicating the effectiveness of the action taken. The reward function was constructed to promote rapid and accurate threat mitigation. A high positive reward was assigned for successfully preventing or containing attacks, e.g., isolating an infected device before malware propagation). At the same time, penalties were given for missed attacks or false alarms that caused service disruption. Action costs were also taken into account—though beneficial to security, certain actions could degrade system performance, e.g., disconnecting a critical server). Thus, rewards were adjusted to encourage a trade-off between protection and availability.

The agent's objective was to learn an optimal policy π^* that maximized the expected cumulative reward over time. To achieve this, value-based and policy-based reinforcement learning approaches were applied by implementing and comparing five DRL algorithms: DQN, PPO, TD3, A3C, and SAC.

For DQN, enhancements such as Double DQN and Dueling DQN were incorporated to improve stability and mitigate Q-value overestimation. Experience replay was employed to store and sample past transitions, which improved training independence.

For PPO, an on-policy algorithm, clipped surrogate loss functions were used to maintain training stability and prevent abrupt policy changes. A3C was implemented in a multi-threaded configuration, allowing experiences to be collected simultaneously from multiple simulated environments and enabling more diverse exploration.

On the off-policy side, TD3 was chosen to handle continuous action spaces. It was preferred over DDPG due to its ability to reduce update variance through twin Q-networks and smoothen target policy. The simulation environment included continuous control elements e.g., adjusting IDS alert thresholds between 0 and 1) to evaluate TD3's effectiveness.

SAC was also integrated as a recent and powerful actor-critic algorithm, selected for its sample efficiency and exploration capabilities through entropy maximization. SAC was designed to optimize immediate rewards and maintain controlled stochasticity in policy selection, reducing the chance of converging to suboptimal strategies.

The SAC implementation used two Q-networks and a stochastic policy network, trained using adaptive entropy-weighted gradient descent. The neural network architecture across all agents was kept consistent to ensure fair comparisons. Each agent utilized a multi-layer neural network to approximate the policy or value function from approximately 100 encoded input features representing the network state.

3.2. Simulation Environment and Data Used

A custom simulation environment was developed to emulate a medium-scale enterprise network of multiple interconnected nodes linked through routers and switches to fulfill the research objectives, including PCs, database servers, and web servers. The attack scenarios were implemented using a flexible simulation platform inspired by well-known open-source frameworks such as CyberBattleSim and CybORG.

A virtual attacker agent was incorporated and programmed with multiple intrusion strategies. These simulated attacks included malware propagation, distributed denial-of-service (DDoS) attacks on core servers, phishing attempts through malicious emails, port-based intrusions, and advanced persistent threats (APTs) involving multi-stage infiltration techniques.

Natural background traffic, such as web browsing and file transfers, was generated concurrently with attack activities to increase realism. This setup required the defense agent to accurately identify malicious behavior embedded within legitimate network flows, enhancing the complexity of the detection task.

In addition to the simulation environment, semi-live data from real network recordings was used to evaluate the model. Specifically, the CIC-IDS2018 dataset was employed, containing labelled benign and malicious traffic instances. The dataset was streamed continuously to the agent in real-time, simulating an operational environment. During this test, the agent was required to process each incoming flow or packet as it arrived and to make immediate decisions, e.g., labeling a session as malicious for isolation), without prior knowledge of future events. This setup thoroughly evaluated the agent's real-time responsiveness and adaptability.

3.3. Training Settings and Parameters

Each DRL agent was trained over a sufficient number of episodes to ensure the convergence of a stable policy. The simulation was organized into episodes, where each episode represented one full day of network activity, including a predefined number of randomized attack events.

A discount factor $\gamma = 0.99$ was used to emphasize long-term rewards, which was particularly important in handling advanced persistent threats (APT), where the consequences of an action may unfold over multiple time steps. For the DQN algorithm, the initial exploration rate was set to $\varepsilon = 0.2$, and was gradually decayed throughout training. In contrast, policy-based methods such as SAC utilized entropy regularization or guided noise to promote exploration.

The learning rate for all neural network models was fixed at 10^{-4}, based on results from preliminary experiments. To account for the differences in data usage strategies, off-policy algorithms (DQN, TD3, SAC) were trained for up to 500,000 interactions, benefiting from experience replay mechanisms. Meanwhile, on-policy algorithms (PPO, A3C) require more training episodes approximately 1,000) due to discarding outdated experiences after each policy update.

All DRL models were implemented using the Stable Baselines library, with customized modifications to adapt them for cybersecurity contexts, including tailored reward functions and the simulation of packet-level delays.

Throughout the training phase, detailed logs were maintained for each agent, capturing key performance indicators such as attack detection rates, false positive rates, decision latency per event, and network throughput stability during defense operations.

