Deep Reinforcement Learning for Cyber Security

Thanh Thi Nguyen and Vijay Janapa Reddi, *Member, IEEE*

***Abstract*— The scale of Internet-connected systems has increased considerably, and these systems are being exposed to cyberattacks more than ever. The complexity and dynamics of cyberattacks require protecting mechanisms to be responsive, adaptive, and scalable. Machine learning, or more specifically deep reinforcement learning (DRL), methods have been proposed widely to address these issues. By incorporating deep learning into traditional RL, DRL is highly capable of solving complex, dynamic, and especially high-dimensional cyber defense problems. This article presents a survey of DRL approaches developed for cyber security. We touch on different vital aspects, including DRL-based security methods for cyber–physical systems, autonomous intrusion detection techniques, and multiagent DRL-based game theory simulations for defense strategies against cyberattacks. Extensive discussions and future research directions on DRL-based cyber security are also given. We expect that this comprehensive review provides the foundations for and facilitates future studies on exploring the potential of emerging DRL to cope with increasingly complex cyber security problems.**

***Index Terms*— Cyber defense, cyber security, cyberattacks, deep learning, deep reinforcement learning (DRL), Internet of Things (IoT), IoT, review, survey.**

I. Introduction

INTERNET of Things (IoT) technologies have been employed broadly in many sectors, such as telecommunications, transportation, manufacturing, water and power management, healthcare, education, finance, government, and even entertainment. The convergence of various information and communication technology (ICT) tools in the IoT has boosted its functionalities and services to users to new levels. ICT has witnessed a remarkable development in terms of system design, network architecture, and intelligent devices in the last decade. For example, ICT has been advanced with the innovations of cognitive radio network (CRN) and 5G cellular network [1], [2], software-defined network (SDN) [3], cloud computing [4], (mobile) edge caching [5], [6], and fog computing [7]. Accompanying these developments is the increasing vulnerability to cyberattacks, which are defined as any type of offensive maneuver exercised by one or multiple computers to target computer information systems, network infrastructures, or personal computer devices. Cyberattacks may be instigated by economic competitors or state-sponsored attackers. There has been, thus, a critical need for the development of cyber security technologies to mitigate and eliminate impacts of these attacks [8].

Artificial intelligence (AI), especially machine learning (ML), has been applied to both attacking and defending in cyberspace. On the attacker side, ML is utilized to compromise defense strategies. On the cyber security side, ML is employed to put up robust resistance against security threats in order to adaptively prevent and minimize the impacts or damages that occurred. Among these ML applications, unsupervised and supervised learning methods have been used widely for intrusion detection [9]–[11], malware detection [12]–[14], cyber–physical attacks [15]–[17], and data privacy protection [18]. In principle, unsupervised methods explore the structure and patterns of data without using their labels, while supervised methods learn by examples based on data's labels. These methods, however, cannot provide *dynamic and sequential responses* against cyberattacks, especially new or constantly evolving threats. In addition, the detection and defending responses often take place after the attacks when traces of attacks become available for collecting and analyzing, and thus, proactive defense solutions are hindered. A statistical study shows that 62% of the attacks were recognized after they have caused significant damages to the cyber systems [19].

Reinforcement learning (RL), a branch of ML, is the closest form of human learning because it can learn by its own experience through exploring and exploiting the unknown environment. RL can model an autonomous agent to take sequential actions optimally without or with limited prior knowledge of the environment, and thus, it is particularly adaptable and useful in real-time and adversarial environments. With the power of *function approximation* and *representation learning*, deep learning has been incorporated into RL methods and enabled them to solve many complex problems [20]–[24]. The combination of deep learning and RL, therefore, indicates excellent suitability for cyber security applications where cyberattacks are increasingly sophisticated, rapid, and ubiquitous [25]–[28].

The emergence of DRL has actually witnessed great success in different fields, from video game domain, e.g., Atari [29], [30], game of Go [31], [32], real-time strategy game StarCraft II [33]–[36], 3-D multiplayer game Quake III Arena Capture the Flag [37], and teamwork game Dota 2 [38] to real-world applications, such as robotics [39], autonomous vehicles (AVs) [40], autonomous surgery [41], [42], natural language processing [43], biological data mining [44], and drug design [45]. DRL methods have also recently been applied to solve various problems in the IoT area. For example, a DRL-based resource allocation framework that integrates networking, caching, and computing capabilities for smart city

Manuscript received July 21, 2020; revised January 16, 2021 and April 16, 2021; accepted October 18, 2021. *(Corresponding author: Thanh Thi Nguyen.)*

Thanh Thi Nguyen is with the School of Information Technology, Deakin University, Melbourne Burwood Campus, Burwood, VIC 3125, Australia (e-mail: thanh.nguyen@deakin.edu.au).

Vijay Janapa Reddi is with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: vj@eecs.harvard.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3121870.

Digital Object Identifier 10.1109/TNNLS.2021.3121870

applications is proposed in [46]. The DRL algorithm, i.e., double dueling deep Q-network (DQN) [47], [48], is used to solve this problem because it involves a large-state space, which consists of the dynamic changing status of base stations, mobile edge caching (MEC) servers and caches. The framework is developed based on the programmable control principle of SDN and the caching capability of *information-centric networking*. Alternatively, Zhu *et al.* [49] explored MEC policies by using the *context awareness* concept that represents the user's context information and traffic pattern statistics. The use of AI technologies at the mobile network edges is advocated to intelligently exploit the operating environment and make the right decisions regarding what, where, and how to cache appropriate contents. To increase the caching performance, a DRL approach, i.e., the asynchronous advantage actor-critic (A3C) algorithm [50], is used to find an optimal policy aiming to maximize the offloading traffic.

Findings from our current survey show that applications of DRL in cyber environments are generally categorized under two perspectives: optimizing and enhancing the communications and networking capabilities of the IoT applications, e.g., [51]–[59], and defending against cyberattacks. This article focuses on the latter where DRL methods are used to solve cyber security problems with the presence of cyberattacks or threats. Section II provides a background of DRL methods, followed by a detailed survey of DRL applications in cyber security in Section III. We group these applications into three major categories, including DRL-based security solutions for cyber–physical systems (CPSs), autonomous intrusion detection techniques, and DRL-based game theory for cyber security. Section IV concludes this article with extensive discussions and future research directions on DRL for cyber security.

II. Deep Reinforcement Learning Preliminary

Different from the other popular branch of ML, i.e., supervised methods learning by examples, RL characterizes an *agent* by creating its own learning experiences through interacting directly with the *environment*. RL is described by concepts of *state*, *action*, and *reward* (Fig. 1). It is a trial and error approach in which the agent takes action at each time step that causes two changes: the current state of the environment is changed to a new state, and the agent receives a reward or penalty from the environment. Given a state, the reward is a function that can tell the agent how good or bad an action is. Based on received rewards, the agent learns to take more good actions and gradually filter out bad actions.

A popular RL method is Q-learning whose goal is to maximize the discounted cumulative reward based on the Bellman equation [60]

$$Q(s_t, a_t) = E\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t, a_t\right]. \quad (1)$$

The discount factor $\gamma \in [0, 1]$ manages the importance levels of future rewards. It is applied as a mathematical trick to analyzing the learning convergence. In practice, the discount is necessary because of partial observability or uncertainty of the stochastic environment.

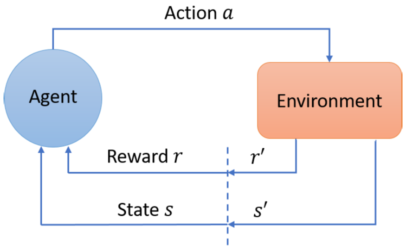

Fig. 1. Interactions between the agent and its environment in RL, characterized by state, action, and reward. Based on the current state s and reward r, the agent will take an optimal action, leading to changes of state and reward. The agent then receives the next state s' and reward r' from the environment to determine the next action, making an iterative process of agent-environment interactions.

Q-learning needs to use a lookup table or Q-table to store expected rewards (Q-values) of actions given a set of states. This requires a large memory when the state and action spaces increase. Real-world problems often involve continuous state or action space, and therefore, Q-learning is inefficient to solve these problems. Fortunately, deep learning has emerged as a powerful tool that is a great complement to traditional RL techniques. Deep learning methods have two typical capabilities, i.e. function approximation and representation learning, which help them to learn a compact low-dimensional representation of raw high-dimensional data effectively [61]. The combination of deep learning and RL was the research direction that Google DeepMind has initiated and pioneered. They proposed DQN with the use of a deep neural network (DNN) to enable Q-learning to deal with high-dimensional sensory inputs [29], [62].

Using DNNs to approximate the Q-function, however, is unstable due to the correlations among the sequences of observations and the correlations between the Q-values $Q(s, a)$ and the target values $Q(s', a')$. Mnih *et al.* [29] proposed the use of two novel techniques, i.e., experience replay memory and target network, to address this problem (Fig. 2). On the one hand, experience memory stores an extensive list of learning experience tuples (s, a, r, s'), which are obtained from the agent's interactions with the environment. The agent's learning process retrieves these experiences randomly to avoid the correlations of consecutive experiences. On the other hand, the target network is technically a copy of the estimation network, but its parameters are frozen and only updated after a period. For instance, the target network is updated after 10 000 updates of the estimation network, as demonstrated in [29]. DQN has made a breakthrough as it is the first time an RL agent can provide a human-level performance in playing 49 Atari games by using just raw image pixels of the game board.

As a *value-based method*, DQN takes a long training time and has limitations in solving problems with continuous action spaces. Value-based methods, in general, evaluate the goodness of an action given a state using the Q-value function. When the number of states or actions is large or infinite, they show inefficiency or even impracticality. Another type of RL, i.e., *policy gradient methods*, has solved this problem effectively. These methods aim to derive actions directly by learning a policy $\pi(s, a)$ that is a probability distribution

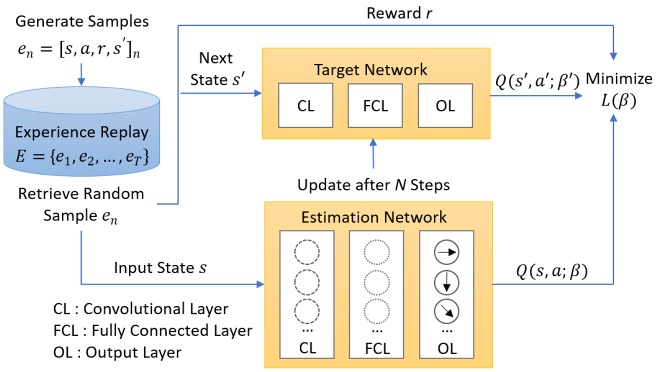

Fig. 2. DQN architecture with the loss function described by $L(\beta) = E[(r + \gamma \max_{a'} Q(s', a'|\beta') - Q(s, a|\beta))^2]$, where β and β' are parameters of the estimation and target DNNs, respectively. Each action taken by the agent will generate an experience, which consists of the current state s, action a, reward r, and next state s'. These learning experiences (samples) are stored in the experience replay memory, which are then retrieved randomly for a stable learning process.

over all possible actions. REINFORCE [64], vanilla policy gradient [65], trust region policy optimization (TRPO) [66] and proximal policy optimization (PPO) [67] are notable policy gradient methods. The gradient estimation, however, often suffers from a large fluctuation [68]. The combination of value-based and policy-gradient methods has been developed to aggregate the advantages and eradicate the disadvantages of these two methods. This kind of combination has constituted another type of RL, i.e., *actor-critic methods*. This structure comprises two components: an actor and a critic that can be both characterized by DNNs. The actor attempts to learn a policy by receiving feedback from the critic. This iterative process helps the actor improve its strategy and converge to an optimal policy. Deep deterministic policy gradient (DDPG) [69], distributed distributional DDPG (D4PG) [70], A3C [50], and unsupervised reinforcement and auxiliary learning (UNREAL) [71] are methods that utilize the actor-critic framework. An illustrative architecture of the popular algorithm A3C is presented in Fig. 3. A3C's structure consists of a hierarchy of a master learning agent (global) and individual learners (workers). Both the master agent and individual learners are modeled by DNNs with each having two outputs: one for the critic and another for the actor. The first output is a scalar value representing the expected reward of a given state $V(s)$, while the second output is a vector of values representing a probability distribution over all possible actions $\pi(s, a)$.

