Recent Advances in Reinforcement Learning Applied to Intrusion Detection

Zheni Utic
Department of Mathematical Sciences
Georgia Southern University
Savannah, USA
zutic@georgiasouthern.edu

Kandethody Ramachandran
Mathematics and Statistics Department
University of South Florida
Tampa, USA
ram@usf.edu

***Abstract*—With the progress of the Internet and the dependence on mobile devices, the importance of advanced technologies and methodologies for protection in the intrusion detection domain escalates to unseen levels. Even though there are many established machine learning security methods, there is no adequate study done in the Reinforcement learning area. This paper aims to create a survey and to outline the recent progress in the Reinforcement learning field applied to intrusion detection and emphasize the importance of further research in this area.**

***Index Terms*—intrusion detection, reinforcement learning, machine learning, survey**

I. Introduction

With the advances of the pandemic situation worldwide, we found ourselves in an unpredictable and unforeseen environment. We all face the decision of whether we should go back to basics or continue living our lives with the same standard and amenities as before, but becoming almost entirely dependent on the Internet. Our current era made us highly reliant on new software and hardware technologies that are vulnerable and need to be steadily protected. Reinforcement learning (RL) is a machine learning approach widely used in robotics, where an agent has to make decisions in an unknown and sometimes changing environment. The utilization of this popular method is minimal in the intrusion detection domain. Consequently, this paper will emphasize the necessity for more research in RL for security applications and outline the recent advances in the method applied to intrusion detection.

II. Outline of the Reinforcement Learning Method in Intrusion Detection

A. Environment as a Markov Decision Process

RL is a machine learning technique that helps a particular agent decide what actions to take in an unknown environment. It allows the agent to observe and analyze a situation and select the best possible action based on an expected maximized return. The agent learns to take the most proper action by representing the environment as a Markov decision process (MDP). MDP is based on a 5-tuple $(S, A, T, R,)$ where S stands for the set of states, A denotes a set of actions , T is a transitional probability: going from one state to another., R is a reward function, and γ is a discount factor.

The agent learns from the environment he can interact with based on a trial and error approach. The process starts with the idea that the agent finds himself in a particular state, then takes action, receives feedback, and transitions to another state.

The possibility for a classification machine learning approach is the goal when the method is applied to intrusion detection. It is the act of representing a particular system as a set of labeled instances, learning the dependence of the factors and its corresponding label, and then predicting the label of a set of unlabeled networks as most accurately as possible. The Reinforcement Learning for Intrusion detection has to know how to assign each instance from the system to a specific label. The classification process occurs based on consecutively evaluating the characteristics of the network and assigning labels. The different labels in general represent the states of the network, whether it is in a normal state or under the possible types of attack. The attacks are usually represented in four main categories Denial-of-Service (DoS), Probing, User to Root (U2R) and Remote to User (R2L). The goal is to create a learning algorithm or procedure that will focus only on the relevant information in the system and stop the process once the classification is done. To achieve this goal, we have to represent the various elements of the intrusion detection system as a Markov Decision Process and apply the Reinforcement learning concept. For example, the action set could be: continue, classify the state of the network as being under a particular type of attack or not and stop. The choice of each action depends on the state that the decision agent is currently in.

The choice of action a , given the state s is called the policy π of the classifying agent [33]. A policy π is defined as the conditional probability of selecting different actions given every state s. The evaluation of policy π can be performed with the creation of the called action-value function. The objective is to create an action-value function $q_\pi(s, a)$ (1) , and we are interested in obtaining the maximum value of the function across all policies, by using the Bellman optimaility equation for the state-value function (4) or the Bellman optimality equation for the action-value function (3) The action-value function under policy π, with a discount factor γ and return R at time t and episode k is:

$$q_\pi(s,a) = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+1+k} | S_t = s, A_t = a], \forall s \in S \quad (1)$$

The state-value function under policy π:

$$v_\pi(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+1+k} | S_t = s], \forall s \in S \quad (2)$$

It represents the cumulative discounted expected return for each episode k at a time t, conditional on the states $S_t = s$ that the agent needs to explore and the possible corresponding actions he has to take.

Bellman optimality equation for q^* is:

$$q^*(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma \underset{a'}{max} q(s',a')] \quad (3)$$

Bellman optimality equation for v^* is:

$$v^*(s) = \underset{a}{max} \sum_{s',r} p(s',r|s,a)[r + \gamma v^*(s')] \quad (4)$$

The learning of the structure process is based on two possible approaches. The first is called a model method, where the agent has to estimate the transition probabilities to solve the MDP. The second approach is called a model-free approach, where the agent optimizes the action-value function without knowing the transitional probabilities.

