Towards An Unsupervised Reward Function For A Deep Reinforcement Learning Based Intrusion Detection System

Bilel Saghrouchni, Frédéric Le Mouël, Bogdan Szanto

Towards An Unsupervised Reward Function For A Deep Reinforcement Learning Based Intrusion Detection System

Bilel Saghrouchni[1,2], Frédéric Le Mouël[1], and Bogdan Szanto[2]

[1]*INSA Lyon, Inria, CITI, UR3720, 69621 Villeurbanne, France*
[2]*SPIE ICS, 69500 Bron, France*

***Abstract*—Intrusion detection systems (IDS) based on deep learning have proven successful, but struggle to learn continuously and detect new attacks over time due to a supervised label-based reward function. In this article, we introduce an unsupervised Deep Double Q Learning (DDQL) method that aims to detect attacks and learn new behaviors through an unsupervised reward function leveraging a normality score inspired by car traffic anomaly detection.**

***Index Terms*—Intrusion Detection System, Deep Reinforcement Learning, Clustering, Unsupervised**

I. Introduction

Intrusion Detection Systems (IDS) are now considered essential tools. They analyse the traffic to detect malicious behaviours. Over the decades, numerous approaches have been adopted for IDS development, starting with signature-based methods.. Despite being efficient and widely used for years, patterns that slightly differ from the ones stored in the database were not detected. It also involves managing a database and updating it regularly. Anomalies-based IDS leverage Machine Learning (ML) and Deep Learning (DL) to detect more complex patterns within the traffic [9] [7] [13]. They have demonstrated significant potential in discerning behavioural variations and identifying attacks. Nevertheless, their ability to detect unfamiliar patterns remains limited [18], and the computational and storage complexities associated with regular learning pose considerable difficulties.

Reinforcement Learning (RL) offers a compelling solution to these constraints with the potential to detect and classify attacks [1] [14] [10]. In RL, an agent refines its policy by making decisions and receiving rewards or penalties based on responses of the environment via a reward function (Figure 1). However, RL faces challenges in real-world scenarios with extensive data volumes and large state spaces [10]. Recent works have shown that Deep Reinforcement Learning (DRL) methods, which leverage neural networks are capable of enhancing the robustness and the security of a network working alone without any human intervention [1]. Unfortunately, very little attention is paid to the reward mechanism and many works on DRL-based IDSs analysing network traffic use a supervised reward function that is often label-based, which prevents the agent from detecting new patterns and training continuously in a real-time environment where the labels are unknown [15] [10].

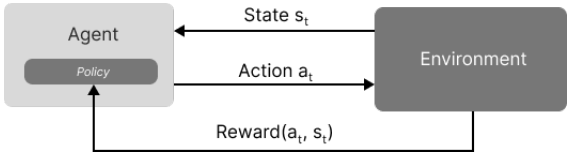

Fig. 1: Reinforcement Learning process

The main goal of our work is to take full advantage of the learning capabilities of reinforcement learning and implement an unsupervised IDS that can be used in real-world conditions and then deal with unknown patterns. In this article, we present an analysis of two well-known intrusion detection datasets, focusing on the features that define network flows. In addition, we introduce a clustering-based method that leverages flow properties to develop an unsupervised reward function.

The paper is organized as follows: Section 2 details the two datasets used for our experiments, including the preprocessing steps. In Section 3, we delve into the application of DRL to intrusions detection and describe the core components of the algorithm. Section 4 is dedicated to the analysis of flow features, which are essential pieces of information on which the agent base its decisions. We introduce our unsupervised reward function in Section 5 followed by a discussion of potential directions for future research in Section 6.

II. Datasets

For our experiments, we chose the NSL-KDD [12] and CICIDS17 [11] datasets, both well-known and covering a wide range of attacks. This choice was made to evaluate our method across a broad spectrum of attacks, rather than restricting it to a specific subset, while allowing comparison with previous studies.

