ADVERSARIAL MULTI-AGENT REINFORCEMENT LEARNING ALGORITHM FOR ANOMALY NETWORK INTRUSION DETECTION SYSTEM

Safa Mohamed, Ridha Ejbali*

Research Team in intelligent Machines (RTIM), National Engineering School of Gabes, University of Gabes, Gabes
Tunisia

* Corresponding Author: e-mail: *safamohamed280@yahoo.fr*

Abstract: With the rapid evolution of cyber-attacks in a dynamic environment, intrusion detection has become a difficult task. The application of new algorithms has become a necessity in detecting and classifying dangerous traffic. We propose a new Adversarial Multi-Agent Reinforcement Learning approach-based Deep SARSA algorithm (AE-Deep SARSA) that can address to solve the problem of imbalanced distribution dataset; the main contribution is to ameliorate the detection of minority classes that often account for rare instances in order to increase the performance of classifier prediction. We carried out our work using imbalanced NSL-KDD dataset that is known as NIDS benchmark datasets and this presented a classifier‘s challenge in our approach. We also validate the performance of our method by comparing with two classic machine learning methods and three related published results. The proposed model outperforms the other models in the Accuracy and F1-score metrics, it also requires less prediction time.

Key words: Anomaly detection, NSL-KDD, NIDS, deep reinforcement learning, deep SARSA.

1. INTRODUCTION

With the growing need for internet use in a wide range of applications, network services are vulnerable to various attacks. Effective security techniques and tools present a necessity to protect them. Intrusion detection is one major research problem in network security. Its aim is to identify unusual access or attacks to secure internal networks. In fact, an intrusion is any activity which affects integrity, privacy, usability and availability of information resources in order to destabilize the network. Network Intrusion Detection Systems (NIDS) is a security technology which is more

important than several other measures such as antivirus software, firewalls and access control schemes because they are intended to strengthen the security in modern computing infrastructure in order to block malicious attacks that could disrupt the functions of the system. A NIDS monitors and analyses the network traffic entering into or exiting from the network and raises alarms if abnormal activities or cyber-attacks are detected [1]. In general, NIDS can be classified into two techniques: Signature detection and anomaly detection [2].

The Signature-based NIDS (SNIDS) is used to identify attacks in a form of signature or pattern (pre-installed rules) stored in a database. It uses the known pattern to detect attacks; one of its advantages is that it rarely produces a false alarm which provides high accuracy detection rates, but it fails to identify any unknown attacks. On the other hand, Anomaly detection-based NIDS (ANIDS) attempts to build the normal behaviour of network, like usual routines of user, the sort of bandwidth generally used, the traffic volume selected, and the protocols used. The goal is to check the deviation from a set of base-line functionalities (normal traffic pattern) such as changes in transactions of the user, traffic volume, bandwidth, etc., and then, it triggers the alert system. However, the anomaly-based system gives high false positive rates, but if well suited, it can detect unknown attacks.

In the last few decades, various machine learning (ML) techniques have been used to build ADNIDS. The objective of these techniques is to classify the traffic as anomalous or normal. However, many NIDS perform a combination of feature selection techniques and traditional machine learning methods to eliminate redundant and noise features of traffics in order to ameliorate classification results [3].

Another category of machine learning methods is called deep learning (DL) which attempts to model high-level abstractions in data using deep networks of supervised and/or unsupervised learning algorithms. In order to learn from multiple levels of abstractions, it uses consecutive layers in hierarchical manners of classification [4]. It has been applied with various approaches for anomaly detection. Although, the majority of these techniques produced promising results, their limitation lies in adaptability to changes in the attack patterns and their distribution in the database which introduces difficulty in detecting intrusion. There is a need for a cost-effective NIDS that automatically learns and adapts to every change in attack pattern in the environment with the least human intervention. Hence, the appearance of a new method called deep reinforcement learning (DRL) combines excellent perception ability of DL with decision making ability of RL. It has shown excellent results in adaptive and autonomous learning for a wide range of problems in robotics, computer games, computer vision, and healthcare [5]. For example, Deep Q-network (DQN) [6] is proposed by the group of DeepMind applied to video game platforms, which successfully combines deep neural networks (DNN); a powerful nonlinear function approximation technique, with the Q-learning algorithm.

Later, a new DRL algorithm called deep SARSA learning has been introduced to find solution to control problems of video games. The experiments' results show

that deep SARSA algorithm gains better scores and faster convergence in breakout and seaquest than deep Q learning [7]. Thanks to all these benefits, we inquire the performance of DRL for anomaly detection. We apply the concept of Adversarial Multi Agent Reinforcement Learning using Deep SARSA algorithm (AE-deep SARSA) for anomaly-based NIDS.