The value loss function of the critic is specified by

$$L_1 = \sum (R - V(s))^2 \tag{2}$$

where $R = r + \gamma V(s')$ is the discounted future reward. In addition, the actor is pursuing minimization of the following policy loss function:

$$L_2 = -\log(\pi(a|s)) * A(s) - \vartheta H(\pi) \tag{3}$$

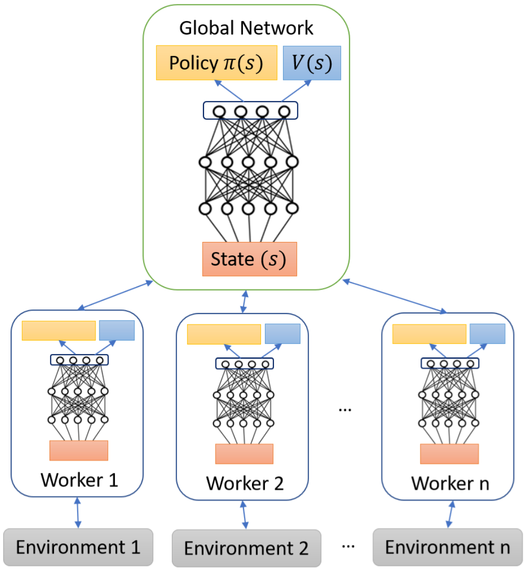

Fig. 3. Learning architecture of A3C, consisting of a global network and a number of worker agents. Each worker initially resets its parameters to those of the global network and interacts with its copy of the environment for learning. Gradients obtained from these individual learning processes will be used to update the global network asynchronously. This increases the learning speed and diversifies the experience learned by the global network as the experiences obtained by individual worker agents are independent.

where $A(s) = R - V(s)$ is the estimated *advantage* function, and $H(\pi)$ is the entropy term, which handles the exploration capability of the agent with the hyperparameter ϑ controlling the strength of the entropy regularization. The advantage function $A(s)$ shows how advantageous the agent is when it is in a particular state. The learning process of A3C is *asynchronous* because each learner interacts with its separate environment and updates the master network independently. This process is iterated, and the master network is the one to use when the learning is finished.

Table I summarizes comparable features of value-based, policy-gradient, and actor-critic methods, and their typical example algorithms. Valued-based methods are more sample efficient than policy-gradient methods because they are able to exploit data from other sources such as experts [72]. In DRL, a value function or a policy function is normally approximated by a universal function approximator, such as a (deep) neural network, which can take either discrete or continuous states as inputs. Therefore, modeling state spaces is more straightforward than dealing with action spaces in DRL. Value-based methods are suitable for problems with discrete action spaces as they evaluate every action explicitly and choose an action at each time step based on these evaluations. On the other hand, policy-gradient and actor-critic methods are more suitable for continuous action spaces because they describe the policy (a mapping between states and actions) as a probability distribution over actions. The continuity characteristic is the main difference between discrete and continuous action spaces. In a discrete action space, actions are characterized as a mutually exclusive set of options, and while in continuous action space, action has a value from a certain range or boundary [73].

III. DRL IN CYBER SECURITY: SURVEY

A large number of applications of RL to various aspects of cyber security have been proposed in the literature, ranging from data privacy to critical infrastructure protection.

TABLE I

SUMMARY OF FEATURES OF DRL TYPES AND THEIR NOTABLE METHODS

DRL types	Value-based	Policy-gradient	Actor-critic
Features	- Compute value of action given a state $Q(s,a)$. - No learned explicit policy. - Sample efficient [72].	- No value function is needed. - Explicit policy is constructed. - Sample inefficient [72].	- Actor produces policy $\pi(s,a)$. - Critic evaluates action by $V(s)$. - Often perform better than value-based or policy-gradient methods.
Typical methods	- DQN [29] - Double DQN [47] - Dueling Q-network [48] - Prioritized Experience Replay DQN [63]	- REINFORCE [64] - Vanilla Policy Gradient [65] - TRPO [66] - PPO [67]	- DDPG [69] - D4PG [70] - A3C [50] - UNREAL [71]
Applications	Suitable for problems with discrete action spaces, e.g., classic control tasks: Acrobot, CartPole, and MountainCar as described and implemented in the popular OpenAI Gym toolkit [74].	More suitable for problems with continuous action spaces, e.g., classic control tasks described and implemented in the OpenAI Gym toolkit: MountainCarContinuous and Pendulum [74] or BipedalWalker and CarRacing problems [75].	

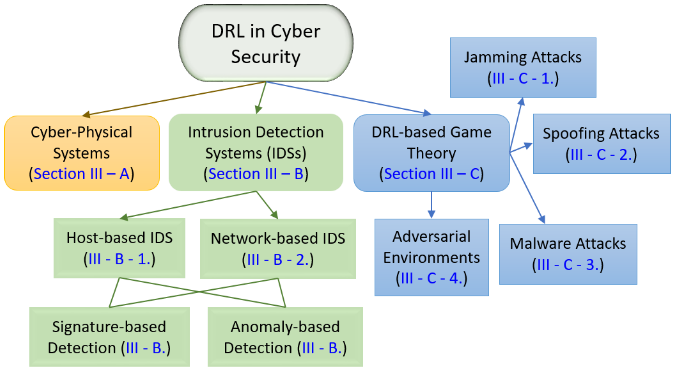
DRL in Cyber Security
Jamming Attacks (III - C - 1.)
Cyber-Physical Systems (Section III – A)
Intrusion Detection Systems (IDSs) (Section III – B)
DRL-based Game Theory (Section III – C)
Spoofing Attacks (III - C - 2.)
Host-based IDS (III - B - 1.)
Network-based IDS (III - B - 2.)
Adversarial Environments (III - C - 4.)
Malware Attacks (III - C - 3.)
Signature-based Detection (III - B.)
Anomaly-based Detection (III - B.)

Fig. 4. Different (sub)sections of the survey on DRL in cyber security.

However, drawbacks of traditional RL have restricted its capability in solving complex and large-scale cyber security problems. The increasing number of connected IoT devices in recent years has led to a significant increase in the number of cyberattack instances and their complexity. The emergence of deep learning and its integration with RL has created a class of DRL methods that are able to detect and fight against sophisticated types of cyberattacks, such as falsified data injection to CPSs [86], deception attack to autonomous systems [93], distributed denial-of-service (DDoS) attacks [114], intrusions to host computers or networks [125], jamming [142], spoofing [157], malware [161], attacks in adversarial networking environments [168], and so on. This section provides a comprehensive survey of state-of-the-art DRL-powered solutions for cyber security, ranging from defense methods for CPSs to autonomous intrusion detection approaches, and game theory-based solutions.

The structure of the survey is presented in Fig. 4. We limit the survey to existing applications of DRL to cyber security. There are other topics in cyber security where DRL has not been applied to and they are therefore discussed in Section IV (Discussions and Future Research Directions). Those potential topics include a multiagent DRL approach to cyber security, combining host-based and network-based intrusion detection systems (IDSs), model-based DRL and combining model-free and model-based DRL methods for cyber security applications, investigating methods that can deal with continuous action spaces in cyber environments, offensive AI, deepfakes, ML poisoning, adversarial ML, human–machine teaming within a human-on-the-loop architecture, bit-and-piece DDoS attacks, and potential attacks by quantum physics-based powerful computers to crack encryption algorithms.

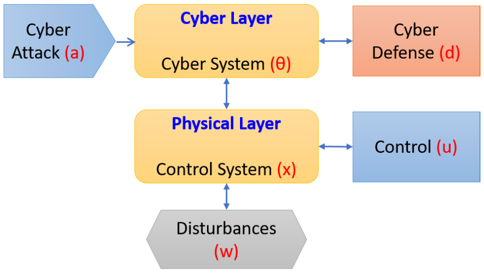

Fig. 5. Dynamics of attack and defense in a CPS. The physical layer is often uncertain with disturbances w, while cyberattack a directly affects the cyber layer where a defense strategy d needs to be implemented. The dynamics of attack-defense characterized by $\theta(t,a,d)$ is injected into the conventional physical system to develop a cyber–physical co-modeling as presented in (4).

A. DRL-Based Security Methods for Cyber–Physical Systems

Investigations of defense methods for CPS against cyberattacks have received considerable attention and interests from the cyber security research community. CPS is a mechanism controlled by computer-based algorithms facilitated by Internet integration. This mechanism provides efficient management of distributed physical systems via the shared network. With the rapid development of the Internet and control technologies, CPSs have been used extensively in many areas, including manufacturing [76], health monitoring [77], [78], smart grid [79]–[81], and transportation [82], [83]. Being exposed widely to the Internet, these systems are increasingly vulnerable to cyberattacks [84]. In 2015, hackers attacked the control system of a steel mill in Germany by obtaining login credentials via the use of phishing emails. This attack caused a partial plant shutdown and resulted in damage of millions of dollars. Likewise, there was a costly cyberattack on a power grid in Ukraine in late December 2015 that disrupted electricity supply to a few hundred thousand end consumers [85].

In an effort to study cyberattacks on CPS, Feng *et al.* [85] characterized the cyber state dynamics by a mathematical model

$$\dot{x}(t) = f(t,x,u,w;\theta(t,a,d)); \quad x(t_0) = x_0 \tag{4}$$

where x, u, and w represent the physical state, control inputs, and disturbances correspondingly (see Fig. 5). In addition, $\theta(t,a,d)$ describes the cyber state at time t with a and d referring to cyberattack and defense, respectively.

The CPS defense problem is then modeled as a two-player zero-sum game by which utilities of players are summed up

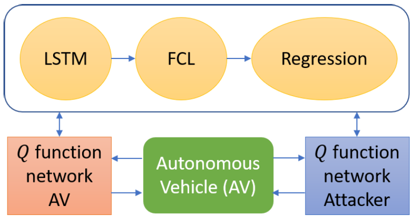

Fig. 6. Architecture of the adversarial DRL algorithm for robust AV control. A DNN consisting of an LSTM, a fully connected layer (FCL), and regression is used to learn long-term dependencies within large datasets, which contain the outcomes of the players' past interactions. The DNN can approximate the Q functions to find optimal actions for players, i.e., AV (defender) and especially attacker, who seeks to inject faulty data to AV sensor readings.

to zero at each time step. The defender is represented by an actor-critic DRL algorithm. Simulation results demonstrate that the proposed method in [85] can learn an optimal strategy to timely and accurately defend the CPS from unknown cyberattacks.

Applications of CPS in critical safety domains such as autonomous automotive, chemical process, automatic pilot avionics, and smart grid require a certain correctness level. Akazaki *et al.* [86] proposed the use of DRL, i.e., double DQN (DDQN) and A3C algorithms, to find falsified inputs (counterexamples) for CPS models. This allows for effective yet automatic detection of CPS defects. Due to the infinite state space of CPS models, conventional methods, such as simulated annealing [87] and cross entropy [88], were found inefficient. Experimental results show the superiority of the use of DRL algorithms against those methods in terms of the smaller number of simulation runs. This leads to a more practical detection process for CPS models' defects despite the great complexity of CPS's software and physical systems.

AVs operating in the *future smart cities* require a robust processing unit of intravehicle sensors, including camera, radar, roadside smart sensors, and intervehicle beaconing. Such reliance is vulnerable to cyber–physical attacks aiming to get control of AVs by manipulating the sensory data and affecting the reliability of the system, e.g., increasing accident risks or reducing the vehicular flow. Ferdowsi *et al.* [89] examined the scenarios, where the attackers manage to interject faulty data to the AV's sensor readings while the AV (the defender) needs to deal with that problem to control AV robustly. Specifically, the *car following model* [90] is considered in which the focus is on autonomous control of a car that follows closely another car. The defender aims to learn the leading vehicle's speed based on sensor readings. The attacker's objective is to mislead the following vehicle to a deviation from the optimal safe spacing. The interactions between the attacker and defender are characterized by a game-theoretic problem. The interactive game structure and its DRL solution are diagramed in Fig. 6. Instead of directly deriving a solution based on the *mixed-strategy Nash equilibrium* analytics, Hochreiter and Schmidhuber *et al.* proposed the use of DRL to solve this dynamic game. Long short-term memory (LSTM) [91] is used to approximate the Q-function for both defending and attacking agents as it can capture the temporal dynamics of the environment.

Likewise, Rasheed *et al.* [92] introduced an adversarial DRL method that integrates LSTM and generative adversarial network (GAN) model to cope with data infusion attacks in AVs equipped with 5G communication links. The adversary attempts to inject faulty data to impact safe distance spacing between AVs, while the AVs manage to minimize this deviation. LSTM is used as a generator, while a convolutional neural network (CNN) is used as a discriminator to resemble a GAN structure, which is able to capture previous temporal actions of the AV and attacker as well as previous distance deviations. A DRL algorithm is proposed based on these observations to select optimal actions (suitable velocity) for the AV to avoid collisions and accidents.