B. Temporal Difference Method for Estimating the Value Functions

Temporal Difference (TD) is a approach utilized to calculate the current state-value function based on the expected reward, and it is dependent on the future values. It is approach that helps to estimate the state-value function. An advantage of estimating the state-value function is that the agent does not have to wait for the final reward to be received. In general, he has to wait for the reward before updating the actions and the states so he can continue in the next state. Once the agent acquires the last reward, he traces the route to the final state and updates each value. The procedure may be described with the following equation:

$$v(s_t) = E_\pi[R_0 + \gamma v_\pi \hat{(}s_t) | S_t = s], \forall s \in S \quad (5)$$

Temporal Difference (TD) is a model-free RL approach. It learns by evaluating the state-value function using the estimated distribution of the current value. It is based on the state-value function (2) under policy π. In equation (6) $R_0 + \gamma v_\pi \hat{(}s_t)$ is an unbiased estimator for $v_\pi(s)$.

$$V(S_t) \leftarrow V(S_t) + \alpha[R_t - V(S_t)] \quad (6)$$

In [21], the authors present a straightforward derivation of a Least-squares temporal difference (TD) algorithm. It is a model-based reinforcement learning approach and implements a TD method for estimating the value functions, integrated with supervised linear regression. V_π can be represented and learned precisely in a small discrete state space X. The classical algorithm starts with the idea that the agent finds himself in a particular state and uses the transition probability so far, together with the reward function. The author creates a Markov chain model with the following sufficient statistics: a vector **n** that reports how many times each state has been visited; a matrix **T** with the counts of the transitional probabilities and a vector **s** that represents the sum of all rewards after leaving a particular state. The decision agent creates and solves a linear system of Bellman equations every time he expects a new estimate of the value function V_π. This model-based approach differs from the $TD(\lambda)$, a model-free approach to the same problem. $TD(\lambda)$ does not keep any statistics the rewards and the transitional probability, but it amends the values function until it converges to an optimal V_π.

Various RL procedures are employed to maximize the expected action-value function. We can categorize them into two classes: on-policy and off-policy methods, depending on whether there is a predefined initial policy when the agent starts the decision-making process.

1) Q-learning Off-policy Method: Q-Learning [2]is the most popular RL off-policy procedure employed in the ID domain. It is found on the idea for value iteration where the agent estimates the action-value function (1), to update all states s and actions a for every iteration. The goal is to optimize the Bellman equation by taking higher rewards R actions. Equation (7) denotes the Q-value where where α is a learning rate and a constant $0 < \alpha < 1$ and γ is again a discount factor $0 < \gamma < 1$.

$$Q'(S_t, A_t) \leftarrow (1-\alpha)Q'(S_t, A_t) + \alpha[R_t + \gamma \max Q(A'_t, S'_t)] \quad (7)$$

The motivation for employing Q-learning as one of the top RL methods applied to ID is that it is a model-free method. The researcher can use the rewards as a controlling tool. Last but not least, as we mentioned before, it is an off-policy approach because it learns the model without following a particular policy [3].

2) SARSA On-policy Method: State – action – reward – state – action (SARSA) is an on-policy RL method in which the decision agent receives feedback from the environment and amends the policy. In SARSA, the Q-values represent the received reward in the next step for taking action A at state S and the reward received from the next state and action [4].

The SARSA concept is similar to the Q-learning approach with some modifications. It can be defined as a quintuple $(S_t, A_t, R_t, S_{t+1}, A_{t+1})$, the method amends the Q-values based on the present state S, the present action A for S, the received reward R for action A, the new state S, and the next action A for the new state.

The earlier Q-value function (7) can be revised as in (8). Even though the equations (7) and (8) look nearly identical. In SARSA the agent takes the next available action, while the Q-learning agent selects the action with the highest estimation value among all following possible actions. Therefore in some

cases the Q-learning may be costly compared to SARSA.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_t + \gamma Q'(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \quad (8)$$

III. Taxonomy of the RL in IDS

The intrusion detection techniques in computer security have attracted many research interests in the literature [19]. The objective of the IDS is to detect attacks or divergences of the specific network properties in a particular system. The traditional methods utilize signatures of learned attacks and compare the evaluated traffic with the signatures. This approach is called a misuse detection method, and it is not suitable for finding new attack types with other signatures. In contrast, anomaly detection does not need prior knowledge of particular signatures. On the one hand, a benefit of anomaly detection is the ability to find new attacks; on the other hand, a drawback of the traditional anomaly detection methods is that they have to deal with a large amount of noise and uncertainty and therefore give false-positive rates. In anomaly detection methods, a primary component of the IDS is a feature(element) received from a sensor at a given time; this could be either a system call in host-based IDSs or a feature in a network-based IDSs. A complete sequence for anomaly detection is a time-series sequence of features classified as either normal or abnormal. The proposed taxonomy will classify the research work for anomaly detection based on the sensor approach as a Network-based IDS or system calls in host-based IDSs. The prevalence of the research work is concentrated in anomaly detection and more precisely network-based IDS. It helps organizations detect breaches in devices and applications. Since the network is an essential part of the IT environment, this is an area where organizations and researchers focus a considerable amount of effort on protecting the system.