3.4. Evaluation Metrics

After training, the agent policies were frozen and tested on new simulation scenarios and live-streamed data. The performance of each DRL agent was evaluated using three categories of metrics to capture both security effectiveness and operational efficiency:

- Detection Accuracy and False Positive Rate: Evaluation was based on confusion matrices, from which the True Positive Rate (TPR) and False Positive Rate (FPR) were derived. Additionally, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was used as a global metric of classification quality, where values approaching 1.0 indicated strong separation between attack and normal traffic. During live-streaming evaluations, where traffic patterns were more dynamic and class imbalance was expected, Precision and Recall were also used to assess the model's ability to maintain high detection quality while minimizing false alarms.
- Latency Response (Time): Latency was defined as the time elapsed from attack initiation to the agent's first effective defensive action. This was especially important for fast-moving attacks such as malware propagation or DDoS. Real-time readiness was characterized by response times in the order of seconds or less.
- Throughput: Throughput was measured as the rate of legitimate data successfully transmitted during defense (in Mbps). The comparison was made between defended and undefended network states. A defense mechanism was considered acceptable if throughput degradation remained below 5%. Instances of false packet drops were also logged as part of the performance audit.

3.5. Summary of Algorithm Characteristics

To support fair evaluation, each algorithm was selected to represent a different class of reinforcement learning paradigm, covering both discrete and continuous control settings. Table 1 compares algorithm types and strengths relevant to discrete and continuous control in cybersecurity environments.

Table 1. Summary of DRL Algorithm Characteristics Used in the Evaluation.

Algorithm	Type	Action Space	Strengths
DQN	Off-policy, value-based	Discrete	Simple and robust; suited for rule toggling
PPO	On-policy, policy-gradient	Discrete/Cont.	Stable training; moderate compute efficiency
TD3	Off-policy, actor-critic	Continuous	Reduces update variance; suitable for tuning
A3C	On-policy, asynchronous	Discrete/Cont.	Fast response; enables parallel learning
SAC	Off-policy, entropy-based	Continuous	Strong exploration; high-dimensional support

This evaluation framework ensures that each algorithm is tested under realistic and dynamic conditions, with results interpreted across effectiveness and operational impact dimensions. The general DRL workflow evaluated in this study is depicted in Figure 1, where agent actions are guided by learned policies and evaluated through feedback from the environment.

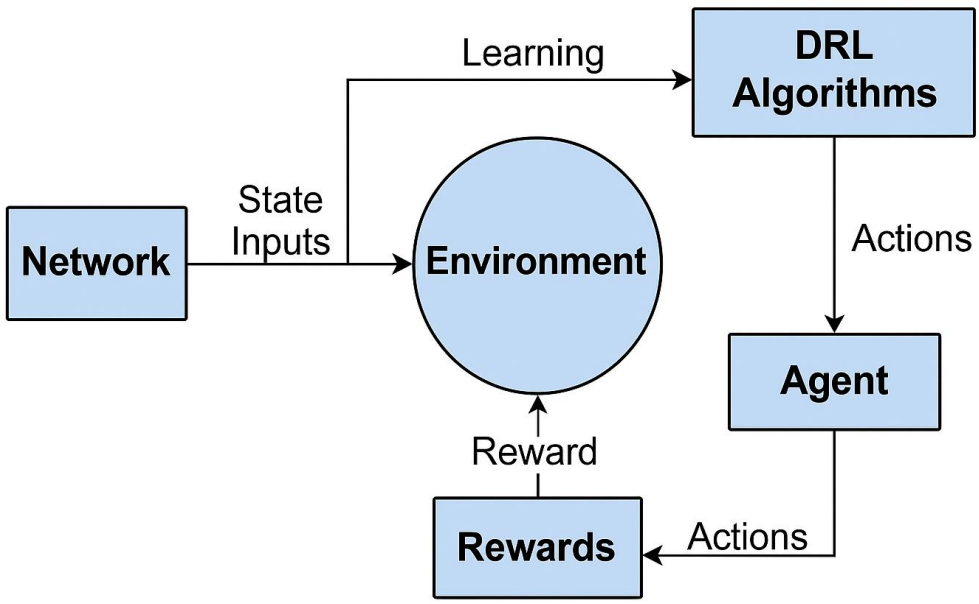

Figure 1. Workflow of a DRL-based intrusion detection system used for evaluation.

4. Results and Discussion

4.1. Results

After completing the training phase and confirming policy convergence for all DRL agents, a series of evaluation experiments were conducted in environments not seen during training. These included compound attack scenarios where multiple attack types occurred simultaneously. The following subsections summarize the most notable findings: security performance, operational efficiency, and live data streaming evaluations.

4.1.1. Attack Detection and Mitigation Performance

The ability of each agent to detect and mitigate threats was evaluated using classification metrics. It was found that the SAC algorithm consistently outperformed others. SAC

achieved the highest AUC-ROC score of 0.95, indicating its superior ability to distinguish malicious behavior under varying thresholds. This is illustrated in Figure 2.