Obviously, before using NSL-KDD and CICIDS17 we need to prepare them to fit the data to our model and reduce computational time. In alignment with common data preparation in DRL studies [17] [1] [10], we apply the following operations:

Dataset	NSL-KDD	CICIDS17
Data Model	Network flows	Network flows
Features	41 (38 continuous, 3 categorical)	81 (68 continuous, 13 categorical)
Total Records	125,973 (train), 22,543 (test)	2,830,743
Labels	23 train, 38 test	15 grouped in 7 categories
Main Classes	Normal, DOS, Probe, R2L, U2R	Normal, Brute Force, DoS, DDoS, Web, Infiltration, Botnet
Protocols	TCP, UDP, ICMP	HTTP, HTTPS, FTP, SSH, Email, etc.
Split	Predefined train/test	70% train, 30% test

TABLE I: Comparison of NSL-KDD and CICIDS17 datasets

- **Encoding:** Categorical features are transformed into numerical ones using Label Encoding and One Hot Encoding.
- **Normalization:** Each feature is normalized by subtracting the mean and dividing by the standard deviation.

III. Deep Reinforcement Learning for Intrusions Detection

Although machine learning and neural networks are frequently utilized for classification tasks, leveraging DRL to tackle these issues presents challenges. Nonetheless, this approach offers key benefits. The reward function contributes to a better control and interpretability during the training phase. Then, the agent acquires the ability to take decisions and adapt in dynamic environments, which is an essential aspect of scenarios involving anomaly detection.

The article focuses on the main components of DRL and explains how our unsupervised reward function becomes integral to the training process.

A. Q-network

In a Deep Q-learning algorithm, the Q-network has an important goal as it aims to approximate the Q-value function. In our proposition we use a simple Feed Forward Neural Network (FFNN) of two hidden layers with ReLU activation for all layers. The decision to use a FFNN was guided by its popularity in DRL studies for intrusion detection and similar problems [8] [3], its straightforwardness and the nature of the environment. To achieve a more stable learning phase and faster convergence, we use two Q-Networks through the Double Deep Q-Learning (DDQL) [19].

B. Environment and state

As in many RL-based IDS, the environment is simulated by sampling a dataset containing flows [1] [17] [14]. Each flow is characterized by a fixed number of features (22 for NSL-KDD, 27 for CICIDS17) representing various aspects of the environment at a given time t. The features of several flows will constitute a state which is a subgroup of flows.

C. Actions

The actions aims to classify network traffic. In our approach, we opted for binary classification with labels of 'attack' and 'normal'. The motivation for this choice is to use a fixed neural network architecture. If multiple attack labels were considered, introducing a new attack would necessitate adding a new output neuron, thereby modifying the neural network's architecture and it is not the focus of this article.

D. Reward function

The reward is the feedback of the environment to an action taken by the agent. It's a major component because it determines the quality of agent's actions and it influences the evolution of the decision policy. In DRL based IDS, most of the reward functions are supervised (Equation 1) and give the agent a good reward if its prediction is equal to the label, and a bad one otherwise [1] [10] [14].

$$\text{Reward}(s_t, a_t) = \begin{cases} -1, & \text{if } a_t \neq l_t \\ 1, & \text{if } a_t = l_t \end{cases} \quad \text{with } l_t \text{ the label for } s_t \tag{1}$$

This method imposes limits on the circumstances in which the agent can operate and reduces its capabilities. Without labels associated with each flow, the agent is not capable of learning more and it becomes unable to detect unknown attacks and behaviours. This motivates our research on an unsupervised reward function.

IV. Features Selection and Analysis

To make the DRL agent converge, the states it observes has to provide the most relevant information that it can use to take a decision. A naive approach could be to incorporate each feature of a flow (Table I) as a state characteristic. However, this

