Robust Enhancement of Intrusion Detection Systems Using Deep Reinforcement Learning and Stochastic Game

Hafsa Benaddi, Khalil Ibrahimi, *Senior Member, IEEE*, Abderrahim Benslimane, *Senior Member, IEEE*, Mohammed Jouhari, and Junaid Qadir, *Senior Member, IEEE*

***Abstract*— The incorporation of advanced networking technologies makes modern systems vulnerable to cyber-attacks that can result in a number of harmful outcomes. Due to the increase of security incidents and massive activities on networks, existing works have mainly focused on designing *Intrusion Detection Systems (IDSs)* based on traditional machine learning and deep learning models. In recent times, state of the art performance has been achieved in various fields through Deep Reinforcement Learning (DRL), which combines deep learning with reinforcement learning. In this paper, we propose a new DRL-based IDS for network traffics using *Markov decision process (MDP)* to improve the IDS decision-making performance. In addition, an extensive analysis of the IDS behavior is provided through modeling the interaction between the well-behaving IDS and attacker players using *Stochastic Game Theory*. Specifically, we used a non-zero-sum stochastic game, where the transitions between states depend on both the IDS and the attacker's actions at each stage of the game. We show that our game reaches a Nash Equilibrium upon convergence to seek the optimal solution, which corresponds to the optimal decision policy where both players maximize their profits. We compared the performance of our proposed DRL-IDS to the baseline benchmark of standard reinforcement learning (RL) and several machine learning algorithms using NSL-KDD dataset. As a result, our proposed DRL-IDS outperforms the existing models by improving both the detection rate and the accuracy while reducing false alarms. Results were provided to demonstrate the convergence of the game theory-based IDS under various settings toward equilibrium. This equilibrium corresponds to the safe state where both players are playing their respective best strategies.**

***Index Terms*—Security, Network traffic, Intrusion Detection System, Deep Reinforcement Learning, Stochastic Game, Q-learning, Performance Evaluation, NSL-KDD.**

I. Introduction

ACTUALLY, the overall worldwide IP traffic is expected to reach 400 exabytes per month while it was only about 120 exabytes per month in 2017 [1]. Thus, effective techniques and tools should be designed to deal with network traffic security such as the threats of critical information disclosure, unauthorized access as well as the disruption of transmitted data. Many worldwide organizations adopted the use of *Network Intrusion Detection Systems* (NIDS) as a key part of their information system. This is due to the power of NIDS on securing the information communicated over the target system through processing the network traffic and classifying the normal and abnormal traffics that can be seen as cyber-attacks. *Intrusion Detection System* (IDS) is an important process in network security that aims to detect and monitor the network from abnormal activities and the threat of intrusions in the network traffic. This is performed by identifying normal and abnormal network activities. In general, we can classify the IDS systems into two types. The first one is called *misuse-based* IDS (also called *signature-based IDS*), where its role consists of detecting the intrusion by observing activities that are similar to those used in known attacks. The ability of the first method to predict new and unknown attacks is limited due to its dependency on a database of general attack activities. The second type (*anomaly detection based IDS*) works by creating a profile of normal network behavior and then by identifying any strange behavior that substantially differs from the pattern of ordinary traffic. With such a strategy, the IDS can identify previously unseen attacks. Various works in the literature have adopted standard *Deep Learning* (DL) techniques such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Deep Neural Networks (DNNs). Such classification schemes enhance the performance of the IDS by ensuring high detection precision of normal and abnormal activities [2], [3], [4]. However, several research works have recommended the use of Deep Learning (DL) for IDS to enable the classification of the high dimensional complex features by classifying the critical examples and giving the optimal activation function to get the best performance of the model [5]–[7], [8]. The advantage of DL arises in complex large-scale settings where the number of states is enormous. The DL models can be used in such settings to efficiently estimate the action values by offering the ability to perform classification of

Manuscript received 31 January 2022; revised 25 April 2022; accepted 12 June 2022. Date of publication 28 June 2022; date of current version 17 October 2022. The review of this article was coordinated by Dr. Zubair Fadlullah. *(Corresponding author: Abderrahim Benslimane.)*

Hafsa Benaddi and Khalil Ibrahimi are with the Ibn Tofail University, Faculty of Sciences, LaRI Laboratory, 6CW7+CV Kenitra, Morocco (e-mail: hafsa.benaddi@uit.ac.ma; ibrahimi.khalil@uit.ac.ma).

Abderrahim Benslimane is with the University of Avignon, CERI/LIA, 84029 Avignon, France (e-mail: abderrahim.benslimane@univ-avignon.fr).

Mohammed Jouhari is with the School of Computer Science, Mohammed VI Polytechnic University, Ben Guerir 43150, Morocco (e-mail: mohammed.jouhari@uit.ac.ma).

Junaid Qadir is with the Department of Computer Science, Engineering, College of Engineering, Qatar University, 2713 Doha, Qatar (e-mail: junaid.qadir@itu.edu.pk).

Digital Object Identifier 10.1109/TVT.2022.3186834

an incident, object monitoring, image captioning, and semantic segmentation in real-time. However, including the concept of Q-Learning of *Reinforcement Learning* (RL) on the IDS has been comprehensively explored for auditing and monitoring the sensor networks. This is used for building a fast decision mechanism where the IDS learns the optimal policy of actions over the set states depending on the considered environment. The interaction between control systems and cyber threats is important in the network traffics [9]–[12]. To date, several approaches have been proposed to keep the networks safe and credible [13]. Especially, game-theoretic approaches [9], [14], [15] have been considered to solve security problems such as jamming attacks and replay attack. In this regard, a non-cooperative evolutionary game is proposed by Shi *et al.* [16] for enhancing the security levels of the systems while considering the honeypot configuration dynamically and iteratively against malicious activities. Likewise, the concept of modeling the interaction between the hackers and the technology as a game is introduced by Mishra and Smirnova in [17] where the game results in providing an efficient configuration of the IDS that ensure either a high detection rate and low false alarm rate. We note that this paper is a substantial extension of our earlier work [18], which focused on designing a Deep Reinforcement Learning-based (DRL)-IDS. In this paper, we extend our previous work by improving the IDS that provides more analysis of our proposed DRL-IDS and additionally model the interaction between the IDS and attacker using a non zero-sum-game with incomplete information where both players can dynamically change behavior based on their strategies and their predicted rewards. Moreover, we obtained an optimal strategy to achieve an ϵ-Nash equilibrium game using a value iteration concept. The main contributions of this paper are summarized next:

1) A hybrid RL and DL based IDS scheme (DRL-IDS) is proposed to improve the performance of the IDS on detecting cyber threats for network traffics in real-time and identifying new and existing malware and threats—e.g., Man-In-the Middle attack, and Denial-of-Service (DoS)—with high precision and gaining maximum rewards. DRL-IDS can identify attacks according to their risk level with high accuracy and precision while ensuring a low false alarm rate. The Deep Q-Network is used to improve the Q-function estimation given by RL incorporated with the IDS. The RL algorithm defines the error target of the estimated Q-function to aid Deep Q-Network decision on estimated state and action.

2) A non zero-sum stochastic game is developed to reach the equilibrium point among players, where each participant tried to take action for achieving an objective of maximizing their profits in the predictability of the subsequent actions. The game theory as proved in the related work, it becomes a new respective of many systems, because, we didn't need the global information than the classical approach (MDP) to react in real time and find the optimal decisions as shown in our numerical results.

3) We provide two distinct analyses. First, the non game-theoretic IDS using NSL-KDD as the incoming traffic data to monitor the real-time network streams in which the performance evaluation of the proposed model-IDS is given with several tests on the proposed environments. Second, we discuss the results obtained from our extensive simulation to study the interaction between players based on our proposed stochastic game in which a safe state of the system is achieved by the game.

The rest of the paper is organized as follows: In Section II, we investigate the related work that has focused on developing approaches-based IDS. In Section III, we describe the formulation of DRL-IDS, a detailed description of NSL-KDD dataset and we model the interaction between two-player a using non-zero-sum stochastic game. In Section IV, we provide the performance evaluation of our proposed IDS models based on the DRL and the stochastic game. Finally, Section V concludes this article.

II. Related Work

Researchers seeking a much more elaborate discussion on state-of-the-art intrusion detection systems and open research problems rely on various detailed works published in the literature. Therefore, in this section, we summarize the most recent related work in the field. Several traditional Deep Learning (DL) approaches have been widely applied to ensure efficient and accurate IDS. Yin *et al.* [19] suggested a DNN for intrusion detection in the network. They used RNN to find the intrusion. The model is made up of two-stage forward propagation and backward propagation. Their suggested approach was designed for binary and multiclass classification and improved jointly the accuracy and the capability to identify the kind of known attacks. Yang *et al.* [20] proposed a hybrid intrusion detection model using a Modified Density Peak Clustering Algorithm (MDPCA) and Deep Belief Networks (DBNs), which uses MDPCA for feature extraction and DBNS for classification. They assessed the model on NSL-KDD and UNSW-NB15 datasets. The experimental results showed that the combined MDPCA with the DBNs model achieved better accuracy, detection rate, and false-positive rate. However, the model does not have a strong modeling ability to detect U2R and R2L attacks. Ahsan *et al.* [21] proposed a hybrid intrusion detection network using CNN and Long Short-Term Memory (LSTM). CNN is used for extracting the feature and LSTM for merging of features. The model showed the highest accuracy on the NSL KDD dataset. Wisanwanichthan *et al.* [22] suggested a Double-Layered Hybrid Approach (DLHA) in network intrusion detection, was employed to detect anomalies and unknown attacks. The DLHA model deployed the first layer to detect DoS and Probe using the Naive Bayes classifier. The second layer to distinguish R2L and U2R from normal instances using SVM confirmed their high detection rate of 93.11% with over 96.67% detection rate of R2L, and 100% of U2R for the NSL-KDD dataset in comparison with several existing IDS techniques. Liu *et al.* [23] proposed a hybrid intrusion detection K-Means, Random Forest (RF), and DL, where k-Means and RF were used for binary classification, and DL approaches such as LSTM and CNN for multiclass classification. The model classification was evaluated through the NSL-KDD and CIC-IDS2017 datasets. The accuracy of 5 types of data in NSL-KDD reached 85.24%. Min *et al.* [24] proposed a network intrusion detection method using a memory-augmented deep auto-encoder (MemAE). The

