Multi-agent Reinforcement Learning for Cybersecurity: Approaches and Challenges

Salvo Finistrella*, Stefano Mariani and Franco Zambonelli

University of Modena and Reggio Emilia, Reggio Emilia, Italy

Abstract

In the face of the rapidly evolving threat landscape, traditional security measures often lag behind with sophisticated cyber attacks. Through a review of existing literature, we examine the shortcomings of conventional cybersecurity methods, highlighting the need for Reinforcement Learning based methods. Our study classifies various RL approaches in cybersecurity, aimed to enhance detection, mitigation, and response capabilities, along two dimensions: the RL technique used, and the network configuration. Moving forward, we emphasise the importance of further research and development to address challenges such as model complexity, sample efficiency, and vulnerabilities to adversarial attacks.

Keywords

Reinforcement learning, Cybersecurity, Multi-agent system, DoS attack mitigation, Intrusion Detection System (IDS)

1. Introduction

On a global scale, projections indicate that the cost of cybercrime will surpass 8 trillion dollars, cementing its status as the world's third-largest and most rapidly expanding economy [1]. Such cost includes damage and destruction of data, stolen money, lost productivity, theft of intellectual property, theft of personal and financial data, fraud, post-attack disruption, forensic investigation, restoration and deletion of hacked data and systems and reputational harm. Given this and the escalating nature of cyber threats, the integration of *Reinforcement Learning* techniques emerges as a promising strategy to fortify cybersecurity defences [2, 3, 4].

RL is an area of machine learning where an active entity (called *agent*) is given the goal of learning a behavioural policy through experience, by interacting with an *environment* through trial and error. While interacting with such an environment, the agent may get *rewards* for useful actions that advance it towards the task to be accomplished, or punishments for actions that steer it away. By aiming at maximising the accumulated rewards, the agent learns which actions lead to favourable outcomes and adjusts its behaviour accordingly [5].

The motivation for employing RL in cybersecurity stems from its ability to introduce *dynamic* and *adaptive* defence mechanisms. Traditional approaches struggle to keep pace with the rapidly evolving threat landscape, whereas RL enables security systems to learn from experience and

WOA 2024: 25th Workshop "From Objects to Agents", July 8-10, 2024, Forte di Bard (AO), Italy

*Corresponding author.

salvo.finistrella@unimore.it (S. Finistrella); stefano.mariani@unimore.it (S. Mariani); franco.zambonelli@unimore.it (F. Zambonelli)

https://smarianimore.github.io (S. Mariani)

0009-0004-8597-9031 (S. Finistrella); 0000-0001-8921-8150 (S. Mariani); 0000-0002-9421-8566 (F. Zambonelli)

adjust their strategies timely, and *autonomously*. By automating response mechanisms, RL algorithms can not only detect, but also analyse and mitigate cyber threats without human intervention, significantly reducing response times and potential damages. Furthermore, RL offers continuous learning capabilities, allowing security systems to adapt to novel threats, by continuously updating their knowledge and strategies based on new data and experiences.

In this article, we present a classification of RL methods tailored to bolster security measures across diverse domains, encompassing single-agent paradigms as well as multi-agent ones. We classify RL techniques as applied to host-based, network-based, and centralised network-based configurations augmented with Software-Defined Networking (SDN, see 4.2). By summarising these approaches, we aim to provide a road-map for practitioners and researchers navigating the complex landscape of RL applications in cybersecurity.

2. Motivation & Background

In today's ever-evolving cyber landscape, traditional security measures often fall short in defending against new threats. The reasons are many.

- **Static defences**: they rely on fixed rules, threat signatures, and predefined attack patterns for detection and prevention. Thus they are inherently limited to recognising known threats and vulnerabilities [6].
- **Lack of contextual understanding**: they operate within rigid frameworks that do not account for the broader context of an attack, failing to adapt to evolving scenarios. For instance, these systems cannot adapt to new attack vectors or understand the nuances of different operational environments, making it difficult to identify and respond to sophisticated and previously unseen attacks [7].
- **Limited scalability**: they may become overwhelmed by the sheer volume of data to analyse and the diversity of threats to address [8].
- **Inadequate response times**: they often rely on manual intervention to address security incidents, introducing obvious delays [8].

These challenges highlight the need for more adaptive and responsive cybersecurity strategies, and RL offers a dynamic and flexible solution to meet these needs. By harnessing RL, security systems can autonomously adjust to emerging threats, which is particularly crucial for countering *zero-day* attacks—exploiting previously unknown vulnerabilities, leaving victims defenceless with no time to prepare or patch the flaw [9].

- **Dynamic Threat Detection**: RL algorithms continuously learn and adapt to new threat patterns through interactions with the environment, improving accuracy and efficiency.
- **Real-Time Threat Analysis**: RL can identify and neutralise threats with unparalleled speed and efficiency.
- **Enhanced Decision-Making**: RL learns from experience to make smarter decisions, analysing data patterns and detecting anomalies in threat identification.

Table 1
Limitations of traditional approaches to cybersecurity (columns), and how RL helps improving (rows).

	Adaptive Defences	Context Understanding	Scalability	Response Times
Dynamic Detection	✓		✓	
Real-time Analysis	✓	✓		✓
Enhanced Decision-making		✓		
Efficiency through Automation			✓	✓
Adversarial Evasion	✓			✓

- **Efficiency Through Automation**: RL can remove manual intervention from basic tasks like malware scanning and network traffic monitoring, enhancing consistency and precision.
- **Minimising False Positives**: RL effectively differentiates between genuine threats and routine activities, reducing false alarms over time.