SAC also maintained a TPR of approximately 0.90 for malware and 0.85 for DDoS, with an average FPR below 0.05. TD3 followed closely with an AUC of approximately 0.92, especially excelling in scenarios requiring continuous actions such as traffic regulation.

PPO achieved a solid AUC of 0.89, showing limitations in handling multi-stage attacks. A3C recorded an AUC of 0.88, offering fast reaction times but suffering slightly due to its on-policy nature. DQN performed lowest with an AUC of 0.85, although this still surpassed traditional methods typically between 0.70–0.80). DQN struggled with continuous or high-dimensional action spaces but benefitted from Double and Dueling extensions that improved feature relevance filtering. These findings are reflected in Figure 3, which compares the classification accuracy of the five algorithms. SAC achieved the highest accuracy (95%), followed by TD3 (92%) and PPO (89%), while DQN remained lowest at 85%.

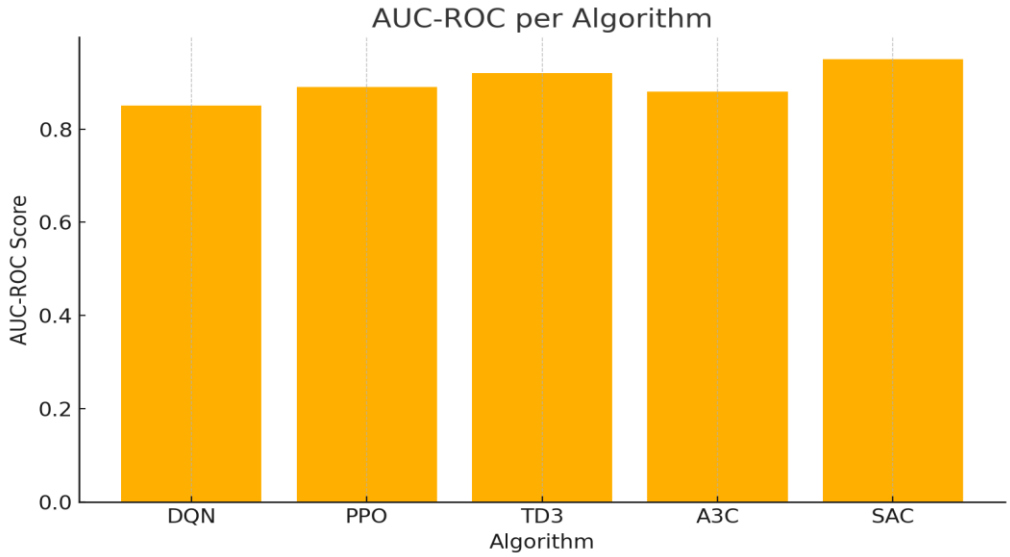

Figure 2. AUC-ROC per algorithm, reflecting classification effectiveness in distinguishing malicious vs. normal traffic.

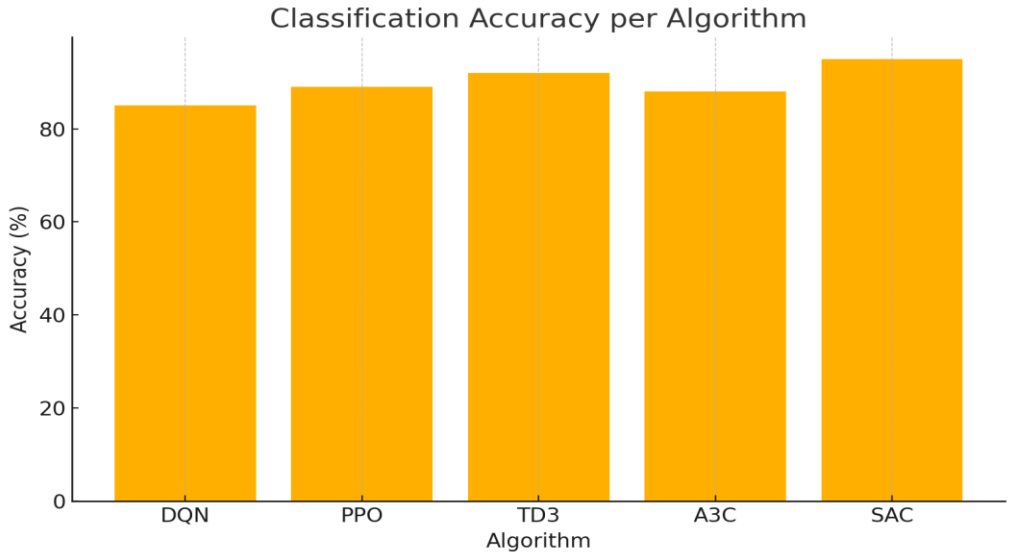

Figure 3. Classification accuracy per algorithm.

Regarding cumulative rewards, SAC consistently achieved the highest average reward per episode, reflecting early threat interception with minimal cost. TD3 and PPO recorded moderate to high rewards. A3C performed well in short episodes but struggled with longer

sequences due to frequent policy overwrites. DQN earned the lowest rewards, indicating less efficient adaptation. The reward trajectories over training episodes are visualized in Figure 4.

