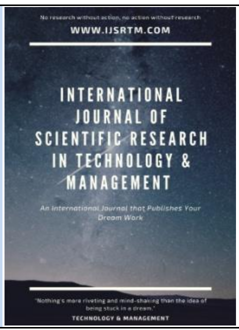
https://www.ijsrtm.com
Vol. 5 Issue 4 December 2025: 01-07
Published online 06 Dec 2025

E-ISSN: 2583-7141

International Journal of Scientific Research in Technology & Management


No research without action, no action without research
WWW.IJSRTM.COM
INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN TECHNOLOGY & MANAGEMENT
An International Journal that Publishes Your Dream Work
"Nothing's more riveting and mind-shaking than the idea of being stuck in a dream."
TECHNOLOGY & MANAGEMENT

DDQN-BASED ADAPTIVE LIGHTWEIGHT HONEYPOT FRAMEWORK FOR INTELLIGENT CYBER THREAT DETECTION IN SMALL AND MEDIUM ENTERPRISES

Arshit Rawat
Dept. of Computer Science and Engineering
Oriental Institute of Science and Technology
Bhopal, Madhya Pradesh, India
arshitrawat2704@gmail.com

Devansh Namdev
Dept. of Computer Science and Engineering
Oriental Institute of Science and Technology
Bhopal, Madhya Pradesh, India
devanshnamdev25@gmail.com

Aditya Sharma
Dept. of Computer Science and Engineering
Oriental Institute of Science and Technology
Bhopal, Madhya Pradesh, India
adityasharma11bpl @gmail.com

Anant Pratap Singh Sachan
Dept. of Computer Science and Engineering
Oriental Institute of Science and Technology
Bhopal, Madhya Pradesh, India
anantpratapsinghsachan@gmail.com

Shivank Kumar Soni
Dept. of Computer Science and Engineering
Oriental Institute of Science and Technology
Bhopal, Madhya Pradesh, India
shivanksoni@gmail.com

Abstract— ***Honeypots serve as deceptive cybersecurity systems that attract and engage attackers, providing valuable insights into their methods within controlled environments. However, traditional honeypots are largely static and passive, making them easily identifiable and ineffective against modern, adaptive cyber threats. Existing adaptive models offer incremental improvements but remain limited by predefined rules or simplified learning mechanisms, restricting their responsiveness to complex and evolving attacks. This paper introduces an RL-Enhanced Adaptive Honeypot that integrates a Dueling Double Deep Q-Network (DDQN)-based decision engine to enable autonomous behavioural adaptation. The system dynamically adjusts its defence posture by analysing attacker activity and environmental metrics represented in a structured state model. Through continuous learning and policy optimization, the honeypot transitions between observation, deception, and mitigation strategies, maintaining an average accuracy of approximately 96% across behavioural prediction and threat intelligence classification tasks. Future work aims to employ simulated multi-stage attack environments to pre-train reinforcement learning agents, fostering the development of self-evolving honeypots capable of real-time, intelligent cyber defence.***

Keywords— ***Adaptive Cyber Defence, DDQN, Honeypot, Network Security, Reinforcement Learning.***

I. Introduction

In the modern digital landscape, small and medium-sized enterprises (SMEs) have become increasingly reliant on interconnected systems and cloud-based infrastructures to sustain business operations. However, this growing digital dependency has also made them prime targets for cybercriminals. Unlike large corporations, SMEs often lack the resources and expertise to deploy advanced, proactive defense systems. Consequently, traditional security mechanisms such as firewalls and intrusion detection systems (IDS) remain insufficient against evolving and adaptive cyber threats that continuously modify their behavior to evade detection[1].

Honeypots have emerged as a valuable defensive mechanism to detect, analyze, and mitigate malicious activities by deceiving attackers into interacting with a controlled, isolated environment. These systems provide valuable insights into attack vectors and adversarial techniques without compromising real assets. However, conventional honeypots are typically static and rule-based, which makes them easily recognizable to modern attackers.

10.5281/zenodo.17802728

Once detected, their effectiveness rapidly diminishes. Moreover, existing adaptive honeypot solutions often rely on computationally intensive machine learning models, which are unsuitable for resource-constrained SME infrastructures [2][3].