Table 1 summarises which RL capabilities enable surpassing what current limitations.

3. Reinforcement Learning in a Cybersecurity Environment

Reinforcement Learning represents a powerful paradigm in the domain of artificial intelligence, enabling agents to learn optimal behaviour through interaction with their environment. Mimicking the trial-and-error learning process observed in humans and animals, RL algorithms iteratively explore and exploit their environment to maximise cumulative rewards. Rewards can be *sparse* or *dense*. The first are given infrequently, making it challenging for the agent to learn desired behaviours. Conversely, dense rewards are provided more frequently, facilitating quicker learning. RL has applications in diverse fields, from robotics and gaming to finance and healthcare, where systems must autonomously adapt to uncertain and dynamic environments.

In the domain of cybersecurity, the core RL concepts such as state and environment observation, action selection, policy optimisation, reward mechanisms, and goal-driven strategies, can be instantiated as follows.

- **Environment state and observation**: RL algorithms rely on observing the (possibly, hidden) state of the environment to make decisions. As depicted in Figure 1, in cybersecurity the environment is the network environment, within which several appliances generate the data constituting the *state*: firewalls, Intrusion Detection Systems (IDSs), Intrusion Prevention Systems (IPSs), proxy servers, sniffers, Operating Systems, and other software. This entails monitoring various data such as network traffic patterns, system logs, software configurations, and user behaviour. The generated observations serve as inputs for the agents to learn.

- **Action selection**: after observing the state, the RL agent selects actions based on its learned policy, there including triggering alarms, deploying patches, updating security configurations, isolating compromised systems, alerting security personnel, block services, and dropping packets. The RL environment evaluates the efficacy of these actions, assessing whether the state of security has improved or deteriorated. *Rewards* or penalties are then issued accordingly (see below).
- **Policy optimisation**: RL agents continuously refine their decision-making policy through *trial and error* and rewards, aiming to maximise short or long-term rewards.
- **Reward mechanisms**: *Rewards* provide feedback to the agent, indicating the efficacy of its actions. In cybersecurity, their primary goal is to incentivise actions leading to successful detection and mitigation of threats while penalising those that fall short. To achieve this, the reward function is meticulously crafted to align with the objectives of the Intrusion Detection System (IDS). Its design could involve rewarding accurate identification and swift response to threats while admonishing false positives and negatives. The specifics of this function's formulation and computation vary, contingent upon the intricacies of the RL algorithm and the objectives of the security system at hand. Subsection 4.3 provides practical examples of rewards.
- **Goal-driven strategies**: RL agents are driven by overarching goals, such as maximising the security posture of the environment. This encourages them to develop strategies that prioritise actions leading to the most significant reduction in *risk* and protection of *assets*.

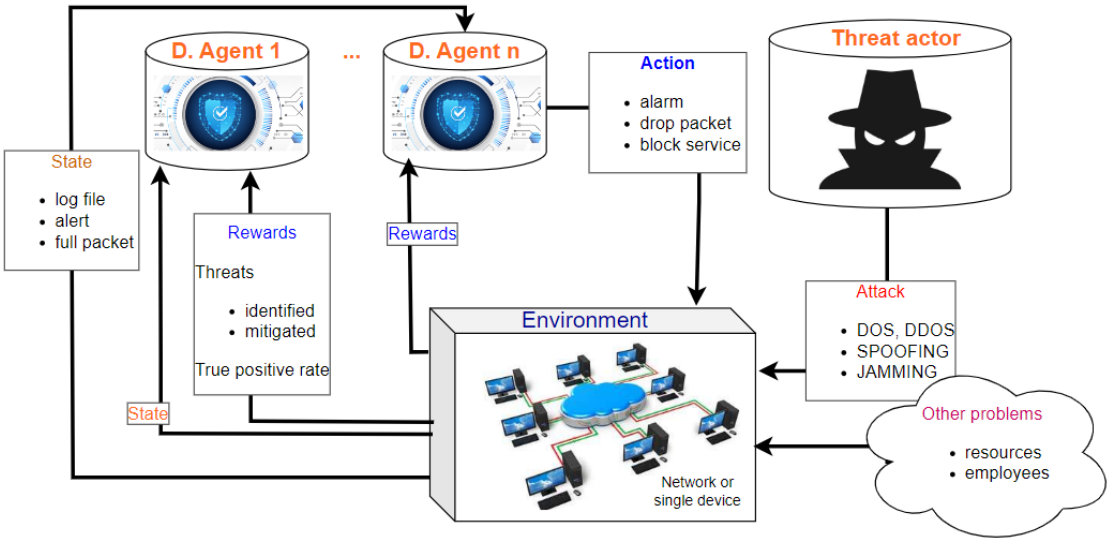

Figure 1: A typical RL environment within the domain of cybersecurity. One or multiple agents interact with a network environment with several appliances available but also susceptible of attacks. A threat actor is also present, in the form of one or multiple active human/software attackers, cyber threat simulation software, or dataset of past attacks.

The RL agents are tasked with defending the environment against threat actors, that can be either actual human/software attackers or emulated through datasets and simulation frameworks. These threat actors strategically exploit the environment to find vulnerabilities, and additional challenges may arise from authorised resources or employees attacked to get access to other parts of the system.