4.1.2. Response Time and Operational Metrics

The average response time (latency) per algorithm is shown in Figure 5. The A3C agent responded fastest, with an average latency of 1.8 seconds, benefiting from asynchronous parallel learning. SAC followed with a response time of 2.5 seconds, maintaining real-time usability. PPO and TD3 exhibited average response times of 3 seconds, while DQN lagged with an average of 4 seconds, especially when handling compound attacks. The average traffic reduction during defense operations was minimal in terms of throughput impact. SAC produced the lowest throughput drop ~3%, as illustrated in Figure 6. PPO and A3C followed with ~4%, while TD3 caused about 5% due to global rate-limiting actions. DQN was the most disruptive, occasionally dropping throughput by up to 6% due to drastic countermeasures like component resets.

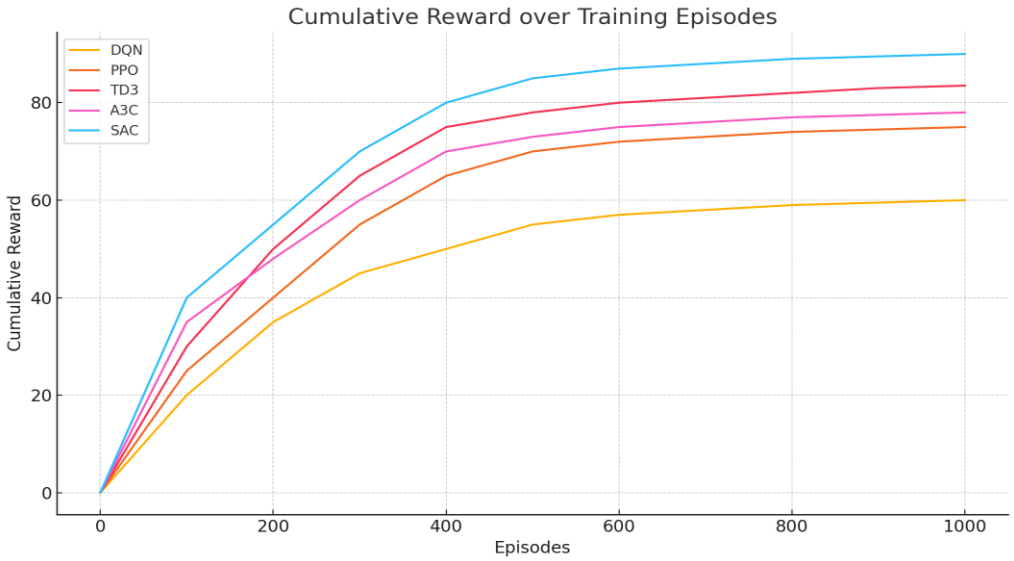

Figure 4. Cumulative reward over training episodes, showing learning progression and policy efficiency.

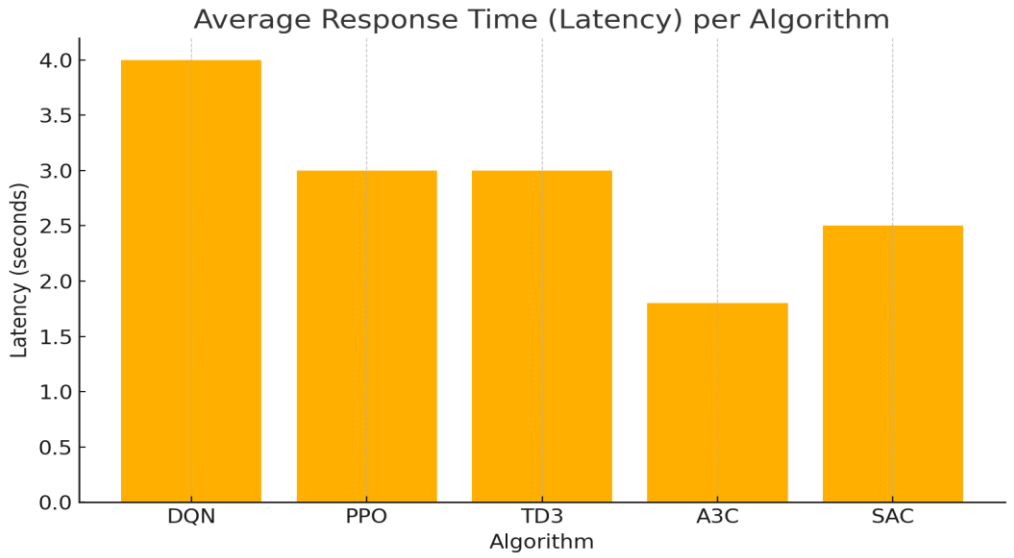

Figure 5. Average response time (latency) for each algorithm.