This work aims to solve the problem of rarity and imbalanced classes in a dataset. Our objective is to increase the detection accuracy of these instances of anomalies or attacks. For this reason, we use the NSL-KDD [8], which is a very unbalanced dataset to improve performance on minority classes.

Our contributions are summarized as follows:

- To propose an Adversarial Multi-Agent Reinforcement Learning algorithm for anomaly-based NIDS suitable for a supervised classification problem that used the labelled dataset in a DRL framework.
- To present a simple, fast and flexible classifier for prediction.
- To apply, for the first time, deep SARSA algorithm to address the training bias associated with an unbalanced dataset, in order to better detection of minority classes.
- Perform a comparison of the results obtained of our proposed model with other models which used NLS-KDD dataset.

The remaining part of this paper is organised as follows: Section 2 presents related works. Section 3 describes the work performed. Section 4 shows and compares the results. And finally Section 5 offers a conclusion and suggests future works to be adopted later.

2. RELATED WORKS

In this section, we discuss some of the related works that have been widely used in anomaly detection and classification applied mainly to NSL KDD dataset. We also propose a detailed explanation of the algorithms needed to apply in RL and DRL.

2.1. Anomaly Detection

Anomaly detection is an active area of research that involves an important number of techniques. Machine learning is one of the most popular methods that can be classified according to availability of labels into supervised, unsupervised and semi-supervised methods [9]:

- Supervised anomaly detection: also called classification method, requires a labelled dataset for the training of the model that refers to the normal and anomalous instances to build the predictive model.
- Unsupervised anomaly detection: contains unlabelled data instances.
- Semi-supervised anomaly detection: the trained model contains instances only available for the normal behaviour or normal class, then the test is used to measure the deviation compared with this model, so, any deviation is thus marked anomalous.

The classical ML models that have been widely applied for ANIDS were the supervised and unsupervised models, which are currently areas of active research. Da Costa et al. [10] and Dhanabaland Shantharajah [11] study the effectiveness of the many classification algorithms and also analyse the relationship of the protocols available in the commonly used network protocol. Thomas and Pavithran [12] have surveyed the various machine learning researches conducted for anomaly-based intrusion detection using the NSL-KDD data set. Other authors study the impact of techniques for selection of relevant characteristics for classification results. For this reason, Ibraheem et al. [13] proposed an intrusion detection approach that used PCA to select the best feature of traffic and MPL to classify the normal and malicious connections.

In order to provide detection with very high performance rates in all attacks, Çavuşoğlu [14] implemented a hybrid system that combines different machine learning and feature selection techniques. It is necessary to prove the increasing use of the deep learning models. Berman et al. [15] presented an excellent review of the application of MLPs, auto encoders, recurrent networks and restricted and Boltzmann machines with different results in prediction accuracy. Mohamed et al. [16] attempted to develop the Denoising-Autoencoder with Dropout based network anomaly detection method for improving intrusion detection. Imamverdiyev et al. [17] built three deep learning and three classic machine learning architectures for network attacks detection problem. Experimental results show that the Bernoulli-Bernoulli (RBM) produced the higher accuracy rate of 73.23 %.

Some machine learning models could be vulnerable to manipulation; this risk is called adversarial machine learning. It is the process in which an adversary attempts to infuse malicious data into the learning algorithm in order to deceive a model to force misclassification [18]. Adversaries can be used in two popular attack methods: poisoning or contaminating attack and evasion attack [19]. The first, aims to modify a part of the training data causing the system to make inaccurate classification decisions. On the contrary, the second assumes that the model has already been trained, so it aims at varying its behaviour to cause the model to make erroneous predictions. In reinforcement learning, an adversarial environment is proposed, whose goal is to establish a classifier that introduces mistakes by performing small perturbations to the training data [20]. Many researchers study the impact of the adversarial learning problem in NIDS [18].

Other researchers started using RL for anomaly detection. In 2007, Servin [21] used a reinforcement learning model with a simulated network environment for Distributed Denial of Service Attacks detection (DDOS). Servin [22] proposed a novel approach to train Multi-agent Reinforcement Learning (MARL) agents to cooperate for detection of DOS and DDOS attacks. Nguyen et al. [23] presented an efficient review of DRL for Cyber security to real or simulated environment.
Caminero et al. [24] also built a first multi-agent adversarial reinforcement learning approach (AE-RL). The method is adequate for adjusting the RL to an IDS intrusion classification problem. The proposed model outperforms the other well-known

machine learning models in the weighted accuracy and F1 score. Suwannala et al. [25] proposed Adversarial/Multi Agent Reinforcement Learning using Deep Q-Learning (AE-DQN) algorithm for anomaly-based NIDS. This algorithm exhibits superior performance in detecting certain types of attacks in NSL-KDD dataset compared to others models.