4. Classification: Framework and Survey

To explore the intersection between RL and cybersecurity, we started by looking at existing surveys. We thus exploited the Scopus computer science database using the query string "reinforcement learning" AND "cybersecurity" AND ("survey" OR "review"). This search yielded 31 results. However, manual inspection revealed that only five amongst them truly were broad surveys focussed on RL applied to general cybersecurity. Other either focussed on a specific application scenario (e.g. IoT) or were not centred around RL (e.g. mostly covered statistical machine learning methods). Thus, we delved deeper into these five surveys [2, 3, 4, 8, 10] and applied snowballing when due to clarify which articles could faithfully and clearly represent a research thread within our proposed classification.

Among these five surveys, three in particular inspired this work. The first, Uprety and Rawat [2] organise their survey according to the nature of attacks in the specific application domain of the Internet of Things (IoT). Adawadkar and Kulkarni [3], instead, focus on IDS and resource optimisation in IoT environments. The researchers identify key parameters for comparing RL-based algorithms, including detection rate, precision, and accuracy, providing valuable insights into the effectiveness of RL in enhancing cybersecurity measures. Finally, Cengiz and Gök [4] concentrate initially on penetration testing and then on Intrusion Detection Systems (IDS), providing valuable insights into the evolving cybersecurity landscape. They conclude by explaining how RL can be applied to various types of attacks. By selectively surveying the available literature and evidence, they offer a comprehensive overview of the role of RL in fortifying defences against emerging threats.

With the similar goal of assessing the state of the art and compare various approaches in RL for cybersecurity, we have formulated a novel classification meant to better introduce researchers and practitioners to the filed, encompassing two key dimensions: *(i)* the architecture of the RL approach employed, and *(ii)* the network configuration adopted for cybersecurity.

4.1. RL techniques dimension

In the context of RL, multiple learning agents can coexist, and they can share learning data or not. Additionally, when multiple agents exist, they can explicitly try to hinder each other learnt policies. These possibilities give rise to 5 categories of RL approaches:

- **Single-agent**. In single-agent systems, there is only one agent operating in the environment and learning. This agent makes decisions and takes actions independently alone, considering only its own information.
- **Centralized multiagent**. In *centralized* multiagent systems, multiple agents exist, but there is a central controller or coordinator that learns a decision-making policy for all

agents based on the information assembled from each agent. This central controller is the only one learning a policy, that is then given to every other agent.

- **Decentralized multiagent**. In *decentralized* multiagent systems, each agent makes its own decisions independently without a central controller. Agents in decentralized systems typically have limited access to information about the environment and the actions of other agents. They must use local information and possibly communication with nearby agents to learn and make decisions. Each agents learns its own policy.
- **Multiagent CTDE**. The CTDE paradigm involves training a multi-agent system in a centralised manner, where a central controller learns the policies or strategies for each agent using global information. However, during execution, each agent operates *independently*, making decisions based on the centrally learned policy but *conditioned* on its own observations, and without direct coordination with the central controller or other agents.
- **Adversarial multi-agent**. In *adversarial* multi-agent systems, agents operate in a competitive environment where each agent's objectives are directly opposed to those of other agents. These systems often involve strategic interactions, where agents must anticipate and react to the actions of other agents in order to achieve their own objectives.

This classification helps to quickly identify the learning scenario and enables further, more fine-grained categorisation based on the specific RL algorithm adopted.

4.2. Network configuration dimension

The other dimension we consider is the network configuration of the cybersecurity measures.

- **Host-based cybersecurity**. The defence system is focused on protecting individual devices (hosts) such as computers, servers, mobile devices, and endpoints. As such, it involves installing security software directly on these devices. This software may include antivirus programs, firewalls, intrusion detection/prevention systems (IDS/IPS) [11], and endpoint protection platforms (EPP). Host-based cybersecurity measures are essential for safeguarding against threats like malware, unauthorised access, and data breaches that may target specific devices (see Figure 2).
- **Network-based cybersecurity**. The focus shifts to securing the communication pathways between different devices and sub-systems within a network. It involves implementing security measures at the network level to detect and prevent unauthorised access, malicious activities, and data breaches. Network-based security solutions include firewalls, intrusion detection/prevention systems (IDS/IPS), virtual private networks (VPNs), and network access control (NAC) systems (see Figure 2). These measures help protect against threats such as unauthorised access attempts, malware propagation, and network-based attacks like DDoS (Distributed Denial of Service) attacks.
- **Network-based cybersecurity centralised with SDN**. This configuration combines network-based cybersecurity measures with Software-Defined Networking (SDN, see

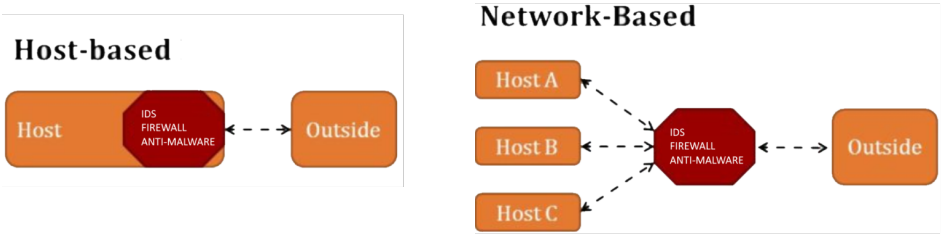

Figure 2: Host based vs network based security configuration.

Figure 3) technology. SDN [12] is an approach to networking that separates the control plane from the data plane, allowing for centralised management and programmability of network resources. Here, security policies and controls are centrally managed and enforced across the network infrastructure through software-defined policies. This enables more dynamic and granular control over network traffic. SDN-based security solutions include centralised firewall management, dynamic access control policies, and real-time threat intelligence integration, among others.

This classification helps to quickly identify what kind of defence is needed.