MemAE model consists of an encoder, a decoder, and a memory module. They reported experimental results on several datasets, such as the UNSW-NB15 and the NSL-KDD datasets. Kunang *et al.* [25] proposed a DL IDS using a pretraining approach with deep autoencoder (PTDAE) combined with DNN. They used DAE as feature extraction and fine-tuning phase using DNN architecture. Their technique provides the best attack detection rate performance. Reinforcement learning (RL) illuminates an ample space of learning problems characteristic of autonomous agents performing various interactions in an environment, such as sequential decision-making problems with gaining rewards. Lately, a few studies have likewise incorporated the utilization of RL for IDS improvement. Sengupta *et al.* [26] proposed a method by altering the Q-learning calculation to learn ideal cut incentive for an alternate property of network traffic. Their model can get high accuracy of 98% and has a quicker handling speed for real-time prediction. Moreover, other few works apply RL-IDS for dispersed network frameworks. Caminero *et al.* [27] incorporated the supervised learning process with a modified RL algorithm based on interaction with a real environment behavior based on an adversarial strategy. They replaced the environment by using AWID and NSL-KDD datasets. Their model aims to optimize the agent's policy function and not define the reward function. Their suggested approach was achieved accuracy 80.16% and F1-score 79.40%. Lopez *et al.* [28] conducted an application of RL for detecting intrusions in the network. Through its trial after effects of utilizing AWID and NSL-KDD datasets. The best outcomes are acquired for the Double Deep Q-Network (DDQN) with an accuracy of 95.70 % and F1 93.94% measurements. Similarly, Suwannalai *et al.* [29] proposed the Adversarial/Multi-Agent Reinforcement Learning utilizing Deep Q-network (AE-DQN) for network IDS problems. Likewise, Sethi *et al.* [30] proposed an RL-based IDS that utilizes Deep Q-Network in numerous disseminated agents and employs an attention process to characterize the network behaviors. Alavizadeh *et al.* [31] collaborated Q-learning-based reinforcement learning with a deep feed-forward neural network approach to avoid malicious behaviors in the network traffic. Moreover, game-theoretical approaches based on traffic anomaly detection are mainly used to study decision drawbacks such as the interaction between decision-makers. It is also used to resolve the issue of the resolution of the equilibrium [32]. Sangeetha *et al.* [33] designed an intrusion detection system for mobile ad hoc networks based on the Bayesian Hybrid Detection model that reduces malicious nodes as a result of an improved reliability detection. Xia *et al.* [34] proposed collaborative intrusion detection schemes based on sharing strategies in smart grids, where two-layer collaborative IDSs were designed. The first layer is used to model the interaction between the IDS and the attackers. The second layer allows efficient resource allocation under harsh resource restriction conditions. Their proposed method uses a stochastic game that quickly reaches the Nash Equilibrium where the IDS maximizes its payoff and results in optimal detection strategies. Motivated by the existing model suggested previously to enhance the IDS performance, we propose a robust IDS model that relies on classifying the attacks based on their risk and impact on the system which helped in the training step. Deep reinforcement learning was deployed instead of the traditional machine learning models used for IDS due to its power on providing an accurate prediction of attacks from observations (dataset rows) and its ability to handle high-dimensional states space. Furthermore, we used the stochastic game to model the interaction between players to distinguish between the malicious and honest players such as the players are the traffic generator and the IDS as motivated by the above related work. Therefore, we considered the classification of attack to reach a Nash equilibrium where the system is safe under the monitoring of our robust IDS. To the best of our knowledge, this is the first time to classify the attack in order to obtain an optimized solution of the game where the IDS maximizes its reward.

III. Proposed Model

A. Problem Formulation

We start by defining the concept of RL, another extension of ML-based on a *Markov decision process* (MDP). The main idea concerning how to take action a to exploit reward R environment by giving a state s to the attacker and IDS. We define two quintuples: Detection (S_d, A_d, P_d, R_d, γ) and Attack (S_a, B_a, P_a, R_a, $\tilde{\gamma}$).

Space of system states: The set of states captured by the IDS is denoted as $S_d = \{v_0 = normal, v_1 = detection, v_2 = noDetection\}$ respectively. Where v_0 means the normal traffic record on the network traffic, v_1 which means detection of attacks by IDS in the traffic, and v_2 indicates that the IDS cannot detect the attacks because these attacks do not exist in the list of known attacks or new attacks that have not yet exploited. Furthermore, The set of states captured by the attacker is represented as $S_a = \{u_0 = Unattack, u_1 = Attack\}$ respectively, where u_0 means the attacker find the system safe, u_1 which means the attacker successfully attacked the system.

Space of actions: Set of possible actions that can be taken by the IDS, it is expressed: $A_d = \{a_0, a_1, a_2, a_3, \ldots, a_m\}$, where a_k indicates the type of reaction of the IDS in the k^{th} attacks class and $k = 0, 1, 2, 3, \ldots m$. Moreover, the actions are ordered ([35]) by the risk level of the estimated attacks with $a_0 < a_1 < a_2 < a_3 < \ldots < a_m$. The action set of the IDS is ordered starting from the action with the lowest cost to the action with the highest cost. Knowing that the IDS can take either software or hardware defense strategy to face the attacker action. The set of possible actions that can be taken by the attacker are given by: $B_a = \{b_0, b_1, b_2, b_3, \ldots, b_n\}$, where b_l indicate the type of reaction of the attacker in the f^{th} level of protection and $f = 0, 1, 2, 3, \ldots n$. Likewise, the actions are ordered by the type of these attacks with $b_0 < b_1 < b_2 < b_3 < \ldots < b_n$.

Transition of states probability: Matrix of transition probabilities observing at time t for $a \in A_d$ is define as:

$$\mathbf{P_d} = \begin{pmatrix} \beta^a_{1,1} & \beta^a_{1,2} & \beta^a_{1,3} \\ \beta^a_{2,1} & \beta^a_{2,2} & \beta^a_{2,3} \\ \beta^a_{3,1} & \beta^a_{3,2} & \beta^a_{3,3}, \end{pmatrix}$$

where $\beta^a_{i,j}$ is the probability to change the state of the IDS depending on the impact of actions. Given by $\beta^a_{i,j} =$

$p(v_{t+1}/v_t) = p(v_j|v_i, a)$ for $i, j = 1, 2, 3$ and: $\sum_{j=1}^{3} \beta^a_{i,j} = 1\ ,\ i = 1, 2, 3\, and\ a \in A$. The matrix of transition probabilities observing at time t for $b \in B_a$ is expressed as:

$$\mathbf{P_a} = \begin{pmatrix} \psi^b_{1,1} & \psi^b_{1,2} \\ \psi^b_{2,1} & \psi^b_{2,2}, \end{pmatrix}$$

where $\psi^b_{i,j}$ the probability to change the state of the attacker according to the impact of actions. Denoted by: $\psi^b_{i,j} = q(u_{t+1}/s_t) = q(u_j|u_i, b)$ for $i, j = 1, 2$ and: $\sum_{j=1}^{2} \psi^b_{i,j} = 1\ ,\ i = 1, 2\, and\ b \in B_a$.

Reward functions: We aim to optimize the system's objective function, which allows us to represent the IDS returns and to feat an action immediately with the location of reward received in the state s and the action a. This reward takes two forms, a negative value R_n when the IDS takes the best action to protect the system even if the investment against the action is too expensive and positive value R_p when the IDS decides the right action. We consider the value of the reward of the IDS as:

$$r^d_t(v_t, a_t) = \begin{cases} R_p & \text{For } v_t = v_0 \text{ and } a_t = a_0; \\ (1 - \alpha_j(a_t))R_p & \text{For } v_t = v_0 \text{ and } a_t \in \{a_1, .., a_m\}; \\ R_p & \text{For } v_t = v_1 \text{ and } a_t = a_k; \\ (1 - \lambda_j(a_t))R_p & \text{For } v_t = v_1 \text{ and } a_t \in \{a_0, .., a_{k-1}\}; \\ R_n & \text{For } v_t = v_1 \text{ and } a_t \in \{a_{k+1}, .., a_m\}; \\ R_p & \text{For } v_t = v_2 \text{ and } a_t = a_m; \\ (1 - \theta_j(a_t))R_p & \text{For } v_t = v_2 \text{ and } a_t \neq a_0, \end{cases} \tag{1}$$

where $0 < \alpha_j(a_t) < 1, 0 < \lambda_j(a_t) < 1$ and $0 < \theta_j(a_t) < 1$ described the penalty coefficients. Let $r^a_t(u_t, b_t)$ be the value of the reward of the attacker which denoted as:

$$r^a_t(u_t, b_t) = \begin{cases} \epsilon R_p & \text{For } u_t = u_0 \text{ and } b_t = b_0; \\ (1 - \phi_j(b_t))R_p & \text{For } u_t = u_0 \text{ and } b_t \in \{b_1, .., b_l\}; \\ \epsilon R_p & \text{For } u_t = u_1 \text{ and } b_t = b_h; \\ (1 - \rho_j(b_t))R_p & \text{For } u_t = u_1 \text{ and } b_t \in \{b_0, .., b_{h-1}\}; \\ R_p & \text{For } u_t = u_1 \text{ and } b_t \in \{b_{h+1}, .., b_l\}, \end{cases} \tag{2}$$

where $0 < \phi_j(b_t) < 1$ and $0 < \rho_j(b_t) < 1$ denoted the penalty coefficients. The action in each state can be either a reward if detected or a cost if undetected. So these equations (1),(2) represent the gains obtained R_p (which means reward positive) in each state this is when the attack is detected and the costs R_n(reward negative) when the attack is undetected. We can consider the average reward of the IDS and the attacker respectively in each time step t as expressed by the following expressions: $R_d(s_t = v, a_t = a) = \sum_{v_{t+1} \in S_d} p(v_{t+1}|v_t, a) r^d_t(v_{t+1}, a)$. $R_a(s_t = u, b_t = b) = \sum_{u_{t+1} \in S_a} q(u_{t+1}|u_t, b) r^a_t(u_{t+1}, b)$.

Discount factors: $\gamma, \tilde{\gamma} \in [0, 1]$ be discount factor for the IDS and the attacker respectively.

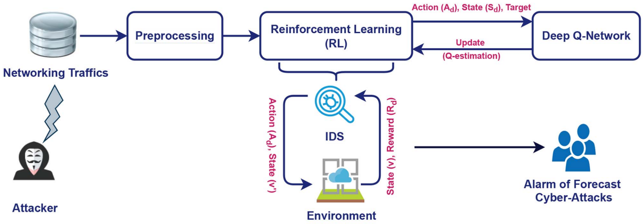

Networking Traffics
Preprocessing
Reinforcement Learning (RL)
Action (A_d), State (S_d), Target
Update (Q-estimation)
Deep Q-Network
Attacker
IDS
Action (A_d), State (v')
State (v), Reward (R_d)
Environment
Alarm of Forecast Cyber-Attacks

Fig. 1. Improving the protection of network data traffics using DRL-IDS.

TABLE I
CLASSES DISTRIBUTION OF NSL-KDD DATASET

Attack Type	Training Set	Testing Set
Normal	67,343	9710
Denial of Service (DoS)	45,927	7458
Probe	11,656	2422
Remote to Local (R2L)	995	2887
User to Root (U2R)	52	67
Total	**125,973**	**22,544**

B. Deep Reinforcement Learning-Based IDS (DRL-IDS) Model

Several types of attacks may occur during the deployment of network traffic, data exchanged among these networks are analyzed by a DRL-based IDS [18] which is needed to provide the best decision in order to secure the network as shown in Fig. 1. In following we will explain the different steps to build this proposal:

Dataset Description: Most state-of-the-art IDS approaches utilized the NSL-KDD dataset as a frequently available dataset on NIDS research [36], [37] for the performance evaluation. This dataset represents a reduced version of the KDD Cup 99 dataset, proposed in 2009 by experts in the field of network intrusion detection to solve some issues appearing in the KDD'99 database [38]. The CSV format of the latter has a reasonable number of training records 125973 and testing records 22544 and possesses TCP/IP connection records (each record consists of 41 attributes characterizing the connection (each attribute can be normal or abnormal records). Table I shows the distribution of training and testing sets for the behaviors of network traffic. The training dataset comprises 23 traffic classes that incorporate 22 abnormal classes and one normal class. The test dataset incorporates 38 abnormal classes, out of which 16 are novel abnormal classes and one normal classThe dataset includes four attack categories which are further divided into 39 subcategories. A full explanation of the testbed configuration and attacks is available in the referenced article [39]. We note that the NSL-KDD dataset is accessible for download on the website [40].