Despite all the improvements, the minority classes are the least detected, thus we need to try to find a balance for all the classes in order to increase accuracy detection.

2.2. Deep reinforcement learning method based on SARSA algorithm

Reinforcement learning (RL) is a branch of machine learning which is different from supervised learning, where an agent autonomously interacts with the learning environment and optimal policy by trial and error in order to maximize the total accumulated rewards. SARSA (State-action-reward-state-action) [26] is an on-policy reinforcement learning algorithm. It means that when updating the current Q value, the next action a' will be taken. Also, the training data are quintuple <s,a,r,s$'$, a'> that will be derived in sequence every update process. The update rule for the Q-value is as follows:

$$Q^{new}(s,a) \leftarrow \mathrm{Q}(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)] \quad (1)$$

Where $\mathrm{Q}^{\mathrm{new}}(s,a)$ refers the new Q-value after the adjustment; $\mathrm{Q}(s,a)$ presents the current Q-value; s,a, r are current state, action and reward;s', a' are the next state and action; γ is the discount factor and α is the learning rate controlling the adjusting size;

In deep reinforcement learning, deep SARSA algorithm used $Q(s,a,\theta)$ as the approximate Q-function with a parameter θ. The parameters θ are the weights of the neural network. It tackles the aforementioned instability and diverge issue by experience, replay and target network. Experience replay smoothingout learning and avoiding oscillations or divergence in the parameter. A target network k is a DNN which has the same structure with the primary network (or value network) yet updates less frequently. A Q-network can be trained by minimising a sequence of loss function $L_i(\theta_i)$ that changes at each iteration i.

$$L_i(\theta_i) = (y_i - Q(s,a,\theta_i))^2 \quad (2)$$

Where $y_i = r + \gamma Q(s',a',\theta_{i-1})$, that refers to the target network for iteration i.Since SARSA is on-policy, it should be noted that during training, the next action a$'$ for estimating estimates the current state-action value is never greedy, it is chosen using the same ε-greedy policy as the action a the one that generated s'[7]:Then the main goal is to optimize the loss function $L_i(\theta_i)$. We obtain the gradient of the loss function by differentiating (2):

$$\nabla L_i(\theta_i) = \big(r + \gamma Q(s',a',\theta_{i-1}) - Q(s,a,\theta_i)\big)\nabla Q(s,a,\theta_i) \quad (3)$$

The Deep SARSA algorithm is trained using the CNN using ε-greedy policy shown in Algorithm 1 [7].

3. PROPOSED METHOD

In this paper, our approach is to implement a new multi-agent adversarial reinforcement learning approach-based deep SARSA algorithm in order to gain higher scores to produce better minority attack categories classification in NSL – KDD dataset. The new framework obtained by the previous modifications allows applying a well-known RL algorithm [7] to classify intrusions using a dataset of pre-recorded intrusion data. We study the performance of our model to compare it with two classic machine learning methods (MLP, CNN-1D) and three literature methods such as RBM, mentioned in [17], AE mentioned in [24] and AE-DQN in [25] which has used the imbalanced NSL-KDD dataset for their performance benchmarking. The detailed development process of our approach is provided in the following sub-section.

3.1. NSL-KDD Dataset

NSL-KDD was an improved version of the KDD Cup 99 dataset [11]. Although NSL-KDD may not be a perfect representation of the real network, it is still an effective benchmark dataset admitted in the NIDS research field. One interesting aspect of NSL-KDD is that it contains rare and imbalanced distribution class in the test and training sets which is an important challenge for the classification. The NSL-KDD dataset includes 125973 training network traffic samples and 22544 test network traffic samples, each record containing 41 attributes unfolding different features of the flow and a label is assigned to each sample either as a normal type or an attack type. These 41 attributes are composed of 38 continuous and 3 categorical (discrete valued). These features have required a pre-processing operation namely normalization, it is used to scale the continuous features to the range [0–1] and one-hot encoding the categorical features. Finally, 122 features are obtained, 38 continuous and 84 with binary values (0, 1) associated to the three one-hot encoded categorical features. The dataset is grouped into five major classes: Normal and four simulated attacks that regroup Probe (Probing attacks), DoS (Denial of Service attacks), U2R (User to Root attack) and R2L (Root to Local attacks) (see Table 1). More details description of the NSL-KDD dataset is provided in [24].

Table 1.Classes distribution in the NSL-KDD [27]

Class	***Dataset***	
	KDD Train+	***KDD Test+***
Probe	*11656*	*2421*
DoS	*45927*	*7460*
R2L	*995*	*2885*
U2R	*52*	*67*
Normal	*67343*	*9711*
Total	*125973*	*22544*

3.2. Methodology

We propose a novel model called AE-Deep SARSA that incorporates a supervised problem which makes use of a labeled dataset into a DRL framework (based on interaction with a simulated environment). In this model, two agents (Environment Agent and Classifier Agent) are trained in parallel using deep SARSA and work in an adversarial model. In the adversarial configuration, the first agent (Environment) tries to augment the complexity of the prediction made by the second agent (Classifier) in order to increase their final performance.