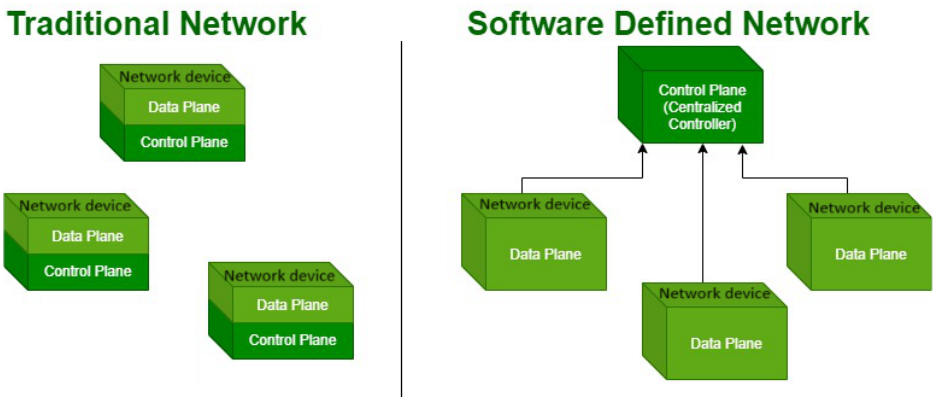

Figure 3: Traditional network vs Software defined network.

4.3. Proposed Classification

By organising our analysis along these two dimensions, our survey endeavours to furnish a comprehensive overview of the strengths, limitations, and potential applications of diverse RL methodologies within diverse network settings. This way, we can construct a holistic portrayal of the current state-of-the-art in RL applications for cybersecurity, depicted in Figure 4, shedding light on emerging trends and challenges.

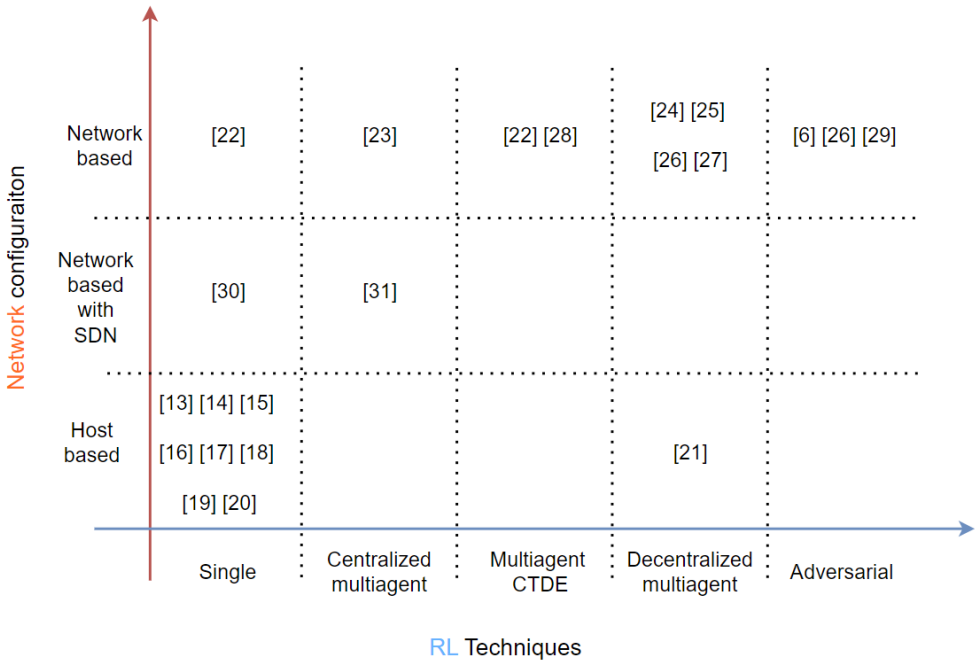

Figure 4: Articles about application of RL in cybersecurity categorised according to the network environment configuration (y-axis) and the RL approach adopted (x-axis). In both axis, decentralisation increases in the direction of the arrow.

Single-agent RL for host-based security. This category gathers the most approaches, as it represents those solutions that are technically easier to set up: a single learning agent learns based on the inputs coming from all the devices in the network, and controls all the security measures therein installed.

Liu et al. [13] present a RL-based approach to enhance the security of wireless networks by mitigating spoofing attacks. The receiver (Bob) acts as an agent using the Q-learning algorithm to make decisions about authenticating packets. Bob's **state** (s_t) represents his current knowledge of the channel conditions and historical authentication results. His **action** (a_t) involves selecting a threshold for authentication to decide whether an incoming packet is legitimate or spoofed. Bob **observes** the packet's physical-layer characteristics, such as signal strength and channel properties. The **reward** function (r_t) provides feedback based on the accuracy of Bob's authentication decisions, rewarding correct identifications and penalising false alarms and missed detections, as follows:

$$r_t = \begin{cases} R_{\text{correct}} & \text{if correct identification (legitimate packet)} \\ R_{\text{false\_alarm}} & \text{if false alarm (legitimate packet classified as spoofed)} \\ R_{\text{missed\_detection}} & \text{if missed detection (spoofed packet not detected)} \\ R_{\text{correct\_rejection}} & \text{if correct rejection (spoofed packet identified)} \end{cases}$$

Through repeated interactions and rewards, Bob learns to improve his authentication policy, thus enhancing the network's resilience to spoofing attacks.