Another classification we will present in this paper is based on the policy for which the action-value function is maximized. As we stated before, the policy is the proposed action that the agent takes, conditional on each state. The RL process could be classified as two types on-policy and off-policy learning methods.

On-Policy methods use a known general policy, and after an action is selected, the model revises the value functions. The policy may be modified after exploration and obtaining feedback from the environment to optimize the action-value function. The off-policy approaches find the policies that optimize the action-value function during the learning process by employing at first some unused hypothetical actions. Those methods can learn from the data, regardless of the policy.

Deep RL (DRL) and neural networks provide many research opportunities in the streaming data field. DRL is a modern approach for IDS, and we will consider it a separate category and part of the RL. Most papers could be classified as in the previous case based on the ID domain and the decision agent's policy. DRL solves some of the complex problems associated with the majority of the machine learning methods.

RL assumes a single decision agent, however sometimes it is possible to have multiagent RL (MARL). The deep reinforcement learning solves the problem of establishing an effective communication between the agents.

IV. Traditional RL methods

In this section we will classify the main research work, that employed the classical RL in IDS, based on the ID domain and the selected policy by the agent.

A. Network-Based Traditional IDS

The Network-Based Traditional RL are mainly off-policy methods, so more research in the on-policy domain is needed. In [5], Cannady suggests an adjusted reinforcement learning method for using adaptive neural networks to ID. The author investigates the detection of denial-of-service (DOS) attacks employing a cerebellar model articulation controller (CMAC) neural network (Albus, 1975). At the first stage, the method begins training the CMAC with simulated attacks, followed by a single test iteration, three hundred DoS attacks are implemented, and the least mean square learning approach is applied to update the weights of CMAC. The reported error was 3.24%. At the second stage, the adaptive learning capabilities of the model to identify new, other than DoS attacks, are tested with five types of UDP Packet Storm attacks. The CMAC is assessed based on the initial performance before and after receiving feedback from the environment. Before receiving feedback from the environment, the reported model error is 15.2%, and after that, it improves to .4%. Based on the provided results, the author claims that the CMAC model can obtain meaningful and rapid learning not only for known but also for unknown attacks and to perform online adaptive learning with significant accuracy.Sengupta, Sen, Sil, and Saha [8] propose a model that incorporates a Q-learning algorithm and rough set theory (RST) for IDS. The purpose of the algorithm is to achieve the highest classification accuracy by classifying the NSL-KDD data set as either "$normal$" or "$anomaly$". Since RST processes discrete data only, the data are discretized by applying cut operation attributes. Using the indiscernibility concept of RST, reduced attribute sets, called "reducts", are obtained, and among the reducts, a single reduct is chosen, which provides the highest classification accuracy. The same reduct does not consistently achieve the highest accuracy for the testing dataset. To overcome the problem, the authors employ discretization and feature selection processes. The Q-learning algorithm is adjusted so that the decision agent can learn the optimum cut value for different attributes to attain higher accuracy. Since only the selected attributes participate in the classification, the proposed algorithm lowers some of the complexity of the Q-learning and obtains 98% accuracy.The authors present two phases of the reward: the initial and final reward matrix. Each attribute is presented as a column and each cut as a row. Classification rules are derived for individual "reducts", and the corresponding accuracy is reported by using a rule-based classifier. The procedure is repeated until two successive states have the same or decreasing accuracy. The reward matrix creates a Q-matrix, where the start state coordinates to a particular cut, and the goal state is defined as

the state at which maximum accuracy is achieved. The authors in [10] employ a Pursuit Reinforcement Competitive Learning (PRCL) approach [11] and [12] for ID. This method applies the idea for immediate reinforcement learning [12], where feedback is received at each step after a decision is made. It uses online clustering, which can perform clustering in real-time with high accuracy in detecting intrusions. The approach is an off-policy method because the agent does not follow any specified policy in the action selection process. The proposed system consists of data pre-processing, the PRCL algorithm, and the performance evaluation phase. Three different methods related to competitive clustering RL are compared, and the accuracy is recorded. Mahardhik, Sudarsono, and Barakbah [13] applied RL to Detect Botnet using PRCL with further rule detection, which has reward and penalty rules to achieve a solution. PRCL can detect Botnet in real-time with high accuracy based on the empirical result. PRCL uses an unsupervised data set to cluster the Botnet and obtain accuracy; it can achieve clustering online.

