electronics

Article

Employing Deep Reinforcement Learning to Cyber-Attack Simulation for Enhancing Cybersecurity

Sang Ho Oh [1], Jeongyoon Kim [2], Jae Hoon Nah [3] and Jongyoul Park [2,*]

[1] Department of Computer Engineering and Artificial Intelligence, Pukyong National University, Busan 48513, Republic of Korea; shoh0320@pknu.ac.kr
[2] Department of Applied Artificial Intelligence, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea; kjy97426@seoultech.ac.kr
[3] Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea; jhnah@etri.re.kr
* Correspondence: jongyoul@seoultech.ac.kr; Tel.: +82-2-970-9776

Abstract: In the current landscape where cybersecurity threats are escalating in complexity and frequency, traditional defense mechanisms like rule-based firewalls and signature-based detection are proving inadequate. The dynamism and sophistication of modern cyber-attacks necessitate advanced solutions that can evolve and adapt in real-time. Enter the field of deep reinforcement learning (DRL), a branch of artificial intelligence that has been effectively tackling complex decision-making problems across various domains, including cybersecurity. In this study, we advance the field by implementing a DRL framework to simulate cyber-attacks, drawing on authentic scenarios to enhance the realism and applicability of the simulations. By meticulously adapting DRL algorithms to the nuanced requirements of cybersecurity contexts—such as custom reward structures and actions, adversarial training, and dynamic environments—we provide a tailored approach that significantly improves upon traditional methods. Our research undertakes a thorough comparative analysis of three sophisticated DRL algorithms—deep Q-network (DQN), actor–critic, and proximal policy optimization (PPO)—against the traditional RL algorithm Q-learning, within a controlled simulation environment reflective of real-world cyber threats. The findings are striking: the actor–critic algorithm not only outperformed its counterparts with a success rate of 0.78 but also demonstrated superior efficiency, requiring the fewest iterations (171) to complete an episode and achieving the highest average reward of 4.8. In comparison, DQN, PPO, and Q-learning lagged slightly behind. These results underscore the critical impact of selecting the most fitting algorithm for cybersecurity simulations, as the right choice leads to more effective learning and defense strategies. The impressive performance of the actor–critic algorithm in this study marks a significant stride towards the development of adaptive, intelligent cybersecurity systems capable of countering the increasingly sophisticated landscape of cyber threats. Our study not only contributes a robust model for simulating cyber threats but also provides a scalable framework that can be adapted to various cybersecurity challenges.

Keywords: cyber-attack simulation; artificial intelligence; cybersecurity; deep reinforcement learning

Citation: Oh, S.H.; Kim, J.; Nah, J.H.; Park, J. Employing Deep Reinforcement Learning to Cyber-Attack Simulation for Enhancing Cybersecurity. *Electronics* **2024**, *13*, 555. https://doi.org/10.3390/electronics13030555

Academic Editors: Krzysztof Szczypiorski and Amna Qureshi

Received: 19 December 2023
Revised: 25 January 2024
Accepted: 29 January 2024
Published: 30 January 2024

1. Introduction

To defend against various cyber-attacks, red team exercises are frequently used to evaluate how well network system defenses react [1]. To simulate sophisticated persistent threats and test the system's defenses against diverse strategies, methods, and techniques employed by advanced attackers, these exercises frequently use adversary profiles [2,3]. Yet, because red team exercises call for particular human expertise, they can be time-consuming and expensive to undertake. Emulators have been created to automate the process and expedite the assault simulation process to increase the effectiveness of red teaming [4]. Additional tools, such as execution payloads, enabling scripts, and staging frameworks, have been developed to streamline the red teaming process and support human experts [3].

Even though these tools are meant to assist human experts who are vital to the planning and decision-making phases of the exercise, such as designing strategies and procedures across various phases of the attack simulation campaign.

The importance of cyber-attack simulation has grown as cyber-attacks become more sophisticated and dynamic [5]. Conventional security systems that rely on rules and signatures that have been predetermined are frequently insufficient to protect against these sophisticated and adaptive attackers [6]. In contrast, machine learning (ML) models have become a more adaptable and flexible approach to cybersecurity [7–9]. ML models can develop over time to better detect and respond to emerging threats by learning from prior data [10–12]. They may offer a stronger protection system that changes in response to changing dangers. In terms of boosting the security of networked systems, the use of ML models in cybersecurity has shown considerable potential.

The goal of our research is to demonstrate deep reinforcement learning (DRL) can be employed effectively in cyber-attack simulation for cybersecurity. We achieve this by conducting a red teaming simulation and evaluating the performance of the DRL agent. By using simulations to model cyber-attacks, we can assess the performance of the DRL algorithm in a controlled environment. Our results demonstrate that the DRL agent was able to learn and execute effective policies for infiltrating a simulated cybersecurity environment.

Our paper is organized as follows. In Section 2, we organized the related works. In Section 3, we outline the methodology, deep Q-network (DQN), actor–critic, proximal policy optimization (PPO), and simulation settings including MITRE ATT&CK [13]. The results from the experiment are shown in Section 4 along with the reward obtained, agents' attack success rate, and the number of iterations they made in each episode. We analyze the findings in Section 5 before outlining our conclusions in Section 6. Finally, Section 7 shows limitations and future works.

2. Related Works

The ML in cybersecurity has become more prevalent, which has raised the risk of cyber-attacks on ML-based software [14,15]. Cyber-attack simulation for cybersecurity uses ML techniques to simulate and model hypothetical cyberattacks on a system to train ML models to detect and respond to these attacks in real-time. This strategy aids researchers in enhancing their overall cybersecurity defenses and preparing for any cyber threats. Traditional ML-based applications, on the other hand, have drawbacks because they are frequently trained on historical data and may not be very generalizable [16–18]. The idea of artificial intelligence (AI)-assisted or self-governing AI red teaming has surfaced as AI develops. In this case, AI can create novel attack strategies against sophisticated cyber systems that human red team experts could not have thought of before using the superior decision-making skills that it has learned through AI training [19,20]. This could transform cyber-attack simulation in cybersecurity and boost the potency of defenses against cyber-attacks.

Owing to the current scenario, we are motivated to train agents to recognize and optimize attack strategy in a simulation-based network using deep reinforcement learning (DRL) methods, an improvement over the conventional ML model. We will be able to find more reliable solutions and better cyber-attack simulations as a result. Reinforcement learning (RL) is a method that can assist in developing autonomous agents capable of making the best consecutive decisions in challenging and unpredictable situations. The potential for RL research in several application sectors has expanded because of open-source learning environments like OpenAIGym [21].

The RL in cyber-attack simulation for cybersecurity has gained popularity recently [22–26]. Researchers have turned to ML approaches like RL to create more resilient and flexible security systems as a result of the sophistication of cyberattacks and the difficulty in creating effective defenses against them. The best ways to protect against attacks, change threats, and improve attack methods can all be learned using RL algorithms. By

regularly engaging in an offensive and defensive game, an RL agent could learn to detect and defend against different types of attacks, like, zero-day exploits [27]. In a range of situations, such as online applications, malware analysis, and network intrusion detection, it has been discovered that this method is effective for identifying and containing cyber-attacks [28,29].

With the provision of adaptive and automated defensive systems that could learn from experience and react in real-time to evolving cyber threats, the RL has the potential to revolutionize cybersecurity [30]. Before RL can be effectively employed in the context of cybersecurity, several important issues must be resolved. One of the most serious challenges is the absence of training data [10,31]. It could be challenging to obtain enough data to effectively train RL models because aggressive cyber-attack situations are frequently unusual and complex. This may lead to the appearance of underfitting models that are unable to accurately capture the complexity of real-world occurrences.

The simulation of intricate and dynamic attack scenarios is another issue [32]. It may be difficult to create precise models that can capture the entire spectrum of probable attack methods and tactics since cyberattacks can be very dynamic and adaptive. This might result in models that are overfitted and overly concentrated on particular attack scenarios, which makes it challenging to generalize to novel or unforeseen attack scenarios. Moreover, there are few open-source RL experimentation platforms for cybersecurity, which makes it difficult for researchers to address real-world problems and advance the state-of-the-art in RL cybersecurity applications [33]. Innovative RL-based solutions for cybersecurity may be difficult to develop and test for researchers without access to expansive and realistic testing environments.