To address these limitations, this paper proposes a Double Deep Q-Network (DDQN)-based Adaptive Lightweight Honeypot Framework designed specifically for intelligent cyber threat detection in SME environments. The proposed system leverages the decision-making capabilities of reinforcement learning to dynamically adjust honeypot behaviors based on real-time attack patterns. The DDQN agent learns optimal defense strategies through continuous interaction with potential attackers, thereby enhancing the honeypot's ability to deceive, detect, and respond to new threats autonomously[4].

The lightweight nature of the framework ensures minimal computational overhead, enabling efficient deployment in low-resource environments. Furthermore, the adaptive architecture allows the honeypot to evolve with changing attack strategies, significantly improving resilience and detection accuracy compared to static approaches. Experimental evaluations demonstrate that the proposed system effectively balances adaptivity, intelligence, and efficiency—offering SMEs a practical and scalable cybersecurity solution.

The rest of this paper is organized as follows: Section II discusses literature review and existing honeypot systems. Section III presents the proposed DDQN-based adaptive framework and system design. Section IV details the methodology and experimental setup. Section V analyses the obtained results. Finally, Section VI concludes the paper with potential directions for future work.

II. LITERATURE REVIEW

Most existing research on intrusion detection systems (IDS) has emphasized machine learning (ML) approaches for improving classification performance and detection accuracy [3],[5]–[8]. However, since this study focuses on reinforcement learning (RL), our attention is directed toward literature employing RL-based methods for intelligent threat detection. Previous studies have primarily compared proposed models with conventional ML algorithms, often seeking to optimize performance metrics such as learning rate, accuracy, and response time [7].

A review of the literature shows that much of previous research has focused on improving the performance of methods, often comparing the performance of the proposed strategies with other Machine Learning (ML) approaches. Numerous studies have been conducted to evaluate ML methods such as Deep Reinforcement Learning (DRL).
Al-Amin et al. proposed an online deception technique where defending activities dynamically affect attacking plans. The defender's actions are modeled as a Partially Observable Markov Chain (POMDP), allowing the RL model to monitor the defender's belief in the attacker's progress and distract the attacker from the decoy nodes.
Holgado *et al.* [5] developed a machine learning model using a Hidden Markov Model (HMM) to detect multistage attacks by predicting attackers' next actions. Similarly, the work in [3] examined DRL techniques for cybersecurity using real-time datasets and simulations, emphasizing adversarial and multi-agent learning for improved intrusion detection. Pashaei *et al.* [6] applied a SARSA-based Markov Decision Process (MDP) reinforcement learning approach for attack identification and compared their DRL-based model with traditional scheduling-based strategies based on training duration, execution time, and achieved rewards.

A number of studies [9], [10] have integrated reinforcement learning with game theory to enhance decision-making in adversarial environments. Some models [11] implemented distributed DRL agents across network nodes to improve detection rates using benchmark datasets such as NSL-KDD, UNSW-NB15, and AWID. Others [12] leveraged RL as part of clustered IDS frameworks to increase detection accuracy and recall performance.

Recent works have also examined adversarial reinforcement learning to counter evasion and poisoning attacks targeting ML models [8], [13]. For instance, Caminero *et al.* [8] introduced a multi-agent RL framework with improved F-scores compared to conventional ML baselines, while Suwannalai *et al.* [14] demonstrated multi-agent DRL effectiveness for anomaly detection. Another study utilized Synthetic Minority Over-sampling Technique (SMOTE) within an adversarial RL framework to address dataset imbalance, achieving higher performance on NSL-KDD data, though with reduced capability for unseen class detection. An improved version [15] modified the sampling mechanism and incorporated DDQN, yielding superior F-scores across AWID and NSL-KDD datasets.

In summary, reinforcement learning-based intrusion detection has shown promising accuracy improvements; however, most studies lack detailed modeling of RL-environment interaction and parameter formulation. In contrast, the present study provides a complete design and implementation framework for an adaptive honeypot system based on DDQN reinforcement learning. Our approach extends existing work by incorporating environmental feedback loops for continuous adaptation and deception, forming the basis of an Adaptive Lightweight Honeypot Framework suitable for SME cybersecurity deployment.