B. Host-Based Traditional RL Methods

1) Off-policy Host-Based Traditional RL methods: Otoum and Kantarci [14] propose a Wireless Sensor Networks (WSN) detection algorithm adopting Q-learning method on a hybrid IDS. The authors simulated a twenty sensors WSN which communicate through Dynamic Source Routing Protocol for Hierarchical Representation Networks. The tested sensor nodes are clustered in four areas of 100mx100m. 20 sensors are selected since results are selected and the the Q-learning algorithm is applied. The results outperform other machine learning methods, based on accuracy, precision and false positive rates.

2) On-policy Host-Based Traditional RL methods: Xu and Luo [15] suggest a kernel-based RL method for sequential behavior modeling in host-based IDSs, using system call sequences. The authors consider a kernel-induced feature space and least-squares temporal-difference (kernel LS-TD) algorithm. The model could be presented as a sequential prediction case, which is resolved by employing reward signals. The authors empirically demonstrate superior performance metrics of their proposed model compared to the Hidden Markov Models(HMMs) and linear TD algorithms. In order to reduce the feature space, a kernel approach is applied, where a high-dimensional nonlinear feature mapping can be created by selecting a Mercer kernel function $k(x_1, x_2)$ in a reproducing kernel Hilbert space (RKHS). The Mercer Theorem [16], is employed to produce the RKHS, which allows computing the inner product of two feature vectors. The kernel-based LS-TD learning algorithm produces dimensions equal to the number of state transition samples. The kernel matrix K and α dimensions have to be decreased to save some computing time. Therefore the authors use an approximately linear dependence (ALD) method [17] to make the K matrix sparse. The authors compare their proposed method with the Hidden Markov Model; both use the Markov reward model, and the results seem to be promising in creating dynamic models and forecasting multistage attacks for IDS.The primary purpose of anomaly detection for the sequences is to solve the MDP with the dynamic behavior of data. Sukhanov, Kovalev, and Styskala [18] suggest an IDS model based on temporal-difference learning for MDP, called Temporal-Difference based Sequence Anomaly Detection 2 (TDSAD2). Their model is different from the classical Temporal Difference model by how the transitional probabilities are estimated. The classical methods require knowing the probability distribution for the transitional probabilities. However, the authors propose the idea that each probability of the transition from one state $s_i = x_t$ to the next state $s_j = x_{t+1}$ is influenced by the entrances of state $x_{t-n} = s_i, n \in N$ in the observed set. The approach has its definition of the reward function. It is calculated with regards to the anomaly sequences and also to normal sequences. The transition probability is updated by a matrix A, representing the pair transitions on dependency on previous states and the number of single states. The authors propose the primary adjustment related to the transitional probabilities estimation, therefore overcoming the weakness of previous TD approaches. They keep estimating the pair transition probabilities instead of estimating the occurrence number and introduce the dependence on the previous entrance of observed states. They also estimate each state to reduce computational power. The so proposed method can be successfully applied to IDS. In another paper [20], Xu proposes a similar anomaly detection approach for sequential data founded on TD learning, where a Markov reward function is presented. The author claims that TD in reinforcement learning can successfully detect abnormal behavior in the case of elaborated sequential processes by estimating the value of the Markov reward function. The advantage of the suggested model is that there is a straightforward labeling procedure utilizing delayed signals. The accuracy can be improved even with a limited training set, and it is superior to the accuracy obtained with Support Vector Machines(SVM) and HMMs. The performance metrics of the presented anomaly detection procedure using TD learning are estimated from a system call data of host-based ID from the MIT Lincoln Lab and the University of New Mexico (UNM). The author suggests a sequential anomaly detection approach for multistage attacks based on temporal-difference (TD) learning. A Markov reward model is created, and it is also demonstrated that the value function in the Markov reward model is analogous to the anomaly probability of the data sequences.

V. Deep RL methods

In the majority of the decision-making situations, the states of the MDP are high-dimensional. Therefore the classical RL methods are not applicable, and deep RL is preferable to solve the agent's problem. Deep reinforcement learning combines deep learning and RL to provide a solution for the MDP by representing the policy as a neural network [34].