Amidst the evolving landscape of cybersecurity threats, concerted efforts are being made to harness the potential of RL to fortify cyber defenses. Ibrahim et al. delve into the realm of cyber-physical systems with a focus on smart grid security, employing an RL-augmented attack graph to pinpoint vulnerabilities through the SARSA RL technique [34]. Dutta et al. advance this frontier with a data-driven DRL framework designed to develop proactive, context-sensitive defense mechanisms that can adapt on-the-fly to changing adversary tactics, all while minimizing operational disruptions [35].

Further contributions include the work of Applebaum et al., who scrutinize the capabilities of RL-trained autonomous agents across diverse network environments [36]. In a similar vein, Elderman et al. explore cybersecurity simulations within networked environments modeled as Markov games with stochastic dynamics and incomplete information [37]. Their work exemplifies the complex interplay between two pivotal agents—the attacker and defender—in sequential decision-making scenarios.

On a practical note, Microsoft's introduction of CyberBattleSim, an experimental research platform based on an OpenAIGym interface, marks a significant step in the direction of RL application in cybersecurity [38]. This Python-based platform lays the groundwork for researchers to conduct trials, test various hypotheses, and forge innovative models to address pressing cybersecurity challenges. As technological advancements continue, we can expect the emergence of more sophisticated RL experimentation environments, further catalyzing the advancement of cutting-edge RL applications in cybersecurity.

This study makes a pivotal contribution to the field of cybersecurity by providing a nuanced comparative analysis of DRL applications for cyber-attack simulations, distinguishing itself from prior research in several key areas. Firstly, it demonstrates the successful deployment of DRL algorithms in simulating and understanding the mechanics of cyber-attacks, which marks a significant step beyond traditional cybersecurity practices. Unlike previous studies, this research showcases the DRL agent's ability to learn and adapt in real-time, offering evidence for the viability of DRL in creating advanced, automated defense mechanisms that can dynamically counter sophisticated cyber threats.

Secondly, by leveraging a realistic cyber-attack scenario from the MITRE ATT&CK database, the study surpasses many existing research works that rely on hypothetical or simplified models. This approach provides a more authentic testing ground for DRL

techniques, enhancing the relevance and transferability of the results to real-world cybersecurity problems, thus paving the way for DRL integration into current cybersecurity applications.

Lastly, the study's comparative examination of DQN, actor–critic, and PPO algorithms against the conventional RL algorithm Q-learning, within the context of a detailed simulation environment, offers fresh insights that previous studies have not captured. It highlights the significance of algorithm selection based on the environment and scenario specifics, a factor that is critical for the development of effective DRL-based cybersecurity solutions. Collectively, these elements position the study as a foundational piece of research, charting a course for future advancements in DRL applications within cybersecurity and contributing to the strategic evolution of cybersecurity methodologies.

3. Materials and Methods

RL algorithms learn approximate solutions for Markov decision processes (MDPs) by acquiring policies that aim to maximize the expected rewards over a specific period [39]. Similar to MDPs, RL leverages state, action, and reward components to interact with the environment, where the agent selects actions based on prior rewards or penalties, as well as its present state. RL algorithms find applications in diverse fields, such as cybersecurity, where exhaustive search methods may not identify optimal solutions.

In this study, we utilized a simulated environment to enable our agent to interact with a realistic cyber-attack scenario provided by MITRE ATT&CK. By employing a simulated environment, we were able to accelerate the agent's policy learning process. This approach allowed our agent to learn from experience and adjust its policies in response to changing cyber threats in real time. The use of a simulated environment also provided a controlled testing environment, allowing us to evaluate the performance of the agent under various conditions. Overall, our study shows the effectiveness of utilizing DRL environments in training agents. Additionally, by incorporating the real-life cyber-attack scenario provided by MITRE ATT&CK, we were able to evaluate the agent's performance in a challenging and relevant context.

3.1. The Reinforcement Learning Algorithms Used

3.1.1. Deep Q-Network (DQN)

When the probability distribution of a state variable is known, the Q-learning approach is used to determine the optimal action to take in a given state. The approach is predicated on the estimation of a value function made up of Q-values that are computed for each state-action pair (s_t, a_t) [40]. The agent assesses the reward for each state-action pair (s_t, a_t) in each iteration t after initializing the Q-values to arbitrary real numbers. The following step of the algorithm updates the Q-values based on the next state-action pair's Q-values, $Q(s_{t+1}, a_{t+1})$, and the immediate reward r_t, with a discount factor γ controlling the impact of future rewards on current rewards as stated in Equation (1). Q-learning is widely used in reinforcement learning, allowing agents to learn optimal policies in various environments by iteratively improving their Q-values.

$$Q(s_t, a_t) \leftarrow r_t + \gamma \max_{a_{t+1}} \{Q(s_{t+1}, a_{t+1})\} \tag{1}$$

Regardless of the policy being used, the Q-values are adjusted to provide an optimal action-value function Q* [41]. Q-learning offers the flexibility to generate state-action combinations using various sampling strategies. The -greedy action selection method, denoted by Equation (2), which has a value between 0 and 1, is one such widely used method.

$$a_t = \begin{cases} \underset{a \in A}{arg\max} Q(s_t, a) & with\ probability\ 1 - \varepsilon, \\ a \sim A & otherwise. \end{cases} \tag{2}$$

A function approximator can be used to estimate Q-values rather than depending on a Q-table [42,43]. The approximator's parameter set is tuned to estimate Q-values as Equation (1), and the neural network is denoted by $Q(s, a; \theta)$ [43,44]; where is the parameter set for the approximator. The Q-network ($Q(s, a; \theta)$) and a target Q-network ($\hat{Q}(s, a; \theta^{-})$) are the two networks involved in DQN [42,45]. The Q-network chooses the optimal action, and the target Q-network creates the target value for updating the Q-network's parameter (θ). A loss function stated in Equation (3) is used to minimize the mean-square error between the target ($\hat{Q}(s', a'; \theta^{-})$) and current Q-networks ($Q(s, a; \theta)$) to update the Q-network at each iteration.

$$L_i(\theta_i) = \mathrm{E}\left[\left(r + \gamma arg\max_{a'}\hat{Q}\left(\mathrm{s}', \mathrm{a}'; \theta^{-}\right) - Q(s, a; \theta_i)\right)^2\right] \quad (3)$$

Achieving a balance between exploration and exploitation is critical in the implementation of DQN, which entails exploring alternative courses of action while exploiting previous knowledge. DQN relies on a database that stores the agent's experiences, where the usage of replay memory constitutes a crucial element [45]. A random selection of experiences from the replay memory is utilized to update the Q-networks [46].

3.1.2. Actor–Critic

Actor–critic methods, as described in [47,48], aim to amalgamate the benefits of actor-only and critic-only approaches. Similar to actor-only methods, actor–critic methods can generate continuous actions [49]. However, they address the substantial variance in policy gradients associated with actor-only methods by incorporating a critic. The primary function of the critic is to assess the current policy dictated by the actor. This assessment can be carried out using commonly employed policy evaluation techniques, such as temporal difference (λ) [50] or residual gradients [51]. The critic approximates and updates the value function based on sampled data. The derived value function, in turn, is utilized to adjust the actor's policy parameters, guiding them towards improved performance. Unlike critic-only methods, actor–critic methods typically retain the favorable convergence properties of policy gradient methods.

The actor–critic algorithm's learning agent is split into two distinct entities: the actor (policy) and the critic (value function). The actor is responsible solely for generating a control input based on the current state, while the critic processes received rewards and evaluates the policy's quality by adapting the value function estimate [49]. Following a series of policy evaluation steps by the critic, the actor undergoes updates informed by information obtained from the critic.