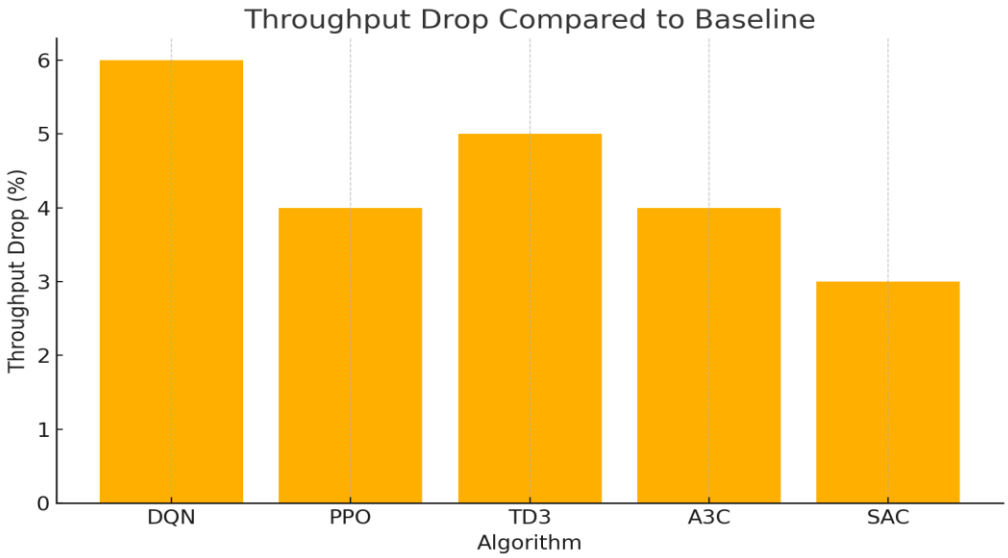

Figure 6. The throughput drop percentage was compared to the baseline (no defense).

Latency added by defense mechanisms remained under 10 milliseconds per packet, indicating negligible performance impact.

4.1.3. Performance on Live Data Streaming

All DRL agents were tested on live-streaming network traffic from the CIC-IDS2018 dataset to evaluate adaptability in realistic settings. During these evaluations, SAC and PPO demonstrated strong generalization capabilities and maintained low FPR, even when confronted with threats not encountered during training. For instance, during a stealthy port scan attack—absent from the training data—the SAC agent identified anomalous behavior within minutes and responded appropriately by blocking suspicious IPs and issuing alerts, effectively containing the threat before it escalated. This behavior highlights the agents' ability to detect subtle deviations and respond to evolving patterns in real-time.

A slight reduction in TPR was observed under high-traffic conditions combined with stealthy attacks, likely due to occasional packet drops under congestion. However, the system maintained stable operation throughout the test. Resource usage remained within acceptable bounds, with CPU utilization peaking around 60% and memory consumption below 70%.

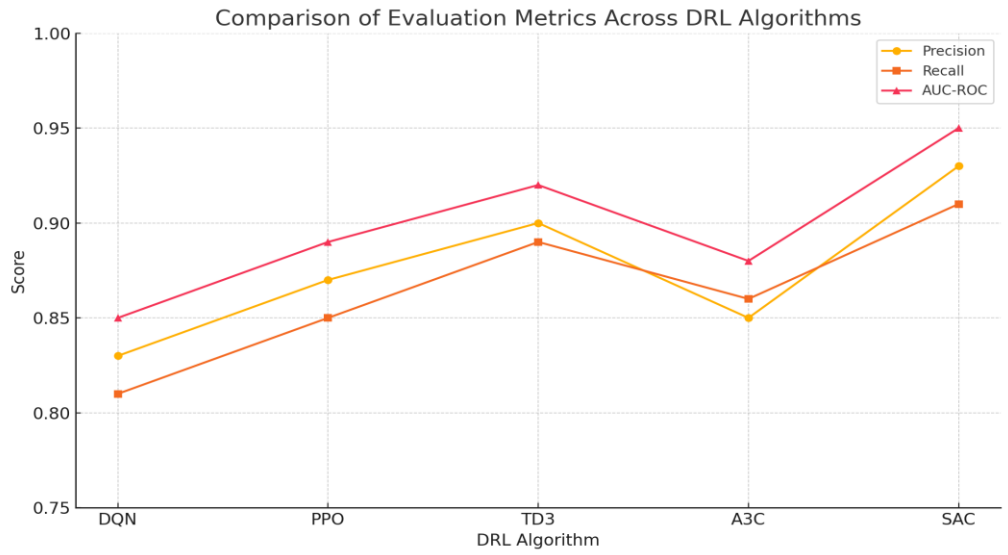

Figure 7. Comparative performance of DRL algorithms on live-streamed data based on Precision, Recall, and AUC-ROC.