A. Network-based Deep RL in IDS

1) Off-policy Network-Based RL Methods: Liu, Yin, and Hu [7] suggest a novel deep Convolution Neural Network

(CNN) Q-learning method to defend against Large-scale Low-Rate Denial-of-Service (LR-DDoS) attacks. In the case of multi-targets LR-DDoS attacks, the attack features and the normal ones are almost identical. The authors apply Deep Convolution Neural Network to rank features at different levels and combine them to produce an output. Q-learning is applied as a decision-making tool, which has to learn long-term characteristics of the whole network.The states are designed as four neurons $(Noutput_1; Noutput_2; Noutput_3; Noutput4)$. The actions are defined within one flow as follows: "rate limit with levels", "well-tune Syn-Receive timer", and "enlarge receive windows". The agent is learning and creating strategies based on the reward function. The authors define pre-state and post-state reward indicators and use them to obtain an overall return function so the agent to aim for higher speed and less package loss; they use an ϵ-greedy policy.The method is tested in a simulated environment to demonstrate efficiency in training, detecting, and preventing attacks. A weakness of the model is that the accuracy is low in sparse data. Bhosale, Mahajan, and Kulkarni [23] present a multiagent system, which is based on influence diagram [24]. Every agent knows the decisions and the actions of the other agents and makes decisions based on that knowledge. There are two probable states: if an intrusion occurs or no intrusion. Bayesian statistics is applied to find the transitional probabilities, and the prior distribution function is denoted by p. The actions of the IDS are alarm (A) or not (NA). The ROC parameters are the probability of an alarm given an intrusion $P(A|I) = H$ or no intrusion, and the probability of an alarm given no intrusion, P(A—NI) = F. The authors combine utility and Bayesian network theories based on an influence diagram. They represent directed acyclic graphs with three types of nodes. The decision nodes show the possible options available to the decision-maker. The chance nodes are random variables from the Bayesian networks. The value nodes represent the utility to be maximized. The reward matrix is shared between all agents in the system. The multiagent decisions are collaborative; they have the same reward or loss of selecting a particular action. Agents pursue achieving a Nash Equilibrium, and none of the agents has an incentive to deviate from it, given that the other players are also following a Nash equilibrium. Shamshirband, Patel, Anuar, Kiah, and Abraham [25] applied a game theory approach for ID and protection systems. The authors proposed a fuzzy Q-learning algorithm [26] and [27] for wireless sensor networks (WSNs) to find an optimal policy for each agent. The algorithm consists of two stages: detection and defense, and it is tested against DDoS attacks. Three agents participate in their model: a base station, sink nodes, and an attacker. The IDS detects future attacks based on a fuzzy Q-learning algorithm. The game begins when the attacker sends enormous traffic of flooding packets to a targeted node from the system. The authors suggest a low energy adaptive clustering hierarchy (LEACH) [28] model to evaluate the accuracy and the energy consumption of their proposed method. Caminero, Lopez-Martin and Carro [1] suggest a two-agents model that incorporates a simulated environment that creates network traffic samples, initiates rewards for each action, employs the classifier, and indicates the possible label for the network, whether there is a normal condition or there occurs any attack type. The rewards are either positive or negative, depending on the correct or incorrect classification of the agent. An essential element of the environment is that it randomly creates new samples from the dataset. The suggested model is Adversarial Environment Reinforcement Learning AE-RL, It uses an off-policy Q-Learning approach for solving the Bellman Equation for the action-value function. According to the authors, the model classifies quickly, incorporates a loss function, produces an adaptable classifier, and handles unbalanced data. The model is based on the idea that two agents have competing objectives: a classifier agent and an environmental agent. They work using the same methods, where the primary agent serves as a decision-maker who has a goal to maximize rewards and produce a classifier for the network. The secondary agent acts as an attack selector. The two agents work in an adversarial mode; they receive inverse reward functions (the gain to the primary agent is a loss for the environment agent and vice versa. Therefore, the environment agent will challenge the classifier agent to make more errors and consider the problematic samples. The Deep Reinforcement Learning (DRL) policy is fast and straightforward; it could be adapted to streaming data for immediate responses and is appropriate for changing environments. Analysis in [29], involved DRL methods such as DQN, double DQN (DDQN), policy gradient, and actor-critic models for network ID. Two data sets are tested to prove that the DDQN method is a superior algorithm. It is also proved that it performs better than some traditional machine learning methods in some cases. RL methods can help IDS respond effectively to environmental modifications. Nevertheless, the question about convergence to optimal policy is still in question in the multiagent system. Saeed, Selamat, Rohani, Krejcar,and Chaudhry, [30] studied the current multiagent IDS architectures that use RL. Sethi, Rupesh, Kumar, Bera, and Madhav [31] propose a context-adaptive IDS that utilizes numerous separate deep reinforcement learning agents distributed across the network for classification purposes. They use the popular data sets to prove the robustness of their model: NSL-KDD, UNSW-NB15, and AWID. Their model is organized so that there is a feature selection step. The network is presented as a two-tuple directed graph consisting of states and nodes, where R. denotes a router node. Their proposed IDS consists of the following DRL components: agent, state, action, and reward. The primary IDS obtains notifications from the agents, receives feedback regarding a potential intrusion, and communicates the information to the agents. The network includes multiple reinforcement learning agents, and it can adapt to the changes in the environment. Their suggested network structure produces high accuracy rates and low false-positive rates. The method is tuned in a way that could detect fine-grained attack types.Hsu and Matsuoka [32] suggest deep reinforcement learning for anomaly detection with two function modes: learning and detection. The detection mode carries a high processing speed, while the learning mode