Fig. 1 presents our framework. Our NSL-KDD dataset requires a pre-processing (one hot encoding and normalization) explained above before implementing and testing our model.

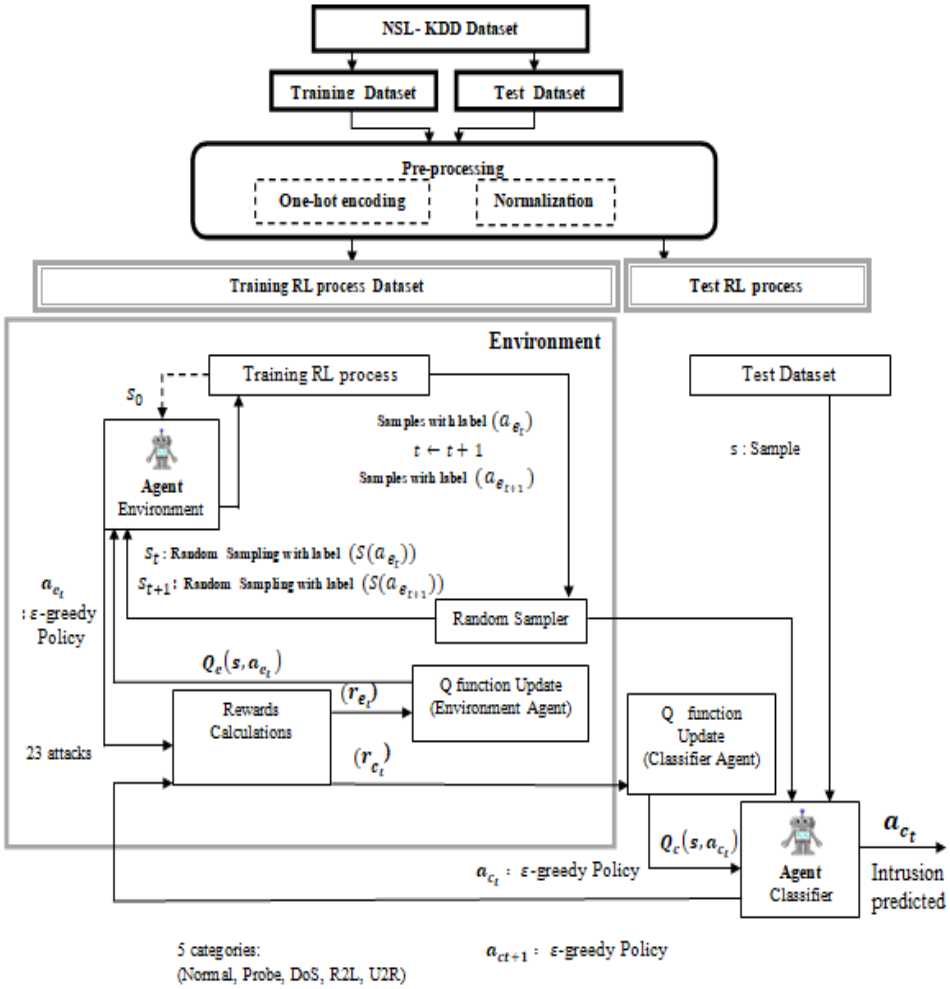

Fig. 1. Proposed AE- deep SARSA structure based NIDS model

Later, two separate processes refer to the training and test or prediction stages. In the training process, the defender agent (Classifier Agent) implements a classifier whose action tries to predict the intrusion label (5 categories: Normal, Probe, DoS, U2R and R2L) from the states (network traffic samples) given in input (see Fig.2).

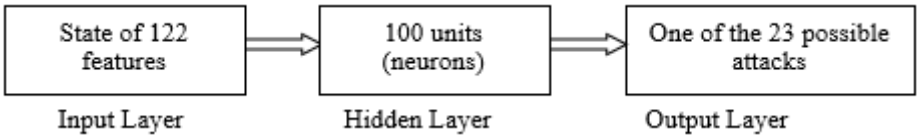

Fig. 2. Architecture for the Environment Agent (attacker agent)

The choice of action is based on ε- greedy policy with the best action is selected with probability ε or a random action with probability 1-ε. On the contrary, the action of attacker agent (Environment Agent) will refer to the class of attack (23 attacks types) used for the type of the next line of data for training based also on ε-greedy exploration (see Fig.3). The two agents received opposite amount of reward from the environment that is depending on prediction. If the prediction of agent defender is correct, it receives a value of +1 and the agent attacker receives a value of -1, otherwise, it gets a negative reward for the Agent Classifier and gets a positive reward for the agent Environment. In the test process, consisting of a given random sample from the test dataset, the goal is to predict the intrusion label result of the action that refers to one of the 5 class classification.