Elnaggar and Bezzo [14] introduce a method to predict and recover from cyber-physical attacks on UAV (Unmanned Aerial Vehicle, aircraft operating without a human pilot on board) using Inverse Reinforcement Learning. The focus is on scenarios where UAVs have to reach a particular position and the attackers try to manipulate the sensor data to disrupt its navigation. The key components of the approach are: **Actions**, which refer to specific movements or adjustments the UAV can make, such as changing direction, speed, or altitude; **States and Observations**, representing the various conditions or positions of the UAVs within its environment, such as the UAV's geographic coordinates, velocity, altitude, along with sensor readings from gyroscope, accelerometer, and GPS; **Policy**, a strategy derived from IRL that guides the system to make decisions that avoid the attacker's goals and maintain a desired operational level; and **Reward**, a function that evaluates the success of actions in maintaining system integrity and achieving goals.

The reward function $R(s, a)$ is designed to reflect the system's objectives and the attacker's interference. For example, it might be formulated as:

$$R(s, a) = -\left(\alpha \cdot d(s, s_{goal}) + \beta \cdot I(s)\right)$$

where $d(s, s_{goal})$ is the distance from the current state s to the goal state s_{goal}, $I(s)$ is an indicator function that penalises unsafe states, and α and β are weighting factors. The algorithm uses Bayesian IRL within a Markov Decision Process framework, applying Monte Carlo Markov Chain sampling to predict the attacker's intentions. The system's effectiveness is demonstrated through simulations involving a UAV navigating a stochastic environment.

Xu et al. proposed a series of works [15, 16, 17] where they introduce TD-SAD (temporal-difference-based sequential anomaly detection) to combat multi-stage cyber attacks in computer systems, showcasing its high detection rates and low false alarm rates across various types of program traces. Building upon this, they extended it to enhance anomaly detection in host-based IDS, demonstrating the effectiveness of RL techniques. Finally, they proposed another method for detecting anomalies in host computers using sequential anomaly detection based on temporal-difference (TD) learning principles, highlighting its efficiency in modelling complex sequential behaviours without prior knowledge of the underlying processes.

Xiao et al. [18] delves into the security vulnerabilities inherent in Mobile Edge Computing (MEC) systems. The paper uses RL to enhance security measures such as secure mobile offloading against smart attacks, lightweight authentication, and collaborative caching schemes.

Feng et al. [19] address the challenge of defending against application-layer distributed denial-of-service (L7 DDoS) attacks, which exploit legitimate-appearing application-layer requests to overwhelm server functions. Traditional DDoS defences struggle with L7 DDoS attacks due to their subtle nature at the transport and network layers. The authors propose a defence mechanism using RL, where an agent learns to mitigate these attacks through a *multi-objective reward function*. This function balances the aggressive mitigation of malicious requests during severe attacks with conservative mitigation to minimise collateral damage to legitimate traffic under normal conditions. Their evaluation demonstrates that the proposed approach effectively mitigates 98.73% of malicious events.

Oh and Iyengar [20] introduces a sequential anomaly detection method using Inverse RL. It models an agent's behaviour through a learned reward function, identifying anomalies when

behaviours deviate from expected patterns. A Bayesian extension to IRL incorporates model uncertainty, enhancing reliability. Key contributions include the application of IRL to anomaly detection, handling varying-length input trajectories in real-time, and empirical validation of effectiveness. The effectiveness of this approach is demonstrated through empirical studies on publicly available real-world data.

Single-agent RL for network-based security. This category includes approaches where a single learning agent is responsible for monitoring and securing the entire network. The agent analyses network traffic and activities in network devices (routers, firewalls, switches, gateways, ...), making decisions to enhance the network's overall security posture.

Liu et al. [21] present a Deep RL approach for mitigating Distributed Denial of Service (DDoS) attacks in SDNs. The system employs a Deep Deterministic Policy Gradient algorithm. The **state** space captures network features from OpenFlow switches; the **action** space configures bandwidth limits for hosts using OpenFlow meters; and the **observation** process continuously monitors network traffic. The reward function, defined as:

$$\text{reward} = \begin{cases} -1 & \text{if Load}_s > U_s \\ \lambda p_b + (1-\lambda)(1-p_a) & \text{if Load}_s \leq U_s \end{cases}$$

penalises server overload and rewards maximising benign traffic while minimising attack traffic. In summary, the DRL-based approach dynamically adjusts bandwidth allocations to effectively mitigate DDoS attacks.

Veluchamy and Kathavarayan [22] proposed the Deep Adaptive RL for Honeypots (DARLH) system that operates in both single-agent and multi-agent paradigms to enhance security in honeypot environments. These are environments featuring decoy systems set up to attract and analyse cyber attackers, by gathering data on attack methods to improve security. At the *single-agent* level, the agent autonomously learns and makes decisions based on its observations of network traffic and system behaviour. This single-agent approach allows for adaptive behaviour and decision-making tailored to the specific environment. At the *multi-agent* level, the system integrates multiple agents, each responsible for monitoring different aspects of the honeypot environment. These agents collaborate to share information, coordinate actions, and collectively contribute to the overall security posture of the system. This two-level architecture combines the adaptability and autonomy of single-agent systems with the collaborative and coordinated capabilities of multi-agent systems, offering a holistic approach to network security.

Multi-agent centralised RL for network-based security. This category encompasses methods where multiple learning agents are coordinated centrally to secure the network. Each agent focuses on a specific aspect of the network, and their actions are managed by a central controller to ensure cohesive and effective security measures.