Autonomous systems can be vulnerable to inefficiency from various sources, such as noises in communication channels, sensor failures, errors in sensor measurement readings, packet errors, and especially cyberattacks. *Deception attack* to autonomous systems is widespread as it is initiated by an adversary whose effort is to inject noises to the communication channels between sensors and the command center. This kind of attack leads to corrupted information being sent to the command center and eventually degrades the system's performance. Gupta and Yang [93] studied the ways to increase the robustness of autonomous systems by allowing the system to learn using adversarial examples. The problem is formulated as a zero-sum game with the players to be the command center (observer) and the adversary. The inverted pendulum problem from Roboschool [67] is used as a simulation environment. The TRPO algorithm is employed to design an observer that can reliably detect adversarial attacks in terms of measurement corruption and automatically mitigate their effects.

B. DRL-Based Intrusion Detection Systems

To detect intrusions, security experts conventionally need to observe and examine audit data, e.g., application traces, network traffic flow, and user command data, to differentiate between normal and abnormal behaviors. However, the volume of audit data surges rapidly when the network size is enlarged. This makes manual detection difficult or even impossible. An IDS is a software or hardware platform installed on host computers or network equipment to detect and report to the administrator abnormal or malicious activities by analyzing the audit data. An intrusion detection and prevention system may be able to take appropriate actions immediately to reduce the impacts of the malevolent activities.

Depending on different types of audit data, IDSs are grouped into two categories: host-based or network-based IDS. *Host-based IDS* typically observes and analyses the host computer's log files or settings to discover anomalous behaviors. *Network-based IDS* relies on a sniffer to collect transmitting packets in the network and examines the traffic data for intrusion detection. Host-based systems normally lack cross-platform support and implementing them requires knowledge of the host operating systems and configurations. Network-based systems aim to monitor traffic over specific network segments, and they are independent of the operating systems and more portable than host-based systems. Implementing network-based systems is, thus, easier and they offer more monitoring power than host-based systems. However, a network-based IDS may have difficulty in handling heavy traffic and high-speed networks because it must examine every packet passing through its network segment.

Regardless of the IDS type, two common detection methods are used: signature-based and anomaly-based detection. *Signature detection* involves the storage of patterns of known attacks and comparing characteristics of possible attacks to those in the database. *Anomaly detection* observes the normal behaviors of the system and alerts the administrator if any activities are found deviated from normality, for instance, the unexpected increase of traffic rate, i.e., number of IP packets per second. ML techniques, including unsupervised clustering and supervised classification methods, have been used widely to build adaptive IDSs [94]–[98]. These methods, e.g., neural networks [99], *k*-nearest neighbors [99], [100], support vector machine (SVM) [99], [101], random forest [102], and recently deep learning [103]–[105], however, normally rely on fixed features of existing cyberattacks so that they are deficient in detecting new or deformed attacks. The lack of prompt responses to dynamic intrusions also leads to ineffective solutions produced by unsupervised or supervised techniques. In this regard, RL methods have been demonstrated effectively in various IDS applications [106]. Sections III-B.1 and III-B.2 review the use of DRL methods in both host-based and network-based IDSs.

1) Host-Based IDS: As the volume of audit data and complexity of intrusion behaviors increase, adaptive intrusion detection models demonstrate limited effectiveness because they can only handle temporally isolated labeled or unlabeled data. In practice, many complex intrusions comprise temporal sequences of dynamic behaviors. Xu and Xie [107] proposed an RL-based IDS that can handle this problem. System call trace data are fed into a *Markov reward process* whose state value can be used to detect abnormal temporal behaviors of host processes. The intrusion detection problem is, thus, converted to a state value prediction task of the Markov chains. The linear *temporal difference (TD) RL* algorithm [108] is used as the state value prediction model where its outcomes are compared with a predetermined threshold to distinguish normal traces and attack traces. Instead of using the errors between real values and estimated ones, the TD learning algorithm uses the differences between successive approximations to update the state value function. Experimental results obtained from using system call trace data show the dominance of the proposed RL-based IDS in terms of higher accuracy and lower computational costs compared with SVM, hidden Markov model, and other ML or data mining methods. The proposed method based on the linear basis functions, however, has a shortcoming when sequential intrusion behaviors are highly nonlinear. Therefore, a kernel-based RL approach using least-squares TD [109] was suggested for intrusion detection in [110] and [111]. Relying on the kernel methods, the generalization capability of TD RL is enhanced, especially in high-dimensional and nonlinear feature spaces. The kernel least-squares TD algorithm is, therefore, able to predict anomaly probabilities accurately, which contributes to improving the detection performance of IDS, especially when dealing with multistage cyberattacks.

2) Network-Based IDS: Deokar and Hazarnis [112] pointed out the drawbacks of both anomaly-based and signature-based detection methods. On the one hand, *anomaly detection* has a high false alarm rate because it may categorize activities, which users rarely perform as an anomaly. On the other hand, *signature detection* cannot discover new types of attacks as it uses a database of patterns of well-known attacks. The authors, therefore, proposed an IDS that can identify known and unknown attacks effectively by combining features of both anomaly and signature detection through the use of log files. The proposed IDS is based on a collaboration of RL methods, association rule learning, and log correlation techniques. RL gives a reward (or penalty) to the system when it selects log files that contain (or do not contain) anomalies or any signs of attack. This procedure enables the system to choose more appropriate log files in searching for traces of the attack.

One of the most difficult challenges in the current Internet is dealing with the *DDoS* threat, which is a DoS attack but has a distributed nature, occurring with a large traffic volume and compromising a large number of hosts. Malialis and Kudenko [113] and Malialis [114] initially introduced the multiagent router throttling method based on the SARSA algorithm [115] to address the DDoS attacks by learning multiple agents to rate-limit or throttle traffic toward a victim server. That method, however, has a limited capability in terms of scalability. They, therefore, further proposed the coordinated team learning design to the original multiagent router throttling based on the divide-and-conquer paradigm to eliminate the mentioned drawback. The proposed approach integrates three mechanisms, namely, task decomposition, hierarchical team-based communication, and team rewards, involving multiple defensive nodes across different locations to coordinately stop or reduce the flood of DDoS attacks. A network emulator is developed based on the work of Yau *et al.* [116] to evaluate throttling approaches. Simulation results show that the resilience and adaptability of the proposed method are superior to its competing methods, i.e., baseline router throttling and additive-increase/multiplicative-decrease throttling algorithms [116], in various scenarios with different attack dynamics. The scalability of the proposed method is successfully experimented with up to 100 RL agents, which has a great potential to be deployed in a large Internet service provider network.

Alternatively, Bhosale *et al.* [117] proposed a multiagent intelligent system [118] using RL and influence diagram [119] to enable quick responses against the complex attacks. Each agent learns its policy based on the local database and information received from other agents, i.e., decisions and events. Shamshirband *et al.* [120], on the other hand, introduced an intrusion detection and prevention system for wireless sensor networks (WSNs) based on a game theory approach and employed a fuzzy Q-learning algorithm [121], [122] to obtain optimal policies for the players. Sink nodes, a base station, and an attacker constitute a three-player game where sink nodes and base station are coordinated to derive a defense strategy against the DDoS attacks, particularly in the application layer. The IDS detects future attacks based on the fuzzy Q-learning algorithm that takes part in two phases: detection and defense

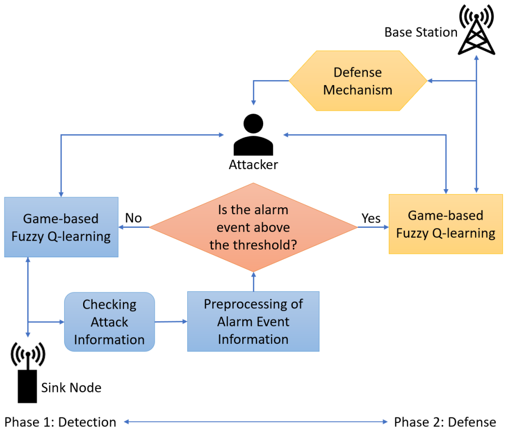
Base Station
Defense Mechanism
Attacker
Game-based Fuzzy Q-learning
No
Is the alarm event above the threshold?
Yes
Game-based Fuzzy Q-learning
Checking Attack Information
Preprocessing of Alarm Event Information
Sink Node
Phase 1: Detection
Phase 2: Defense

Fig. 7. Two-phase intrusion detection and prevention system based on a game theory approach and fuzzy Q-learning. In Phase 1, the sink node uses fuzzy Q-learning to detect anomalies caused by the attacker to victim nodes. The malicious information is preprocessed and checked against a threshold by the sink node before passing to Phase 2, where the base station also employs fuzzy Q-learning to select optimal defense actions.

(Fig. 7). The game commences when the attacker sends an overwhelming volume of flooding packets beyond a specific threshold as a DDoS attack to a victim node in the WSN. Using the low energy adaptive clustering hierarchy (LEACH), which is a prominent WSN protocol [123], the performance of the proposed method is evaluated and compared with that of existing soft computing methods. The results show the efficacy and viability of the proposed method in terms of detection accuracy, energy consumption, and network lifetime.

In another approach, Caminero *et al.* [124] proposed a model, namely, adversarial environment using RL, to incorporate the RL theory in implementing a classifier for network intrusion detection. A simulated environment is created, where random samples drawn from a labeled network intrusion dataset are treated as RL states. The adversarial strategy is employed to deal with unbalanced datasets as it helps to avoid training bias via an oversampling mechanism, and thus, decreases classification errors for underrepresented classes. Likewise, a study in [125] applied DRL methods, such as DQN, DDQN, policy gradient, and actor-critic models, for network intrusion detection. With several adjustments and adaptations, DRL algorithms can be used as a supervised approach to classifying labeled intrusion data. DRL policy network is simple and fast, which is suitable for online learning and rapid responses in modern data networks with evolving environments. Results obtained on two intrusion detection datasets show DDQN is the best algorithm among the four employed DRL algorithms. DDQN's performance is equivalent and even better than many traditional ML methods in some cases. Recently, Saeed *et al.* [126] examined the existing multiagent IDS architectures, including several approaches that utilized RL algorithms. The adaptation capability of RL methods can help IDS to respond effectively to changes in the environments. However, the optimal solution is not guaranteed because the convergence of a multiagent system is hard to obtain.

C. DRL-Based Game Theory for Cyber Security

Traditional cyber security methods, such as firewall, antivirus software, or intrusion detection are normally passive, unilateral, and lagging behind dynamic attacks. Cyberspace involves various cyber components, and thus, reliable cyber security requires the consideration of interactions among these components. Specifically, the security policy applied to a component has a certain impact on the decisions taken by other components. Therefore, the decision space increases considerably, with many what-if scenarios when the system is large. Game theory has been demonstrated effectively in solving such large-scale problems because it can examine many scenarios to derive the best policy for each player [127]–[131]. The utility or payoff of a game player depends not only on its actions but also on other players' activities. In other words, the efficacy of cyber defending strategies must take into account the attacker's strategies and other network users' behaviors. Game theory can model the conflict and cooperation between intelligent decision-makers, resembling activities of cyber security problems, which involve attackers and defenders. This resemblance has enabled game theory to mathematically describe and analyze the complex behaviors of multiple competitive mechanisms. In the following, we present game-theoretic models involving multiple DRL agents that characterize cyber security problems in different attacking scenarios, including jamming, spoofing, malware, and attacks in adversarial environments.

1) Jamming Attacks: Jamming attacks can be considered as a special case of DoS attacks, which are defined as any event that diminishes or eradicates a network's capacity to execute its expected function [132]–[134]. Jamming is a serious attack in networking and has attracted a great interest of researchers who used ML or especially RL to address this problem, e.g., [135]–[141]. The recent development of deep learning has facilitated the use of DRL for jamming handling or mitigation. Xiao *et al.* [142] studied security challenges of the MEC systems and proposed an RL-based solution to provide secure offloading to the edge nodes against jamming attacks. MEC is a technique that allows cloud computing functions to take place at the edge nodes of a cellular network or generally of any network. This technology helps to decrease network traffic and reduce overhead and latency when users request to access contents that have been cached in the edges closer to the cellular customer. MEC systems, however, are vulnerable to cyberattacks because they are physically located closer to users and attackers with less secure protocols compared with cloud servers or database centers. In [142], the RL methodology is used to select the defense levels and important parameters, such as offloading rate and time, transmit channel, and power. As the network state space is large, the authors proposed the use of DQN to handle high-dimensional data, as shown in Fig. 8. DQN uses a CNN to approximate the Q-function that requires high computational complexity and memory. To mitigate this disadvantage a transfer learning method named hotbooting technique is used. The *hotbooting method* helps to initialize weights of CNN more efficiently by using experiences that have been learned in similar circumstances. This reduces the learning time by avoiding random explorations at the start of each episode. Simulation results demonstrate that the proposed method is effective in terms of enhancing the security and user privacy of MEC systems and it can protect the systems in confronting with different types of smart attacks with low overhead.