Preprocessing: During this step, the recorded dataset collected from the network traffic is registered with a different type that can represent a challenge for training the model. However, this dataset consists of redundant and invalid records that can be removed in the input [41]. Typically, the preprocessing of the NSL-KDD dataset, including the defects as mentioned earlier, is outlined in detail as follow:

Data labeling: We categorized the NSL-KDD dataset into 4 types of attacks, including DoS, Probe, R2L, U2R, and one normal data type. Then we mapped all the attribute values of these data types to clear the data and train our agent.

One-hot encoding: The features are most likely to be numeric or boolean, not strings of characters. Therefore, the categorical features are represented by a one-hot encoding, which requires all symbolic data to be transformed into a real vector that enhances the set of features.

Normalisation: Numerical data were mapped to the interval $[0, 1]$ with a min-max normalization without warping the differences in the value ranges. This step is useful only when the features differ. This step has the benefit of speeding up some deep learning applications. Following these preprocessing steps, the NSL-KDD dataset was finally transformed into 42 features, which helped us improve the outcome of the simulation, especially the training process.

Classification of Attacks: Traditional IDSs proposed in the literature do not provide sufficient security to detect new types of attacks because they are not adaptive and scalable. With the appearance of new kinds of attacks and the massive rise in network traffic, traditional IDS offer limited protection in terms of security and privacy. Critical infrastructure faces both classical and unknown cyber-intrusions, where the classical attacks work to collect, retrieve, and maintain access to the target data by using various techniques; the most popular attacks exploit software vulnerabilities and flaws. On the other hand, unknown attacks can be defined as newly discovered malicious. However, we classified the risk of attacks based on their appearance on the dataset, such as the most frequent attack is considered as the critical attack (DoS: critical, Prob: high, R2L: medium, U2R: low). Note that in the NSL-KDD dataset there is no feature indicating the risk level of the attacks.

Reinforcement Learning based IDS: In our case we modeled the IDS in such way to select randomly an a_t, then the environment take samples of reward $r_t^d(v_t, a_t)$ depending on the state of arrival v_{t+1}; consequently, the IDS obtains its reward on the next state v_{t+1}. Furthermore, π is a given policy from v_t to v_{t+1} that specifies the action a_t that will be taken in each state v_t. Then, the strategy will be updated in each modification of the observation of the IDS in the environment, it produces samples path $(v_0, a_0, r_0^d), (v_1, a_1, r_1^d), \ldots$. We define $\pi = (\pi_1, \pi_2, \ldots)$ as the vector of the optimal policies, the target of each data stream is to obtain π_t that represents the best policy. The state of the system v is a sufficient input to select the best action; thus, the maximum expected sum of the IDS rewards at t is given by: $\pi^* = \text{argmax}_{a \in A}\{r_t^d(v_t, a_t) + \sum_{v' \in S} P_t(v'|v, a)V_{I-1-t}^D(v')\}$. To choose the best state, we define the optimal value function V_{i+1}^D of our IDS in-state v as the expected cumulative reward from the policy π^* which can be calculated in each step i by: $V_{i+1}^D(v) = \text{argmax}_{a \in A}\{r_{I-1-i}^d(v_t, a_t) + \sum_{v' \in S} P_{I-1-i}(v'|v, a)V_i^D(v')\}$. Using the concept of Q-Learning, we will determine a time step size to define the action a_t performed at the next time step. The Q-learning algorithm performs the updates according to the optimal policy π^* corresponding to the best action a in each state v, even if it is not these optimal actions that the IDS realizes, with the learning rate $0 < \alpha < 1$. This latter defines how quickly or slowly our agent learns to take accurate action during the learning process and influences its learning behavior. $Q(v_t, a_t) = Q(v_t, a_t) + \alpha[r_t^d + \gamma max_{a_t \in A}\{Q(v_{t+1}, a_t) - Q(v_t, a_t)\}]$. The IDS predicts the state value function V_{i+1}^D to update the pair (v, a) in each iteration to determine which step has the best reward. Then, Q-Learning constructs a Q-table whose lines represent the states v, and the columns represent the actions a. In each v_t, the IDS realizes an action a_t, observes the reward r_t^d of this action as well as the next state as (v_{t+1}), and updates the estimated value of $\hat{Q}$ following Bellman equation: $\hat{Q}(v_{t+1}, a_{t+1}) = (1 - \alpha)Q(v_t, a_t) + \alpha[r_t^d + \gamma max_{a' \in A}\{Q(v', a')\}]$.

Deep Q-Network based IDS: We incorporated the *Deep Q-Network* (DQN) to our model using the Q-function that we estimated previously so that we can find the best decision of attack prediction and also improve the effectiveness by estimating the action values over the set of states by the non-linear function $Q(v_t, a_t; \theta) \approx \hat{Q}(v_{t+1}, a_{t+1})$. The parameter θ refers to the neuron's weights, which is updated in each iteration step i to train the Q-Network. We present below the improvement provided by this implementation:

1. Use a feedforward pass for the current state s to get predict Q-values for all actions;

2. Apply the experiences replay like an historical of the interaction process of the IDS in over-time t as $f_t = (v_t, a_t, r_t^d, v_{t+1})$ into the replay dataset $H_t = \{h(1), h(2), \ldots, h(t)\}$ which can help the network to learn the various transitions of the ancient experiences.

3. Update Deep Q-Network on the records from the training data (v, a, r^d, v') around the target Q-value by optimizing the loss function at each iteration i denoted as follows: $L_i(\theta_i) = E[(z_i - Q(v, a; \theta_i))^2]$, which $z_i = r_t^d + \gamma \text{argmax}_{a'} Q(v', a'; \theta_{i-1})$. Where θ_{i-1} network parameters of the previous network.

4. Update the weights using back-propagation using the gradient of the loss function with respect to the parameters θ as shown above:

$$\nabla_{\theta_i} L_i(\theta_i) = E[(z_i - Q(v, a; \theta_i))\nabla_{\theta_i} Q(v, a; \theta_i)]. \quad (3)$$

The system performance metrics are based on the Algorithm 1.

C. Stochastic Game With Incomplete Information Model

A stochastic game process can be considered as several static non-cooperative games that are repeated over time. Each of them is called a "state," making stochastic "transitions" between the game states. In this type of game, the participants can switch their actions strategies based on the other players' past actions and transitions state. We can use this game efficiently to evaluate and analyze the IDS reaction to monitor the system from malicious activities regarding the transitions of the attackers. Formally, we define the stochastic game as a segment of tuples $G =< N, S, M, U, \mathbb{P}, \gamma, \tilde{\gamma} >$, each parameter described as follows: $N = \{N_d, N_a\}$ is the set of players, N_d and N_a represent the IDS and the attacker participants. S is the set of the game states. Let S_d describes the IDS behavior from its constrained

Algorithm 1: Pseudo Code of the Non-Game-Theroretic-Based DRL-IDS.

```
Data: Collected dataset $Y$
Input: Initialize State, Action, environment, parameters $\theta$
Initialize the target Q-network
Initialize Replay memory H space
Output: return vector $Q(v_t, a_t; \theta)$
while $|\hat{Q}_{i+1} - \hat{Q}_i| < \sigma$ do
    for steps← $1$ to $M$ do
        for features $X \leftarrow 1$ to $N$ do
            $v = v_0$ (starting of State $v$)
            for $t$← $0$ to $T - 1$ do
                - Select a random action $a_t$ with the random probability $p$ based on $\epsilon$-strategy as:
                $a_t = \mathrm{argmax}_a\, Q(v, a_k; \theta)$
                - Apply $a_t$ and the IDS observe the reward $r_t^d$ and the next state
                - Observe chosen reward $r_t$ and Store the tuple $(v_t, a_t, r_t^d, v_{t+1})$ in $H$
                - Choose a Mini Batch arbitrary with this selected features $(v_l, a_l, r_l, v_{l+1})$ from $H$
                if $v_{l+1}$ is a terminal State then
                    $\mu_l = r_l$
                end
                else
                    $\mu_l = r_l + \delta \mathrm{argmax}_{a'}\, Q(v', a'; \theta)$
                end
                - Calculate the gradient of the loss function based on (3).
            end
        end
    end
end
```

actions set A_d, and the attacker chooses a type of attack independently from its actions set B_a at time t. Where $S = S_d \times S_a$ the aggregated states space where the actions taken by each participant result in the next state transition probability $s^{(t+1)}$ such as $s \in S$. $M = \{A_d, B_a\}$ is the set of actions such as $A_d = \{a_0, \ldots, a_{k-1}\}$ is a set of IDS actions where $k = 4$, such as a_0 means low level of monitoring, a_1 means medium level of monitoring, a_2 high level of monitoring and a_3 critical level of monitoring. $B_a = \{b_0, \ldots, b_{f-1}\}$ is a set of attacker actions space where $f = 3$, such as b_0 stands for not attacks, b_1 a passive attacks, and b_2 an active attacks. $U = \{U_d, U_a\}$ is the set of payoff functions matrices corresponding to the strategy space G. Where U_d depict the payoffs of the IDS. U_a represents the expected payoffs of the attacker corresponding to their strategy actions spaces A_d and B_a, respectively. $\mathbb{P}$ is the state transition probability set at t, where the transitions between the IDS and attacker is independent of each other. Therefore, each player controls its state separately $\mathbb{P}$ from $S \times M$.

Utilities Function: The utilities of the IDS and the attacker at time t depend on the state of the game and the actions taken by each player. Let $U_d^{(t)}$ the utility function of the IDS at stage t. The goal of the IDS is to maximise the model's effectiveness by minimizing the damages of malicious activities. However, the utility can be described $U_d^{(t)}$ as follow: $U_d^{(t)}(a^{(t)}, b^{(t)}, s^{(t)}) = F(a^{(t)}|b^{(t)}, s^{(t)}) \times R_d(v^{(t)}, a^{(t)}) - D(b^{(t)}|a^{(t)}, s^{(t)}) \times C_d$. Let $U_a^{(t)}$ be the utility of the attacker at time t.
$U_a^{(t)}(a^{(t)}, b^{(t)}, s^{(t)}) = D(b^{(t)}|a^{(t)}, s^{(t)})) \times R_a(u^{(t)}, b^{(t)}) - F(a^{(t)}|b^{(t)}, s^{(t)}) \times C_a$.