3.1.3. Proximal Policy Optimization (PPO)

PPO has emerged as the predominant on-policy policy optimization algorithm owing to its robust performance and straightforward implementation [52]. Instead of directly maximizing this lower bound, PPO adopts the objective of maximizing the surrogate objective while imposing a constraint on the proximity of the subsequent policy to the current one. To achieve this, PPO employs a heuristic approach, focusing on the following objective during each policy update:

$$L_k^{PPO}(\pi) = \begin{pmatrix} \mathbb{E} \\ (s,a) \sim d^{\pi_k} \end{pmatrix}\left[\min\left(\frac{\pi(a|s)}{\pi_k(a|s)}A^{\pi_k}(s,a), \mathrm{clip}\left(\frac{\pi(a|s)}{\pi_k(a|s)}, 1-\epsilon, 1+\epsilon\right)\right)\right] \quad (4)$$

where clip (x, l, u) = min(max(x, l), u). As evident from the second term in this objective, PPO limits the variance between successive policies by eliminating the motivation for the probability ratio $\pi(a|s)/\pi_k(a|s)$ to exceed the predefined clipping range $[1-\epsilon, 1+\epsilon]$. Ultimately, the external minimization ensures that Equation (4) functions as a lower bound for the surrogate objective [52].

3.2. MITRE ATT&CK Scenario

The MITRE ATT&CK framework is a comprehensive knowledge base and architecture that facilitates understanding of the strategies, tactics, and operational procedures adopted by cyber adversaries across the entire cyber kill chain [53]. Developed by the MITRE Corporation, a federally funded nonprofit organization that conducts research and development and offers government research and development services, the ATT&CK framework leverages actual field observations to provide a globally accessible knowledge repository of adversary tactics and strategies. The framework is continually updated and maintained by a community of cybersecurity experts from government, industry, and academia. ATT&CK provides a common language for cybersecurity professionals to share threat intelligence and collaborate on defense strategies. By providing a detailed understanding of adversary behavior, ATT&CK helps organizations to better defend against cyber-attacks and improve their overall cybersecurity posture. ATT&CK has become an important reference for cybersecurity professionals worldwide, and its impact is visible in the development of security solutions and the integration of threat intelligence into security operations [54].

This study implemented and simulated a scenario of the Ajax security team, a real-life cyber-attack, by replicating the tactics and techniques used [55]. The techniques of in Ajax security team were implemented as actions, and process acquisition was used to construct a simulation framework.

3.2.1. States

The MDPs consist of states, actions, and rewards. The state represents the status of the personal computer (PC) in the network. In this study, we present four factors that affect the PC from cyber-attack as indicated in Table 1. The state components include (1) the number of ports that are blocked by firewalls, (2) a list of authorized PCs that can obtain each other's administrator privileges without credentials, (3) vulnerabilities that can expose credentials of PCs connected to the user's PC and (4) credentials that allow the user to obtain administrator privileges. Without keyboard security, keylogging is possible, and credentials may be stored in web browsers. Opening a malicious email may result in a certain probability of virus infection. In DRL, agents are endowed with a comprehensive understanding of the environmental state to enable effective decision-making. The state comprises a set of values that characterize the environmental features pertinent to the agent's objectives.

Table 1. State components.

No.	State
1	Number of ports
2	List of authorized PCs
3	Keyboard Security
4	Web Credential

3.2.2. Actions

The attacker has four options for actions as shown in Table 2. (1) Try port access through open ports, (2) spoofing login by impersonating an authorized PC's IP address, (3) key logging to obtain credentials when keyboard security is not in place, and (4) accessing web-stored credentials. First, choose an exploit attack to target the vulnerability of the PC object and use a port scan to find open ports to attack the target PC object. The attacker uses four attributes as the state: the number of connected nodes and open ports discovered in the previous attacks, the presence of keyboard security, and the presence of web credentials. Based on the current state, the attacker selects the most rational action to attack the target PC.

Table 2. Action components.

No.	Action
1	Opened Port Attack
2	Spoofing
3	Key logging
4	Access Web Credential

In DRL for cyber-attack simulation, multiple parameters, including the selection of a vulnerability or user credentials, must be specified to execute actions effectively. Each action is subject to specific prerequisites, such as identifying the target host or having the necessary credentials. The outcomes of each action can potentially result in the discovery of new hosts, the acquisition of sensitive information, or the compromise of additional hosts.

In our exploration of cybersecurity scenarios, we have drawn upon the MITRE ATT&CK framework, specifically focusing on the Ajax security team scenario. While the MITRE ATT&CK framework offers a diverse array of attack techniques, we found ourselves confronted with the intricate nature of our computing environment, making it impractical to employ all available techniques. Consequently, we made strategic decisions to narrow our focus to a select set of representative attack techniques. These chosen techniques include open port attacks, spoofing, keylogging, and accessing web credentials. By concentrating on these specific methods, we aim to gain insights into the vulnerabilities and potential security breaches that may arise within our system, ensuring a more targeted and comprehensive approach to cybersecurity analysis.

3.2.3. Rewards

In DRL-based cyberattack simulation, the reward system provides feedback to the agent on the effectiveness of its actions in the environment. The reward function incorporates the cost of the agent's actions and their impact on the penetration testing process. The environmental reward value for the agent's present action is evaluated based on the cost of the action and its variance. In our study, a positive reward (+1) is given if the agent successfully owned the node, while a negative reward (-1) is issued otherwise. This feedback guides the agent towards optimal behavior and enhances its ability to adapt to dynamic environmental conditions.

3.3. Simulation Framework

In this study, the design and implementation of the simulated cyber-attack scenario were conducted using the widely adopted PyTorch deep learning framework. To construct the simulated environment, classes for the PC, attacker, and environment were defined and effectively employed. The environment was composed of a client PC and four target PCs, as illustrated in Figure 1. The attacker acquired essential attack information from the client PC and proceeded to initiate a sequence of attacks aimed at infiltrating and gaining control over the target PCs. The simulation concluded when the attacker successfully compromised all PCs within the network environment. This approach allowed for a controlled and systematic examination of cyber-attack dynamics, providing valuable insights into the vulnerabilities and potential security breaches within the simulated network.

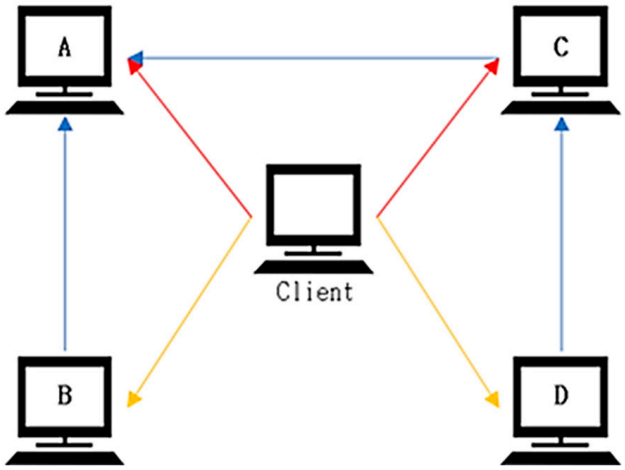

Figure 1. Simulated Network Environment.

The simulation process is divided into two main stages: node (PC) discovery and node attack. The node discovery stage, as illustrated in Figure 2 can be enumerated as following steps:

1. Node Discovery Initiation:
   - The attacker begins the process in the node discovery stage.
   - The objective is to search for a target node to initiate an attack.
2. Exploitation Attempt:
   - The attacker attempts to exploit a vulnerability in the targeted PC.
   - Success in exploiting the vulnerability grants access to credentials for the discovered nodes.
3. Credential Acquisition:
   - If the exploitation is successful, the attacker obtains the necessary credentials for accessing the discovered nodes.
   - These credentials are crucial for further actions within the network.
4. Failed Exploitation Scenario:
   - In case the attacker fails to exploit the vulnerability, they are unable to obtain credentials.
   - However, they can still identify and list connected nodes during this phase.
5. Preparation for Further Attacks:
   - Before proceeding to attack the discovered nodes, the attacker must go through this process.
   - The completion of this process is essential for discovering additional nodes and acquiring the credentials needed for network access.