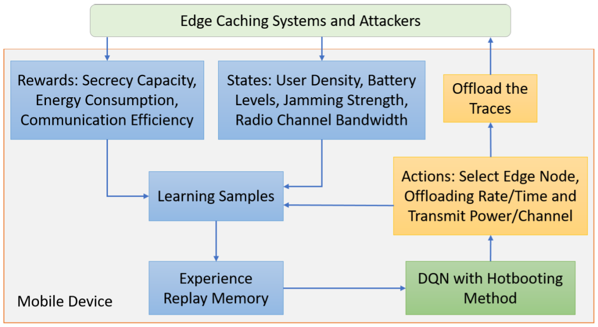
Edge Caching Systems and Attackers
Rewards: Secrecy Capacity, Energy Consumption, Communication Efficiency
States: User Density, Battery Levels, Jamming Strength, Radio Channel Bandwidth
Offload the Traces
Learning Samples
Actions: Select Edge Node, Offloading Rate/Time and Transmit Power/Channel
Experience Replay Memory
DQN with Hotbooting Method
Mobile Device

Fig. 8. Secure offloading method in MEC based on DQN with hotbooting technique. The DQN agent's actions are to find optimal parameters, such as offloading rate, power, and channel for the mobile device to offload the traces to the edge node accordingly. The attackers may deploy jamming, spoofing, DoS, or smart attacks to disrupt this process. By interacting with the edge caching systems, the agent can evaluate the reward of the previous action and obtain a new state, enabling it to select the next optimal action.

On the other hand, Aref *et al.* [143] introduced a multiagent RL method to deal with antijamming communications in *wideband autonomous cognitive radios* (WACRs). WACRs are advanced radios that can sense the states of the radio frequency spectrum and network, and autonomously optimize their operating mode corresponding to the perceived states. Cognitive communication protocols, however, may struggle when there are unintentional interferences or malicious users who attempt to interrupt reliable communications by deliberate jamming. Each radio's effort is to occupy the available common wideband spectrum as much as possible and avoid the sweeping signal of a jammer that affects the entire spectrum band. The multiagent RL approach proposed in [143] learns an optimal policy for each radio to select appropriate subband, aiming to avoid jamming signals and interruptions from other radios. Comparative studies show the significant dominance of the proposed method against a random policy. A drawback of the proposed method is the assumption that the jammer uses a fixed strategy in responding to the WACRs strategies, although the jammer may be able to perform adaptive jamming with the cognitive radio technology. In [144], when the current spectrum subband is interfered with by a jammer, Q-learning is used to optimally select a new subband that allows uninterrupted transmission as long as possible. The reward structure of the Q-learning agent is defined as the amount of time that the jammer or interferer takes to interfere with the WACR transmission. Experimental results using the hardware-in-the-loop prototype simulation show that the agent can detect the jamming patterns and successfully learns an optimal subband selection policy for jamming avoidance. The obvious drawback of this method is the use of a Q-table with a limited number of environment states.

The access right to spectrum (or more generally resources) is the main difference between CRNs and traditional wireless technologies. RL in general or Q-learning has been investigated to produce optimal policy for cognitive radio nodes to interact with their radio frequency environment [145]. Attar *et al.* [146] examined RL solutions against attacks on both CRN architectures, i.e., infrastructure-based, e.g., the IEEE 802.22 standard, and infrastructure-less, e.g., *ad hoc* CRN. The adversaries may attempt to manipulate the spectrum sensing process and cause the main sources of security threats in infrastructure-less CRNs. The external adversary node is not part of the CRN, but such attackers can affect the operation of an *ad hoc* CRN via jamming attacks. In an infrastructure-based CRN, an exogenous attacker can mount incumbent emulation or perform sensor-jamming attacks. The attacker can increase the local false-alarm probability to affect the decision of the IEEE 802.22 base station about the availability of a given band. A jamming attack can have both short-term and long-term effects. Wang *et al.* [147] developed a game-theoretic framework to battle against jamming in CRNs where each radio observes the status and quality of available channels and the strategy of jammers to make decisions accordingly. The CRN can learn optimal channel utilization strategy using minimax-Q learning policy [148], solving the problems of how many channels to use for data and to control packets along with the channel switching strategy. The performance of minimax-Q learning represented via spectrum-efficient throughput is superior to the myopic learning method, which gives high priority to the immediate payoff and ignores the environment dynamics and the attackers' cognitive capability.

In CRNs, *secondary users* (SUs) are obliged to avoid disruptions to communications of *primary users* (PUs) and can only gain access to the licensed spectrum when it is not occupied by PUs. Jamming attacks are emergent in CRNs due to the opportunistic access of SUs as well as the appearance of *smart jammers*, which can detect the transmission strategy of SUs. Xiao *et al.* [149] studied the scenarios where a smart jammer aims to disrupt the SUs, rather than PUs. The SUs and jammer, therefore, must sense the channel to check the presence of PUs before making their decisions. The constructed scenarios consist of a secondary source node supported by relay nodes to transmit data packets to secondary receiving nodes. The smart jammer can learn quickly the frequency and transmission power of SUs, while SUs do not have full knowledge of the underlying dynamic environment. The interactions between SUs and jammer are modeled as a cooperative transmission power control game, and the optimal strategy for SUs is derived based on the Stackelberg equilibrium [150]. The aim of SU players is to select appropriate transmission powers to efficiently send data messages in the presence of jamming attacks. The jammer's utility gain is the SUs' loss and vice versa. RL methods, i.e., Q-learning [60] and WoLF-PHC [151], are used to model SUs as intelligent agents for coping with the smart jammer. WoLF-PHC stands for the combination of the win or learns fast algorithm and the policy hill-climbing method. It uses a varying learning rate to foster convergence to the game equilibrium by adjusting the learning speed [151]. Simulation results show the improvement in the antijamming performance of the proposed method in terms of the *signal-to-interference-plus-noise ratio* (SINR). The optimal strategy achieved from the Stackelberg game can minimize the damage created by the jammer in the worst case scenario.

Recently, Han *et al.* [152] introduced an antijamming system for CRNs using the DQN algorithm based on a *frequency-spatial* antijamming communication game. The game simulates an environment of numerous jammers that inject jamming signals to disturb the ongoing transmissions of SUs. The SU should not interfere with the communications of PUs and must defeat smart jammers. This communication system is 2-D that utilizes both frequency hopping and user mobility. The RL state is the radio environment consisting of PUs, SUs, jammers, and serving base stations/access points. The DQN is used to derive an optimal frequency hopping policy that determines whether the SU should leave an area of heavy jamming or choose a channel to send signals. Experimental results show the superiority of the DQN-based method against the Q-learning-based strategy in terms of faster convergence rate, increased SINR, lower cost of defense, and improved utility of the SU. DQN with the core component CNN helps to speed the learning process of the system, which has a large number of frequency channels, compared with the benchmark Q-learning method.

To improve the work of Han *et al.* [152], Liu *et al.* [153] also proposed an antijamming communication system using a DRL method but having different and more extensive contributions. Specifically, Liu *et al.* [153] used the raw spectrum information with temporal features, known as *spectrum waterfall* [154] to characterize the environment state, rather than using the SINR and PU occupancy as in [152]. Because of this, Liu *et al.*'s [153] model does not necessitate prior knowledge about the jamming patterns and parameters of the jammer but rather uses the local observation data. This prevents the model from the loss of information and facilitates its adaptability to a dynamic environment. Furthermore, Liu *et al.*'s [153] work does not assume that the jammer needs to take the same channel-slot transmission structure with the users as in [152]. The recursive CNN is utilized to deal with a complex infinite environment state represented by spectrum waterfall, which has a recursion characteristic. The model is tested using several jamming scenarios, which include sweeping jamming, comb jamming, dynamic jamming, and intelligent comb jamming. A disadvantage of both Han *et al.*'s and Liu *et al.*'s methods is that they can only derive an optimal policy for one user, which inspires a future research direction focusing on multiple users' scenarios.

2) Spoofing Attacks: Spoofing attacks are popular in wireless networks where the attacker claims to be another node using the faked identity, such as media access control, to gain access to the network illegitimately. This illegal penetration may lead to man-in-the-middle, or DoS attacks [155]. Xiao *et al.* [156], [157] modeled the interactions between the legitimate receiver and spoofers as a zero-sum authentication game and utilized Q-learning and Dyna-Q [158] algorithms to address the spoofing detection problem. The utility of the receiver or spoofer is computed based on the Bayesian risk, which is the expected payoff in the spoofing detection. The receiver aims to select the optimal test threshold in PHY-layer spoofing detection while the spoofer needs to select an optimal attacking frequency. To prevent collisions, spoofers are cooperative to attack the receiver. Simulation and experimental results show the improved performance of the proposed methods against the benchmark method with a fixed test threshold. A disadvantage of the proposed approaches is that both action and state spaces are quantized into discrete levels, bounded within a specified interval, which may lead to locally optimal solutions.

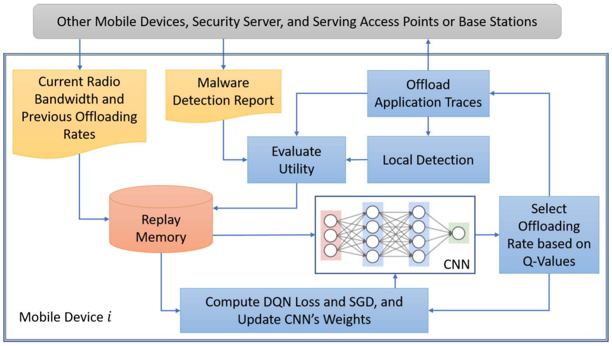
Other Mobile Devices, Security Server, and Serving Access Points or Base Stations
Current Radio Bandwidth and Previous Offloading Rates
Malware Detection Report
Offload Application Traces
Evaluate Utility
Local Detection
Replay Memory
CNN
Select Offloading Rate based on Q-Values
Compute DQN Loss and SGD, and Update CNN's Weights
Mobile Device *i*

Fig. 9. Cloud-based malware detection using DQN, where the stochastic gradient descent (SGD) method, is used to update weights of the CNN. Malicious detection is performed in the cloud server with more powerful computational resources than mobile devices. The DQN agent helps to select optimal task offloading rates for mobile devices to avoid network congestion and detection delay. By observing the network status and evaluating utility based on malware detection reports from the server, the agent can formulate states and rewards, which are used to generate a sequence of optimal actions, i.e., dynamic offloading rates.

3) Malware Attacks: One of the most challenging malware of mobile devices is the *zero-day attacks*, which exploit publicly unknown security vulnerabilities, and until they are contained or mitigated, hackers might have already caused adverse effects on computer programs, data, or networks [159], [160]. To avoid such attacks, the traces or log data produced by the applications need to be processed in real time. With limited computational power, battery life, and radio bandwidth, mobile devices often offload specific malware detection tasks to security servers at the cloud for processing. The security server with powerful computational resources and a more updated malware database can process the tasks quicker, more accurately, and then send a detection report back to mobile devices with less delay. The offloading process is, therefore, a key factor affecting the *cloud-based malware detection* performance. For example, if too many tasks are offloaded to the cloud server, there would be radio network congestion that can lead to long detection delays. Wan *et al.* [161] enhanced the mobile offloading performance by improving the previously proposed game model in [162]. The Q-learning approach used in [162] to select optimal offloading rate suffers the curse of high dimensionality when the network size is increased, or a large number of feasible offloading rates is available for selection. Wan *et al.* [161], thus, advocated the use of hotbooting Q-learning and DQN and showed the performance improvement in terms of malware detection accuracy and speed compared with the standard Q-learning. The cloud-based malware detection approach using DQN for selecting the optimal offloading rate is shown in Fig. 9.

4) Attacks in Adversarial Environments: Traditional networks facilitate the direct communications between client application and server where each network has its switch control that makes the network reconfiguration task time-consuming and inefficient. This method is also disadvantageous because the requested data may need to be retrieved from more than one database involving multiple servers. SDN is a next-generation networking technology as it can reconfigure the network adaptively. With the control being programmable with a global view of the network architecture, SDN can manage and optimize network resources effectively.