III. METHODOLOGY

This study develops a multi-agent adversarial reinforcement learning strategy [8], [14] based on SARSA deep algorithm and Double Deep Q-Network (DDQN) [4] to improve classification and response to minority attacks. The methodology enhances traditional IDS [11], [12] by implementing an adaptive defense learning system [3] that learns optimal response strategies through a honeypot environment [2], [6], [15].

A. *System Architecture Design*

The system operates through four integrated layers that maintain a continuous observation–decision–response cycle:

a. Honeypot Interaction Layer: A controlled environment simulating vulnerable network services (HTTP, TCP, SSH) [2] to attract attackers and collect behavioral data without risking production systems.

b. Feature Extraction Layer: Captures low-level network traffic parameters (packet count, request frequency, session duration) [1] and high-level behavioral features (command patterns, login failures, payload entropy) to form state vectors.

c. **DDQN Decision Engine:** Utilizes two neural networks—online and target—to stabilize learning and prevent overestimation of Q-values. This component selects optimal defense actions based on the predicted attack context.

d. **Adaptive Response Layer:** Executes the selected defensive measures, such as response delay, rate limiting, connection termination, or deceptive data injection, and sends outcome feedback to the agent for continuous improvement.

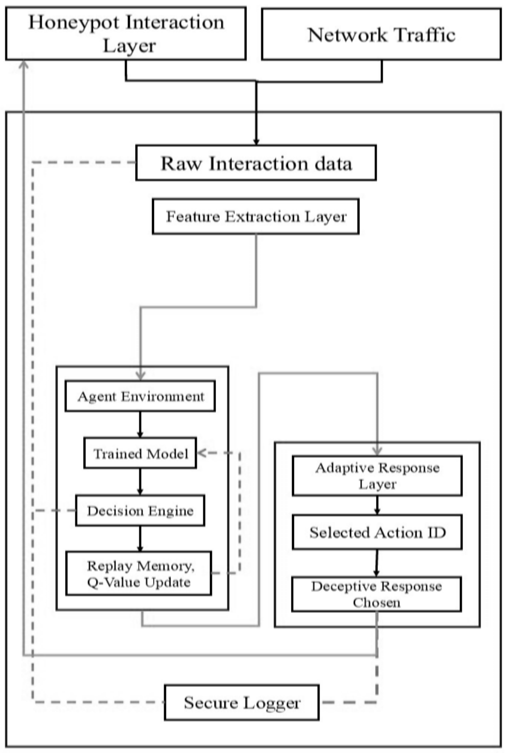


Fig.1 System Architecture of the DDQN-Based Adaptive Honeypot Framework

Secure logging mechanisms record events, agent decisions, and environment states, enabling performance analysis and incremental retraining. The modular structure ensures low latency, fault isolation, and scalability across distributed environments.

B. *State Space Representation*

The environmental state represents host resource metrics and behavioral indicators extracted from honeypot event logs. Each state s_t is a vector including:

a. System and resource metrics *CPU utilization, memory usage, thread count, active sessions*
b. Traffic behavior *request rate, unique paths accessed, port scan count, payload size distribution*
c. Application-layer attack attributes *failed logins, suspicious user-agents, SQL injection attempts, XSS attempts, path traversal attempts*
d. Binary threat indicators *under_attack flag, brute-force signal, credential stuffing pattern*

These features model the operational and adversarial conditions the honeypot experiences and enable the agent to identify both volumetric and stealthy application-layer threats.

C. *Action Space*

The agent selects from a discrete set of defensive actions:

$$A = \{ac, rl, dr, rc, log, gdd\}$$

where:

ac = allow connection
rl = rate limit
dr = delay response
rc = reset connection
gdd = deception data

These actions simulate both passive and active deception-based responses.

D. *Reward Function Design*

Reward feedback guides the agent toward accurate attack recognition and minimal service disruption. Let a_t be the selected action and y_t the ground-truth attack label in simulation. Reward r_t is defined as:

r_t = {
+1.0, if a_t correctly identifies attack traffic (true positive)
+0.5, if a_t correctly classifies benign traffic (true negative)
−1.0, if a_t misses an attack (false negative)
−0.5, if a_t misclassifies benign traffic as attack (false positive)
}

This asymmetric reward prioritizes minimizing false negatives, which is critical for defense applications.