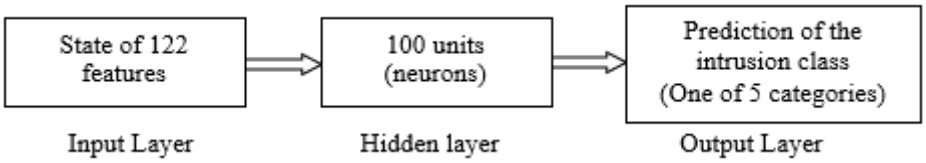

Fig. 3. Architecture for the Classifier Agent (defender agent)

Algorithm 1 illustrates detailed stages of our model; arbitrarily initializes the Q values of the environment and the classifier agents. For every episode, a random state of the dataset s_0 is selected in order to supply the Q function. Then, from the current state, the environment selects an action a_{e_t} based ε-greedy policy. According to this action, the environment chooses the current state randomly from the dataset s_t where the label is a_{e_t}, after the agent classifier classifies this state based ε-greedy policy. This action a_{c_t} is transmitted to the environment, then it will be compared with the truth label. Meanwhile, the value of reward r_{ct} is given and that depends on the correct classification. This value is opposite to the agent environment r_{et}, also the environment produced the next state s_{t+1}. After the environment agent replaces s_{t+1} by one of random samples where the label is a_{et+1}, the agent classifier

classifies this state based on the same policy and attribute of the action a_{ct+1}. Finally, it updates the Q function of the classifier and the environment agents based on a quadratic loss function according to the deep SARSA algorithm.

Algorithm 1 : Proposed AE-Deep SARSA algorithm

$Q_c(s, a_{c_t})$: Arbitrarily initialize
$Q_e(s, a_{e_t})$: Arbitrarily initialize
For episode do
Initialize state s_0 = A random sample of dataset;
Environment Agent: choose action a_{e_t} using ε-greedy policy derived from $Q_e(s_t, a_{e_t})$
Replace s_t by one of random samples where label is a_{e_t}
Repeat for each time step t= 1, M
Classifier Agent: choose action a_{c_t} using ε-greedy policy derived from $Q_c(s_t, a_{c_t})$
Obtain $(r_{ct}, r_{et}, s_{t+1})$
Environment Agent: choose next action a_{et+1} using ε-greedy policy derived from $Q_e(s_t, a_{e_t})$
Replace s_{t+1} by one of random samples where label is a_{et+1}
Classifier Agent: choose next action a_{ct+1} using ε-greedy policy derived from $Q_c(s_t, a_{c_t})$
Perform a gradient descent step on:
$(r_{et} + \gamma Q_e(s_{t+1}, a_{et+1}) - Q_e(s_t, a_{e_t}))^2$
Perform a gradient descent step on:
$(r_{ct} + \gamma Q_c(s_{t+1}, a_{ct+1}) - Q_c(s_t, a_{c_t}))^2$

The algorithm progresses with a simple neural network with one layer and 100 units trained with 100 episodes and the number of iterations for episode is equal to 100. Since, the states are not correlated with each other, the discount factor is close to zero ($\gamma = 0.001$) and the learning rate is ($\alpha = 0.2$).

These two-agents progress along with a decreasing ε-greedy policy starting with a high exploration that decreases along the episodes. The important point in this algorithm is that it is based on policy method, which used the same policy as that used to make decisions during learning.

4. RESULT

In this section, we evaluate the performance of different models such as the AE-deep SARSA algorithm, MPL with 3 hidden layers (512, 128 and 64 node) and the one-dimensional deep CNN (CNN 1-D), in addition to three related published results. We developed our NIDS model using Python programming language following different metrics (see Table 2) and to the confusion matrix. Considering very unbalanced dataset, F1-score becomes a better metric for prediction performance than the others.

Table 2.Evaluation metrics

Metric	***Formula***
Accuracy	$A = \frac{TP + TN}{TP + TN + FP + FN}$
Recall	$R = \frac{TP}{(TP+FN)} \times 100\%$
Precision	$P = \frac{TP}{(TP + FP)} \times 100\%$
F1-score	$\mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{(\mathrm{P} + \mathrm{R})}$

TP (True Positive): represents an Attack datum that is correctly classified as an attack.
FP (False Positive): is the equivalent of normal datum that is incorrectly classified.
TN (True Negative): is a normal datum that is correctly classified as normal.
FN (False Negative): refers to an attack datum that is incorrectly classified as normal

Fig. 4 represents the performance of our proposed AE-deep SARSA algorithm in terms of true positive, false negative and false positive at 100 episodes. We can see that the proposed algorithm produced strong performance in the detection of normal sample, DoS and Probe attacks, it also offers a reasonable performance in the detection of R2L attacks and U2R attacks despite their presence in reduced numbers of sample data.