RL has been demonstrated broadly in the literature as a robust method for SDN controlling, e.g., [163]–[167].

Although RL's success in SDN controlling is abundant, the attacker may be able to falsify the defender's training process if it is aware of the network control algorithm in an adversarial environment. To deal with this problem, Han *et al.* [168] proposed the use of *adversarial RL* to build an *autonomous defense system for SDN*. The attacker selects important nodes in the network to compromise, for example, nodes in the backbone network or the target subnet. By propagating through the network, the attacker attempts to eventually compromise the critical server, while the defender prevents the server from being compromised and preserve as many unaffected nodes as possible. To achieve those goals, the RL defender takes four possible actions, consisting of "isolating," "patching," "reconnecting," and "migrating." Two types of DRL agents are trained to model defenders, i.e., DDQN and A3C, to select appropriate actions given different network states. The reward is characterized based on the status of the critical server, the number of preserved nodes, migration costs, and the validity of the actions taken. That study considered the scenarios where attackers can penetrate the learning process of RL defenders by flipping reward signs or manipulating states. These causative attacks poison the defender's training process and cause it to perform suboptimal actions. The adversarial training approach is applied to reduce the impact of poisoning attacks with its eminent performance demonstrated via several experiments using the popular network emulator Mininet [169].

In an adversarial environment, the defender may not know the private details of the attacker, such as the type of attack, attacking target, frequency, and location. Therefore, the defender, for example, may allocate substantial resources to protect an asset that is not a target of the attacker. The defender needs to dynamically reconfigure defense strategies to increase the complexity and cost for the intruder. Zhu *et al.* [170] introduced a model where the defender and attacker can repeatedly change the defense and attack strategies. The defender has no prior knowledge about the attacker, such as launched attacks and attacking policies. However, it is aware of the attacker classes and can access the system utilities, which are jointly contributed by the defense and attack activities. Two interactive RL methods are proposed for cyber defenses in [170], namely, adaptive RL and robust RL. The adaptive RL handles attacks that have a diminishing exploration rate (*nonpersistent attacker*) while the robust RL deals with intruders who have a constant exploration rate (*persistent attacker*). The interactions between defender and attacker are illustrated via the attack and defense cycles, as in Fig. 10. The attackers and defenders do not take actions simultaneously but asynchronously. On the attack cycle, the attacker evaluates previous attacks before launching a new attack if necessary. On the defense cycle, after receiving an alert, the defender carries out a meta-analysis on the latest attacks and calculates the corresponding utility before deploying a new defense if needed. An advantage of this system model is that it does not assume any underlying model for the attacker but instead treats attack strategies as black boxes.

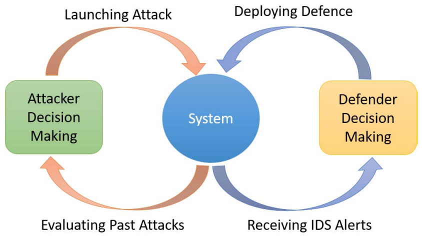

Fig. 10. Defender and attacker interact via the IDS in an adversarial environment, involving defense and attack cycles. Using these two cycles, a defender and an attacker can repeatedly change their defense and attack strategies. This model can be used to study defense strategies for different classes of attacks, such as buffer over-read attacks [171] and code reuse attacks [172].

Alternatively, Elderman *et al.* [173] simulated cyber security problems in networking as a stochastic Markov game with two agents, one attacker, and one defender, with incomplete information and partial observability. The attacker does not know the network topology but attempts to reach and get access to the location that contains a valuable asset. The defender knows the internal network but does not see the attack types or positions of intruders. This is a challenging cyber security game because a player needs to adapt its strategy to defeat the unobservable opponents [174]. Different algorithms, e.g., Monte Carlo learning, Q-learning, and neural networks are used to learn both defender and attacker. Simulation results show that Monte Carlo learning with the softmax exploration is the best method for learning both attacking and defending strategies. Neural network algorithms have a limited adversarial learning capability, and thus they are outperformed by Q-learning and Monte Carlo learning techniques. This simulation has a disadvantage that simplifies the real-world cyber security problem into a game of only two players with only one asset. In real practice, there can be multiple hackers simultaneously penetrating a server that holds valuable data. In addition, a network may contain useful data in different locations instead of in a single location as simulated.

IV. Discussions and Future Research Directions

DRL has emerged over recent years as one of the most successful methods of designing and creating human or even superhuman AI agents. Many of these successes have relied on the incorporation of DNNs into a framework of traditional RL to address complex and high-dimensional sequential decision-making problems. Applications of DRL algorithms, therefore, have been found in various fields, including IoT and cyber security. Computers and the Internet today play crucial roles in many areas of our lives, e.g., entertainment, communication, transportation, medicine, and even shopping. Lots of our personal information and important data are stored online. Even financial institutions, e.g., banks, mortgage companies, and brokerage firms, run their business online. Therefore, it is essential to have a security plan in place to prevent hackers from accessing our computer systems. This article has presented a comprehensive survey of DRL methods and their applications to cyber security problems, with notable examples summarized in Table II. The adversarial environment of cyber systems has instigated various proposals of game

TABLE II
SUMMARY OF TYPICAL DRL APPLICATIONS IN CYBER SECURITY

Applications	Goals/Objectives	Algorithms	States	Actions	Rewards
Robustness-guided falsification of CPS [86]	Find falsifying inputs (counterexamples) for CPS	Double DQN and A3C	Defined as the output of the system.	Choose the next input value from a set of piecewise-constant input signals.	Characterized by a function of past-dependent life-long property, output signal and time.
Security and safety in autonomous vehicle systems [89]	Maximize the robustness of AV dynamics control to cyber-physical attacks that inject faulty data to sensor readings.	Q-learning with LSTM	AV's own position and speed along with distance and speed of some nearby objects, e.g., the leading AV.	Take appropriate speeds to maintain safe spacing between AVs.	Using a utility function that takes into account the deviation from the optimal safe spacing.
Increasing robustness of the autonomous system against adversarial attacks [93]	Devise filtering schemes to detect corrupted measurements (deception attack) and mitigate the effects of adversarial errors.	TRPO	Characterized by sensor measurements and actuation noises.	Determine which estimation rule to use to generate an estimated state from a corrupted state.	Defined via a function that takes state features as inputs.
Secure offloading in mobile edge caching [142]	Learn a policy for a mobile device to securely offload data to edge nodes against jamming and smart attacks.	DQN with hotbooting transfer learning technique.	Represented via a combination of user density, battery levels, jamming strength, and radio channel bandwidth.	Agent's actions include choosing an edge node, selecting offloading rate and time, transmit power and channel.	Computed based on secrecy capacity, energy consumption, and communication efficiency.
Anti-jamming communication scheme for CRN [152]	Derive an optimal frequency hopping policy for CRN SUs to defeat smart jammers based on a frequency-spatial anti-jamming game.	DQN that employs CNN	Consist of presence status of PUs and SINR information at time $t-1$ received from serving base station or access point.	SUs take action to leave a geographical area of heavy jamming obstructed by smart jammers or choose a frequency channel to send signals.	Represented via a utility function based on SINR and transmission cost.
Anti-jamming communication method [153], improving the previous work in [152]	Propose a smart anti-jamming scheme similar to [152] with two main differences: spectrum waterfall is used as the state, and jammers can have different channel-slot transmission structure with users.	DQN with recursive CNN due to recursion characteristic of spectrum waterfall.	Using temporal and spectral information, i.e., spectrum waterfall containing both frequency and time domain information of the network environment.	Agent's action is to select a discretized transmission frequency from a predefined set.	Defined by a function involving SINR-based transmission rate and cost for frequency switching.
Spoofing detection in wireless networks [156], [157]	Select the optimal authentication threshold.	Q-learning and Dyna-Q	Include false alarm rate and missed detection rate of the spoofing detection system at time $t-1$	Action set includes the choices of different discrete levels of the authentication thresholds bounded within a specified interval.	Using a utility function calculated based on the Bayesian risk, which is the expected payoff in spoofing detection.
Mobile offloading for cloud-based malware detection [161], improving the previous work in [162]	Improve malware detection accuracy and speed.	Hotbooting Q-learning and DQN.	Consist of current radio bandwidth and previous offloading rates of other devices.	Select optimal offloading rate level for each mobile device.	Represented by a utility function calculated based on the detection accuracy, response speed, and transmission cost.
Autonomous defense in SDN [168]	Tackle the poisoning attacks that manipulate states or flip reward signals during the training process of RL-based defense agents.	Double DQN and A3C	Represented by an array of zeros and ones showing the state of the network (whether a node is compromised or a link is switched on/off). Array length is equal to several nodes plus several links.	Attackers learn to select a node to compromise while a defender can take four actions: isolate, patch, reconnect and migrate to protect server and preserve as many nodes as possible.	Modelled based on the status of the critical server, number of preserved nodes, migration cost and the validity of actions taken.
Secure mobile crowdsensing (MCS) system [193]	Optimize payment policy to improve the sensing performance against faked sensing attacks by formulating a Stackelberg game.	DQN	Consist of the previous sensing quality and the payment policy.	Select the server's optimal payment vector to smartphone users.	Using a utility function that involves the total payment to users and the benefit of server from sensing reports of different accuracy levels.
Automated URL-based phishing detection [198]	Detect malicious websites (URLs)	DQN	Characterized by the vector space representation of website features such as HTTPS protocols, having IP address, prefix or suffix in URLs.	Select either 0 or 1, corresponding to a benign or phishing URL.	Based on the classification action, the reward equates to 1 or -1 if the URL is classified correctly or incorrectly.

theory models involving multiple DRL agents. We found that this kind of application occupies a major proportion of papers in the literature relating to DRL for cyber security problems.

A. Challenges and Future Work on Applying DRL for CPS Security Solutions

An emerging area is the use of DRL for security solutions for CPSs [175], [176]. The large-scale and complex nature of CPSs, e.g., in environmental monitoring networks, electrical smart grid systems, transportation management networks, and cyber manufacturing management systems, require security solutions to be responsive and accurate. This has been addressed by various DRL approaches, e.g., TRPO algorithm [93], LSTM-Q-learning [89], DDQN, and A3C [86]. One of the great challenges in implementing DRL algorithms for CPS security solutions is the lack of realistic CPS simulations. For example, the work in [86] had to use the MATLAB/Simulink CPS modeling and embed it into the OpenAI Gym environment. This implementation is expensive in terms of computational time due to the overhead caused by integrating the MATLAB/Simulink CPS simulation in the OpenAI Gym library. More proper simulations of CPS models embedded directly in DRL-enabled environments are, thus, encouraged in future work. Another common challenge in applying DRL algorithms is to transfer the trained policies from simulations to real-world environments. While simulations are cheap and safe for training DRL agents, the reality gap due to modeling impreciseness and errors make the transfer challenging. This is more critical for CPS modeling because of the complexity, dynamics, and large scale of CPS. Research in this direction, i.e., sim-to-real transfer for DRL-based security solutions for CPS, is worth investigating as it can help to reduce time, costs, and increase safety during the training process of DRL agents, and eventually reduce costly mistakes when executing in real-world environments.

B. Challenges and Future Work on Applying DRL for IDS

Although there have been a large number of applications of traditional RL methods for IDSs, there has been a small amount of work on DRL algorithms for this kind of application. This is probably because the integration of deep learning and RL methods has just been sustained very recently. The complexity and dynamics of intrusion detection problems are expected to be solved effectively by DRL methods, which combine the powerful representation learning and function approximation capabilities of deep learning and the optimal sequential decision-making capability of traditional RL. Applying DRL for IDS requires simulated or real-intrusion environments for training agents interactively. This is a great challenge because using real environments for training is costly, while simulated environments may be far from reality. Most of the existing studies on DRL for intrusion detection relied on game-based settings (e.g., Fig. 7) or labeled intrusion datasets. For example, the work in [125] used two datasets of labeled intrusion samples and adjusted the DRL machinery for it to work on these datasets in a supervised learning manner. This kind of application lacks a live environment and proper interactions between DRL agents and the environment. There is, thus, a gap for future research on creating more realistic environments, which are able to respond in real time to actions of the DRL agents and facilitate the full exploitation of the DRL's capabilities to solve complex and sophisticated cyber intrusion detection problems. Furthermore, as host-based and network-based IDSs have both advantages and disadvantages, combining these systems could be a logical approach. DRL-based solutions for this kind of integrated system would be another interesting future study.