The behaviors of the IDS depends on the value of $F(a|b, s)$ that describes the success prediction of the IDS when the attacker varies the type of attack, and $D(b|a, s))$ is the damage incurred by the attacker:

$$F(a|b_2, s) = \begin{cases} 1 & \text{if}\, a = a_3; \\ 1 - \epsilon & \text{if}\, a = a_2; \\ 1 - \frac{1}{2} \times \epsilon & \text{if}\, a = a_1; \\ 1 - \frac{1}{\epsilon} & \text{if}\, a = a_0. \end{cases}$$

With ϵ depicted the sensibility of the IDS and $0 < \epsilon < 1$

$$F(a|b_1, s) = \begin{cases} 0 & \text{if}\, a = a_3; \\ 1 & \text{if}\, a = a_2; \\ 1 - \epsilon & \text{if}\, a = a_1; \\ 1 - \frac{1}{2} \times \epsilon & \text{if}\, a = a_0. \end{cases}$$

When the attacker is not doing the attacks, we will have the lack of effectiveness of the IDS, such as $F(a|b_0, s) = 0$.

However, the attacker applied a damage $D(b|a, s))$ if the IDS behave following $F(a|b, s)$, so, the damage can be calculated as follow:

$$D(b|a, s)) = \begin{cases} 0 & \text{if}\, b = b_0; \\ 1 - \xi & \text{if}\, b = b_1; \\ 1 & \text{if}\, b = b_2. \end{cases}$$

Wherein ξ denoted the level of damage of the attacker and $0 < \xi < 1$.

State Transition Probability: According to transition probabilities, the stochastic game moves from one state to another, simultaneously controlled by the players' actions. Furthermore, each initialization of time steps depends on the current state and the subsequent potential state. The transition probability is given by the probability $\mathbb{P}$: $\mathbb{P}(s^{(t+1)}|a^{(t)}, b^{(t)}, s^{(t)}) = \prod_{v \in S_d} \mathbb{P}_d(v^{(t+1)}|a^{(t)}, b^{(t)}, s^{(t)}) \prod_{u \in S_a} \mathbb{P}_a(u^{(t+1)}|a^{(t)}, b^{(t)}, s^{(t)})$ This transition probability relies on the performance of the IDS and the success damage of the attacker, which can be learned directly from experience or indirectly in an empiric way.

Expected Payoffs: Let $x^{(t)}(s)$ and $y^{(t)}(s)$ be the probabilities in time step t and state s of players N_d and N_a to choose respectively the actions $a \in A_d$ and $b \in B_a$. We consider $\mathbf{x}$ and $\mathbf{y}$ as distribution vectors of strategies of the IDS and the attacker at t: $\mathbf{x} = (x_0^{(t)}(s), x_1^{(t)}(s), x_2^{(t)}(s), x_3^{(t)}(s)); x_i^{(t)}(s) \geq 0$; $\sum_{i=0}^{3} x_i^{(t)}(s) = 1$. $\mathbf{y} = (y_0^{(t)}(s), y_1^{(t)}(s), y_2^{(t)}(s)); y_j^{(t)}(s) \geq 0; \sum_{j=0}^{2} y_j^{(t)}(s) = 1$. Where player N_d takes action $a_d \in A_d$ with probability $x_i^{(t)}$ and player N_a takes action $b_a \in B_a$ with probability $y_j^{(t)}$. Hence, the utility of IDS N_d is the sum of discounted payoffs for endless instant: $w_d^{\gamma}(s, \mathbf{x}, \mathbf{y}) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{s,\mathbf{x},\mathbf{y}}^{s} U_d^{(t)}(a, b, s)$. Furthermore, the utility of the attacker N_a can be expressed as: $w_a^{\tilde{\gamma}}(s, \mathbf{x}, \mathbf{y}) = \sum_{t=0}^{\infty} \tilde{\gamma}^t \mathbb{E}_{s,\mathbf{x},\mathbf{y}}^{s} U_a^{(t)}(a, b, s)$, Where $\gamma, \tilde{\gamma} \in [0, 1]$ are the discount factors that models respectively the level of the IDS to monitor against any attack strictly for its own benefit and the attacker to take action separately.

Nash Equilibrium of Game: In this part, we focus our effort on finding a solution to our proposed game while respecting the requirement of our system in terms of security level. The Nash equilibrium can be described in general for arbitrary strategies as a strategy profile such as no player has an incentive to

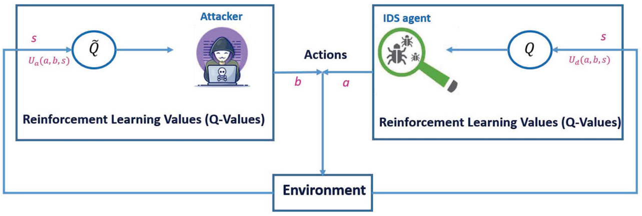

Attacker
s
$\tilde{Q}$
$U_a(a,b,s)$
Reinforcement Learning Values (Q-Values)
Actions
b
a
IDS agent
Q
s
$U_d(a,b,s)$
Reinforcement Learning Values (Q-Values)
Environment

Fig. 2. The Attacker-IDS Stochastic Game Model.

diverge unilaterally from his chosen strategy. We formulate our definition as follow:

Definition 1: Let $\mathbf{w}_d^{\gamma} = [w_d^{\gamma}(s_1), \ldots, w_d^{\gamma}(s_N)]$ and $\mathbf{w}_a^{\tilde{\gamma}} = [w_a^{\tilde{\gamma}}(s_1), \ldots, w_a^{\tilde{\gamma}}(s_N)]$. The Nash equilibrium action is a pair of strategies $(\mathbf{x}^*, \mathbf{y}^*)$, that can be expressed as : $\mathbf{w}_d^{\gamma}(s, \mathbf{x}^*, \mathbf{y}^*) \geq \mathbf{w}_d^{\gamma}(s, \mathbf{x}_1^*, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N^*, \mathbf{y}^*), \quad 0 \leq \mathbf{x}_n \leq 1.\mathbf{w}_a^{\tilde{\gamma}}(s, \mathbf{x}^*, \mathbf{y}^*) \geq \mathbf{w}_a^{\tilde{\gamma}}(s, \mathbf{x}^*, \mathbf{y}_1^*, \ldots, \mathbf{y}_m, \ldots, \mathbf{y}_M^*), 0 \leq \mathbf{y}_m \leq 1$. For all strategies $\mathbf{x}, \mathbf{y}$ and state s, no participants have the interest in modifying its profiles in the balance. Within the ϵ-Nash equilibrium is a stationary strategy that can verify the following formula: $\mathbf{w}_d^{\gamma}(s, \mathbf{x}^*, \mathbf{y}^*) - \mathbf{w}_d^{\gamma}(s, \mathbf{x}_1^*, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N^*, \mathbf{y}^*) \geq \epsilon, 0 \leq \mathbf{x}_n \leq 1. \quad \mathbf{w}_a^{\tilde{\gamma}}(s, \mathbf{x}^*, \mathbf{y}^*) - \mathbf{w}_a^{\tilde{\gamma}}(s, \mathbf{x}^*, \mathbf{y}_1^*, \ldots, \mathbf{y}_m, \ldots, \mathbf{y}_M) \geq \epsilon, 0 \leq \mathbf{y}_m \leq 1$.

Definition 2: Nash equilibrium of the auxiliary game $\dot{\Gamma}$ with the equilibrium payoffs $(w_d^{\gamma}(s, \mathbf{x}^*, \mathbf{y}^*), w_a^{\tilde{\gamma}}(s, \mathbf{x}^*, \mathbf{y}^*))$ is the strategy profile $(\mathbf{x}^*(s), \mathbf{y}^*(s))$. For each $s \in S$, the payoffs of the players N_d and N_a are defined respectively as follows: $\dot{\Gamma}(s, a, b) = U_d + \gamma \sum_{s' \in S} \mathbb{P}(s'|a, b, s)$ $w_d^{\gamma}(s,' \mathbf{x}, \mathbf{y}).\ddot{\Gamma}(s, a, b) = U_a + \tilde{\gamma} \sum_{s' \in S} \mathbb{P}(s'|a, b, s) w_a^{\tilde{\gamma}}(s,' \mathbf{x}, \mathbf{y})$.

In order to find the Nash equilibrium solutions, we use the solutions of the fixed-point equations (4),(5), then $\mathbf{w}_a^{\gamma}, \mathbf{w}_b^{\tilde{\gamma}}$ the unique solutions are given by solving the next systems:

$$\mathbf{w}_a^{\gamma} = NE\{\dot{\Gamma}(a, b, \mathbf{w}_a^{\gamma})\}, \tag{4}$$

$$\mathbf{w}_b^{\tilde{\gamma}} = NE\{\ddot{\Gamma}(a, b, \mathbf{w}_b^{\gamma})\}, \tag{5}$$

where NE is the Nash payoff function that yields non-zero-sum game.

Theorem 1: Any non-zero-sum stochastic game with finite discounting has at least one equilibrium point in stationary strategies (As proved in [42]).

Optimal Decision Making Policy: In the previous section, we modeled the two-player non-zero-sum stochastic game. In such a game, the IDS aims to maximize its expected payoff by minimizing the cost of damage incurred by the attacker. In contrast, the attacker aims to maximize his expected payoff. Consequently, the Q-value is maximized. Hence, we give details on how to solve this game from the side of the IDS and attacker to get optimal decision policy by a value iteration scheme. We also described the expected total value of a stochastic game by $V^t(s)$ at a state s. At a given stage t, we denote a $Q^t(a, b, s)$ as an expected discount cost when the IDS takes action a, and the attacker takes action b in the current state s as shown in Fig. 2.

The optimal value in this game can be obtained as follow:

$$V^t(s) = \max_{\mathbf{x}} \max_{\mathbf{y}} \mathbb{E} \left\{ \sum_{a \in A_d} y^T Q_t(a, b, s) x \right\}, \tag{6}$$

TABLE II
EVALUATION INDICATORS

Situations	Actual Record	Predicted Record
True positive (Tp)	anomalous	anomalous
True negative (Tn)	normal	normal
False positive (Fp)	normal	anomalous
False negative (Fn)	anomalous	normal

Let η, $\tilde{\eta}$ be an equilibrium of the game $\dot{\Gamma}(V^{t+1}, \tilde{V}^{t+1})$ for $0 < t < T$, The value functions for every $s \in S$ given by the following formula, wherein $Q_t(a, b, s)$ is updated by:

$$Q_i^t(a, b, s) = \mathbb{E}_{\eta_i^t, \eta_{-i}^t, \tilde{\eta}^t} \Bigg[U_d(a, b, s) + \gamma_i \sum_{s' \in S} \mathbb{P}(s'|a, b, s) V_i^{t+1}(s') \Bigg], \tag{7}$$

$$\tilde{Q}_j^t(a, b, s) = \mathbb{E}_{\eta^t, \tilde{\eta}_j^t, \tilde{\eta}_{-j}^t} \Bigg[U_a(a, b, s) + \tilde{\gamma}_j \sum_{s' \in S} \mathbb{P}(s'|a, b, s) \tilde{V}_j^{t+1}(s') \Bigg], \tag{8}$$

However, $(\eta, \tilde{\eta})$ is the ϵ-Nash equilibrium of every $s \in S$. Then, η, $\tilde{\eta}$ are optimal stable strategies for the IDS and the attacker respectively.