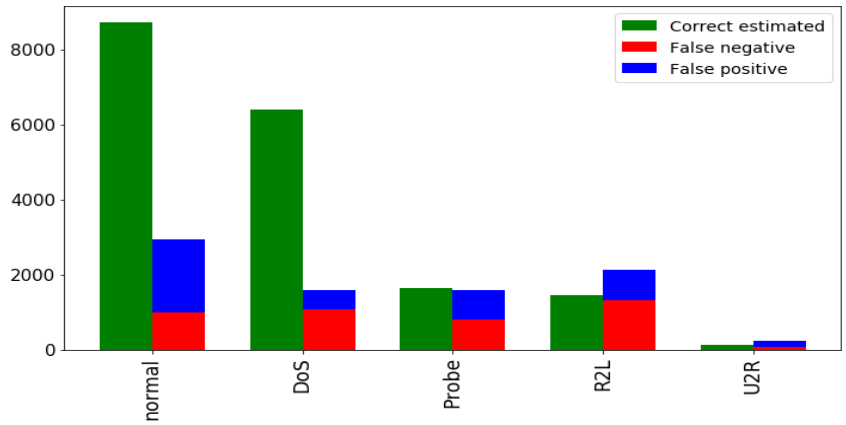

Fig. 4. Our AE-deep SARSA model performance

Table 3 illustrates multi-classification performance for our study (AE-deep Sarsa, CNN-1D and MLP) applied to the NSL-KDD dataset. The experimental results show that the three models produce high accuracy value for the different labels (Normal,Dos, Probe, U2R and R2L). However, the proposed AE-deep Sarsa is the only one that provides the best result of f1-score, precision and recall metrics especially of minority classes (R2L, U22R).

Table 3.Multi-classification performance scores of AE-deep SARSA, CNN-1D, MLP models

	Accuracy		
Class	***AE-deep SARSA***	***CNN-1D***	***MLP***
Normal	*0.870165*	*0.868967*	*0.834413*
DoS	*0.929249*	*0.933685*	*0.91682*
Probe	*0.929604*	*0.928407*	*0.923749*
R2L	*0.905873*	*0.861471*	*0.889682*
U2R	*0.989842*	*0.980837*	*0.972498*
	F1-score		
Class	***AE-deep SARSA***	***CNN-1D***	***MLP***
Normal	*0.856301*	*0.856477*	*0.824503*
DoS	*0.889059*	*0.893146*	*0.865331*
Probe	*0.673255*	*0.677716*	*0.652376*
R2L	***0.57577***	*0.372009*	*0.413582*
U2R	***0.52588***	*0.191011*	*0.124294*
	Precision		
Class	***AE-deep SARSA***	***CNN-1D***	***MLP***
Normal	*0.818335*	*0.810856*	*0.75863*
DoS	*0.923688*	*0.956375*	*0.931787*
Probe	*0.671182*	*0.655972*	*0.639065*
R2L	***0.640285***	*0.416667*	*0.589382*
U2R	***0.640285***	*0.152695*	*0.0866142*
	Recall		
Class	***AE-deep SARSA***	***CNN-1D***	***MLP***
Normal	*0.897961*	*0.907537*	*0.902904*
DoS	*0.856932*	*0.837758*	*0.807723*
Probe	*0.675341*	*0.837758*	*0.666254*
R2L	***0.523066***	*0.335997*	*0.318562*
U2R	***0.635***	*0.255*	*0.22*

From Table 4, we can see that the high values of the aggregated accuracy metrics (0.8124) and F1-score (0.8103) are obtained by AE-deep SARSA algorithm

compared to the CNN-1D and MLP algorithms and the three related published results (RBM, AE-RL and AE-DQN). In terms of execution time, CNN-1D has a longer runtime, longer than that of the models (MLP and AE-RL). In contrast, our AE-deep SARSA spends less runtime (0.46 seconds) thanks to the characteristics of the classifier used (simple, fast and flexible classifier).