C. Exploring Capabilities of Model-Based DRL Methods

Most DRL algorithms used for cyber defense so far are model-free methods, which are sample inefficient as they require a large quantity of training data. These data are difficult to obtain in real cyber security practice. Researchers generally utilize simulators to validate their proposed approaches, but these simulators often do not characterize the complexity and dynamics of the real-cyber space of the IoT systems fully. Model-based DRL methods are more appropriate than model-free methods when training data are limitedly available because, with model-based DRL, it can be easy to collect data in a scalable way. Exploration of model-based DRL methods or the integration of model-based and model-free methods for cyber defense is, thus, an interesting future study. For example, function approximators can be used to learn a proxy model of the actual high-dimensional and possibly partial observable environment [177]–[179], which can be then employed to deploy planning algorithms, e.g., Monte-Carlo tree search techniques [180], to derive optimal actions. Alternatively, model-based and model-free combination approaches, such as model-free policy with planning capabilities [181], [182] or model-based lookahead search [31], can be used as they aggregate advantages of both methods. On the other hand, the current literature on applications of DRL to cyber security often limits discretizing the action space, which restricts the full capability of the DRL solutions to real-world problems. An example is an application of DRL for selecting optimal mobile offloading rates in [161] and [162], where the action space has been discretized although a small change of the rate would primarily affect the performance of the cloud-based malware detection system. Investigation of methods that can deal with continuous action spaces in cyber environments, e.g., policy gradient and actor-critic algorithms, is another encouraging research direction.

D. Training DRL in Adversarial Cyber Environments

AI can not only help defend against cyberattacks but can also facilitate dangerous attacks, i.e., *offensive AI*. Hackers can take advantages of AI to make attacks smarter and more sophisticated to bypass detection methods to penetrate computer systems or networks. For example, hackers may employ algorithms to observe normal behaviors of users and employ the users' patterns to develop untraceable attacking strategies. ML-based systems can mimic humans to craft convincing fake messages that are utilized to conduct large-scale phishing attacks. Likewise, by creating a highly realistic fake video or audio messages based on AI advances (i.e., *deepfakes* [183]), hackers can spread false news in elections or manipulate financial markets [184]. Alternatively, attackers can poison the data pool used for training deep learning methods (i.e., *ML poisoning*) or attackers can manipulate the states or policies, falsify part of the reward signals in RL to trick the agent into taking suboptimal actions, resulting in the agent being compromised [185]. These kinds of attacks are difficult to prevent, detect, and fight against as they are part of a battle between AI systems. *adversarial ML*, especially supervised methods, have been used extensively in cyber security [186] but very few studies have been found on using adversarial RL [187]. The adversarial DRL or DRL algorithms trained in various adversarial cyber environments are worth comprehensive investigations as they can be a solution to battle against the increasingly complex offensive AI systems [188]–[190].

E. Human–Machine Teaming With Human-on-the-Loop Models

With the support of AI systems, cyber security experts no longer examine a huge volume of attack data manually to detect and defend against cyberattacks. This has many advantages because the security teams alone cannot sustain the volume. AI-enabled defense strategies can be automated and deployed rapidly and efficiently but these systems alone cannot issue creative responses when new threats are introduced. Moreover, human adversaries are always behind cybercrime or cyberwarfare. Therefore, there is a critical need for human intellect teamed with machines for cyber defenses. The traditional *human-in-the-loop* model for human–machine integration struggles to adapt quickly with cyber defense system because autonomous agent carries out part of the task and need to halt to wait for human's responses before completing the task. The modern *human-on-the-loop* model would be a solution for a future human–machine teaming cyber security system. This model allows agents to autonomously perform the task whilst humans can monitor and intervene operations of agents only when necessary. How to integrate human knowledge into DRL algorithms [191] under the human-on-the-loop model for cyber defense is an interesting research question.

F. Exploring Capabilities of Multiagent DRL Methods

As hackers utilized more and more sophisticated and large-scale approaches to attack computer systems and networks, the defense strategies need to be more intelligent and large-scale as well. Multiagent DRL is a research direction that can be explored to tackle this problem. Game theory models for cyber security reviewed in this article have involved multiple agents but they are restricted to a couple of attackers and defenders with limited communication, cooperation, and coordination among the agents. These aspects of multiagent DRL need to be investigated thoroughly in cyber security problems to enable an effective large-scale defense plan. Challenges of multiagent DRL itself then need to be addressed, such as nonstationarity, partial observability, and efficient multiagent training schemes [192]. On the other hand, the RL methodology has been applied to deal with various cyberattacks, e.g., jamming, spoofing, false data injection, malware, DoS, DDoS, brute force, Heartbleed, botnet, Web attack, and infiltration attack [193]–[198]. However, recently emerged or new types of attacks have been largely unaddressed. One of these new types is the *bit-and-piece DDoS attack*. This attack injects small junk into legitimate traffic of over a large number of IP addresses so that it can bypass many detection methods as there is so little of it per address. Another emerging attack, for instance, is attacking from the computing cloud to breach systems of companies who manage IT systems for other firms or host other firms' data on their servers. Alternatively, hackers can use quantum physics-based powerful computers to crack encryption algorithms that are currently used to protect various types of invaluable data [184]. Consequently, a future study on addressing these new types of attacks is encouraged.

References

[1] I. Kakalou, K. E. Psannis, P. Krawiec, and R. Badea, "Cognitive radio network and network service chaining toward 5G: Challenges and requirements," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 145–151, Nov. 2017.

[2] Y. Huang, S. Li, C. Li, Y. T. Hou, and W. Lou, "A deep-reinforcement-learning-based approach to dynamic eMBB/URLLC multiplexing in 5G NR," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6439–6456, Jul. 2020.

[3] P. Wang, L. T. Yang, X. Nie, Z. Ren, J. Li, and L. Kuang, "Data-driven software defined network attack detection: State-of-the-art and perspectives," *Inf. Sci.*, vol. 513, pp. 65–83, Mar. 2020.

[4] A. Botta, W. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Generat. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[5] O. Krestinskaya, A. P. James, and L. O. Chua, "Neuromemristive circuits for edge computing: A review," *IEEE Trans. neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 4–23, Jan. 2020.

[6] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[7] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[8] B. Geluvaraj, P. M. Satwik, and T. A. Kumar, "The future of cybersecurity: Major role of artificial intelligence, machine learning, and deep learning in cyberspace," in *Proc. Int. Conf. Comput. Netw. Commun. Technol.*, 2019, pp. 739–747.

[9] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[10] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *Proc. 10th Int. Conf. Cyber Conflict (CyCon)*, May 2018, pp. 371–390.

[11] Y. Xin *et al.*, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[12] N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, "Machine learning aided Android malware classification," *Comput. Elect. Eng.*, vol. 61, pp. 266–274, Jul. 2017.

[13] R. M. H. Babu, R. Vinayakumar, and K. P. Soman, "A short review on applications of deep learning for cyber security," 2018, *arXiv:1812.06292*. [Online]. Available: http://arxiv.org/abs/1812.06292

[14] D. Berman, A. Buczak, J. Chavis, and C. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, Apr. 2019.

[15] S. Paul, Z. Ni, and C. Mu, "A learning-based solution for an adversarial repeated game in cyber–physical power systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4512–4523, Nov. 2020, doi: 10.1109/TNNLS.2019.2955857.

[16] D. Ding, Q.-L. Han, Y. Xiang, C. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, Jan. 2018.

[17] M. Wu, Z. Song, and Y. B. Moon, "Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods," *J. Intell. Manuf.*, vol. 30, no. 3, pp. 1111–1123, Mar. 2019.

[18] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning," 2018, *arXiv:1801.06275*. [Online]. Available: http://arxiv.org/abs/1801.06275

[19] A. Sharma, Z. Kalbarczyk, J. Barlow, and R. Iyer, "Analysis of security data from a large computing organization," in *Proc. IEEE/IFIP 41st Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2011, pp. 506–517.

[20] N. D. Nguyen, T. Nguyen, and S. Nahavandi, "System design perspective for human-level agents using deep reinforcement learning: A survey," *IEEE Access*, vol. 5, pp. 27091–27102, 2017.

[21] Z. Sui, Z. Pu, J. Yi, and S. Wu, "Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2358–2372, Jun. 2021, doi: 10.1109/TNNLS.2020.3004893.

[22] A. Tsantekidis, N. Passalis, A.-S. Toufa, K. Saitas-Zarkias, S. Chairistanidis, and A. Tefas, "Price trailing for financial trading using deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 2837–2846, Jul. 2021, doi: 10.1109/TNNLS.2020.2997523.

[23] T. T. Nguyen, N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, and C. P. Lim, "A multi-objective deep reinforcement learning framework," 2018, *arXiv:1803.02965*. [Online]. Available: http://arxiv.org/abs/1803.02965

[24] X. Wang, Y. Gu, Y. Cheng, A. Liu, and C. L. P. Chen, "Approximate policy-based accelerated deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 1820–1830, Jun. 2020.

[25] M. H. Ling, K.-L.-A. Yau, J. Qadir, G. S. Poh, and Q. Ni, "Application of reinforcement learning for security enhancement in cognitive radio networks," *Appl. Soft Comput.*, vol. 37, pp. 809–829, Dec. 2015.

[26] Y. Wang, Z. Ye, P. Wan, and J. Zhao, "A survey of dynamic spectrum allocation based on reinforcement learning algorithms in cognitive radio networks," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 493–506, Mar. 2019.

[27] X. Lu, L. Xiao, T. Xu, Y. Zhao, Y. Tang, and W. Zhuang, "Reinforcement learning based PHY authentication for VANETs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3068–3079, Mar. 2020.

[28] M. Alauthman, N. Aslam, M. Al-Kasassbeh, S. Khan, A. Al-Qerem, and K.-K. R. Choo, "An efficient reinforcement learning-based botnet detection approach," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102479.

[29] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[30] N. D. Nguyen, S. Nahavandi, and T. Nguyen, "A human mixed strategy approach to deep reinforcement learning," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 4023–4028.

[31] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[32] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[33] O. Vinyals *et al.*, "StarCraft II: A new challenge for reinforcement learning," 2017, *arXiv:1708.04782*. [Online]. Available: http://arxiv.org/abs/1708.04782

[34] P. Sun *et al.*, "TStarBots: Defeating the cheating level builtin AI in StarCraft II in the full game," 2018, *arXiv:1809.07193*. [Online]. Available: http://arxiv.org/abs/1809.07193

[35] Z.-J. Pang, R.-Z. Liu, Z.-Y. Meng, Y. Zhang, Y. Yu, and T. Lu, "On reinforcement learning for full-length game of StarCraft," 2018, *arXiv:1809.09095*. [Online]. Available: http://arxiv.org/abs/1809.09095

[36] V. Zambaldi *et al.*, "Relational deep reinforcement learning," 2018, *arXiv:1806.01830*. [Online]. Available: http://arxiv.org/abs/1806.01830

[37] M. Jaderberg *et al.*, "Human-level performance in first-person multiplayer games with population-based deep reinforcement learning," 2018, *arXiv:1807.01281*. [Online]. Available: http://arxiv.org/abs/1807.01281

[38] OpenAI. (Mar.1, 2019)OpenAI Five. [Online]. Available: https://openai.com/five/

[39] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3389–3396.

[40] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2034–2039.

[41] T. Nguyen, N. D. Nguyen, F. Bello, and S. Nahavandi, "A new tensioning method using deep reinforcement learning for surgical pattern cutting," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 1339–1344, doi: 10.1109/ICIT.2019.8755235.

[42] N. D. Nguyen, T. Nguyen, S. Nahavandi, A. Bhatti, and G. Guest, "Manipulating soft tissues by deep reinforcement learning for autonomous robotic surgery," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2019, pp. 1–7, doi: 10.1109/SYSCON.2019.8836924.

[43] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2469–2489, Jul. 2019.

[44] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2063–2079, Jun. 2018.

[45] M. Popova, O. Isayev, and A. Tropsha, "Deep reinforcement learning for de novo drug design," *Sci. Adv.*, vol. 4, no. 7, Jul. 2018, Art. no. eaap7885.

[46] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[47] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[48] Z. Wang *et al.*, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.

[49] H. Zhu, Y. Cao, W. Wang, T. Jiang, and S. Jin, "Deep reinforcement learning for mobile edge caching: Review, new features, and open issues," *IEEE Netw.*, vol. 32, no. 6, pp. 50–57, Nov. 2018.