A comparative evaluation using Precision, Recall, and AUC-ROC was conducted to capture detection performance under live conditions, often involving imbalanced and dynamic traffic distributions. As shown in Figure 7, SAC again outperformed the other algorithms, achieving a Precision of 0.93, Recall of 0.91, and an AUC-ROC of 0.95. These results indicate both accuracy in detection and consistency across thresholds. Conversely, DQN recorded the lowest performance across all three metrics, reflecting its difficulty in handling novel or complex traffic patterns.

With these findings, SAC can be considered the most reliable and adaptive algorithm among those tested, followed by TD3 and PPO. Although DQN was the least effective in this setting, it outperformed many traditional approaches. Overall, the agents demonstrated a strong balance between detection performance, response speed, and operational efficiency, reinforcing the applicability of DRL-based solutions for real-time intrusion detection.

4.2. Discussion

The evaluation results have confirmed the effectiveness of comparing modern and traditional reinforcement learning algorithms in adaptive cybersecurity defense. Several key insights were identified, which are discussed below.

4.2.1 Impact of Algorithm Choice on Performance

It was observed that the selection of the reinforcement learning algorithm significantly influenced the defense agent's performance. The SAC algorithm consistently outperformed the others across most metrics. Its integration of entropy maximization with off-policy learning enabled broader exploration while preserving long-term learning stability. This contributed to SAC's ability to balance effective threat detection with minimal disruption to network operations. In contrast, DQN encountered limitations due to its value-based architecture and reduced generalization capacity in high-dimensional state-action spaces. These findings are consistent with prior research, which indicated that value-based methods typically require extensive interaction to perform effectively in complex environments.

Algorithms such as PPO and A3C also yielded strong results. PPO benefited from stable policy updates, while A3C's asynchronous training enhanced responsiveness, particularly in fast-changing scenarios. However, A3C's performance declined in long-horizon episodes where consistent policy retention was needed. TD3 achieved competitive outcomes, likely due to its enhancements over DDPG—including twin Q-networks and policy smoothing—which improved learning stability. These observations affirm that algorithm selection should be tailored to the problem context: environments requiring rapid responses may favor PPO or A3C, while those involving continuous control benefit more from SAC or TD3.

4.2.2. Adaptability to Unseen Threats and Novel Environments

All DRL agents, particularly SAC and PPO, demonstrated adaptability to unfamiliar attack patterns during live-streaming evaluations. These agents could generalize to previously unseen or modified threats by learning behavioral patterns instead of relying solely on static attack signatures.

This behavior underscores the advantage of DRL over traditional intrusion detection systems, which require periodic retraining to remain effective. As reported by Yang et al. [15], the generalization capability of DRL is critical in addressing zero-day threats, and the conducted experiments support this recommendation by validating the model in realistic, evolving threat environments.

4.2.3 Response Time and Operational Viability

The trained DRL agents were observed to operate within real-time constraints, with average response times of just a few seconds and negligible performance degradation in network throughput. The slight decline in data transfer rate (~3–5%) was considered acceptable given the security benefits provided.

These findings emphasize the practical feasibility of deploying DRL agents in production settings. Further improvements in latency may be achieved through model optimization or specialized hardware acceleration, such as FPGAs. Nevertheless, the agents exhibited reliable real-time behavior even when evaluated on general-purpose systems. It should also be noted that some prior studies have overlooked operational metrics such as latency and throughput. In contrast, the current study incorporated these metrics as part of the evaluation, aligning with practical requirements for intrusion detection deployment.

4.2.4 Challenges and Future Work

Despite encouraging outcomes, several challenges remain. A primary limitation concerns the interpretability of agent decisions. Like most deep learning systems, DRL agents function based on internal representations that are difficult to explain. This may hinder trust and acceptance by human operators, especially when actions result in service disruptions. Future work could explore Explainable Reinforcement Learning (XRL) to provide interpretable insights into agent decision-making processes.

This study did not address security concerns regarding DRL itself, but they remain an open issue. Recent literature has shown that DRL agents may be vulnerable to adversarial manipulations, and future research should consider this risk before real-world deployment. Moreover, the current single-agent framework could be extended into a multi-agent setting, where attacker and defender agents co-evolve. This would represent a more realistic adversarial environment and could benefit from adversarial reinforcement learning to improve robustness.

Lastly, integration with generative deep learning models (e.g., GANs or Transformers) may allow the simulation of rare or complex attacks during training and enhance the diversity of learned defense strategies. This direction, also suggested by Yang et al. [15] holds promise for both academic research and practical application.

5. Conclusions

This study presented a comprehensive comparative evaluation of adaptive cybersecurity defense mechanisms using DRL. The research assessed each agent's strengths under dynamic and realistic network conditions by integrating advanced algorithms such as SAC and A3C alongside established methods like DQN, PPO, and TD3 within a unified framework.

Experimental findings demonstrated that SAC consistently delivered superior performance across key metrics, achieving an AUC-ROC of 0.95, high precision, and minimal impact on system throughput. A3C showed the fastest response time, while TD3 and PPO offered balanced detection accuracy and training stability. These results confirm the feasibility of deploying DRL-based agents for real-time defense, providing a strong balance between detection accuracy, response speed, and service continuity.