IV. Experimental Results and Analysis

A. Performance evaluation metrics: In order to study the efficiency of our proposed DRL-IDS to predict the system's state and estimate the quality of attack detection in the network environment, a set of performance metrics is applied such as accuracy, detection rate, false negative rate, recall, and F-measure which relies on the following metrics: *Tp* (True positives), *Tn* (True negatives), *Fp* (False positives), *Fn* (False negatives) that constructed the confusion matrix in Table II.These metrics are defined as follow:

Accuracy (AC): The aptness to predict accurately and to detect all known and newly malicious activities.

Precision (PR): Configurable hyper-parameter used to detect correctly the intrusive actions of nodes trying to attack the system which is expressed as follow: $PR = \frac{Tp}{Tp+Fp} * 100\%$.

False positive rate (FPR): The normal behaviors that are detected as abnormal activity, described in the above formula: $FPR = \frac{Fp}{Fp+Tn} * 100\%$.

Recall (RC): The ratio of accurate prediction over the overall anomaly detected in the analyzed records, equivalent to the Detection Rate (DR) which can be calculated as: $RC = DR = \frac{Tp}{Tp+Fn} * 100\%$.

F-measure (FM): A measurement of the precision calculated as the weighted harmonic mean of the detection rate and precision of the model. FM formulated by: $FM = 2 * \frac{PR*RC}{PR+RC} * 100\%$.

B. Experimental setup: The simulation results provided in the following sections are obtained using an Intel(R) Core(TM) i5-5200 U CPU machine with 8 GB memory. In the first part of our study concerning DRL, we performed our simulation using

TABLE III
PARAMETERS INPUTS SETTINGS

Parameters	Value
Learning rate	0.4, 0.04, 0.004, 0.0004, 0.00004
batch size	32
Episodes	37790 (30%), 88180 (70%), 125973 (100%)
Episode length	1 timestep
buffer size	50000
Discount factor	0.999
Type of features	Normal, DoS, R2L, U2R, PROBE.
Tested dataset	NSL-KDD
Number of actions of the IDS k	4
Number of actions of the attacker f	3
Profit of both players (%)	high-90,low-50
Cost of IDS C_d (%)	10
Cost of the attacker C_a (%)	15
Trust rang	[0,1]
Iteration	500
Sensibility ϵ	0.1
Discount Factor for the IDS γ level	high-0.3, low-0.05
Discount Factor for the attacker $\tilde{\gamma}$	high-0.3, low-0.05
Damage ξ	0.1

the Stable Baselines framework, composed of a set of RL algorithms stable implementation on the top of OpenAI Baselines. Using this framework gives a credential of our results since we worked on a structured environment that respects RL properties and OpenAI Baselines requirements. Using this framework, we defined our customized environment to handle the NSL-KDD dataset as an observation and modeled our reward function for efficient agent learning. The agent gets a reward for the current episode only if his action corresponds to the label, knowing that we considered in our case each episode include only one timestep. We used the Google Colab platform to implement our game model and analyze the numerical results in the second part. This part consists of implementing our mathematical formula to study our proposed modulation's performance. The parameters values used in our contribution are listed in Table III.

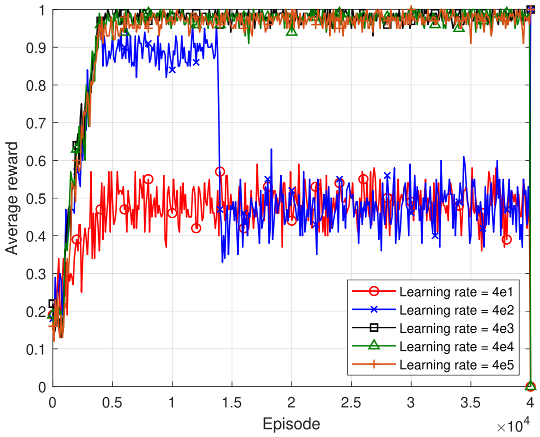


Fig. 3. DRL convergence for different Learning rate value.

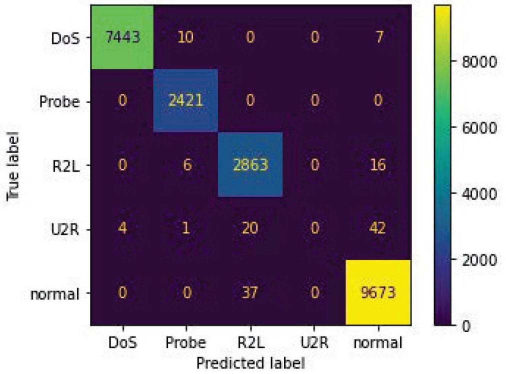
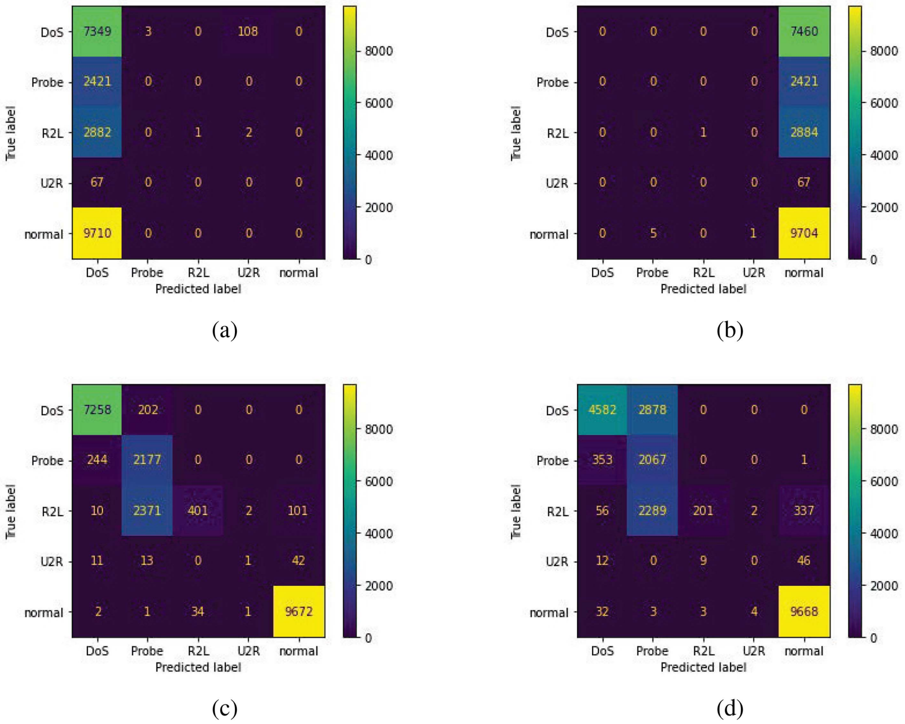


Fig. 4. Confusion matrix of our detection model on NSL-KDD dataset.

C. Proposed DRL-IDS evaluation: We analyzed and discussed the impact of the proposed DRL-IDS and its performance compared to the existing IDSs in the literature. We analyzed our proposed IDS model based on DRL by simulating its convergence using different learning rates. Furthermore, we trained our model on different portions of the NSL-KDD dataset. Our simulation considers a DRL agent with two-layer fully connected and a Deep RL with seven hidden layers. We then compared the obtained results with the existing model in the literature. Through this study, we have the detection of data streams cover three states: normal, detection, and noDetection. We consider the transition probability matrix between states P_{d1} which is symmetric probability, then we obtain similar probability of P_{d2} and P_{d3} as: $\mathbf{P_{d1}} = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0.4 \end{pmatrix}$. To find the best optimal actions that can improve the performance of the proposed scheme, we adopted the technique of deep Q-network. The complexity of the Q-learning algorithm depends on many possible aspects, such as the high number of actions and the state transition probability. Hence, deep Q-learning uses DL to approximate the Q-function. Consequently, the convergence time is affected by many factors, including the learning rate, the mini-batch size, and the number of convolutional layers. However, the evaluation of the target Q-network is the only one trained relying on the gradient descent method, and we substitute the target Q-network for every mini-batch size by the updated trained Q-value. Among the tests that we performed for our implementation, we considered simulating the convergence of the model. Initially, we trained the model using a learning rate of α=4e-1 that is depicted in Fig 3. by the red line, and we can easily see that this case is the worst since the model converges at the lowest reward value corresponding to the lowest accuracy. The blue line in this figure depicts the training using the learning rate 4e-2; in this value, the model is not stable, as shown in the figure. The reward converges at 0.9 and unexpectedly goes down to 0.5 after 15000 episodes. Later, we trained the model on a learning rate of 4e-3 corresponding to the best accuracy, as shown in the following figure. The remaining simulations converge to the maximum, but they don't reach 1 and are unstable. Consequently, they ensure good accuracy about 0.8 and 0.9 but not the best as obtained by the learning rate of 4e-3. Fig 4 shows the outcome of our best simulation test using a learning rate of 4e3. It shows the confusion matrix for testing on KDD test+ of our proposed agent on detecting different attacks. Our agent detects all probe attacks, which are 2421 attacks in the test dataset. While 2863 R2L attacks can be detected by our agent and fail to identify the remaining R2L attack of the test dataset. This figure shows that our agent fails in detecting U2R attacks. In addition, the majority



(a) (b)

(c) (d)

Fig. 5. Confusion matrix of DRL agent trained using different learning rates.

of Dos attacks and normal activities can be identified using the DRL-IDS. All these results will be supported by showing the accuracy, the precision, the F1-score, and the recall in Fig 6. As we can see, our proposed DRL-IDS cannot detect U2R attacks due to the inefficient U2R sample in the dataset used for training the agent. This means that our agent requires sufficient samples of the specific attack for training to identify this type of attack accurately. Our trained agent can identify the most dangerous attack, which is, in our case, the DoS attack. Moreover, our DRL-IDS agent outperforms the existing model in literature by providing the most accurate classification of DoS attack and identifying well the normal traffic. Fig 5. shows confusion matrices of different tests conducted toward finding the best DRL training parameters. In this case, we varied the learning rate on the training process of our agent as shown in the previous Fig 3. each learning corresponds to a different learning outcome. As we mentioned previously, the write learning rate in our case is equal to 4e-3 while the other tentative either converge in lower reward or gives unstable convergence, which is the case of the learning rates 4e-1, 4e-2, 4e-4, and 4e5 depicted respectively in Fig 5(a), (b), (c), (d), (a) shows the training results for a learning rate of 4e-1. Our agent in this experiment can only identify the DoS attacks, such as 7349 were identified while failed on identified only 201 DoS attacks. The other attacks' classes can not be identified using our DRL agent trained with this learning rate. Fig 5. (b) shows the confusion matrix of agent training using 4e-2 as the learning rate. In this scenario, our agent detects normal traffic well while completely failing to detect the other classes of attack. In these two latter cases, the behavior of our model will be explained by the low accuracy resulting from training with the imperfect parameters. DRL agent trained using 4e-4 learning rate can detect the majority of DoS and Probe attacks as well as the normal traffic as shown in Fig. 5(c) Where fewer DoS attacks and fewer Probe attacks are detected using agent trained on 4e-5 while normal traffic can be easily identified, this outcome can be easily explained when considering the model's accuracy-such accuracy will be less than that of the agent trained on 4e-4 learning rate Fig. 5(d). To adjust the training parameters of our DRL-IDS agent, we split the NSL-KDD dataset into different portions used for training, knowing that we have a separate dataset for testing. We should note that each row of the dataset corresponds to an episode. In general, each episode is composed of a set of time steps where the agent takes action to win a reward at the end of the episode. Although in our case, the agent plays one time and gets a reward which means that the episode is composed of a single time step. Fig 6. shows results of training using different dataset portions, accuracy, precision, F1 score, recall are all used to identify the best dataset portion to train our agent efficiently and get the best results. Green, orange, and gray bars depict the results obtained for training on 30%, 70%, and full dataset. The green bar shows the best results compared to the others, which means that training our agent using only 30% of the dataset results in a DRL-IDS agent that can provide the most accurate classification. We can explain this by the number of experience required to atteint the maximum convergence of the DRL agent. Experience is the raw data received by the agent as an observation to take action at each episode, which is the decision if the entered raw data corresponds to a specific attack or is normal data. We observed in our study that adding more data raws for training only destabilizes the agent and reduces its performance; as shown in our simulation results, this is due to the unbalanced NSL-KDD dataset. Knowing that learning rate, batch size, and