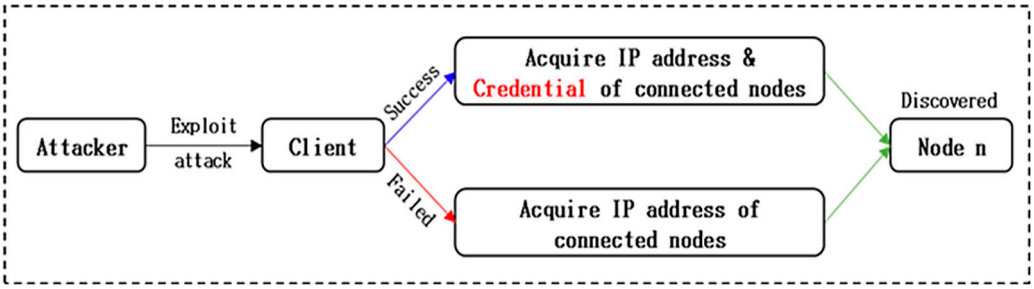

Figure 2. Process of Discovering Node.

The node attack stage, as illustrated in Figure 3 can be enumerated as following steps:

1. Focus on Discovered Nodes:
   - In the second stage of the simulation, the attacker shifts their focus to attacking the nodes previously discovered.
2. Immediate Control with Credentials:
   - If the attacker already possesses credentials for a discovered node, they can directly take control of it.
   - No additional attacks are required in this case.
3. Spearphishing Attack:
   - If the attacker lacks credentials for a node, they initiate a spearphishing attack as the initial step.
   - The goal is to gain access to the targeted node.
4. Node Infection Identification:
   - After the spearphishing attack, the attacker identifies whether the targeted node is infected or not.
5. Cycle of Attack Methods:
   - In the absence of credentials, the attacker employs a cycle of four attack methods:
   - Key logging.
   - Credential theft from web browsers.
   - Open port attack.
   - Spoofing.
6. Iterative Attack Process:
   - The attacker alternates between these attack methods until they successfully obtain the credentials for the targeted node.
7. Gaining Control:
   - Upon successful credential acquisition, the attacker gains control of the node.
8. Utilization for Further Attacks:
   - With control over the node, the attacker can utilize it as a base to conduct additional attacks on other nodes within the network.

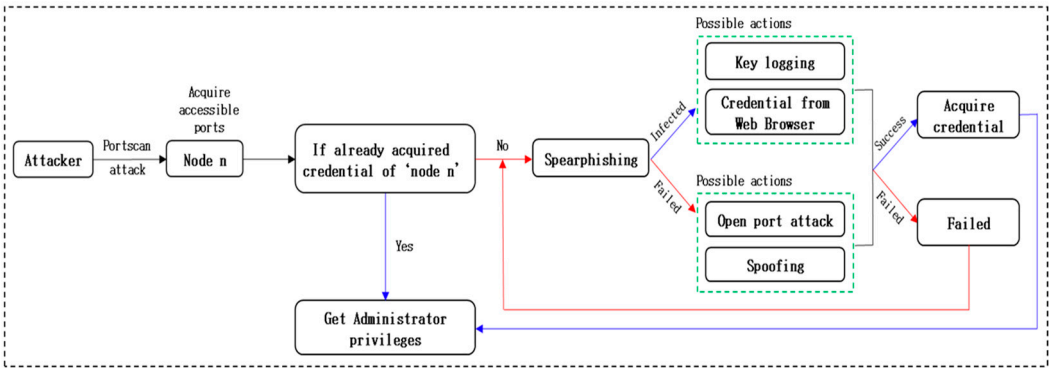

Figure 3. Process of Attacking Node.

It's crucial to emphasize that the attack methods utilized in this simulation are grounded in real-world cyber-attack tactics and techniques. This approach ensures a realistic representation of the actual threats encountered by computer networks. By simulating these attacks in a controlled environment, the opportunity arises to develop and test effective defense strategies against them.

The simulation involves a two-stage process: node discovery and node attack. This method offers a comprehensive and realistic environment for evaluating the performance of

deep reinforcement learning agents in cyber-attack scenarios. In our simulation framework, the successful acquisition of administrator privileges through an attack on a node (PC) is defined as "owning" that node. The corresponding rewards in the simulation reflect this; a successful ownership yields a reward of +1 for the attacker, while a failed attempt results in a reward of −1. This reward structure enables the evaluation of the agent's performance and effectiveness in navigating and securing the simulated network environment.

4. Results

This section outlines the outcomes of a comparative analysis between DRL agents and a baseline RL algorithm, gauged through cumulative rewards. The findings underscore the efficacy of DRL methodologies in empowering agents to formulate more potent attack strategies. The simulation results elucidate the DRL agents' adeptness in acquiring and executing successful methods for infiltrating a simulated cyber environment, thereby accentuating the considerable potential of DRL algorithms within the cybersecurity domain. Graphical representations within this section portray the evolution of rewards garnered by agents across episodes, attack success rates, and iteration counts, serving to illustrate the agents' progressive mastery in attaining elevated rewards throughout episodes.

The study incorporates three DRL algorithms—DQN, actor–critic, and PPO—contrasted against the tabular Q-learning algorithm serving as the baseline. Q-learning, being a conventional RL algorithm devoid of neural network utilization, sets the benchmark. The comparative analysis furnishes insights into the advantageous attributes of DRL methodologies for cybersecurity tasks when compared with the performance of the traditional Q-learning algorithm.

4.1. Comparison of Reward Obtained by RL Algorithms

Figure 4 illustrates the average reward achieved by the DRL algorithm within the network environment employed for the experiment. Initially, the DRL algorithm yields a low average reward as the attacker employs random actions in attempting to compromise the target node, resulting in diminished success rates. However, with progressing training epochs, the attacker incrementally acquires knowledge of the optimal policy, favoring actions with heightened probabilities of success. Consequently, the average reward from the DRL algorithm escalates, signifying an augmented success rate in seizing control of the target node.

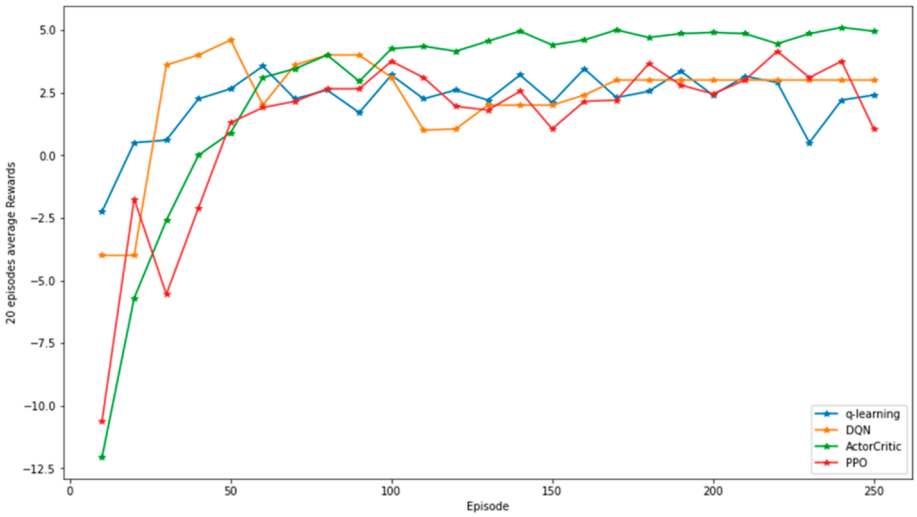

Figure 4. Average Rewards Obtained by Each RL Agent.

Examining the graph reveals fluctuations in the algorithm's average reward, attributed to sporadic entrapment in local minima where the optimal policy remains elusive. Never-

theless, as training perseveres, the algorithm can overcome these local minima, ultimately advancing its overall performance.

In the outcome analysis, the tabular Q-learning algorithm exhibits superior initial performance in comparison to DRL algorithms due to its utilization of dynamic programming, storing information in a table without employing a neural network. Nevertheless, as the volume of accumulated data increases and training progresses through 100 and 150 episodes, the performance of Q-learning converges with that of alternative algorithms. Furthermore, in scenarios where the state and action space expand, Q-learning demands protracted training periods and substantial memory resources, rendering it considerably more unstable in comparison to other algorithms.