Table 4.Performance metrics of different models

	Our study			***Others Works***		
	Proposed AE-deep SARSA	***CNN-1D***	***MLP***	***RBM [17]***	***AE-RL [24]***	***AE-DQN [25]***
Accuracy	*0.8124*	*0.7867*	*0.7686*	*0.7323*	*0.8016*	*0.80*
F1-score	*0.8103*	*0.7843*	*0.7631*	*0.7530*	*0.7940*	*0.79*
Prediction time (Second)	*0.46*	*1.64*	*1.03*	-	*0.50*	-

Looking at the confusion matrix (Fig.5, Fig.6, Fig.7, Fig.8) of the AE- deep SARSA model for the NIDS dataset, compared to two well-known algorithms and one of three related published, we can see that the number of correct estimates for two minority classesR2L and U2R have a significant increase for our proposed AE-deep SARSA algorithm. Then; the rest of the models present a low detection for these two classes. So, AE-deep SARSA tries to improve the classification of the less frequent classes that reduce the false negatives for these less frequent classes.

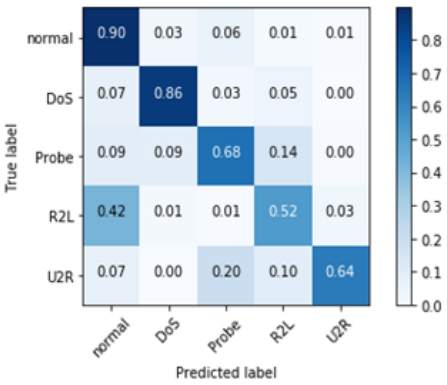

Fig. 5. Confusion Matrix for AE-deep SARSA

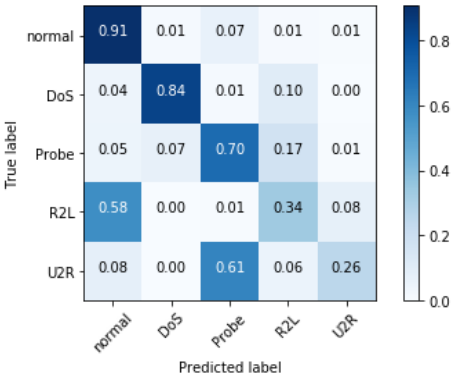

Fig. 6.Confusion Matrix for CNN-1D

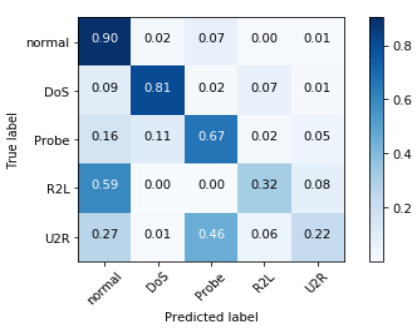

Fig. 7.Confusion Matrix for MLP

Actual Class \ Predicted Class	Normal	DoS	Probe	R2L	U2R
Normal	**0.92**	0.01	0.07	0.00	0.00
DoS	0.05	**0.86**	0.02	0.07	0.01
Probe	0.05	0.10	**0.84**	0.01	0.00
R2L	0.66	0.00	0.05	**0.24**	0.05
U2R	0.41	0.00	0.36	0.03	**0.19**

Fig. 8.Confusion Matrix forAE-DQN Model [25]

5. CONCLUSION

NIDS is a critical service that ensures the security of network. The main objective is to adapt to the detection of various types of attacks. Difficulties have been imposed and were caused by unbalanced and complex dataset. Different methods in the existing literature especially deal with this problem, but do not achieve the best performance. In this paper, we attempted to develop an AE-deep SARSA algorithm for intrusion detection that integrates on adversarial Reinforcement learning and supervised models. The strong point of this proposed new model is that it gives a high prediction performance of the two minority attacks class (U2R, R2L) with a reasonable runtime.

As part of future work, we propose to evaluate our model with other datasets and investigate how to combine off-policy deep Q-learning (DQN) with on-policy deep SARSA algorithm [28].

Acknowledgment. The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

REFERENCES

[1] Vartouni, A. M., Kashi, S. S., Teshnehlab, M.An anomaly detection method to detect web attacks using Stacked Auto-Encoder, *6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 2018. pp. 131-134.

[2] Rhodes, B. C., Mahaffey, J. A., Cannady, J. D. Multiple self-organizing maps for intrusion detection, In: *Proceedings of the 23rd national information systems security conference*, 2000. pp. 16-19.

[3] Qu, F., Zhang, J., Shao, Z., Qi, S. An intrusion detection model based on deep belief network, In: *Proceedings of the 2017 VI International Conference on Network, Communication and Computing*, December, 2017.pp. 97-101.

[4] Lin, W. H., Lin, H. C., Wang, P., Wu, B. H., Tsai, J. Y. Using convolutional neural networks to network intrusion detection for cyber threats, In: *International Conference on Applied System Invention (ICASI)*, 2018. pp. 1107-1110.