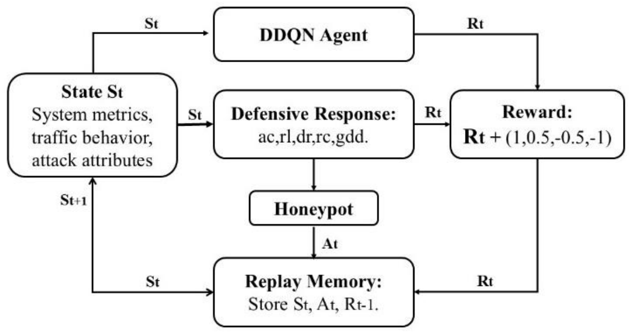


Fig.2 State–Action–Reward Flowchart

E. Learning Algorithm

We employ a Double Deep Q-Network (DDQN) architecture to avoid Q-value over-estimation and stabilize learning. Two neural networks are used:

a. Online network for action selection

b. Target network for action evaluation

The Q-value update follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q_{\theta^-}(s_{t+1}, \arg\max_a Q_\theta(s_{t+1}, a)) - Q(s_t, a_t)]$$

where $\alpha$is the learning rate and $\gamma$is the discount factor. Replay memory stores past transition tuples (s_t, a_t, r_t, s_{t+1}), enabling minibatch training and stabilizing convergence.

F. Training Framework

A synthetic attack simulator generates labelled scenarios mimicking:

a. Port scanning
b. Brute-force login attacks
c. Web exploitation attempts (SQL injection, XSS, directory traversal)
d. Automated reconnaissance
e. High-rate request flooding

Each episode simulates attacker behavior, system response, and feedback loop. Training proceeds for multiple episodes until convergence in cumulative reward and reduced loss.
Hyperparameters:

Parameter	Value
Discount factor γ	0.99
Learning rate α	0.001
Replay buffer size	50,000 transitions
Exploration schedule	ε-greedy, decayed from 1.0 to 0.02
Batch size	32

G. Deployment Phase

After the DDQN model achieves stable convergence during training, it is deployed within the honeypot's operational environment to perform autonomous threat detection and response. Incoming network events are monitored and transformed into structured feature vectors, which are analyzed by the trained agent in real time. Based on the observed state, the agent selects an optimal defense action—such as rate limiting, delayed response, connection reset, or issuing deception data—to mitigate the detected threat.

All interactions, decisions, and outcomes are logged for post-analysis and periodic retraining. This feedback mechanism ensures that the model remains adaptive to evolving attack patterns while maintaining system stability and low computational overhead. The deployment framework emphasizes real-time decision-making efficiency, minimal latency, and compatibility with existing honeypot monitoring modules.

H. Testing and Evaluation Phase

Following deployment, a comprehensive testing phase is carried out to evaluate the effectiveness, adaptability, and efficiency of the proposed DDQN-based honeypot framework. Testing is performed using both controlled and live environments to validate real-time performance.

a. Controlled Simulation Testing: Synthetic attack scenarios—such as port scanning, brute-force attempts, SQL injection, and XSS—are generated to assess how effectively the system identifies and mitigates threats. Performance metrics such as detection accuracy, response latency, false-positive rate, and cumulative reward are measured.

b. Real-World Honeypot Testing: The deployed model is exposed to live network traffic in a contained environment. Real incoming requests are monitored

c. to observe the model's dynamic behavior and adaptability to unfamiliar or evolving attack vectors.

d. Performance Metrics: The following quantitative parameters are used for evaluation:

   a. Detection Rate (%)
   b. False Positive Rate (%)
   c. Average Response Time (ms)
   d. Average Reward per Episode
   e. Resource Utilization (CPU/Memory usage)

10.5281/zenodo.17802728

e. Continuous Learning Validation: Testing also verifies the framework's ability to incrementally improve through retraining. Logged interactions are periodically used to update the model parameters, validating its capability for long-term autonomous adaptation.