Contrastingly, the deep Q-network (DQN) algorithm, despite employing soft updates, demonstrates a propensity to become ensnared in local minima beyond 150 episodes as shown in Table 3. Once trapped in such minima, performance persists even with continued training unless the algorithm successfully extricates itself. Conversely, the actor–critic algorithm, a policy-based approach, exhibits robust performance in the experiment by enabling more flexible decision-making compared to value-based algorithms such as Q-learning and DQN.

Table 3. Average Reward Obtained by RL Algorithms.

Algorithms	**Episodes**				
	50	**100**	**150**	**200**	**250**
Q-learning	2.5	2.6	2.3	2.2	2.1
DQN	4.5	2.7	2.4	2.6	2.6
Actor–critic	0.2	4.2	3.8	4.4	4.8
PPO	−2.5	3.0	1.6	2.4	1.7

This section presents the outcomes of an experiment employing three DRL algorithms, specifically DQN, actor–critic, and PPO. The presentation includes a depiction of cumulative rewards garnered by the agents across individual episodes, revealing a discernible pattern wherein the agents systematically enhance their ability to attain higher rewards as each episode unfolds. The findings substantiate that DRL methodologies empower the agents to adeptly assimilate efficient attack techniques, thereby accentuating the considerable potential of DRL algorithms within the domain of cybersecurity. In aggregate, the implemented DRL policies demonstrate proficiency in executing strategic maneuvers conducive to the successful infiltration of the simulated cyber environment.

4.2. Comparison of Success Rate Obtained by RL Algorithms

The evaluation of DRL algorithm efficacy transpired within a network environment, with the results presented in Figure 5. Success rate, denoting the attacker's proficiency in infiltrating and taking control of all PCs within the simulated network, served as the metric. The graphical representation illustrates a congruent trend between the success rate and reward (refer to Figure 4), wherein elevated success rates correlate with increased rewards. This parallelism across algorithms substantiates the effectiveness of DRL methodologies in achieving the objective of infiltrating a simulated cyber environment. The success rate, being a pivotal metric, assumes significance in appraising the performance of DRL algorithms within cybersecurity applications, serving as a reflective measure of the agent's capability to fulfill its objectives.

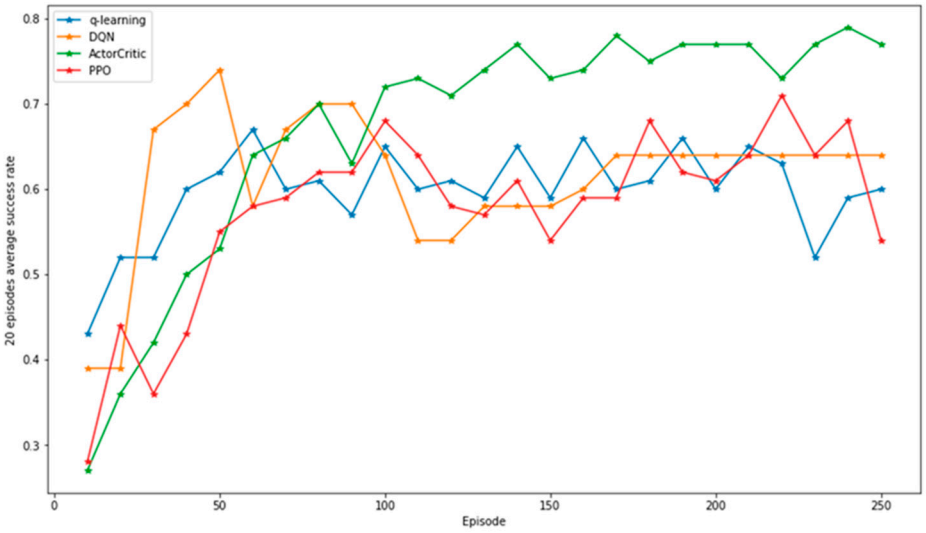

Figure 5. Average Success Rate Obtained by Each RL Agent.

In the preliminary phase of training, specifically at 50 episodes, the DQN algorithm manifests the highest average success rate, registering at 0.72, surpassing the rates of 0.61, 0.53, and 0.52 for Q-learning, actor–critic, and PPO, respectively, as detailed in Table 4. The expeditious learning exhibited by the DQN algorithm during this phase is attributed to its deterministic action selection mechanism. Throughout the learning process, the algorithm systematically opts for actions with the highest Q-value with a probability of 1-ε, enabling commendable performance even with limited learning iterations. However, in instances of encountering local minima, the algorithm's performance becomes constrained, impeding its capacity to transcend such minima and resulting in marginal performance enhancements despite continued learning. Conclusively, by the conclusion of the training regimen, the actor–critic algorithm attains the pinnacle with the highest average success rate of 0.78.

Table 4. Average Success Rate Obtained by RL Algorithms.

Algorithms	Episodes				
	50	100	150	200	250
Q-learning	0.61	0.65	0.58	0.60	0.59
DQN	0.72	0.62	0.57	0.63	0.63
actor–critic	0.53	0.71	0.73	0.75	0.78
PPO	0.52	0.68	0.53	0.61	0.54

A comprehensive assessment of algorithmic performance necessitates the concurrent evaluation of both the success rate and the obtained reward by the agent. In totality, the DRL algorithms implemented in this experimental paradigm have substantiated their efficacy in proficiently training an agent to achieve predefined objectives within a simulated cyber environment. The outcomes imply the viability of DRL methodologies in augmenting the capabilities of cybersecurity systems, empowering agents to assimilate knowledge and adapt to evolving attack scenarios effectively.

4.3. Comparison of Iteration per Episode by RL Algorithms

This section provides insight into how the different algorithms perform in terms of learning efficiency. Notably, the learning speed diverges among algorithms, and an iteration is defined as an attempted action aimed at completing a single episode. Preceding the learning phase, the agent engages in random actions, resulting in a higher iteration count. Figure 6 illustrates that the actor–critic algorithm exhibits optimal learning efficiency among the assessed algorithms, necessitating the fewest iterations to conclude each episode.

Conversely, the DQN algorithm exhibits limited improvement in learning efficiency, as the iteration count remains nearly constant throughout the learning trajectory.

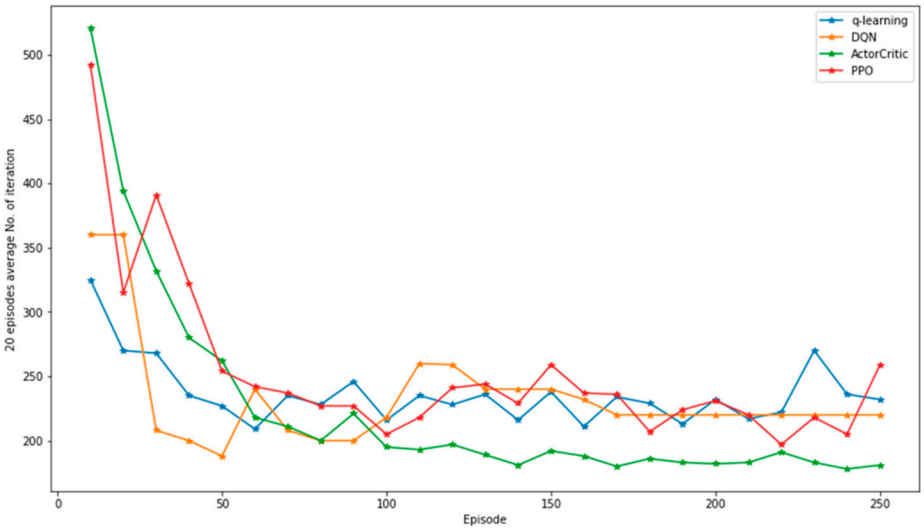

Figure 6. Number of Iterations Taken by Each RL Algorithm.

Significantly, the number of iterations required to complete an episode serves as an indicator of the agent's adeptness in traversing complex state and action spaces. A reduction in iteration count signifies an enhancement in the agent's performance, indicating its capacity to discern optimal actions leading to elevated rewards.

Furthermore, an agent's capability to efficiently learn from its environment assumes pivotal importance in realizing intended objectives in real-world scenarios. Consequently, the insights gleaned from this section hold practical relevance, particularly in domains such as cybersecurity, where agents grapple with the need for informed decision-making in intricate and dynamic environments.