maintains a high accuracy rate for classifying the upcoming network traffic. Their model can detect unknown network behavior patterns in real-time and has the option to self-update. The authors apply it to three different data sets: NSL-KDD, UNSW-NB15, and actual campus network traffic. They also compared their results with some traditional machine learning methods. The structure of the DRL algorithm is established using deep Q-Learning. The DRL agent evaluates the accuracy by monitoring the reward function. When the reward decreases, the agent revises the model with recent data to enhance the execution of the ID. There is a switch flag so that the method can switch between the two modes. Yang, Liang, Li, Wen, and Gao [35] propose a sample generation method of encrypted traffic. The system starts processing data, simulates encrypted malicious traffic, then applies a deep Q-network (DQN) combined with a deep convolution generative adversarial network (DCGAN). After the data is processed, a classification module is employed to classify encrypted traffic, where the fluctuation of the accuracy mode is taken under consideration. Kim, Yoon, and Lim [36] suggest a traffic sampling system for multiple traffic analyzers on a software-defined network (SDN). The decision agent in the presented model follows a deep deterministic policy. The problem is defined as a discrete MDP with continuous action spaces, and deep Q-learning is applied. These simulation results are tested in an empirical SDN environment. In another paper, [37] the authors propose network ID with deep auto-encoder in the Q-network, which detects network "anomalies ." Their off-policy model is with an experience replay and considers actions such as acceptance or denial, based on the "normal" or "anomalous" classification of the network. The Q-learning agent can learn how to predict anomalies during the training period.Suwannalai and Polprasert [38] propose an Adversarial Multi-Agent Reinforcement Learning using Deep Q-Learning. The trained model is established based on the NSL-KDD and tested with the KDDTest+ dataset. Their suggested model yields 80% accuracy and 79% macro F1 score.

2) On-policy Network-Based RL Methods: Malialis and Kudenko [22] propose a Multiagent Router Throttling (MRT) method based on SARSA against DDoS attacks. They also suggest an approach that incorporates three steps: task decomposition, hierarchical team-based communication, and team rewards. A hundred agents are implemented in their experiment, proving the successful potential in the IDS in an extensive internet provider network. Safa and Ejbali [36] apply a multi-agent adversarial reinforcement learning approach based on a deep SARSA algorithm for classification of the NSL –KDD data set. They study the performance and compare it with two classic machine learning methods. Their work creates a deep SARSA algorithm that integrates adversarial Reinforcement learning and supervised models. The model's primary purpose is to adapt to the detection of different attacks and give a high prediction performance with a reasonable runtime.

B. Host-Based Deep RL in IDS

The Host-Based Deep RL methods we collected are only off-policy, so research in the on-policy area is needed. Sengupta, Sen, Sil, and Saha [9] propose a model that incorporates a Q-learning algorithm and rough set theory (RST) for IDS. The purpose of the algorithm is to achieve the highest classification accuracy by classifying the NSL-KDD data set as either "$normal$" or "$anomaly$". Since RST processes discrete data only, the data are discretized by applying cut operation attributes. Using the indiscernibility concept of RST, reduced attribute sets, called reducts, are obtained, and among the reducts, a single reduct is chosen, which provides the highest classification accuracy. The same reduct does not consistently achieve the highest accuracy for the testing dataset. To overcome the problem, the authors employ discretization and feature selection processes. The Q-learning algorithm is adjusted so that the decision agent can learn the optimum cut value for different attributes to attain higher accuracy. Since only the selected attributes participate in the classification, the proposed algorithm lowers some of the complexity of the Q-learning and obtains 98% accuracy.The authors present two phases of the reward: the initial and final reward matrix. Each attribute is presented as a column and each cut as a row. Classification rules are derived for individual reducts, and the corresponding accuracy is reported by using a rule-based classifier. The procedure is repeated until two successive states have the same or decreasing accuracy. The reward matrix creates a Q-matrix, where the start state coordinates to a particular cut, and the goal state is defined as the state at which maximum accuracy is achieved.