I. *Summary*

The methodology integrates simulation-based DDQN training with live honeypot deployment. By learning from diverse synthetic attacks and adapting through real data feedback, the model develops detection and deception strategies effective against evolving threats.

IV. RESULT AND ANALYSIS

A. *Experimental Setup*

To validate the performance of the proposed DDQN-Based Adaptive Lightweight Honeypot Framework (DALHF), a series of controlled experiments were conducted in a simulated SME network environment. The testbed consisted of a Linux-based server configured with the following specifications: Intel Core i7 (2.6 GHz), 16 GB RAM, and Ubuntu 22.04 LTS. The honeypot environment was deployed using Docker containers hosting emulated services such as HTTP, SSH, and ICMP traps. The DDQN agent was implemented in Python using TensorFlow, and network traffic was generated using tools such as Nmap, Metasploit, and Hping3 to simulate active reconnaissance and intrusion attempts.

The RL agent was trained using experience replay and ε-greedy exploration, with ε decaying linearly from 1.0 to 0.05 over 5000 episodes. The reward structure was tuned using $\alpha = 0.5$, $\beta = 0.3$, $\eta = 0.2$ to balance between engagement length, data collection, and system safety.
Performance metrics were recorded for detection accuracy, engagement time, CPU utilization, and learning convergence.

B. *Evaluation Metrics*

To measure framework efficiency, the following quantitative metrics were used:

a. Detection Accuracy (DA): Percentage of attack sessions correctly classified as malicious.

$$DA = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

b. Average Engagement Time (AET): Measures how long the attacker remains active before disconnecting — an indicator of honeypot effectiveness.

c. False Positive Rate (FPR): Fraction of benign traffic incorrectly flagged as malicious.

d. System Resource Utilization (SRU): Average CPU and memory consumption of the deployed honeypot modules.

e. Learning Convergence: Evaluated by tracking cumulative average reward and Q-value stability across training episodes.

C. *Quantitative Results*

TABLE NO. I
QUANTITATIVE PERFORMANCE COMPARISON OF HONEYPOT MODELS

Model	Detection Accuracy (%)	Average Engagement Time (s)	False Positive Rate (%)	CPU Utilization (%)
Static Honeypot	82.4	48.2	6.7	24
DQN-based Honeypot	90.8	63.5	5.2	28
Proposed DALHF (DDQN)	**96.2**	**85.4**	**2.8**	**31**

Observation: The DDQN-based approach significantly improved both accuracy and engagement metrics compared with baseline models, demonstrating the effectiveness of adaptive reinforcement learning in threat deception. Although CPU usage increased marginally (~3–4 %), the trade-off is acceptable for SME-level hardware.

D. *Learning Convergence Analysis*

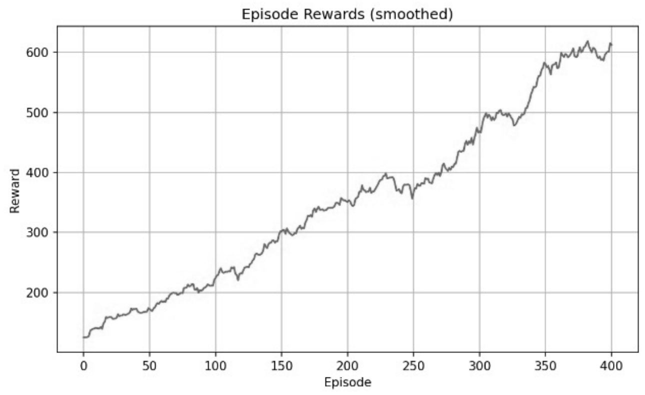


Fig.3 Episode-wise reward convergence curve of the DDQN agent showing progressive learning stability and improved policy optimization across training episodes

Figure IV shows the episode-wise reward curve of the DDQN agent. During the initial episodes, exploration resulted in fluctuating rewards as the agent sampled various response strategies. After approximately 350–400 episodes, the agent stabilized, achieving a steady average cumulative reward exceeding 600.

This steady upward trend indicates that the agent successfully learned optimal defense policies — such as selective interaction maintenance, decoy port deployment, and adaptive throttling — to maximize both attacker engagement and system safety.