The primary objective of this experiment is to assess the efficacy of diverse algorithms in training agents to accomplish predefined objectives. As depicted in Table 5, the actor–critic algorithm emerges as the most efficient, demanding the fewest iterations—171—to conclude an episode, in contrast to Q-learning, DQN, and PPO, which necessitated 240, 223, and 262 iterations, respectively. The constraints on the DQN algorithm's performance stem from its vulnerability to local minima, impeding effective performance enhancement. Additionally, the intrinsic value-based nature of the DQN algorithm engenders a dearth of diversity in its actions, consistently opting for the action with the highest cumulative reward value. This proclivity restricts its capacity to explore alternative strategies, further limiting its adaptive capabilities.

Table 5. Average Iteration per Episode by RL Algorithms.

Algorithms	**Episodes**					
	20	**50**	**100**	**150**	**200**	**250**
Q-learning	326	227	220	243	232	240
DQN	359	173	221	242	223	223
actor–critic	531	262	202	191	178	171
PPO	496	260	213	261	231	262

These findings offer valuable perspectives on the performance dynamics exhibited by distinct algorithms throughout the learning process, serving as a guiding framework for the judicious selection of the most suitable algorithm tailored to a specific task. Furthermore, these outcomes stand to benefit researchers and practitioners within the field of RL,

providing essential insights to inform the development of efficient algorithms and facilitate the optimization of their performance across diverse applications.

4.4. Convergence Test of RL Algorithms

In this section, the convergence test was conducted across multiple episodes provides insightful data on the learning efficacy of four prominent DRL algorithms. The convergence measure, apparently reflecting the algorithms' ability to learn from the environment and make optimal decisions, was used to gauge the performance and efficacy of these methods over a series of 250 episodes as shown in Figure 7.

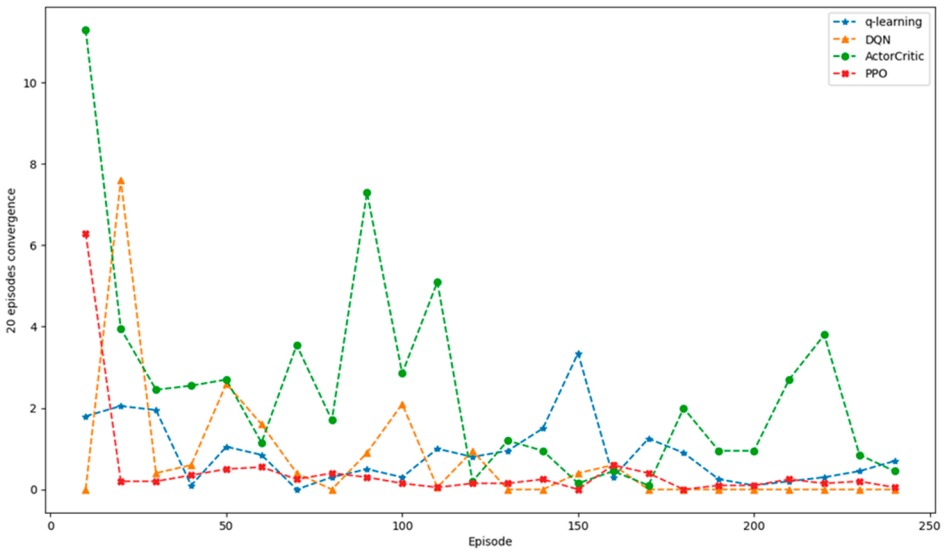

Figure 7. Convergence Test of D/RL Algorithms Used.

As shown in Table 6, the Q-Learning starts with a convergence measure of 1.89 at episode 20, drops to its lowest at episode 100 with 0.39, peaks significantly at episode 150 to 3.52, and then decreases again, reaching 0.56 by episode 250. DQN has no convergence measure at episode 20, which could indicate it started with optimal performance or the metric was not recorded. The performance worsens by episode 50 with a value of 2.74, slightly decreases by episode 100, and then improves considerably, reaching its best performance by episode 250 with a convergence measure of 0.14. The DQN shows a trend of improvement over time, implying that it might be more stable and learn more effectively as more episodes are completed. The actor–critic has the highest initial convergence measure at episode 20 with 11.3, suggesting a poor start compared to the others. However, it shows an improvement by episode 50, a slight increase by episode 100, and then a substantial improvement, reaching its lowest convergence measure at episode 150 with 0.47. The value goes up again at episode 200 and then drops by episode 250, indicating some volatility but overall trending towards better performance. Lastly, PPO begins with a convergence measure of 6.3 at episode 20, improves rapidly by episode 50, and continues to improve or maintain a low convergence measure throughout the remaining episodes, ending with 0.16 by episode 250. PPO demonstrates stable and consistent improvement out of all the algorithms, suggesting it is effective at learning and adapting over time with minimal performance fluctuations.

Table 6. Average Convergence Measure per Episode by RL Algorithms.

Algorithms	Episodes					
	20	**50**	**100**	**150**	**200**	**250**
Q-learning	1.89	1.52	0.39	3.52	0.23	0.56
DQN	0	2.74	2.25	0.68	0.21	0.14
actor–critic	11.3	2.91	3.24	0.47	1.14	0.35
PPO	6.3	0.31	0.27	0.12	0.24	0.16

In summary, the convergence test not only sheds light on the learning efficacy of the algorithms but also highlights the importance of algorithmic stability and adaptability in various learning environments. While each algorithm has its strengths, the PPO's superior performance in this test suggests that it could be the preferred choice in situations where rapid and consistent learning is essential. Future research could focus on exploring the mechanisms behind the observed performance patterns, particularly the factors contributing to PPO's robustness and actor–critic's variability, to further understand and exploit these algorithms' capabilities in complex and dynamic environments.

5. Discussion

In this study, we successfully designed and implemented a real-world cyber-attack scenario to DRL algorithms. This scenario was designed to simulate a real-world network environment and included a range of vulnerabilities and obstacles that an attacker might encounter. Our study shows that the DRL algorithms can be employed in cybersecurity approaches. By training agents to learn from experience and make optimal decisions, DRL algorithms can offer more adaptive and efficient approaches to dealing with complex cybersecurity problems. These approaches can be used to identify vulnerabilities and develop more effective defense mechanisms to mitigate potential threats. The successful implementation of our study not only provides valuable insights into the potential of DRL algorithms for cybersecurity but also demonstrates the importance of interdisciplinary research in addressing real-world problems. Our study involved the collaboration of cybersecurity experts and machine learning researchers, which was essential in designing and implementing an effective experimental framework.

The experimental results presented in this study demonstrate the effectiveness of using DRL algorithms in the field of cybersecurity. The comparison of different DRL algorithms, including DQN, actor–critic, and PPO, with the baseline algorithm of tabular Q-learning, showed that DRL algorithms were more efficient and effective in achieving the desired outcome of the simulated cyber environment. The graphs and tables presented in the study showed that DRL algorithms were able to acquire and execute effective tactics for accomplishing the goal of infiltrating a simulated cyber environment. Moreover, the results indicated that the DRL policy managed to acquire efficient attack techniques, demonstrating the potential of DRL algorithms in the field of cybersecurity.

The results showed that the actor–critic algorithm was the most efficient algorithm in terms of reward and success rate. The agent trained using the actor–critic algorithm was able to achieve a success rate of 0.78, requiring the fewest iterations (171) to complete an episode, and the highest average reward obtained by 4.8. This is likely due to its policy-based approach, which allows for more flexible decision-making compared to value-based algorithms such as Q-learning and DQN. However, it is important to note that the DQN algorithm initially showed the highest average success rate, indicating that it is effective at quickly learning optimal actions in a deterministic environment. Nevertheless, its performance is limited by local minima, which can cause the algorithm to get stuck and prevent it from effectively improving its performance. Therefore, in our study, the actor–critic algorithm was better able to explore different strategies and make more efficient use of the training data.

The baseline Q-learning algorithm showed better initial performance in obtaining reward than the other DRL algorithms, likely due to its use of dynamic programming,

which records it in a table without using a neural network. However, as data accumulates and training progresses, the performance of Q-learning catches up to that of the other algorithms. The use of a table to record state-action values can also be limiting, as the size of the table grows exponentially with the increase in the state and action space.