[50] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[51] Y. Zhang, J. Yao, and H. Guan, "Intelligent cloud resource management with deep reinforcement learning," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 60–69, Nov./Dec. 2017.

[52] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2375–2385, Aug. 2017.

[53] R. Shafin *et al.*, "Self-tuning sectorization: Deep reinforcement learning meets broadcast beam optimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4038–4053, Jun. 2020.

[54] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 362–370, Sep. 2018.

[55] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric IoT," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 3, pp. 781–796, Jul. 2020, doi: 10.1109/TETC.2018.2805718.

[56] Y. He, C. Liang, F. R. Yu, and Z. Han, "Trust-based social networks with computing, caching and communications: A deep reinforcement learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 66–79, Jan. 2020, doi: 10.1109/TNSE.2018.2865183.

[57] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019, doi: 10.1109/COMST.2019.2916583.

[58] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, and Y. Zhang, "Blockchain and deep reinforcement learning empowered intelligent 5G beyond," *IEEE Netw.*, vol. 33, no. 3, pp. 10–17, May/Jun. 2019.

[59] A. S. Leong, A. Ramaswamy, D. E. Quevedo, H. Karl, and L. Shi, "Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems," *Automatica*, vol. 113, Mar. 2020, Art. no. 108759.

[60] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[61] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[62] V. Mnih *et al.*, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[63] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*. [Online]. Available: http://arxiv.org/abs/1511.05952

[64] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.

[65] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.

[66] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[67] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347

[68] C. Wu *et al.*, "Variance reduction for policy gradient with action-dependent factorized baselines," 2018, *arXiv:1803.07246*. [Online]. Available: http://arxiv.org/abs/1803.07246

[69] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: http://arxiv.org/abs/1509.02971

[70] G. Barth-Maron *et al.*, "Distributed distributional deterministic policy gradients," 2018, *arXiv:1804.08617*. [Online]. Available: http://arxiv.org/abs/1804.08617

[71] M. Jaderberg *et al.*, "Reinforcement learning with unsupervised auxiliary tasks," 2016, *arXiv:1611.05397*. [Online]. Available: http://arxiv.org/abs/1611.05397

[72] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2775–2785.

[73] M. Lapan, *Deep Reinforcement Learning Hands-on: Apply Modern RL Methods, With Deep Q-Networks, Value Iteration, Policy Gradients, TRPO, AlphaGo Zero and More*. Birmingham, U.K.: Packt Publishing, 2018.

[74] (Dec. 14, 220). *OpenAI Gym Toolkit Documentation. Classic Control: Control Theory Problems From the Classic RL Literature*. [Online]. Available: https://gym.openai.com/envs/#classic_control

[75] (Dec. 14, 220). *OpenAI Gym Toolkit Documentation. Box2D: Continuous Control Tasks in the Box2D Simulator*. [Online]. Available: https://gym.openai.com/envs/#box2d

[76] L. Wang, M. Törngren, and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *J. Manuf. Syst.*, vol. 37, pp. 517–527, Oct. 2015.

[77] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Syst. J.*, vol. 11, no. 1, pp. 88–95, Mar. 2017.

[78] P. M. Shakeel, S. Baskar, V. R. S. Dhulipala, S. Mishra, and M. M. Jaber, "Maintaining security and privacy in health care system using learning based deep-Q-networks," *J. Med. Syst.*, vol. 42, no. 10, p. 186, 2018.

[79] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, "A survey on smart grid cyber-physical system testbeds," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 446–464, 1st Quart., 2017.

[80] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, "Evaluation of reinforcement learning-based false data injection attack to automatic voltage control," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2158–2169, Mar. 2018.

[81] Z. Ni and S. Paul, "A multistage game in smart grid security: A reinforcement learning solution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2684–2695, Sep. 2019, doi: 10.1109/TNNLS.2018.2885530.

[82] A. Ferdowsi, A. Eldosouky, and W. Saad, "Interdependence-aware game-theoretic framework for secure intelligent transportation systems," *IEEE Internet Things J.*, early access, Sep. 1, 2020, doi: 10.1109/JIOT.2020.3020899.

[83] Y. Li *et al.*, "Nonlane-discipline-based car-following model for electric vehicles in transportation-cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 38–47, Jan. 2018.

[84] C. Li and M. Qiu, *Reinforcement Learning for Cyber-Physical Systems: With Cybersecurity Case Studies*. Boca Raton, FL, USA: CRC Press, 2019.

[85] M. Feng and H. Xu, "Deep reinforecement learning based optimal defense for cyber-physical system in presence of unknown cyber-attack," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–8.

[86] T. Akazaki, S. Liu, Y. Yamagata, Y. Duan, and J. Hao, "Falsification of cyber-physical systems using deep reinforcement learning," in *Proc. Int. Symp. Formal Methods*, 2018, pp. 456–465.

[87] H. Abbas and G. Fainekos, "Convergence proofs for simulated annealing falsification of safety properties," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2012, pp. 1594–1601.

[88] S. Sankaranarayanan and G. Fainekos, "Falsification of temporal properties of hybrid systems using the cross-entropy method," in *Proc. 15th ACM Int. Conf. Hybrid Syst., Comput. Control (HSCC)*, 2012, pp. 125–134.

[89] A. Ferdowsi, U. Challita, W. Saad, and N. B. Mandayam, "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 307–312.

[90] X. Wang, R. Jiang, L. Li, Y.-L. Lin, and F.-Y. Wang, "Long memory is important: A test study on deep-learning based car-following model," *Phys. A, Stat. Mech. Appl.*, vol. 514, pp. 786–795, Jan. 2019.

[91] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[92] I. Rasheed, F. Hu, and L. Zhang, "Deep reinforcement learning approach for autonomous vehicle systems for maintaining security and safety using LSTM-GAN," *Veh. Commun.*, vol. 26, Dec. 2020, Art. no. 100266.

[93] A. Gupta and Z. Yang, "Adversarial reinforcement learning for observer design in autonomous systems under cyber attacks," 2018, *arXiv:1809.06784*. [Online]. Available: http://arxiv.org/abs/1809.06784

[94] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in *Proc. 7th Int. Conf. Emerg. Secur. Technol. (EST)*, Sep. 2017, pp. 138–143.

[95] S. Jose, D. Malathi, B. Reddy, and D. Jayaseeli, "A survey on anomaly based host intrusion detection system," *J. Phys., Conf. Ser.*, vol. 1000, Apr. 2018, Art. no. 012049.

[96] S. Roshan, Y. Miche, A. Akusok, and A. Lendasse, "Adaptive and online network intrusion detection system using clustering and extreme learning machines," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1752–1779, Mar. 2018.

[97] S. Dey, Q. Ye, and S. Sampalli, "A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks," *Inf. Fusion*, vol. 49, pp. 205–215, Sep. 2019.

[98] D. Papamartzivanos, F. G. Mármol, and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems," *IEEE Access*, vol. 7, pp. 13546–13560, 2019.

[99] W. Haider, G. Creech, Y. Xie, and J. Hu, "Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks," *Future Internet*, vol. 8, no. 4, p. 29, Jul. 2016.

[100] P. Deshpande, S. C. Sharma, S. K. Peddoju, and S. Junaid, "HIDS: A host based intrusion detection system for cloud computing environment," *Int. J. Syst. Assurance Eng. Manage.*, vol. 9, no. 3, pp. 567–576, Jun. 2018.

[101] M. Nobakht, V. Sivaraman, and R. Boreli, "A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow," in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 147–156.

[102] P. A. A. Resende and A. C. Drummond, "A survey of random forest based methods for intrusion detection systems," *ACM Comput. Surv.*, vol. 51, no. 3, p. 48, 2018.

[103] G. Kim, H. Yi, J. Lee, Y. Paek, and S. Yoon, "LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," 2016, *arXiv:1611.01726*. [Online]. Available: http://arxiv.org/abs/1611.01726

[104] A. Chawla, B. Lee, S. Fallon, and P. Jacob, "Host based intrusion detection system with combined CNN/RNN model," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2018, pp. 149–158.

[105] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Inf. Sci.*, vol. 513, pp. 386–396, Mar. 2020.

[106] A. Janagam and S. Hossen, "Analysis of network intrusion detection system with machine learning algorithms (deep reinforcement learning algorithm)," M.S. thesis, Dept. Comput., Blekinge Inst. Technol., Stockholm, Sweden, 2018.

[107] X. Xu and T. Xie, "A reinforcement learning approach for host-based intrusion detection using sequences of system calls," in *Proc. Int. Conf. Intell. Comput.*, 2005, pp. 995–1003.

[108] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[109] X. Xu, "A sparse kernel-based least-squares temporal difference algorithm for reinforcement learning," in *Proc. Int. Conf. Natural Comput.*, 2006, pp. 47–56.

[110] X. Xu and Y. Luo, "A kernel-based reinforcement learning approach to dynamic behavior modeling of intrusion detection," in *Proc. Int. Symp. Neural Netw.*, 2007, pp. 455–464.

[111] X. Xu, "Sequential anomaly detection based on temporal-difference learning: Principles, models and case studies," *Appl. Soft Comput.*, vol. 10, no. 3, pp. 859–867, Jun. 2010.

[112] B. Deokar and A. Hazarnis, "Intrusion detection system using log files and reinforcement learning," *Int. J. Comput. Appl.*, vol. 45, no. 19, pp. 28–35, 2012.

[113] K. Malialis, "Distributed reinforcement learning for network intrusion response," Doctoral Dissertation, Dept. Comput. Sci., Univ. York, York, U.K., 2014.

[114] K. Malialis and D. Kudenko, "Distributed response to network intrusions using multiagent reinforcement learning," *Eng. Appl. Artif. Intell.*, vol. 41, pp. 270–284, May 2015.

[115] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.

[116] D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yam, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 29–42, Feb. 2005.

[117] R. Bhosale, S. Mahajan, and P. Kulkarni, "Cooperative machine learning for intrusion detection system," *Int. J. Sci. Eng. Res.*, vol. 5, no. 1, pp. 1780–1785, 2014.

[118] A. Herrero and E. Corchado, "Multiagent systems for network intrusion detection: A review," in *Computational Intelligence in Security for Information Systems*. Berlin, Germany: Springer, 2009, pp. 143–154.

[119] A. Detwarasiti and R. D. Shachter, "Influence diagrams for team decision analysis," *Decis. Anal.*, vol. 2, no. 4, pp. 207–228, Dec. 2005.

[120] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah, and A. Abraham, "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks," *Eng. Appl. Artif. Intell.*, vol. 32, pp. 228–241, Jun. 2014.

[121] P. Muñoz, R. Barco, and I. de la Bandera, "Optimization of load balancing using fuzzy Q-Learning for next generation wireless networks," *Expert Syst. Appl.*, vol. 40, no. 4, pp. 984–994, Mar. 2013.

[122] S. Shamshirband, N. B. Anuar, M. L. M. Kiah, and A. Patel, "An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique," *Eng. Appl. Artif. Intell.*, vol. 26, no. 9, pp. 2105–2127, Oct. 2013.

[123] S. Varshney and R. Kuma, "Variants of LEACH routing protocol in WSN: A comparative analysis," in *Proc. 8th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2018, pp. 199–204.

[124] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Comput. Netw.*, vol. 159, pp. 96–109, Aug. 2019.

[125] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112963.

[126] I. A. Saeed, A. Selamat, M. F. Rohani, O. Krejcar, and J. A. Chaudhry, "A systematic state-of-the-art analysis of multi-agent intrusion detection," *IEEE Access*, vol. 8, pp. 180184–180209, 2020.

[127] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," in *Proc. 43rd Hawaii Int. Conf. Syst. Sci.*, 2010, pp. 1–10.

[128] S. Shiva, S. Roy, and D. Dasgupta, "Game theory for cyber security," in *Proc. 6th Annu. Workshop Cyber Secur. Inf. Intell. Res. (CSIIRW)*, 2010, p. 34.

[129] K. Ramachandran and Z. Stefanova, "Dynamic game theories in cyber security," in *Proc. Int. Conf. Dyn. Syst. Appl.*, vol. 7, 2016, pp. 303–310.

[130] Y. Wang, Y. Wang, J. Liu, Z. Huang, and P. Xie, "A survey of game theoretic methods for cyber security," in *Proc. IEEE 1st Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2016, pp. 631–636.

[131] Q. Zhu and S. Rass, "Game theory meets network security: A tutorial," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 2163–2165.

[132] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 4, pp. 42–56, 4th Quart., 2009.