VI. Conclusion

Intrusion Detection nowadays is an inevitable essential aspect of our daily life. The RL is a valuable, adaptable, and automatic mechanism that perfectly fits the ID goals and can assist the administrators in protecting the computer systems. The purpose of this survey is to outline the recent advances in RL applied to ID and to serve as an essential instrument and methodology searching tool for researchers in academia, industry, or governmental agencies. The proposed taxonomy could help individuals specializing in RL and experts in ID find an appropriate model for their needs or understand the recent advancements in the particular domain and also could outline the areas where more research work is needed.

References

[1] G. Caminero, M. Lopez-Martin and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," Computer Networks, vol. 159, pp. 96–109, May 2019 1955. http://dx.doi.org/10.1016/j.comnet.2019.05.013

[2] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-learning. Machine Learning 8, 3 (01 May 1992), 279–292,. http://dx.doi.org/10.1007/BF00992698

[3] Chris Gaskett, David Wettergreen, and Alexander Zelinsky, 1999. Q-learning in continuous state and action spaces.In Advanced Topics in Artificial Intelligence, Norman Foo (Ed.), Springer Berlin, Berlin, 417–428.

[4] D. Kumar, N. Logganathan, and V. P. Kafle, 2018, Double SARSA based machine learning to improve quality of video streaming over HTTP through wireless networks. In 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K).1–8., http://dx.doi.org/10.23919/ITU-WT.2018.8597682

[5] Cannady, J. "Applying CMAC-Based Online Learning to Intrusion Detection." Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, 2000, https://doi.org/10.1109/ijcnn.2000.861503

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] Liu, Zengguang, et al. "CPSS LR-DDoS Detection and Defense in Edge Computing Utilizing DCNN Q-Learning." IEEE Access, vol. 8, 2020, pp. 42120–42130., https://doi.org/10.1109/access.2020.2976706.

[8] Sengupta, Nandita, et al. "Designing of on Line Intrusion Detection System Using Rough Set Theory and Q-Learning Algorithm." Neurocomputing, vol. 111, 2013, pp. 161–168., https://doi.org/10.1016/j.neucom.2012.12.023.

[9] Mohanty, D., Sethi, K., Prasath, S., Rout, R. R., Bera, P. (2021). Intelligent intrusion detection system for smart grid applications. 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). doi:10.1109/cybersa52016.2021.9478200

[10] Tiyas, Indah Yulia, et al. "Reinforced Intrusion Detection Using Pursuit Reinforcement Competitive Learning." EMITTER International Journal of Engineering Technology, vol. 2, no. 1, 2014, https://doi.org/10.24003/emitter.v2i1.16.

[11] Arai, Kohei. "Pursuit Reinforcement Competitive Learning: PRCL Based Online Clustering with Learning Automata." International Journal of Advanced Research in Artificial Intelligence, vol. 5, no. 10, 2016, https://doi.org/10.14569/ijarai.2016.051006.

[12] Likas, Aristidis, "A Reinforcement Learning Approach to Online Clustering." Neural Computation, vol. 11, no. 8, 1999, pp. 1915–1932., https://doi.org/10.1162/089976699300016025.

[13] Mahardhika, et al., "Botnet Detection Using on-Line Clustering with Pursuit Reinforcement Competitive Learning ." EMITTER International Journal of Engineering Technology, vol. 6, no. 1, 2018, pp. 1–21., https://doi.org/10.24003/emitter.v6i1.207.

[14] Otoum, Safa, et al. "Empowering Reinforcement Learning on Big Sensed Data for Intrusion Detection." ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, https://doi.org/10.1109/icc.2019.8761575.

[15] Xu, Xin, and Yirong Luo. "A Kernel-Based Reinforcement Learning Approach to Dynamic Behavior Modeling of Intrusion Detection." Advances in Neural Networks – ISNN 2007, 2007, pp. 455–464., $https://doi.org/10.1007/978-3-540-72383-7_54$.

[16] Schölkopf, Bernhard, and Alexander J. Smola. "A Short Introduction to Learning with Kernels." Advanced Lectures on Machine Learning, 2003, pp. 41–64., $https://doi.org/10.1007/3-540-36434-x_2$.