The results also show that the attack success rate and reward obtained by the agents are strongly correlated, with higher success rates leading to higher rewards. This finding further reinforces the effectiveness of DRL algorithms in achieving the objective of infiltrating a simulated cyber environment. The success rate is a crucial metric to evaluate the performance of DRL algorithms in cybersecurity applications as it indicates the agent's ability to accomplish its objective. By successfully infiltrating all the PCs in the network environment, the agent demonstrates its competence in accomplishing the task assigned to it. Therefore, the success rate serves as an important measure of the agent's efficacy in real-world cyber-attack scenarios. The strong correlation between success rate and reward also highlights the importance of optimizing the performance of the agent in both aspects to ensure its effectiveness.

Lastly, in the convergence test evaluating the learning efficacy of DRL algorithms, PPO demonstrated superior performance, exhibiting rapid improvement, and maintaining the lowest convergence measures consistently across all episodes. Q-Learning, while showing overall learning improvement, displayed significant performance fluctuations, particularly with a sharp peak at episode 150, indicating potential instability. DQN revealed a steady increase in learning efficacy over time, suggesting it is well-suited for tasks that allow for extended training periods. The actor–critic algorithm, despite a less efficient start, indicated a capability for substantial performance gains, though it also exhibited some volatility in learning progress. These findings underscore the varying dynamics of algorithmic learning efficacy offering valuable insights for selecting and tuning algorithms in environments where stability and adaptability are crucial.

In conclusion, the results of the experiment demonstrate the potential of DRL algorithms to be utilized in the field of cybersecurity. The findings provide insights into the performance of different algorithms in achieving the objective of infiltrating a simulated cyber environment and can inform the selection of the most appropriate algorithm for a given task. Further research can focus on the optimization of the DRL algorithms for specific tasks and the development of hybrid algorithms that combine the strengths of different DRL algorithms.

6. Conclusions

In this study, we successfully implemented a real-world cyber-attack scenario and demonstrated the potential of employment of DRL algorithms in the field of cybersecurity. With three DRL algorithms (DQN, actor–critic, and PPO) and a baseline RL algorithm (tabular Q-learning), we were able to evaluate and compare their performance in training agents to achieve the goal of infiltrating a simulated cyber environment.

The experimental results show that the actor–critic algorithm outperformed the other algorithms in terms of average success rate, learning efficiency, and reward obtained. This highlights the potential of policy-based algorithms such as actor–critic in making more flexible choices and achieving better performance in complex tasks. However, it is worth noting that the DQN algorithm initially showed faster learning speed than other algorithms due to its deterministic action selection. Nevertheless, it also faced limitations in performance due to the problem of local minima. This highlights the importance of selecting an appropriate algorithm that suits the specific task and environment.

In conclusion, we provided evidence that DRL algorithms can be a valuable tool in the field of cybersecurity. The experimental results demonstrate the potential of using DRL algorithms to train agents to achieve the goal of infiltrating a simulated cyber environment. The findings of this study contribute to the development of more effective and efficient defense strategies against cyber-attacks, which are becoming increasingly prevalent and

sophisticated. Finally, the study provides a valuable contribution to the growing body of research on the application of DRL algorithms in the field of cybersecurity.

7. Limitations and Future Works

While the integration of DRL in cybersecurity holds significant promise, certain limitations and challenges need to be addressed to enhance the effectiveness of these models. One key challenge lies in the dynamic nature of cyber threats requires the development of more sophisticated simulation environments that can dynamically adapt to real-time updates of attack patterns and network configurations. To overcome the resource-intensive nature of DRL training, future research should explore resource-efficient strategies, such as model compression and parameter sharing. Additionally, enhancing the transferability of DRL models from simulations to real-world environments demands the investigation of robust transfer learning strategies. Adversarial defense mechanisms, interpretable DRL models, and continuous learning architectures are also crucial focal points for future works, along with the development of validation and assurance frameworks to ensure the ethical and legal compliance of automated decision-making in cybersecurity. Collaborative efforts between cybersecurity experts and machine learning researchers will play a pivotal role in advancing these initiatives and addressing the multifaceted challenges at the intersection of DRL and cybersecurity.

Author Contributions: S.H.O. and J.K. conceived and conducted the experiments. J.H.N. and J.P. verified and advised the experimental outcome. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Research Foundation of Korea (NRF) (No. RS-2023-00221365) and Institute for Information & Communications Technology Promotion (IITP) (No. 2021-0-00796) grant funded by the Korea government (MSIT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author (status: privacy).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Enoch, S.Y.; Huang, Z.; Moon, C.Y.; Lee, D.; Ahn, M.K.; Kim, D.S. HARMer: Cyber-attacks automation and evaluation. *IEEE Access* **2020**, *8*, 129397–129414. [CrossRef]
2. Bahrami, P.N.; Dehghantanha, A.; Dargahi, T.; Parizi, R.M.; Choo KK, R.; Javadi, H.H. Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures. *J. Inf. Process. Syst.* **2019**, *15*, 865–889.
3. Li, L.; Fayad, R.; Taylor, A. Cygil: A cyber gym for training autonomous agents over emulated network systems. *arXiv* **2021**, arXiv:2109.03331.
4. Yoo, J.D.; Park, E.; Lee, G.; Ahn, M.K.; Kim, D.; Seo, S.; Kim, H.K. Cyber attack and defense emulation agents. *Appl. Sci.* **2020**, *10*, 2140. [CrossRef]
5. Sarker, I.H. Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview. *Secur. Priv.* **2022**, *6*, e295. [CrossRef]
6. Alshamrani, A.; Myneni, S.; Chowdhary, A.; Huang, D. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1851–1877. [CrossRef]
7. Franco, M.F.; Sula, E.; Huertas, A.; Scheid, E.J.; Granville, L.Z.; Stiller, B. SecRiskAI: A Machine Learning-Based Approach for Cybersecurity Risk Prediction in Businesses. In Proceedings of the 2022 IEEE 24th Conference on Business Informatics (CBI), Amsterdam, The Netherlands, 15–17 June 2022; IEEE: New York, NY, USA, 2022; Volume 1, pp. 1–10.
8. Meliboev, A.; Alikhanov, J.; Kim, W. Performance evaluation of deep learning based network intrusion detection system across multiple balanced and imbalanced datasets. *Electronics* **2022**, *11*, 515. [CrossRef]
9. Talukder, M.A.; Hasan, K.F.; Islam, M.M.; Uddin, M.A.; Akhter, A.; Yousuf, M.A.; Alharbi, F.; Moni, M.A. A dependable hybrid machine learning model for network intrusion detection. *J. Inf. Secur. Appl.* **2023**, *72*, 103405. [CrossRef]