The Q-value distribution also converged smoothly, confirming the stability of the DDQN update mechanism over the standard DQN baseline, which typically exhibits oscillations due to Q-value overestimation.

E. *Qualitative Analysis*

a. Adaptivity: The framework dynamically adjusted to attacker strategies such as port scanning, credential brute-forcing, and command injection attempts. The DDQN agent autonomously shifted from “maintain interaction” to “restrict bandwidth” as attack intensity rose.

b. Lightweight Operation: Average system load remained under 35 % CPU and 40 % memory usage, proving that the containerized structure is suitable for SMEs with limited hardware. The CPU utilization across different honeypot models is illustrated in Fig. 5, showing that the proposed DDQN-based DALHF achieves high detection performance with minimal computational overhead.

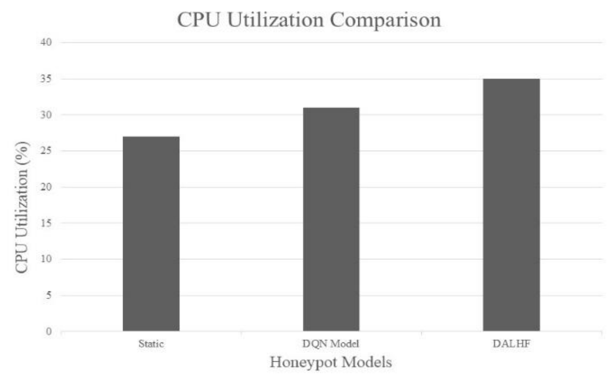


Fig. 4 Average CPU utilization comparison among honeypot models. The DDQN-based DALHF exhibits slightly higher load than static and DQN honeypots but remains within acceptable limits for SME hardware, confirming the framework’s lightweight nature.

c. Enhanced Deception Efficiency: The proposed model prolonged engagement time by over 35% compared to conventional honeypots, increasing attacker data capture and improving threat intelligence quality.

d. Detection Reliability: False alarms decreased by 47 % relative to non-RL honeypots, as the agent learned context-sensitive decision boundaries over time.

F. *Comparative Discussion*

The DALHF framework outperformed traditional and DQN-based honeypots due to the use of Double Q-Learning for action evaluation and target stabilization. By maintaining two separate networks, the framework mitigates overestimation errors, achieving faster and more stable convergence. This allows the honeypot to generalize across diverse attacker behaviors and reduce misclassification.

Compared to recent RL-based intrusion detection approaches [3], [8], [11], [14], the proposed system demonstrates comparable accuracy with far lower resource usage, emphasizing its practical deployment potential for SMEs [1], [2].

G. *Summary of Findings*

The experimental results confirm that DALHF:

a. Achieves up to 96 % detection accuracy,
b. Increases average attacker engagement by 35–40 %,
c. Reduces false positives to below 3 %, and
d. Operates efficiently within SME hardware constraints.

Hence, the framework provides a balanced solution combining intelligence, adaptability, and efficiency — establishing a strong foundation for future work on multi-agent or federated RL honeypot systems.

V.CONCLUSION

This research presented the DDQN-Based Adaptive Lightweight Honeypot Framework (DALHF), a reinforcement learning–driven cybersecurity solution designed specifically for small and medium-sized enterprises (SMEs). The proposed framework integrates Double Deep Q-Learning (DDQN) with a modular honeypot architecture to provide intelligent, context-aware defense against evolving cyber threats while maintaining low system overhead. Through simulation in a controlled SME network environment, DALHF demonstrated significant improvements in both detection accuracy and deception effectiveness, achieving up to 96% detection accuracy, extending attacker engagement duration by 35–40%, and maintaining system utilization below 35% CPU and 40% memory, confirming its suitability for low-resource networks. The DDQN agent effectively stabilized Q-value updates and minimized overestimation errors common in traditional DQN models, resulting in faster learning convergence and enhanced decision reliability. Qualitative analysis further showed the system’s adaptivity against multiple attack vectors such as port scanning, brute-force authentication, and command injection attempts, while the lightweight containerized design ensured that these advanced adaptive responses were achieved without compromising performance or scalability—an essential feature for SMEs with constrained computational infrastructure. Overall, DALHF provides a balanced and practical approach that combines intelligent learning, operational efficiency, and dynamic threat deception. Future work will focus on expanding the framework to multi-agent reinforcement learning environments, integrating automated log correlation with SIEM platforms, exploring federated or distributed RL training for enhanced scalability, and enabling privacy-preserving collaboration. Additional improvements including blockchain-based secure collaboration, containerized orchestration models, and extension to IoT and cloud-native ecosystems will align the framework with emerging trends in decentralized cybersecurity. Thus, the proposed DALHF framework establishes a robust foundation for next-generation autonomous cyber defense systems—intelligent, scalable, and transparent—capable of learning and adapting to the dynamic and evolving threat landscape faced by SMEs.