[133] S. Hu, D. Yue, X. Xie, X. Chen, and X. Yin, "Resilient event-triggered controller synthesis of networked control systems under periodic DoS jamming attacks," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4271–4281, Dec. 2019, doi: 10.1109/TCYB.2018.2861834.

[134] H. Boche and C. Deppe, "Secure identification under passive eavesdroppers and active jamming attacks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 472–485, Feb. 2019.

[135] Y. Wu, B. Wang, K. J. R. Liu, and T. C. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 4–15, Jan. 2012.

[136] S. Singh and A. Trivedi, "Anti-jamming in cognitive radio networks using reinforcement learning algorithms," in *Proc. 9th Int. Conf. Wireless Opt. Commun. Netw. (WOCN)*, Sep. 2012, pp. 1–5.

[137] Y. Gwon, S. Dastangoo, C. Fossa, and H. T. Kung, "Competing mobile network game: Embracing antijamming and jamming strategies with reinforcement learning," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2013, pp. 28–36.

[138] W. G. Conley and A. J. Miller, "Cognitive jamming game for dynamically countering ad hoc cognitive radio networks," in *Proc. MILCOM IEEE Mil. Commun. Conf.*, Nov. 2013, pp. 1176–1182.

[139] K. Dabcevic, A. Betancourt, L. Marcenaro, and C. S. Regazzoni, "A fictitious play-based game-theoretical approach to alleviating jamming attacks for cognitive radios," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 8158–8162.

[140] F. Slimeni, B. Scheers, Z. Chtourou, and V. Le Nir, "Jamming mitigation in cognitive radio networks using a modified Q-learning algorithm," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2015, pp. 1–7.

[141] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[142] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 116–122, Jun. 2018.

[143] M. A. Aref, S. K. Jayaweera, and S. Machuzak, "Multi-agent reinforcement learning based cognitive anti-jamming," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[144] S. Machuzak and S. K. Jayaweera, "Reinforcement learning based anti-jamming with wideband autonomous cognitive radios," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2016, pp. 1–5.

[145] M. D. Felice, L. Bedogni, and L. Bononi, "Reinforcement learning-based spectrum management for cognitive radio networks: A literature review and case study," in *Handbook of Cognitive Radio*. Singapore: Springer, 2019, pp. 1849–1886.

[146] A. Attar, H. Tang, A. V. Vasilakos, F. R. Yu, and V. C. M. Leung, "A survey of security challenges in cognitive radio networks: Solutions and future research directions," *Proc. IEEE*, vol. 100, no. 12, pp. 3172–3186, Dec. 2012.

[147] B. Wang, Y. Wu, K. J. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, Apr. 2011.

[148] M. L. Littman, "Markov games as a framework for multiagent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 157–163.

[149] L. Xiao, Y. Li, J. Liu, and Y. Zhao, "Power control with reinforcement learning in cooperative cognitive radio networks against jamming," *J. Supercomput.*, vol. 71, no. 9, pp. 3237–3257, Sep. 2015.

[150] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang, "Coping with a smart jammer in wireless networks: A Stackelberg game approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 4038–4047, Aug. 2013.

[151] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artif. Intell.*, vol. 136, no. 2, pp. 215–250, Apr. 2002.

[152] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2087–2091.

[153] X. Liu, Y. Xu, L. Jia, Q. Wu, and A. Anpalagan, "Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 998–1001, Mar. 2018.

[154] W. Chen and X. Wen, "Perceptual spectrum waterfall of pattern shape recognition algorithm," in *Proc. 18th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan. 2016, pp. 382–389.

[155] K. Zeng, K. Govindan, and P. Mohapatra, "Non-cryptographic authentication and identification in wireless networks," *IEEE Wireless Commun.*, vol. 17, no. 5, pp. 56–62, Oct. 2010.

[156] L. Xiao, Y. Li, G. Liu, Q. Li, and W. Zhuang, "Spoofing detection with reinforcement learning in wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–5.

[157] L. Xiao, Y. Li, G. Han, G. Liu, and W. Zhuang, "PHY-layer spoofing detection with reinforcement learning in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 10037–10047, Dec. 2016.

[158] R. S. Sutton, "Integrated architecture for learning, planning, and reacting based on approximating dynamic programming," in *Proc. 7th Int. Conf. Mach. Learn.*, 1990, pp. 216–224.

[159] X. Sun, J. Dai, P. Liu, A. Singhal, and J. Yen, "Using Bayesian networks for probabilistic identification of zero-day attack paths," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2506–2521, Oct. 2018.

[160] Y. Afek, A. Bremler-Barr, and S. L. Feibish, "Zero-day signature extraction for high-volume attacks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 691–706, Apr. 2019, doi: 10.1109/TNET.2019.2899124.

[161] X. Wan, G. Sheng, Y. Li, L. Xiao, and X. Du, "Reinforcement learning based mobile offloading for cloud-based malware detection," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[162] Y. Li, J. Liu, Q. Li, and L. Xiao, "Mobile cloud offloading for malware detections with learning," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 197–201.

[163] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for RSU clouds in support of the internet of vehicles," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 133–144, Apr. 2015.

[164] R. Huang, X. Chu, J. Zhang, and Y. H. Hu, "Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: A prototype," *Int. J. Distrib. Sensor Netw.*, vol. 2015, pp. 1–12, Oct. 2015.

[165] S. Kim, J. Son, A. Talukder, and C. S. Hong, "Congestion prevention mechanism based on Q-leaning for efficient routing in SDN," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2016, pp. 124–128.

[166] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2016, pp. 25–33.

[167] A. Mestres *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, Sep. 2017.

[168] Y. Han *et al.*, "Reinforcement learning for autonomous defence in software-defined networking," in *Proc. Int. Conf. Decis. Game Theory Secur.*, 2018, pp. 145–165.

[169] B. Lantz and B. O'Connor, "A mininet-based virtual testbed for distributed SDN development," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 365–366, 2015.

[170] M. Zhu, Z. Hu, and P. Liu, "Reinforcement learning algorithms for adaptive cyber defense against heartbleed," in *Proc. 1st ACM Workshop Moving Target Defense (MTD)*, 2014, pp. 51–58.

[171] J. Wang, M. Zhao, Q. Zeng, D. Wu, and P. Liu, "Risk assessment of buffer 'heartbleed' over-read vulnerabilities," in *Proc. 45th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2015, pp. 555–562.

[172] B. Luo, Y. Yang, C. Zhang, Y. Wang, and B. Zhang, "A survey of code reuse attack and defense," in *Proc. Int. Conf. Intell. Interact. Syst. Appl.*, 2018, pp. 782–788.

[173] R. Elderman, L. J. J. Pater, A. S. Thie, M. M. Drugan, and M. M. Wiering, "Adversarial reinforcement learning in a cyber security simulation," in *Proc. 9th Int. Conf. Agents Artif. Intell.*, 2017, pp. 559–566.

[174] K. Chung, C. A. Kamhoua, K. A. Kwiat, Z. T. Kalbarczyk, and R. K. Iyer, "Game theory with learning for cyber security monitoring," in *Proc. IEEE 17th Int. Symp. High Assurance Syst. Eng. (HASE)*, Jan. 2016, pp. 1–8.

[175] F. Wei, Z. Wan, and H. He, "Cyber-attack recovery strategy for smart grid based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2476–2486, May 2020.

[176] X. Liu, J. Ospina, and C. Konstantinou, "Deep reinforcement learning for cybersecurity assessment of wind integrated power systems," 2020, *arXiv:2007.03025*. [Online]. Available: http://arxiv.org/abs/2007.03025

[177] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in Atari games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2863–2871.

[178] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015, *arXiv:1511.05440*. [Online]. Available: http://arxiv.org/abs/1511.05440

[179] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7559–7566.

[180] C. B. Browne *et al.*, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[181] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel, "Value iteration networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2154–2162.

[182] R. Pascanu *et al.*, "Learning model-based planning from scratch," 2017, *arXiv:1707.06170*. [Online]. Available: http://arxiv.org/abs/1707.06170

[183] T. T. Nguyen, Q. V. H. Nguyen, C. M. Nguyen, D. Nguyen, D. Thanh Nguyen, and S. Nahavandi, "Deep learning for deepfakes creation and detection: A survey," 2019, *arXiv:1909.11573*. [Online]. Available: http://arxiv.org/abs/1909.11573

[184] M. Giles. (Jan. 4, 2019). *Five Emerging Cyber-Threats to Worry About in 2019*. MIT Technology Review. [Online]. Available: https://www.technologyreview.com/2019/01/04/66232/five-emerging-cyber-threats-2019/

[185] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," in *Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, 2017, pp. 262–275.

[186] V. Duddu, "A survey of adversarial machine learning in cyber warfare," *Defence Sci. J.*, vol. 68, no. 4, pp. 356–366, 2018.

[187] T. Chen, J. Liu, Y. Xiang, W. Niu, E. Tong, and Z. Han, "Adversarial attack and defense in reinforcement learning-from AI security view," *Cybersecurity*, vol. 2, no. 1, p. 11, Dec. 2019.

[188] I. Ilahi *et al.*, "Challenges and countermeasures for adversarial attacks on deep reinforcement learning," 2020, *arXiv:2001.09684*. [Online]. Available: http://arxiv.org/abs/2001.09684

[189] L. Tong, A. Laszka, C. Yan, N. Zhang, and Y. Vorobeychik, "Finding needles in a moving haystack: Prioritizing alerts with adversarial reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 946–953.

[190] J. Sun *et al.*, "Stealthy and efficient adversarial attacks against deep reinforcement learning," 2020, *arXiv:2005.07099*. [Online]. Available: http://arxiv.org/abs/2005.07099

[191] T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Multi-agent deep reinforcement learning with human strategies," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 1357–1362, doi: 10.1109/ICIT.2019.8755032.

[192] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[193] L. Xiao, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 35–47, Jan. 2018.

[194] Y. Liu, M. Dong, K. Ota, J. Li, and J. Wu, "Deep reinforcement learning based smart mitigation of DDoS flooding in software-defined networks," in *Proc. IEEE 23rd Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Sep. 2018, pp. 1–6.

[195] Y. Xu, G. Ren, J. Chen, X. Zhang, L. Jia, and L. Kong, "Interference-aware cooperative anti-jamming distributed channel selection in UAV communication networks," *Appl. Sci.*, vol. 8, no. 10, p. 1911, Oct. 2018.

[196] F. Yao and L. Jia, "A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1024–1027, Aug. 2019, doi: 10.1109/LWC.2019.2904486.

[197] Y. Li *et al.*, "On the performance of deep reinforcement learning-based anti-jamming method confronting intelligent jammer," *Appl. Sci.*, vol. 9, no. 7, p. 1361, 2019.

[198] M. Chatterjee and A.-S. Namin, "Detecting phishing websites through deep reinforcement learning," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 227–232.

Thanh Thi Nguyen received the Ph.D. degree in mathematics and statistics from Monash University, Clayton, VIC, Australia, in 2013.

He was a Visiting Scholar with the Department of Computer Science, Stanford University, Stanford, CA, USA, in 2015, and the Edge Computing Laboratory, John A. Paulson School of Engineering and Applied Sciences, Harvard University, Massachusetts, USA, in 2019. He is currently a Senior Lecturer with the School of Information Technology, Deakin University, Burwood, VIC, Australia. He received an Alfred Deakin Post-Doctoral Research Fellowship in 2016, a European–Pacific Partnership for the ICT Expert Exchange Program Award from the European Commission in 2018, and an Australia–India Strategic Research Fund Early- and Mid-Career Fellowship from the Australian Academy of Science in 2020. He has expertise in various areas, including artificial intelligence, deep learning, deep reinforcement learning, cyber security, IoT, and data science.

Vijay Janapa Reddi (Member, IEEE) received the Ph.D. degree in computer science from Harvard University, Cambridge, MA, USA, in 2010.

He is currently an Associate Professor with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, where he is the Director of the Edge Computing Laboratory. His research interests include computer architecture and system-software design, with an emphasis on the context of mobile and edge computing platforms based on machine learning.

Dr. Reddi was a recipient of multiple awards, including the Best Paper Award at the 2005 International Symposium on Microarchitecture and the 2009 International Symposium on High-Performance Computer Architecture, the IEEE's Top Picks in Computer Architecture Awards in 2006, 2010, 2011, 2016, and 2017, the Google Faculty Research Awards in 2012, 2013, 2015, and 2017, the Intel Early Career Award in 2013, the National Academy of Engineering Gilbreth Lecturer Honor in 2016, and the IEEE TCCA Young Computer Architect Award in 2016.