[17] Engel, Y., et al. "The Kernel Recursive Least-Squares Algorithm." IEEE Transactions on Signal Processing, vol. 52, no. 8, 2004, pp. 2275–2285., https://doi.org/10.1109/tsp.2004.830985.

[18] Sukhanov, A. , Kovalev, S., Styskala, V., "Advanced Temporal-Difference Learning for Intrusion Detection." 13th IFAC and IEEE Conference on Programmable Devices and Embedded Systems: PDES 2015", Volume 48, Issue 4, 2015, pp. 43-48.

[19] Steinwart,I., Hush, D., Scovel, C., "A classification framework for anomaly detection", Journal of Machine Learning Research", Issue 6, 2005, pp. 211–232.

[20] Xu, Xin. "Sequential Anomaly Detection Based on Temporal-Difference Learning: Principles, Models and Case Studies." Applied Soft Computing, vol. 10, no. 3, 2010, pp. 859–867., https://doi.org/10.1016/j.asoc.2009.10.003.

[21] Boyan, Justin. "Technical Update: Least-Squares Temporal Difference Learning." Machine Learning, vol. 49, 2002, pp. 233–246.

[22] Malialis,K, and Kudenko, D.,"Distributed response to network intrusions using multi-agent reinforcement learning," Engineering Applications of Artificial Intelligence, vol. 41, pp. 270-284, 2015.

[23] Bhosale,R., Mahajan,S., and P. Kulkarni, "Cooperative machine learning for intrusion detection system," International Journal of Scientific and Engineering Research, vol. 5, no. 1, pp. 1780-1785, 2014.

[24] Detwarasiti,A. and Shachter, R. "Influence diagrams for team decision analysis," Decision Analysis, vol. 2, no. 4, pp. 207-228, 2005.

[25] Shamshirband,S.,Patel, A.,Anuar, N. B., Kiah, M. L. M. and Abraham, A., "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks," Engineering Applications of Artificial Intelligence, vol. 32, pp. 228-241, 2014.

[26] Mu˜noz,P., Barco, R. and de la Bandera, I. "Optimization of load balancing using fuzzy Q-learning for next generation wireless networks," Expert Systems with Applications, vol. 40, no. 4, pp. 984-994, 2013.

[27] Shamshirband,S., Anuar, N.B., Kiah, M. L. M. and A. Patel, "An appraisal and design of a multiagent system based cooperative wireless intrusion detection computational intelligence technique," Engineering Applications of

[28] Varshney, S., and Kuma, R., "Variants of LEACH routing protocol in WSN: A comparative analysis," 8th International Conference on Cloud Computing, Data Science and Engineering (Confluence),pp. 199-204., 2018;

[29] Lopez-Martin,M., Carro, B. and Sanchez-Esguevillas, A., "Application of deep reinforcement learning to intrusion detection for supervised problems," Expert Systems with Applications, vol. 141, 112963, 2020.

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

[31] Sethi, K., Sai Rupesh, E., Kumar, R., Bera, P.; Venu Madhav, Y. (2019). A context-aware robust intrusion detection system: A reinforcement learning-based approach. International Journal of Information Security, 19(6), 657-678. doi:10.1007/s10207-019-00482-7

[32] Hsu, Y., Matsuoka, M., A deep reinforcement learning approach for Anomaly Network Intrusion Detection System. 2020 IEEE 9th International Conference on Cloud Networking (CloudNet). doi:10.1109/cloudnet51028.2020.9335796

[33] Sutton, R. S., Bach, F., Barto, A. G. (2018). Reinforcement learning: An introduction. Massachusetts: MIT Press.

[34] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J. (2018). An introduction to deep reinforcement learning. doi:10.1561/9781680835397

[35] Yang, J., Liang, G., Li, B., Wen, G., Gao, T. (2021). A deep-learning- and reinforcement-learning-based system for encrypted network malicious traffic detection. Electronics Letters, 57(9), 363-365. doi:10.1049/ell2.12125

[36] Kim, S., Yoon, S., Lim, H. (2021). Deep reinforcement learning-based traffic sampling for multiple traffic analyzers on software-defined networks. IEEE Access, 9, 47815-47827. doi:10.1109/access.2021.3068459

[37] Kim, C., Park, J. (2019). Designing online network intrusion detection using deep auto-encoder Q-learning. Computers Electrical Engineering, 79, 106460. doi:10.1016/j.compeleceng.2019.106460

[38] Suwannalai, E.,Polprasert, C. (2020). Network intrusion detection systems using adversarial reinforcement learning with Deep Q-Network. 2020 18th International Conference on ICT and Knowledge Engineering (ICTKE). doi:10.1109/ictke50349.2020.9289884