The study addressed a critical gap in existing research regarding the limited application of recent DRL algorithms in complex and evolving environments. With the expansion of algorithm diversity, the use of more realistic test environments, and the adoption of a multi-metric evaluation framework, the research offers validated insights into the practicality and effectiveness of DRL in cybersecurity. This contribution supports the development of intelligent and self-adaptive intrusion detection systems suitable for real-world deployment.

Future research may explore multi-agent architectures, improve model interpretability through explainable reinforcement learning, and strengthen resilience against adversarial manipulation. Additionally, integration with generative models or deployment in large-scale infrastructures such as enterprise networks or cloud systems may further enhance the capability of DRL-based cybersecurity defense.

Author Contribution: Conceptualization: A.A.H. and F.T.J.; Methodology: A.A.H.; Software: A.A.H.; Validation: A.A.H. and F.T.J.; Formal analysis: A.A.H.; Investigation: A.A.H.; Resources: A.A.H.; Data curation: A.A.H.; Writing—original draft preparation: A.A.H.; Writing—review and editing: A.A.H. and F.T.J.; Visualization: A.A.H.; Supervision: F.T.J.; Project administration: A.A.H.; Funding acquisition: F.T.J.; All authors have read and agreed to the published version of the manuscript

Funding: Please add: This research received no external funding.

Data Availability Statement: The data used in this study include publicly available datasets, such as CIC-IDS2018, which can be accessed at: https://www.unb.ca/cic/datasets/ids-2018.html.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] A. Çetin and S. Öztürk, "Comprehensive Exploration of Ensemble Machine Learning Techniques for IoT Cybersecurity Across Multi-Class and Binary Classification Tasks," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 4, pp. 371–384, Feb. 2025, doi: 10.62411/faith.3048-3719-51.

[2] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. El Makhtoum, "OMIC: A Bagging-Based Ensemble Learning Framework for Large-Scale IoT Intrusion Detection," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 4, pp. 401–416, Feb. 2025, doi: 10.62411/faith.3048-3719-63.

[3] C. S. Htwe, Z. T. T. Myint, and Y. M. Thant, "IoT Security Using Machine Learning Methods with Features Correlation," *J. Comput. Theor. Appl.*, vol. 2, no. 2, pp. 151–163, Aug. 2024, doi: 10.62411/jcta.11179.

[4] M. A. Rahman, G. A. Francia, and H. Shahriar, "Leveraging GANs for Synthetic Data Generation to Improve Intrusion Detection Systems," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 4, pp. 429–439, Feb. 2025, doi: 10.62411/faith.3048-3719-52.

[5] P. H. Hussan and S. M. Mangj, "BERTPHIURL : A Teacher-Student Learning Approach Using DistilRoBERTa and RoBERTa for Detecting Phishing Cyber URLs," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 4, 2025, doi: 10.62411/faith.3048-3719-71.

[6] A. Alaa Hammad, M. Adnan Falih, S. Ali Abd, and A. Rashid Ahmed, "Detecting Cyber Threats in IoT Networks: A Machine Learning Approach," *Int. J. Comput. Digit. Syst.*, vol. 17, no. 1, pp. 1–25, Jan. 2025, doi: 10.12785/ijcds/1571020041.

[7] N. Khlif, N. Khraief, and S. Belghith, "Comparative Analysis of Modified Q-Learning and DQN for Autonomous Robot Navigation," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 3, pp. 296–308, Dec. 2024, doi: 10.62411/faith.3048-3719-49.

[8] M. A. Setiawan, D. R. I. M. Setiadi, E. Z. Astuti, T. Sutojo, and N. A. Setiyanto, "Exploring Deep Q-Network for Autonomous Driving Simulation Across Different Driving Modes," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 3, pp. 217–227, Oct. 2024, doi: 10.62411/faith.3048-3719-31.

[9] S. Nugroho, D. R. I. M. Setiadi, and H. M. M. Islam, "Exploring DQN-Based Reinforcement Learning in Autonomous Highway Navigation Performance Under High-Traffic Conditions," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 274–286, Feb. 2024, doi: 10.62411/jcta.9929.

[10] A. Pathirana *et al.*, "A Reinforcement Learning-Based Approach for Promoting Mental Health Using Multimodal Emotion Recognition," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 124–142, Sep. 2024, doi: 10.62411/faith.2024-22.

[11] A. A. Hammad, S. R. Ahmed, M. K. Abdul-Hussein, M. R. Ahmed, D. A. Majeed, and S. Algburi, "Deep Reinforcement Learning for Adaptive Cyber Defense in Network Security," in *Proceedings of the Cognitive Models and Artificial Intelligence Conference*, May 2024, pp. 292–297. doi: 10.1145/3660853.3660930.