Janakiraman and Deva Priya [23] propose a Deep RL approach exploiting Long Short Term Memory (LSTM) networks for mitigating DDoS attacks in fog-assisted cloud environments. Multiple agents collaborate in a centralised network-based approach to identify and mitigate DDoS attacks at the network layer. By utilising SDN controllers (see Section 4.2), the system is able to analyse network traffic and differentiate between legitimate and malicious packets. The

LSTM component is used for its ability to handle time-dependent data and effectively categorise incoming packets. The reward in this context is defined as the successful identification and mitigation of DDoS attacks while minimising false positives and maintaining the availability of legitimate network services. A detailed discussion on the reward structure and its implications can be found in Section 3 of the paper.

Multi-agent RL for host-based security. In this section, we explore research efforts focused on leveraging the collective intelligence and collaborative decision-making of multiple agents to fortify cybersecurity measures on a single device.

Dasgupta et al. [24] focus on detecting and mitigating GPS spoofing attacks, crucial in transportation cyber-physical systems. They propose a deep RL based method, using in-vehicle sensor data and signal processing to detect spoofing attacks turn-by-turn. The **State (S)** includes the positions and movements of vehicles as well as the signals received from GPS satellites, and encapsulates information about the system's susceptibility to spoofing attacks. The **Action (A)** in this scenario corresponds to the responses that the system can take to detect and mitigate GPS spoofing attacks. These may include adjusting the navigation algorithms, re-calibrating sensors, or deploying countermeasures to verify the authenticity of GPS signals. The **Observation (O)** consists of the data collected from in-vehicle sensors and GPS receivers, as well as the feedback from the detection and mitigation mechanisms. These observations provide insights into the effectiveness of the system's response to spoofing attacks and guide further decision-making processes. The **Reward (R)** signal in this context reflects the immediate benefits or costs associated with the system's actions in response to spoofing attacks. Rewards could be based on successfully detecting and mitigating spoofing attempts, minimising disruptions to navigation systems, or avoiding accidents caused by misleading GPS information. This study demonstrates the potential of RL-based approaches to bolster cybersecurity in transportation systems vulnerable to GPS spoofing attacks. Employing a multi-agent system, the method enhances detection accuracy through collaborative decision-making among agents.

Decentralised multi-agent RL for network-based security. This category involves approaches where multiple learning agents work independently but collaboratively to secure the network. Each agent is responsible for a segment of the network, making autonomous decisions while communicating with other agents to maintain overall network security.

Malialis and Kudenko [25] propose a framework that involves deploying multiple agents within the network to coordinate responses against threats. These agents use RL to dynamically adjust router throttling mechanisms, effectively mitigating DDoS attacks' impact on network performance and availability. The approach is *decentralised*, as each agent operates independently, making decisions based on local observations and interactions with the environment. However, they collaborate indirectly by collectively improving the overall network resilience through their individual actions.

Bhagyashree Deokar [26] propose a cooperative learning method for IDS based on multi-agent systems. The system architecture involves multiple agents distributed across different hosts, each responsible for monitoring network connections and system log files. These agents collaborate in a *decentralised* manner by sharing information and making local decisions based

on their observations, contributing to a collective decision about whether an intrusion has occurred. The decision-making process utilises influence diagrams and Bayesian networks to model uncertainty and optimise decision outcomes.

Bhosale et al. [27] propose an approach to IDS by leveraging a multi-agent framework and RL techniques. It addresses the limitations of traditional single-agent IDS, which struggle to handle the complexity and real-time demands of modern network security. By employing a multi-agent system, each agent possesses partial information and collaborates with others to improve decision-making capabilities. The decision-making process is facilitated by influence diagrams, which represent probabilistic relationships between events and guide local decision-making. This approach leans towards *decentralisation*, as agents collaborate but maintain their autonomy in decision-making.

Shamshirband et al. [28] use Cooperative Game-based Fuzzy Q-learning (G-FQL) for detecting and preventing intrusions, particularly DDoS attacks, in wireless sensor networks (WSNs). G-FQL integrates game theory, fuzzy Q-learning, and a cooperative defence strategy involving sink nodes, a base station, and attackers. The cooperative game mechanism allows the sensor nodes to act as rational decision-makers, collaborating to detect and defend against attacks. Fuzzy Q-learning reinforces the nodes' self-learning abilities, providing them with incentive functions to protect vulnerable sensor nodes. This approach is a mix of *centralisation* and *decentralisation*, as the system involves centralised elements such as the base station coordinating overall strategy, while individual sensor nodes operate autonomously but collaborate within the overarching framework.

Adversarial RL for network-based security. This category focus on a specific RL setting, termed *adversarial*, where RL agents learn a policy that is actively disrupted by hostile agents (termed "threat actors" or "other problems" in Figure 1, which become the adversarial agents). These agents are trained to anticipate and counteract sophisticated attacks, thereby enhancing the network's resilience against adversarial threats.

Caminero et al. [29] incorporate a multi-agent technique by integrating a classifier, acting as the agent, with a simulated environment. This environment generates network traffic samples and provides rewards based on the classifier's predictive accuracy. The classifier's objective is to predict the correct intrusion label for the given network samples, while the environment's goal is to actively increase the difficulty of predictions by behaving adversarially, challenging the classifier to learn from the most difficult cases. Adversarial Environment using RL introduces an innovative approach for intrusion detection in network security. It employs a multi-agent setup by combining reinforcement learning principles with a classifier serving as the primary agent. The classifier aims to predict intrusion labels for network traffic samples, while the adversarial environment generates scenarios that challenge the classifier by increasing the difficulty of predictions. By maximising rewards obtained from the environment, the classifier learns to adapt to these challenges, leading to enhanced performance in detecting and classifying intrusions within network traffic. This dynamic interaction between the classifier and the adversarial environment forms the core of Adversarial Environment using RL, enabling it to effectively address the evolving threats and complexities in network security.