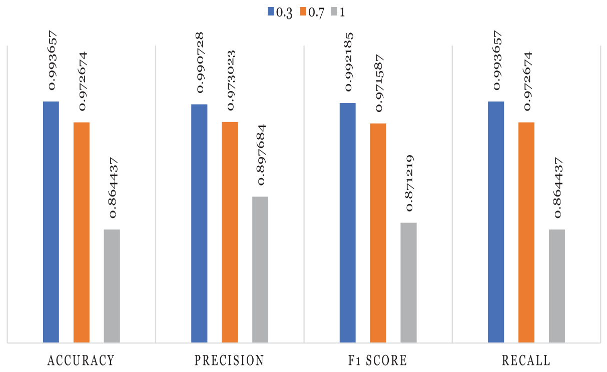


Fig. 6. Performance metrics of DRL-IDS trained on different NSL-KDD portion.

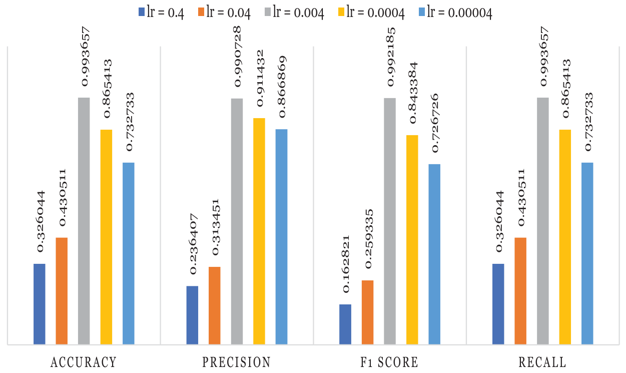


Fig. 7. Performance evaluation of DRL-IDS using different learning rates.

the number of episodes is the main factors that should be tuned to obtain the perfect convergence of our agent in the training process. On the other hand, Fig 7. shows different performance parameters of testing our proposed DRL-based IDS, such as the accuracy, precision, F1-score, and recall. This figure shows the outcome of testing our pre-trained agent using different learning rates in the same direction as Fig 5. to explain the behavior of the system. The blue bar depicts the agent trained using a learning rate of 0.4, the worst one, as shown in the confusion matrix. Agent trained using 0.04 learning rate presented by orange bar achieve an accuracy of 43.05% and precision of 31.34%. Otherwise, the best results are depicted by the green bar corresponding to the pre-trained agent using a learning rate of 0.004; an accuracy of 99.36 is ensured, which is the highest one. While this agent can provide a decision with a precision of 99.07%, these results support the previous ones, meaning that 0.004 is the best learning rate to train our DRL agent on 30% of the dataset. Other findings show promising results in terms of accuracy and F1-score but still under the performance of the previous agent.

1) Impact of Attack Scenarios: We benchmarked our proposed DRL-IDS by evaluating its performance metrics in various attack scenarios as described in the table IV. The results demonstrated that DRL-IDS could accurately detect different kinds of attacks. However, its performance is better for DoS attacks, which its high occurrence can explain in the NSL-KDD dataset. The results in this table show the good performance of our model for the attack classes like DoS, Probe, and R2L. But, the model considers the attack of the type U2R as missing/neglected. Because the number of existing attack of this class is very low as depicted in Table I) and the model use 30% of training data including this class. This means that our agent requires more samples of the specific attack for training to accurately identify this type of attack. Our trained agent can identify the most dangerous attack, which is, in our case, the DoS attack.

TABLE IV
DRL-IDS PERFORMANCE EVALUATION METRICS IN DIFFERENT ATTACK CATEGORIES ON NSL-KDD TEST SET

Class	PR	RC	FM
DoS	1.00	1.00	1.00
Probe	0.99	1.00	1.00
U2R	0.00	0.00	0.00
R2L	0.98	0.99	0.99
Macro average	0.79	0.80	0.80
Weighted average	0.99	0.99	0.99

TABLE V
COMPARISON OF ATTACK DETECTION OF EXPERIMENTAL RESULTS WITH OTHER EXISTING STUDIES (%)

Approaches	DoS	Probe	R2L	U2R
RNN [19]	83.49	83.40	24.69	11.50
MDPCA-DBN [20]	81.09	73.94	17.25	6.50
CNN-LSTM [21]	98.8	0.0	94.6	99,4
DLHA [22]	92.4	90.87	96.67	100
Hybrid K-means+RF [23]	90.42	91.53	73.84	25.79
MemAE [24]	94.23	95.70	85.86	95.79
PTDAE [25]	99.36	100	48.25	18.92
Siam-IDS [43]	85.37	48.66	33.25	56.72
Our Proposed DRL-IDS	100	99.0	98.0	0.0

2) Comparison with state-of-the-art: The outcomes of the performance evaluation of our proposed DRL-IDS model are compared with the state-of-the-art applied for intrusion detection system using the NSL-KDD dataset. We compared the results of the detection rate (DR) of each attack in the dataset with hybrid approaches such as MDPCA-DBN, Siam-IDS, CNN-LSTM, PTDE, MemAE, and hybrid k-means and Random Forest. Furthermore, Reinforcement learning (RL) approaches such as Adversarial Environment Reinforcement Learning(AE-RL), Double Deep-Q-Network(DDQN), Adversarial/Multi-Agent Reinforcement Learning using Deep Q-Learning (AE-DQN), Attention mechanism-based Deep-Q-Network (A-DQN), Deep Q-Learning (DQL) as provided in [27], [28], [29], [30], [31]. The remaining comparison is based on the Accuracy (AC), False Positive Rate (FPR), Precision (PR), Recall (RC), and F1-score (FM). In Table V the comparison reports that our proposed DRL-IDS is not the best model to detect U2R, as our findings are about 0%, while others show higher scores. In contrast, our model can successfully identify any attack compared to others with low detection ratings on DoS, Probe, and R2L. Our model significantly exceeds all other approaches by achieving the highest detection rates of 100% in DoS, 99% in Probe, and 98% in R2L. As demonstrated in Table VI, our model is also more accurate compared to other methods, whereas it has less FPR. However, the lowest AC achieved by the DQL model is approximately 78.07%. Both DDQN and A-DQN models have an accuracy of 89.78% and 97.2%, respectively. In addition, it was observed that our DRL-IDS model offered the best performance based

TABLE VI
PERFORMANCE EVALUATION METRICS WITH EXISTING STUDIES ON RL APPROACHES

Approaches	AC	FPR	PR	RC	FM
AE-RL [27]	0.8016	-	-	-	0.794
DDQN [28]	0.8978	-	0.8944	0.9303	0.912
AE-DQN [29]	0.80	-	-	-	0.79
A-DQN [30]	0.972	0.0142	0.965	0.991	0.978
DQL [31]	0.7807	-	0.7784	0.7676	0.8141
Our proposed DRL-IDS	0.9936	0,0022	0.9907	0.9936	0.9921

TABLE VII
COMPUTATIONAL TIME FOR TRAINING AND PREDICTION OF APPROACHES

Approaches	Training time (sec)	Predicting time (sec)
AE-RL [27]	1090.13	0.50
DDQN [28]	507.01	0.55
DQN [28]	290.5	0.54
Policy Gradient [28]	352.48	0.65
Actor Critic [28]	1725	0.83
DQL [31]	1260	NA
Our proposed DRL-IDS	90.402	0.32

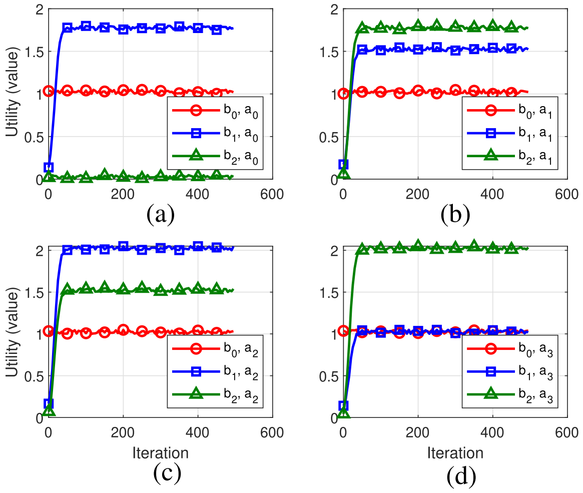


Fig. 8. The Q-value of the IDS against the opponent actions of the attacker such as figures (Fig. (a), (b), (c), and (d)) correspond respectively to the IDS taking the actions a_0, a_1, a_2, and a_3.

on FM(F1-score). This is because of the efficient modulation that allowed the DRL agent to learn well during the training process and make accurate decisions to protect the system from attacks during the inference step. During the training process of our DRL-IDS agent, we identified that once the RL agent is trained, the decision-maker gives the accurate intrusion label (which is the agent action) for the considered state (depicted as the intrusion features). This can be viewed only as a simple neural network model that performs fast inference to provide the most accurate prediction and be used in an industrial production environment. This pattern can be confirmed in Table VII, which denotes the training and predicting times for our model with such RL approaches. It can be seen that our model has a smaller training and predicting time than other approaches. Such as 90.402 seconds spent on training, and 0.32 seconds on testing overall $KDDTest^+$.