[5] Li, H., Zhang, Q., & Zhao, D. Deep reinforcement learning-based automatic exploration for navigation in unknown environment, *IEEE transactions on neural networks and learning systems*, vol. 31, No. 6, 2019, pp. 2064-2076.

[6] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. *arXiv preprint* arXiv:1312.5602, 2013.

[7] D. Zhao, H. Wang, K. Shao, et al., Deep reinforcement learning with experience replay based on SARSA, *IEEE Computational Intelligence*, 2017.

[8] M. Tavallaee et al., A detailed analysis of the KDD CUP 99 data set, *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA)*, 2009, pp 53–58.

[9] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., Han, K. Enhanced network anomaly detection based on deep neural networks., *IEEE Access*, Vol. 6, 2018, pp. 48231-48246.

[10] da Costa, K. A. P., Papa, J. P., de Oliveira-Lisboa, C., Munoz, R., & de Albuquerque, V. H. C. Internet of things: A survey on machine learning-based intrusion detection approaches. *Computer Networks*, vol. 151, 2019, pp. 147–157. doi: 10.1016/j. comnet.2019.01.023.

[11] Dhanabal, L., &Shantharajah, S. P. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, No. 6, 2015, pp. 446-452.

[12] Thomas, R., &Pavithran, D. A survey of intrusion detection models based on nsl-kdd data set, In *2018 Fifth HCT Information Technology Trends (ITT)*, November, 2018. pp. 286-291, IEEE.

[13] Ibraheem, N. B., Jawhar, M. M., Osman, H. M. Principle Components Analysis and Multi Layer Perceptron Based Intrusion Detection System, *AL-Rafidain Journal of Computer Sciences and Mathematics*, Vol. 10, 2013, pp.127-135.

[14] Çavuşoğlu, Ü. A new hybrid approach for intrusion detection using machine learning methods, Applied Intelligence, Vol. 49, 2019, pp. 2735-2761.

[15] Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbett, C. L. A survey of deep learning methods for cyber security. Information, Vol. 10, No. 4, 2019, pp. 122. doi: 10.3390/ info10040122.

[16] Mohamed, S., Ejbali, R., &Zaied, M. DenoisingAutoencoder with Dropout based Network Anomaly Detection, *ICSEA*, 2019, 110.

[17] Y. Imamverdiyev, F. Abdullayeva, Deep learning method for denial of serviceattack detection based on restricted boltzmann machine, *Big Data*, Vol. 6, No. 2, 2018, pp. 159–169.

[18] PACHECO, Yulexis et SUN, Weiqing. Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets, *ICISSP*, 2021. pp. 160-171.

[19] Alhajjar, E., Maxwell, P., & Bastian, N. D. Adversarial machine learning in network intrusion detection systems. *arXiv preprint* arXiv:2004.11898.

[20] S. Huang et al., Adversarial attacks on neural network policies, arXiv: 1702. 02284 [cs.LG], 2017.

[21] A. Servin, *Towards Traffic Anomaly Detection via Reinforcement Learning and Data Flow*, Department of Computer Science, University of York, United Kingdom, 2007.

[22] A. Servin, *Multi-Agent Reinforcement Learning for Intrusion Detection*, Ph.D. thesis, University of York, 2009.

[23] Nguyen, T.T., &Reddi, V.J. Deep reinforcement learning for cyber security, *arXiv*: 2019,1906.05799 [cs.CR].

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

[25] Suwannalai, Ekachai, and ChantriPolprasert. Network Intrusion Detection Systems Using Adversarial Reinforcement Learning with Deep Q-network, 1*8th International Conference on ICT and Knowledge Engineering (ICT&KE)*, 2020, IEEE.

[26] Rummery, G., &Niranjan, M. *On-line Q-learning using connectionist system*s (Technical Report CUED/FINFENG-TR 166), 1994, Cambridge University, UK.

[27] Defazio, A. and T. Graepel, A comparison of learning algorithms on the Arcade Learning Environment, 2014, *arXiv preprint* arXiv:1410.8620.

[28] XU, Zhi-xiong, CAO, Lei, CHEN, Xi-liang, et al. Deep reinforcement learning with sarsa and q-learning: A hybrid approach. *EICE TRANSACTIONS on Information and Systems*, Vol. 101, No 9, 2018, pp. 2315-2322.

Information about the authors:

Safa Mohamed–PhD Student at the National Engineering School of Gabes, University of Gabes, Tunisia. Member of Research Team in intelligent Machines (RTIM). Research interests include intrusion detection systems, information security, machine learning and deep learning.

Ridha Ejbali – Associate Professor at the Faculty of Sciences of Gabes, University of Gabes, Tunisia. Member of Research Team in intelligent Machines (RTIM). Research interests include machine vision, machine learning, deep learning, pattern recognition, classification and information security.

Manuscript received on 14 July 2021