Turner et al. [30] focus on modelling the interactions between attackers and defenders in a network environment. Attackers aim to exploit vulnerabilities, while defenders seek to mitigate risks. Multi-Agent Reinforcement Learning (MARL) involves competition between RL agents representing attackers and defenders, each learning to optimise its strategy based on feedback received from the environment. Co-evolution involves evolving populations of strategies for attackers and defenders simultaneously, with each population adapting to the strategies of its opponent over time. The paper compares the effectiveness of these approaches in generating robust solutions for cybersecurity challenges, emphasising the importance of balancing exploration and exploitation in learning strategies.

5. Limitations, Challenges, and Open Issues

While RL holds promise for addressing cybersecurity challenges, it also presents certain limitations, challenges, and open issues, summarised in Table 2.

Limitations. A first limitation of RL approaches regarding cybersecurity environments is *complexity*. Cybersecurity environments often exhibit high-dimensional and dynamic characteristics, leading to computationally intensive training processes. The complexity arises from the need to represent diverse network states, attacker behaviours, and defensive actions accurately. As a result, RL algorithms may encounter scalability issues, prolonged training times, and resource constraints, limiting their practical applicability in real-world cybersecurity scenarios.

Another limitation, generally applicable to RL but even more so to cybersecurity, is *sample efficiency*. Many RL algorithms require extensive training data to learn effective policies, posing challenges in resource-constrained cybersecurity settings where collecting sufficient labelled data is difficult. A notable example in cybersecurity is detecting zero-day attacks.

Challenges. A first challenge for RL is posed by *adversarial attacks*, that exploit vulnerabilities in RL-based cybersecurity systems, like poisoning the training data with fake one, to manipulate the system's behaviour. In particular, steering learning agents towards sub-optimal policies. Another challenge is achieving robust *generalisation* across diverse cyber threats and environments. For instance, ensuring that an RL-based IDS trained on one network architecture performs effectively when deployed in a different one. Finally, ensuring available of *quality data*

Table 2
Limitations, challenges, and open issues of RL approaches applied to cybersecurity.

	Complexity	Sample Efficiency	Generalisation	Adversarial Attacks	Transfer Learning	Explainability
Limitation	✓	✓				
Challenge			✓	✓		
Open Issue					✓	✓

is another challenge, requiring data collection strategies that reflect real-world cyber threats, environments, and defence mechanisms.

Open Issues. Amongst the open issues still to be fully investigated in RL applied to cybersecurity, at least two emerge strongly from the surveyed literature: *explainability* and *transferability*. The former amounts to ensuring that RL-based cybersecurity systems are understandable and transparent to human beings. Achieving explainability involves making the decisions and actions of RL algorithms interpretable to stakeholders. For instance, in autonomous threat response systems, explainability ensures that security analysts can comprehend the reasoning behind the system's actions and trust its recommendations. Transferability instead amounts to transferring knowledge and policies learned in one cybersecurity context to another. This would obviously improve efficiency and effectiveness, as there would be less need to re-training RL systems from scratch. For instance, leveraging knowledge from detecting malware to enhance intrusion detection in network traffic.

6. Conclusion & Future Works

In this paper, we have discussed the potential of RL to enhance cybersecurity defences by offering adaptive and dynamic mechanisms to combat emerging threats. We highlighted limitations of traditional approaches, motivated why RL can help surpassing them, and then proposed a bi-dimensional classification to help researchers enter the field or get a bird-eye view.

While our analysis demonstrates the promise of RL in improving detection, mitigation, and response capabilities, the surveyed literature also identify challenges that must be addressed. These include complexity, sample efficiency, and vulnerabilities to adversarial attacks. Moving forward, it is imperative to focus on developing robust and explainable RL-based defence mechanisms, as well as exploring techniques for knowledge transfer and generalisation across diverse cyber threats and environments. By addressing these challenges, we can harness the full potential of RL to fortify cybersecurity defences and mitigate emerging threats effectively.

References

[1] S. Morgan, Cybercrime to cost the world 8 trillion annually in 2023, 2022. URL: https://web.archive.org/web/20240429123425/https://cybersecurityventures.com/cybercrime-to-cost-the-world-8-trillion-annually-in-2023/.

[2] A. Uprety, D. B. Rawat, Reinforcement learning for iot security: A comprehensive survey, IEEE Internet of Things Journal 8 (2021) 8693–8706. doi:10.1109/JIOT.2020.3040957.

[3] A. M. K. Adawadkar, N. Kulkarni, Cyber-security and reinforcement learning — a brief survey, Engineering Applications of Artificial Intelligence 114 (2022) 105116. doi:10.1016/j.engappai.2022.105116.

[4] E. Cengiz, M. Gök, Reinforcement learning applications in cyber security: A review, Sakarya University Journal of Science 27 (2023) 481–503. doi:10.16984/saufenbilder.1237742.

[5] J. H. Connell, K. Sridhar Mahadevan, *Robot Learning*, Robotica 17 (1999) 229–235. doi:10.1017/S0263574799271172.

[6] Z. Hu, P. Chen, M. Zhu, P. Liu, Reinforcement Learning for Adaptive Cyber Defense Against Zero-Day Attacks, Springer International Publishing, Cham, 2019, pp. 54–93. doi:10.1007/978-3-030-30719-6_4.

[7] M. Macas, C. Wu, W. Fuertes, A survey on deep learning for cybersecurity: Progress, challenges, and opportunities, Computer Networks 212 (2022) 109032. doi:10.1016/j.comnet.2022.109032.