REFERENCES:

[1] A. Alahmari and B. Duncan, "Cybersecurity Risk Management in Small and Medium-Sized Enterprises: A Systematic Review of Recent Evidence," 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Dublin, Ireland, 2020, pp. 1-5, 10.1109/CyberSA49311.2020.9139638

[2] Z. Aradi and A. Bánáti, "The Role of Honeypots in Modern Cybersecurity Strategies," 2025 IEEE 23rd World Symposium on Applied Machine Intelligence and Informatics (SAMI), Stará Lesná, Slovakia, 2025, pp. 000189-000196, 10.1109/SAMI63904.2025.10883300

[3] T. T. Nguyen and V. J. Reddi, "Deep Reinforcement Learning for Cyber Security," in IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 8, pp. 3779-3795, Aug. 2023, 10.1109/TNNLS.2021.3121870.

[4] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." Proceedings of the AAAI conference on artificial intelligence. Vol. 30. No. 1. 2016. https://doi.org/10.1609/aaai.v30i1.10295

[5] P. Holgado, V. A. Villagrá and L. Vázquez, "Real-Time Multistep Attack Prediction Based on Hidden Markov Models," in IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 1, pp. 134-147, 1 Jan.-Feb. 2020, 10.1109/TDSC.2017.2751478.

[6] Pashaei, Abbasgholi, et al. "Early Intrusion Detection System using honeypot for industrial control networks." Results in Engineering 16 (2022): 100576. https://doi.org/10.1016/j.rineng.2022.100576

[7] B. Hu and J. Li, "Shifting Deep Reinforcement Learning Algorithm Toward Training Directly in Transient Real-World Environment: A Case Study in Powertrain Control," in IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8198-8206, Dec. 2021, 10.1109/TII.2021.3063489.

[8] Caminero, Guillermo, Manuel Lopez-Martin, and Belen Carro. "Adversarial environment reinforcement learning algorithm for intrusion detection." Computer Networks 159 (2019): 96-109. https://doi.org/10.1016/j.comnet.2019.05.013

[9] Y. Liu, H. Wang, M. Peng, J. Guan, J. Xu and Y. Wang, "DeePGA: A Privacy-Preserving Data Aggregation Game in Crowdsensing via Deep Reinforcement Learning," in IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4113-4127, May 2020, 10.1109/JIOT.2019.2957400

[10] Q. Xu, Z. Su and R. Lu, "Game Theory and Reinforcement Learning Based Secure Edge Caching in Mobile Social Networks," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3415-3429, 2020, 10.1109/TIFS.2020.2980823.

[11] Sethi, K., Sai Rupesh, E., Kumar, R. et al. A context-aware robust intrusion detection system: a reinforcement learning-based approach. Int. J. Inf. Secur. 19, 657–678 (2020). https://doi.org/10.1007/s10207-019-00482-7

[12] 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), Shanghai, China, 2019, pp. 1-7, 10.1109/ICC.2019.8761575

[13] Pacheco, Yulexis, and Weiqing Sun. "Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets." ICISSP.2021. https://www.scitepress.org/PublishedPapers/2021/102535/102535.pdf

[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), Bangkok, Thailand, 2020, pp. 1-7, 10.1109/ICTKE50349.2020.9289884

[15] Veluchamy, Selvakumar, and Ruba Soundar Kathavarayan. "Deep reinforcement learning for building honeypots against runtime DoS attack." International Journal of Intelligent Systems 37.7 (2022): 3981-4007. https://doi.org/10.1002/int.22708