D. Stochastic Game Impact on the System Safe: We evaluated the IDS performance using the stochastic game, where this later consists of modeling the interaction between the IDS and the attacker. To obtain an efficient solution for our system, we used a value iteration algorithm to solve the equations (7) and (8) consequently to find the game value and obtain optimal stationary strategies. We recall the actions taken by the IDS are $a_0 = 0$, $a_1 = 1$, $a_2 = 2$, $a_3 = 3$ correspond to the level of protection low, medium, high, and critical respectively. The attacker actions $b_0 = 0$, $b_1 = 1$, and $b_2 = 2$ denote respectively that the attacker does not attack, attacks passively, and attack actively. We assume in our case that the transition probabilities between states of the game are unknown to the system. In the following, we split several scenarios to illustrate the convergence of our proposed model. Fig. 8 shows the Q-value of the IDS using his actions against the attacker's actions corresponding to b_0, b_1, and b_2. We can see that the IDS's Q-value converges after the first 10 rounds for the four scenarios, such as Fig. 8(a) shows the result of the IDS playing action low protection in the function of the attacker's actions. The IDS will lose against the attacker when the latter plays the action b_2 while the IDS wins the game by maximizing his Q-value when the attacker plays b_1 since the passive attack performed by the attacker is not dangerous for our system while on the other side the IDS kept their resources for upcoming dangerous attacks. Fig. 8(b) shows that the IDS maximizes its Q-value and wins the game by playing the action a_1 against b_2. At the same time, the IDS loses the game when playing b_0. This is due to the energy spent in detecting the attack while the attacker plays b_0 corresponding to no attacker, which is called a true negative rate. From Fig. 8(c) corresponding to the IDS playing the action a_2, we can see that the IDS wins the game against the attacker when the latter plays the action b_1 while losing the game against b_0 corresponding to a false-negative rate. Fig. 8(d) shows that the IDS only maximizes its Q-value when the attacker is performing an active attack depicted by b_2 (Detection Rate). Similarly, to the previous results Fig. 8(c), the IDS loses the game against the attacker playing either b_0 or b_1. This is because the IDS takes high-risk and critical actions against passive and low risque attacks which results in a negative reward for the IDS that fails in predicting the risque of the attacker's action. In Fig. 9 we plot the Q-value of the attacker regarding the actions of the IDS such as Fig. 9(a), (b), (c), and (d) correspond respectively to the IDS taking the action low detection, medium, high and critical detection. In general, the attacker can only maximize its Q-value when performing an active attack (b_2) regardless of the IDS action. In contrast, the attacker obtains the best Q-value when the IDS takes action a_0 corresponding to the low detection Fig. 9(a) after converging to the max Q-value. The optimal Q-value decreases regarding the IDS actions as shown in figures Fig. 9(b), (c), and (d). In this case, the action b_2 against the action a_0 of the IDS presents the best scenario of the attacker's gain. In Fig. 9(b) the attacker gain less with the action b_2 due to the detection level of the IDS playing a_1. The max Q-value of the attacker keeps decreasing regarding IDS' actions, such as the lowest Q-value correspond to the critical detection of this later. Fig. 10(a) confirms the existence of the Nash equilibrium in the game (a_0^*, b_1^*) and (a_3^*, b_2^*), where both points correspond respectively to compromised system s_4 and safe system s_1, such as the first point present the scenario where the attacker perform

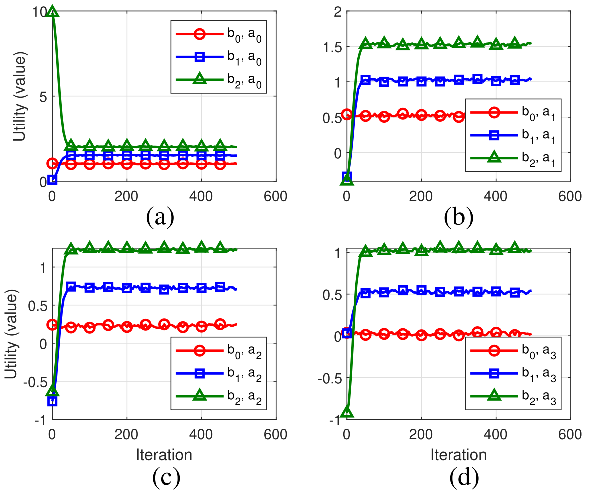


Fig. 9. The Q-value of the attacker against the opponent actions of the IDS such as figures (a), (b), (c), and (d) correspond respectively to the IDS taking the actions a_0, a_1, a_2, and a_3 against attacker taking the actions b_0, b_1, and b_2.

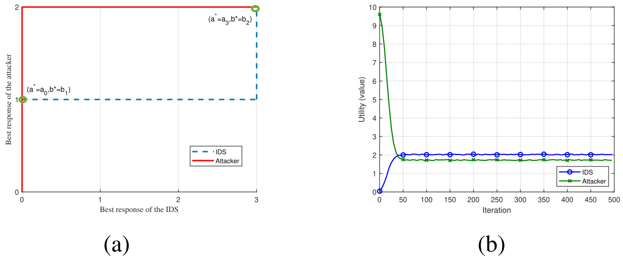


Fig. 10. (a) Best reply of IDS (player I) and the best reply of attacker (player II) curves show the existing Nash Equilibrium solutions namely ($a^* = a_0, b^* = b_1$) and ($a^* = a_3, b^* = b_2$). (b) The Q-values of both players at the Nash equilibrium. (a_3^*, b_2^*).

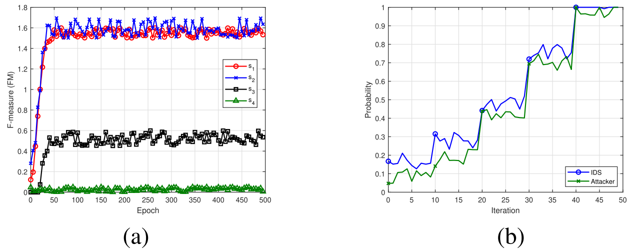


Fig. 11. (a) The IDS-game value is plotted in function of the iteration times at different states of the game. (b) Probability of system being safe.

passive attack and the IDS apply a low detection strategy. The second one corresponds to the attacker performing an active attack and the IDS performs a critical detection which allows them to protect the system against the attacker. These results confirm the expected system protection. At the equilibrium, each player gets its maximum Q-value depending on others' actions, as depicted in Fig. 10(b). This is attained only if both players kept unchanged their best strategy.

In Fig. 11(a) we plot the game value $V(s)$ for $s = s_1, s_2, s_3, s_4$ for the stationary mixed strategies. After 500 iterations the game value of each state are given by $V(s_1) = 1.5311$, $V(s_2) = 1.5967$, $V(s_3) = 0.5030$, $V(s_4) = 0.0373$ which correspond to the mixed equilibrium points $(x^*(s_i), y^*(s_i))$ where $x^*(s_i)$ present the IDS actions probability at state s_i and $y^*(s_i)$ is the attacker actions probability at state s_i. Such as the mixed strategies at the equilibrium are given by (0.03343351, 0.06995044, 0.15010229, 0.74651377), (0.529 44866, 0.30220828, 0.08202081, 0.08632225), (0.17142309, 0.18300888, 0.26373177, 0.38183626), (0.5737554, 0.357 35559, 0.05597631, 0.0129127) for s_1, s_2, s_3 and s_4 respectively, for the IDS and (0.02211347, 0.10832202, 0.86956452), (0.75766459, 0.12885015, 0.11348526), (0.732 69738, 0.19175167, 0.07555095), (0.08460271, 0.3667609, 0.54863639) for the attacker. Fig. 11(b) shows the probability that the system is at the safe s_2 and not safe s_4 states corresponding respectively to the best strategy of the IDS and the attacker. This later represents the probability of the attacker attaining the equilibrium (blue line). We can see that the probability of both players reaches the value one after 40 iterations; this is due to the convergence of our system as shown in figures Fig. 8(d) and Fig. 9(a), where attacker scheme converge after 40 iterations. In addition, we remark that the IDS probability is more significant than the attacker and attain one faster. Anomaly activity is successfully detected in the system corresponds to the state s_1 (true positive), the state s_2 means the system successfully detected acceptable activity (true negative). A false positive state s_3 is when normal activity identified as an anomaly. The last state, false-negative s_4, is the most dangerous one. This is when malicious activity is identified as normal.

V. Conclusion

This paper presents a robust IDS based on DRL and the analyses of the interaction between the IDS and attacker using a non zero-sum stochastic game model. The aim is to detect and identify known and unknown malicious behaviors from the network traffic based on the NSL-KDD dataset. We have implemented the IDS with and without the game where the proposed DRL-based single IDS approach built based on MDP has shown significant improvements in terms of accuracy, precision, recall, and F1-score. Likewise, using the concept of a value iteration in the game, we have obtained better protection of the network traffics to deal with the active attacker. The obtained results show that our proposed model can provide a high-security level of the system compared to the existing approaches and handle the network's time-variant property under any type of attack. Therefore, our future work will focus on evaluating the model on the other datasets such as BoT-IoT and IoT-23 intrusion detection and developing an extended model to improve the network security against large-scale cyber-attacks and show the impact of heterogeneity on the IDS performance and computational complexity. Furthermore, it can be interesting to extend this work to a general non-zero-sum stochastic game by considering multiple attackers and IDS scenarios and finding the safe situation of the system.

References

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (VNI): Complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowl. Netw. Presentation*, 2018. Accessed: Jan. 5, 2022. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf

[2] G. Singh and N. Khare, “A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques,” *Int. J. Comput. Appl.*, pp. 1–11, Feb. 2021, https://doi.org/10.1080/1206212X.2021.1885150

[3] Y. Zhao, Y. Xun, and J. Liu, “ClockIDS: A real-time vehicle intrusion detection system based on clock skew,” *IEEE Internet Things J.*, doi: 10.1109/JIOT.2022.3151377.

[4] Y. Xun, J. Liu, and Y. Zhang, “Side-channel analysis for intelligent and connected vehicle security: A new perspective,” *IEEE Netw.*, vol. 34, no. 2, pp. 150–157, Mar./Apr. 2020.

[5] I. H. Sarker, “Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective,” *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–16, 2021.

[6] Y. Xun, Y. Zhao, and J. Liu, “VehicleEIDS: A novel external intrusion detection system based on vehicle voltage signals,” *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2124–2133, Feb. 2022.

[7] N. Marchang, R. Datta, and S. K. Das, “A novel approach for efficient usage of intrusion detection system in mobile ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1684–1695, Feb. 2017.

[8] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, “Automobile driver fingerprinting: A new machine learning based authentication scheme,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1417–1426, Feb. 2020.

[9] N. Haddadou, A. Rachedi, and Y. Ghamri-Doudane, “A job market signaling scheme for incentive and trust management in vehicular ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3657–3674, Aug. 2015.

[10] A. Kaci and A. Rachedi, “Toward a machine learning and software defined network approaches to manage miners’ reputation in blockchain,” *J. Netw. Syst. Manage.*, vol. 28, no. 3, pp. 478–501, 2020.

[11] A. Rachedi and A. Hasnaoui, “Advanced quality of services with security integration in wireless sensor networks,” *Wireless Commun. Mobile Comput.*, vol. 15, no. 6, pp. 1106–1116, 2015.

[12] A. Kaci, T. Bouabana-Tebibel, A. Rachedi, and C. Yahiaoui, “Toward a Big Data approach for indexing encrypted data in cloud computing,” *Secur. Privacy*, vol. 2, no. 3, 2019, Art. no. e65.

[13] N. S. Bhati and M. Khari, “A survey on hybrid intrusion detection techniques,” in *Research in Intelligent and Computing in Engineering*. Berlin, Germany: Springer, 2021, pp. 815–825.