[8] T. T. Nguyen, V. J. Reddi, Deep reinforcement learning for cyber security, IEEE Transactions on Neural Networks and Learning Systems 34 (2023) 3779–3795. doi:10.1109/TNNLS.2021.3121870.

[9] Y. Guo, A review of machine learning-based zero-day attack detection: Challenges and future directions, Computer Communications 198 (2023) 175–185. doi:10.1016/j.comcom.2022.11.001.

[10] P. Dixit, S. Silakari, Deep learning algorithms for cybersecurity applications: A technological and status review, Computer Science Review 39 (2021) 100317. URL: https://www.sciencedirect.com/science/article/pii/S1574013720304172. doi:https://doi.org/10.1016/j.cosrev.2020.100317.

[11] D. Denning, An intrusion-detection model, IEEE Transactions on Software Engineering SE-13 (1987) 222–232. doi:10.1109/TSE.1987.232894.

[12] S. Shin, L. Xu, S. Hong, G. Gu, Enhancing network security through software defined networking (sdn), in: 25th International Conference on Computer Communication and Networks (ICCCN), 2016, pp. 1–9. doi:10.1109/ICCCN.2016.7568520.

[13] J. Liu, L. Xiao, G. Liu, Y. Zhao, Active authentication with reinforcement learning based on ambient radio signals, Multimedia Tools and Applications 76 (2017) 3979–3998. doi:10.1007/s11042-015-2958-x.

[14] M. Elnaggar, N. Bezzo, An irl approach for cyber-physical attack intention prediction and recovery, in: 2018 Annual American Control Conference (ACC), 2018, pp. 222–227. doi:10.23919/ACC.2018.8430922.

[15] X. Xu, T. Xie, A reinforcement learning approach for host-based intrusion detection using sequences of system calls, in: Advances in Intelligent Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 995–1003. doi:10.1007/11538059_103.

[16] X. Xu, Y. Luo, A kernel-based reinforcement learning approach to dynamic behavior modeling of intrusion detection, in: Advances in Neural Networks – ISNN 2007, Springer Berlin Heidelberg, 2007, pp. 455–464. doi:10.1007/978-3-540-72383-7_54.

[17] X. Xu, Sequential anomaly detection based on temporal-difference learning: Principles, models and case studies, Applied Soft Computing 10 (2010) 859–867. doi:10.1016/j.asoc.2009.10.003.

[18] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, M. Guizani, Security in mobile edge caching with reinforcement learning, IEEE Wireless Communications 25 (2018) 116–122. doi:10.1109/MWC.2018.1700291.

[19] Y. Feng, J. Li, T. Nguyen, Application-layer ddos defense with reinforcement learning, in: IEEE/ACM 28th International Symposium on Quality of Service (IWQoS), 2020, pp. 1–10. doi:10.1109/IWQoS49365.2020.9213026.

[20] M.-h. Oh, G. Iyengar, Sequential anomaly detection using inverse reinforcement learning, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, 2019, p. 1480–1490. doi:10.1145/3292500.3330932.

[21] Y. Liu, M. Dong, K. Ota, J. Li, J. Wu, Deep reinforcement learning based smart mitigation of ddos flooding in software-defined networks, in: IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2018, pp. 1–6. doi:10.1109/CAMAD.2018.8514971.

[22] S. Veluchamy, R. S. Kathavarayan, Deep reinforcement learning for building honeypots against runtime dos attack, Int. J. Intell. Syst. 37 (2022) 3981–4007. doi:10.1002/INT.22708.

[23] S. Janakiraman, M. Deva Priya, A deep reinforcement learning-based ddos attack mitigation scheme for securing big data in fog-assisted cloud environment, Wireless Personal Communications 130 (2023) 2869–2886. doi:10.1007/s11277-023-10407-2.

[24] S. Dasgupta, T. Ghosh, M. Rahman, A reinforcement learning approach for global navigation satellite system spoofing attack detection in autonomous vehicles, Transportation Research Record 2676 (2022) 318–330. doi:10.1177/03611981221095509.

[25] K. Malialis, D. Kudenko, Distributed response to network intrusions using multiagent reinforcement learning, Engineering Applications of Artificial Intelligence 41 (2015) 270–284. doi:10.1016/j.engappai.2015.01.013.

[26] A. H. Bhagyashree Deokar, Intrusion detection system using log files and reinforcement learning, International Journal of Computer Applications 45 (2012) 28–35. doi:10.5120/7026-9675.

[27] R. Bhosale, D. S. Mahajan, P. A. Kulkarni, Cooperative machine learning for intrusion detection system, International Journal of Scientific & Engineering Research 5 (2014). URL: https://www.ijser.org/researchpaper/Cooperative-Machine-Learning-For-Intrusion-Detection-System.pdf.

[28] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah, A. Abraham, Cooperative game theoretic approach using fuzzy q-learning for detecting and preventing intrusions in wireless sensor networks, Engineering Applications of Artificial Intelligence 32 (2014) 228–241. URL: https://www.sciencedirect.com/science/article/pii/S0952197614000311. doi:doi.org/10.1016/j.engappai.2014.02.001.

[29] G. Caminero, M. Lopez-Martin, B. Carro, Adversarial environment reinforcement learning algorithm for intrusion detection, Computer Networks 159 (2019) 96–109. doi:10.1016/j.comnet.2019.05.013.

[30] M. J. Turner, E. Hemberg, U.-M. O'Reilly, Analyzing multi-agent reinforcement learning and coevolution in cybersecurity, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 1290–1298. doi:10.1145/3512290.3528844.