10. Ahsan, M.; Nygard, K.E.; Gomes, R.; Chowdhury, M.M.; Rifat, N.; Connolly, J.F. Cybersecurity threats and their mitigation approaches using Machine Learning—A Review. *J. Cybersecur. Priv.* **2022**, *2*, 527–555. [CrossRef]
11. Sarker, I.H.; Kayes, A.S.M.; Badsha, S.; Alqahtani, H.; Watters, P.; Ng, A. Cybersecurity data science: An overview from machine learning perspective. *J. Big Data* **2020**, *7*, 41. [CrossRef]
12. Haider, N.; Baig, M.Z.; Imran, M. Artificial Intelligence and Machine Learning in 5G Network Security: Opportunities, advantages, and future research trends. *arXiv* **2020**, arXiv:2007.04490.
13. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. *Mitre Att&ck: Design and Philosophy*; Technical Report; The MITRE Corporation: Bedford, MA, USA, 2018.
14. Sen, R.; Heim, G.; Zhu, Q. Artificial Intelligence and Machine Learning in Cybersecurity: Applications, Challenges, and Opportunities for MIS Academics. *Commun. Assoc. Inf. Syst.* **2022**, *51*, 28. [CrossRef]
15. Pinto, A.; Herrera, L.C.; Donoso, Y.; Gutierrez, J.A. Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors* **2023**, *23*, 2415. [CrossRef]
16. Duddu, V. A survey of adversarial machine learning in cyber warfare. *Def. Sci. J.* **2018**, *68*, 356. [CrossRef]
17. Shaukat, K.; Luo, S.; Varadharajan, V.; Hameed, I.A.; Chen, S.; Liu, D.; Li, J. Performance comparison and current challenges of using machine learning techniques in cybersecurity. *Energies* **2020**, *13*, 2509. [CrossRef]
18. Piplai, A.; Anoruo, M.; Fasaye, K.; Joshi, A.; Finin, T.; Ridley, A. Knowledge guided Two-player Reinforcement Learning for Cyber Attacks and Defenses. In Proceedings of the International Conference on Machine Learning and Applications, Nassau, Bahamas, 12–14 December 2022.
19. Applebaum, A.; Miller, D.; Strom, B.; Korban, C.; Wolf, R. Intelligent, automated red team emulation. In Proceedings of the 32nd Annual Conference on Computer Security Applications, Los Angeles, CA, USA, 5–8 December 2016; pp. 363–373.
20. Meier, R.; Lavrenovs, A.; Heinäaro, K.; Gambazzi, L.; Lenders, V. Towards an AI-powered Player in Cyber Defence Exercises. In Proceedings of the 2021 13th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 25–28 May 2021; IEEE: New York, NY, USA, 2021; pp. 309–326.
21. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. *arXiv* **2016**, arXiv:1606.01540.
22. Caminero, G.; Lopez-Martin, M.; Carro, B. Adversarial environment reinforcement learning algorithm for intrusion detection. *Comput. Netw.* **2019**, *159*, 96–109. [CrossRef]
23. Chen, T.; Liu, J.; Xiang, Y.; Niu, W.; Tong, E.; Han, Z. Adversarial attack and defense in reinforcement learning-from AI security view. *Cybersecurity* **2019**, *2*, 11. [CrossRef]
24. Bhattacharya, A.; Ramachandran, T.; Banik, S.; Dowling, C.P.; Bopardikar, S.D. Automated adversary emulation for cyber-physical systems via reinforcement learning. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
25. Zhou, S.; Liu, J.; Hou, D.; Zhong, X.; Zhang, Y. Autonomous penetration testing based on improved deep q-network. *Appl. Sci.* **2021**, *11*, 8823. [CrossRef]
26. Al Amin, M.A.R.; Shetty, S.; Kamhoua, C. Cyber Deception Metrics for Interconnected Complex Systems. In Proceedings of the 2022 Winter Simulation Conference (WSC), Singapore, 11–14 December 2022; IEEE: New York, NY, USA, 2022; pp. 473–483.
27. Huang, Y.; Huang, L.; Zhu, Q. Reinforcement learning for feedback-enabled cyber resilience. *Annu. Rev. Control.* **2022**, *53*, 273–295.
28. Rathore, H.; Sahay, S.K.; Nikam, P.; Sewak, M. Robust android malware detection system against adversarial attacks using q-learning. *Inf. Syst. Front.* **2021**, *23*, 867–882. [CrossRef]
29. Sethi, K.; Madhav, Y.V.; Kumar, R.; Bera, P. Attention based multi-agent intrusion detection systems using reinforcement learning. *J. Inf. Secur. Appl.* **2021**, *61*, 102923. [CrossRef]
30. Nguyen, T.T.; Reddi, V.J. Deep reinforcement learning for cyber security. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 3779–3795. [CrossRef] [PubMed]
31. Ambalavanan, V. Cyber threats detection and mitigation using machine learning. In *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*; IGI Global: Hershey, PL, USA, 2020; pp. 132–149.
32. Standen, M.; Lucas, M.; Bowman, D.; Richer, T.J.; Kim, J.; Marriott, D. Cyborg: A gym for the development of autonomous cyber agents. *arXiv* **2021**, arXiv:2108.09118.
33. Walter, E.; Ferguson-Walter, K.; Ridley, A. Incorporating deception into cyberbattlesim for autonomous defense. *arXiv* **2021**, arXiv:2108.13980.
34. Ibrahim, M.; Elhafiz, R. Security Analysis of Cyber-Physical Systems Using Reinforcement Learning. *Sensors* **2023**, *23*, 1634. [CrossRef] [PubMed]
35. Dutta, A.; Chatterjee, S.; Bhattacharya, A.; Halappanavar, M. Deep Reinforcement Learning for Cyber System Defense under Dynamic Adversarial Uncertainties. *arXiv* **2023**, arXiv:2302.01595.
36. Applebaum, A.; Dennler, C.; Dwyer, P.; Moskowitz, M.; Nguyen, H.; Nichols, N.; Park, N.; Rachwalski, P.; Rau, F.; Webster, A.; et al. Bridging automated to autonomous cyber defense: Foundational analysis of tabular q-learning. In Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security, Los Angeles, CA, USA, 11 November 2022; pp. 149–159.
37. Elderman, R.; Pater, L.J.; Thie, A.S.; Drugan, M.M.; Wiering, M.A. Adversarial Reinforcement Learning in a Cyber Security Simulation. In Proceedings of the 9th ICAART, Porto, Potugal, 24–26 February 2017; pp. 559–566.

38. Seifert, C.; Betser, M.; Blum, W.; Bono, J.; Farris, K.; Goren, E.; Grana, J.; Holsheimer, K.; Marken, B.; Neil, J.; et al. *CyberBattleSim, version 1.1*; Microsoft Defender Research Team: Redmond, WA, USA, 2021.
39. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
40. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
41. Clifton, J.; Laber, E. Q-learning: Theory and applications. *Annu. Rev. Stat. Its Appl.* **2020**, *7*, 279–301. [CrossRef]
42. Nair, A.; Srinivasan, P.; Blackwell, S.; Alcicek, C.; Fearon, R.; De Maria, A.; Panneershelvam, V.; Suleyman, M.; Beattie, C.; Petersen, S. Massively parallel methods for deep reinforcement learning. *arXiv* **2015**, arXiv:1507.04296.
43. Kumar, R.; Aggarwal, R.K.; Sharma, J.D. Energy analysis of a building using artificial neural network: A review. *Energy Build.* **2013**, *65*, 352–358. [CrossRef]
44. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
45. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
46. Fan, J.; Wang, Z.; Xie, Y.; Yang, Z. A theoretical analysis of deep Q-learning. In Proceedings of the Learning for Dynamics and Control, Online, 11–12 June 2020; pp. 486–489.
47. Witten, I.H. An adaptive optimal controller for discrete-time Markov environments. *Inf. Control.* **1977**, *34*, 286–295. [CrossRef]
48. Barto, A.G.; Sutton, R.S.; Anderson, C.W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybern.* **1983**, *13*, 834–846. [CrossRef]
49. Grondman, I.; Busoniu, L.; Lopes, G.A.; Babuska, R. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2012**, *42*, 1291–1307. [CrossRef]
50. Sutton, R.S. Learning to predict by the methods of temporal differences. *Mach. Learn.* **1988**, *3*, 9–44. [CrossRef]
51. Baird, L. Residual algorithms: Reinforcement learning with function ap-proximation. In *Machine Learning Proceedings 1995*; Morgan Kaufmann: Burlington, MA, USA, 1995; pp. 30–37.
52. Queeney, J.; Paschalidis, Y.; Cassandras, C.G. Generalized proximal policy optimization with sample reuse. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 11909–11919.
53. Alexander, O.; Belisle, M.; Steele, J. *MITRE ATT&CK®for Industrial Control Systems: Design and Philosophy*; The MITRE Corporation: Bedford, MA, USA, 2020; p. 29.
54. Strom, B.E.; Battaglia, J.A.; Kemmerer, M.S.; Kupersanin, W.; Miller, D.P.; Wampler, C.; Whitley, S.; Wolf, R.D. *Finding Cyber Threats with ATT&CK-Based Analytics*; Technical Report No. MTR170202; The MITRE Corporation: Bedford, MA, USA, 2017.
55. The MITRE Corporation. Ajax Security Team, The MITRE Corporation. 2016. Available online: https://attack.mitre.org/groups/G0130/ (accessed on 5 December 2022).