[12] H. Shen, K. Zhang, M. Hong, and T. Chen, "Towards Understanding Asynchronous Advantage Actor-Critic: Convergence and Linear Speedup," *IEEE Trans. Signal Process.*, vol. 71, pp. 2579–2594, 2023, doi: 10.1109/TSP.2023.3268475.

[13] J. Adamczyk, V. Makarenko, S. Tiomkin, and R. V. Kulkarni, "Average-Reward Reinforcement Learning with Entropy Regularization," *arXiv*. Jan. 15, 2025. [Online]. Available: http://arxiv.org/abs/2501.09080

[14] Israa Saad Mohammed, "DCITD: A Deep Q-Network Approach for Cyber Image Threats Detection," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 17s, pp. 436–449, Mar. 2025, doi: 10.52783/jisem.v10i17s.2748.

[15] W. Yang, A. Acuto, Y. Zhou, and D. Wojtczak, "A Survey for Deep Reinforcement Learning Based Network Intrusion Detection," *arXiv*. Sep. 25, 2024. [Online]. Available: http://arxiv.org/abs/2410.07612

[16] Y. Badr, "Enabling intrusion detection systems with dueling double deep Q -learning," *Digit. Transform. Soc.*, vol. 1, no. 1, pp. 115–141, Aug. 2022, doi: 10.1108/DTS-05-2022-0016.

[17] E. Walter, K. Ferguson-Walter, and A. Ridley, "Incorporating Deception into CyberBattleSim for Autonomous Defense," *arXiv*. Aug. 31, 2021. [Online]. Available: http://arxiv.org/abs/2108.13980

[18] F. M. Zennaro and L. Erdődi, "Modelling penetration testing with reinforcement learning using capture-the-flag challenges: Trade-offs between model-free learning and a priori knowledge," *IET Inf. Secur.*, vol. 17, no. 3, pp. 441–457, May 2023, doi: 10.1049/ise2.12107.

[19] Q. Yao, Y. Wang, X. Xiong, P. Wang, and Y. Li, "Adversarial Decision-Making for Moving Target Defense: A Multi-Agent Markov Game and Reinforcement Learning Approach," *Entropy*, vol. 25, no. 4, p. 605, Apr. 2023, doi: 10.3390/e25040605.

[20] C. Lei, D.-H. Ma, and H.-Q. Zhang, "Optimal Strategy Selection for Moving Target Defense Based on Markov Game," *IEEE Access*, vol. 5, pp. 156–169, 2017, doi: 10.1109/ACCESS.2016.2633983.

[21] E. Muhati and D. B. Rawat, "Asynchronous Advantage Actor-Critic (A3C) Learning for Cognitive Network Security," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, Dec. 2021, pp. 106–113. doi: 10.1109/TPSISA52974.2021.00012.

[22] L. Chavali, T. Gupta, and P. Saxena, "SAC-AP: Soft Actor Critic based Deep Reinforcement Learning for Alert Prioritization," *arXiv*. Jul. 27, 2022. [Online]. Available: http://arxiv.org/abs/2207.13666

[23] R. Ozalp, A. Ucar, and C. Guzelis, "Advancements in Deep Reinforcement Learning and Inverse Reinforcement Learning for Robotic Manipulation: Toward Trustworthy, Interpretable, and Explainable Artificial Intelligence," *IEEE Access*, vol. 12, pp. 51840–51858, 2024, doi: 10.1109/ACCESS.2024.3385426.

[24] D. S. Diop, S. Y. Luis, M. P. Esteve, S. L. T. Marín, and D. G. Reina, "Decoupling Patrolling Tasks for Water Quality Monitoring: A Multi-Agent Deep Reinforcement Learning Approach," *IEEE Access*, vol. 12, pp. 75559–75576, 2024, doi: 10.1109/ACCESS.2024.3403790.

[25] K. Ohashi, K. Nakanishi, N. Goto, Y. Yasui, and S. Ishii, "Orthogonal Adversarial Deep Reinforcement Learning for Discrete- and Continuous-Action Problems," *IEEE Access*, vol. 12, pp. 151907–151919, 2024, doi: 10.1109/ACCESS.2024.3479089.

[26] M. Yavuz and Ö. C. Kivanç, "Optimization of a Cluster-Based Energy Management System Using Deep Reinforcement Learning Without Affecting Prosumer Comfort: V2X Technologies and Peer-to-Peer Energy Trading," *IEEE Access*, vol. 12, pp. 31551–31575, 2024, doi: 10.1109/ACCESS.2024.3370922.

[27] T. Zhou, Y. Yakuwa, N. Okamura, H. Hochigai, T. Kuroda, and I. Eguchi Yairi, "Dueling Network Architecture for GNN in the Deep Reinforcement Learning for the Automated ICT System Design," *IEEE Access*, vol. 13, pp. 21870–21879, 2025, doi: 10.1109/ACCESS.2025.3534246.