[14] Y. Du, J. Xia, J. Ma, and W. Zhang, “An optimal decision method for intrusion detection system in wireless sensor networks with enhanced cooperation mechanism,” *IEEE Access*, vol. 9, pp. 69498–69512, 2021.

[15] B. Subba, S. Biswas, and S. Karmakar, “A game theory based multi layered intrusion detection framework for wireless sensor networks,” *Int. J. Wireless Inf. Netw.*, vol. 25, no. 4, pp. 399–421, 2018.

[16] L. Shi, X. Wang, and H. Hou, “Research on optimization of array honeypot defense strategies based on evolutionary game theory,” *Mathematics*, vol. 9, no. 8, 2021, Art. no. 805.

[17] B. Mishra and I. Smirnova, “Optimal configuration of intrusion detection systems,” *Inf. Technol. Manage.*, vol. 22, pp. 231–244, 2021.

[18] H. Benaddi, K. Ibrahimi, A. Benslimane, and J. Qadir, “A deep reinforcement learning based intrusion detection system (DRL-IDS) for securing wireless sensor networks and Internet of Things,” in *Proc. Int. Wireless Internet Conf.*, Springer, 2019, pp. 73–87.

[19] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[20] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, “Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks,” *Appl. Sci.*, vol. 9, no. 2, 2019, Art. no. 238.

[21] M. Ahsan and K. E. Nygard, “Convolutional neural networks with LSTM for intrusion detection,” in *Proc. CATA*, 2020, pp. 69–79.

[22] T. Wisanwanichthan and M. Thammawichai, “A double-layered hybrid approach for network intrusion detection system using combined naive bayes and SVM,” *IEEE Access*, vol. 9, pp. 138432–138450, 2021.

[23] C. Liu, Z. Gu, and J. Wang, “A hybrid intrusion detection system based on scalable k-means random forest and deep learning,” *IEEE Access*, vol. 9, pp. 75729–75740, 2021.

[24] B. Min, J. Yoo, S. Kim, D. Shin, and D. Shin, “Network anomaly detection using memory-augmented deep autoencoder,” *IEEE Access*, vol. 9, pp. 104695–104706, 2021.

[25] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, “Attack classification of an intrusion detection system using deep learning and hyperparameter optimization,” *J. Inf. Secur. Appl.*, vol. 58, 2021, Art. no. 102804.

[26] N. Sengupta, J. Sen, J. Sil, and M. Saha, “Designing of on line intrusion detection system using rough set theory and q-learning algorithm,” *Neurocomputing*, vol. 111, pp. 161–168, 2013.

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

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

[29] E. Suwannalai and C. Polprasert, “Network intrusion detection systems using adversarial reinforcement learning with deep q-network,” in *Proc. IEEE 18th Int. Conf. ICT Knowl. Eng. (ICTKE)*, 2020, pp. 1–7.

[30] K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, “Attention based multi-agent intrusion detection systems using reinforcement learning,” *J. Inf. Secur. Appl.*, vol. 61, 2021, Art. no. 102923.

[31] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, “Deep q-learning based reinforcement learning approach for network intrusion detection,” *Computers*, vol. 11, no. 3, p. 41, 2022.

[32] Z. Wang, S. Xu, G. Xu, Y. Yin, M. Zhang, and D. Sun, “Game theoretical method for anomaly-based intrusion detection,” *Secur. Commun. Netw.*, vol. 2020, p. 10, 2020, Art. no. 8824163, https://doi.org/10.1155/2020/8824163

[33] V. Sangeetha, M. Vaneeta, S. S. Kumar, P. K. Pareek, and S. Dixit, “Efficient intrusion detection of malicious node using bayesian hybrid detection in MANET,” in *Proc. IOP Conf. Series: Mater. Sci. Eng.*, IOP Publishing, 2021, vol. 1022, Art. no. 012077.

[34] Z. Xia, J. Tan, K. Gu, and W. Jia, “Detection resource allocation scheme for two-layer cooperative IDSs in smart grids,” *J. Parallel Distrib. Comput.*, vol. 147, pp. 236–247, 2021.

[35] P. Maillé, P. Reichl, and B. Tuffin, “Of threats and costs: A game-theoretic approach to security risk management,” in *Performance Models and Risk Management in Communications Systems*. ser. Springer Optimization and Its Applications, vol. 46, New York, NY: Springer, 2011, pp. 33–54, https://doi.org/10.1007/978-1-4419-0534-5_2

[36] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, 2021, Art. no. e4150.

[37] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “A survey of network-based intrusion detection data sets,” *Comput. Secur.*, vol. 86, pp. 147–167, 2019.

[38] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *Proc. IEEE Symp. Comput. Intell. Secur. defense Appl.*, 2009, pp. 1–6.

[39] M. Ghurab, G. Gaphari, F. Alshami, R. Alshamy, and S. Othman, “A detailed analysis of benchmark datasets for network intrusion detection system,” *Asian J. Res. Comput. Sci.*, vol. 7, no. 4, pp. 14–33, 2021.

[40] “NSL-KDD dataset,” Accessed: Dec. 15, 2021. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[41] H. S. Obaid, S. A. Dheyab, and S. S. Sabry, “The impact of data preprocessing techniques and dimensionality reduction on the accuracy of machine learning,” in *Proc. 9th Annu. Inf. Technol., Electromechanical Eng. Microelectronics Conf.*, 2019, pp. 279–283.

[42] A. M. Fink, “Equilibrium in a stochastic n-person game,” *J. Sci. Hiroshima Univ., series ai (mathematics)*, vol. 28, no. 1, pp. 89–93, 1964.

[43] P. Bedi, N. Gupta, and V. Jindal, “Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network,” *Procedia Comput. Sci.*, vol. 171, pp. 780–789, 2020.

Hafsa Benaddi received the bachelor’s degree in software engineering from the Faculty of Sciences and Techniques, Moulay Ismail University, Errachidia, Morocco, in 2015, and the Master’s of Science degree in cryptography and information security from the Faculty of Sciences, Mohammed V University, Rabat, Morocco, in 2017. She is currently working toward the Ph.D. degree in mathematics, computer science, and applications with the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. She was the recipient of the Best Paper Award from Wicon’19 Conference in Taiwan. Her research interests include performance evaluation of intrusion detection systems, wireless sensor networks, and blockchain solutions.

Khalil Ibrahimi (Senior Member, IEEE) was born in Kenitra, Morocco. He received the B.Sc. degree in mathematical sciences and the M.Sc. degree in engineering, telecommunications, and multimedia from the Faculty of Sciences, Mohammed V University, Rabat, Morocco, in September 2003 and December 2005, respectively, and the Ph.D. degree in computer sciences from the University of Avignon, Avignon, France, and Mohammed V University, in November 2009. In 2010, he was an Assistant Professor (ATER part-time) with the CERI/LIA of University of Avignon. From 2010 to 2015, he was an Assistant Professor with the Faculty of Sciences, LaRIT Laboratory, IBN-TOFAIL University, Kenitra, Morocco. From 2015 to 2021, he was an Associate Professor (HdR) with the Modeling of Information and Communication Systems (MISC) Laboratory, IBN-TOFAIL University, where he was responsible of Research Team (SMCS): Security and Modeling of Communication Systems. He is currently a Full Professor with LaRI Laboratory (Research in Informatics), IBN-TOFAIL University, and responsible of Research Team (DaWNet): Data Science, Intelligent System, Wireless Network and Security. His research interests particularly include performance evaluation and resources allocation of next generation networks (3G, beyond 3G and 4G), social network, underwater sensor network (UWSN), Security and Blockchains. Dr. Ibrahimi was the Chair of RAWSN 2016 Workshop, Local Chair of WINCOM'17, Organizer of WINCOM 2020. He was the recipient of the Best POSTER Award from MSWIM 13 Conference and the Best PAPER Award from the WiCON-2019 International Conference. He was a reviewer of many international journals, such as *Computer Communications*, *Cybernetics and Information Technologies*, *International Journal of Conservation Science*, IEEE ACCESS, comcom, and MDP, and conferences Globecom, ICC, and MSWIM.

Abderrahim Benslimane (Senior Member, IEEE) received the B.S. degree in computer science from the University of Nancy, Nancy, France, in 1987, the DEA (M.S. degree) and the Ph.D. degree in computer science from the Franche-Comte University of Besanon, Besanon, France, in 1989 and 1993, respectively. Since 2001, he has been a Full Professor of computer-science with the Avignon University, Avignon, France. He is currently the Vice Dean of the Faculty of Sciences and Technology and the Head of the Master Degree SICOM, Communicating Systems. He has been nominated in 2020 as IEEE VTS Distinguished Lecturer. He has the French award for Doctoral supervision and Research during 2017–2021. Since September 1994, he has been as an Associate Professor with the University of Technology of Belfort-Montbliard, Belfort, France. He has more than 220 refereed international publications, such as books, conference proceedings, journals and conferences, and more than 20 Special issues. All publications are in his research topics. He supervised more than 20 Ph.D thesis and more than 40 M.Sc. research thesis. He was the recipient of the title to supervise researches (HDR 2000) from the University of Cergy-Pontoise, Cergy, France. He has been nominated IEEE ComSoc Steering Chair of Multimedia Communications TC during 2022–2024 and was the Vice Chair during 2020–2022. During 2017–2019, he was the past Chair of the ComSoc Technical Committee of Communication and Information Security. He is the EiC of Inderscience *International Journal of Multimedia Intelligence and Security* (IJMIS), the Area Editor of Security in IEEE INTERNET OF THINGS JOURNAL, the Editorial Member of IEEE TRANSACTION ON MULTIMEDIA, *IEEE Wireless Communication Magazine*, IEEE SYSTEM JOURNAL, *Elsevier Ad Hoc Networks*, *Springer Wireless Network Journal,* and the Past Area Editor of *Wiley Security and Privacy journal* during 2017–2019. He has been the Co-Founder and is the General-Chair of the IEEE WiMob since 2005 and iCOST and MoWNet international conference since 2011.

Mohammed Jouhari received the B.Sc. degree in physics and the M.Sc. degree in signals processing and telecommunication from the Faculty of Sciences, Mohammed V University, Rabat, Morocco, in 2011 and 2013, respectively, and the Ph.D. degree from Ibn Tofail University, Kenitra, Morocco, in 2019. He was a Postdoctoral Fellow with Computer Science and Engineering Department, Qatar University, Qatar. He is currently a Postdoctoral Researcher with the School of Computer Science, Mohammed 6 Polytechnic University, Ben Guerir, Morocco. His research interests include wireless communication, underwater acoustic sensor networks, distributed machine learning, Internet of Things, and deep reinforcement learning.

Junaid Qadir (Senior Member, IEEE) is currently a Professor with Information Technology University (ITU), Lahore, Pakistan, where he is also the Director of the IHSAN Lab that focuses on deploying ICT for development, and is engaged in systems and networking research. His research interests include the application of algorithmic, machine learning, optimization techniques in networks, wireless networks, cognitive networking, software-defined networks, and cloud computing. He was the recipient of the Highest National Teaching Award in Pakistan and the Higher Education Commission's (HEC) Best University Teacher Award (during 2012–2013). He is a Senior Member at ACM.