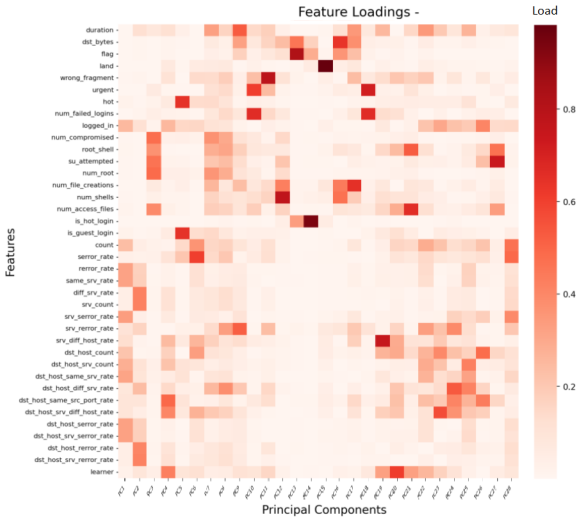

Fig. 2: NSL-KDD's PC explained

could lead to high-dimensional states populated with irrelevant features that do not contribute to efficient intrusion detection and could increase learning times. To overcome this problem, once flow data is formatted conveniently, we apply Principal Component Analysis (PCA) as suggested in [16] keeping dimensions representing 97% of the variance. Then we end-up with 22 principal components for the NSL-KDD dataset and 27 for the CICIDS17 dataset. These principal components (PC)

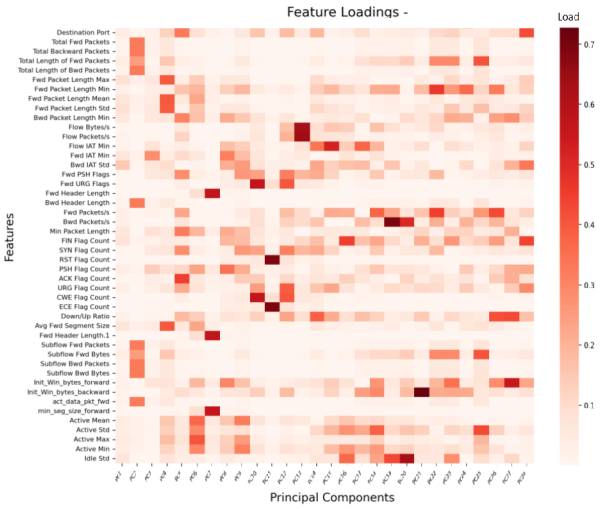

Fig. 3: CICIDS17's PC explained

result from linear combinations of the original dataset features. States are now described using these new components. In the Figure 2 and 3, we detailed how the PC are built and which features they use. For the NSL-KDD dataset (Figure 2) indicates that error-rate features are critical for anomaly detection. In the CICIDS17 dataset (Figure 3), features based on packet characteristics — such as length, size, and temporal properties — emerge as the most significant indicators of intrusive activities. This analysis helps us gain better insight into which features provide the most information about a flow and what the agent observes to make decisions.

V. Towards an unsupervised reward function

The reward function is a key component of the DRL algorithm as it steers learning and future actions. The way we design it can have a significant impact on the performance of the agent whether in terms of correctness or convergence time [6].

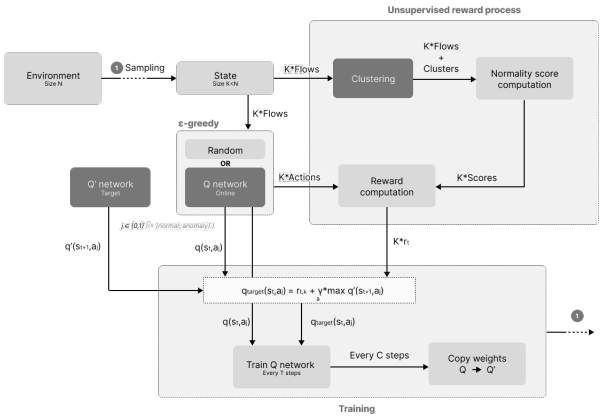

Fig. 4: Architecture of the unsupervised reward function within the Double Deep Q-Learning algorithm

In intrusion detection problems using reinforcement learning, the reward is often a simple comparison between the labels and the agent's actions. The challenge is maintaining this behavior in cases where ground-truth labels are unavailable. Research in anomaly detection for uni-dimensional vehicular traffic flows has led to the development of a clustering-based reward function, referenced in [4]. Our ongoing research is focused on utilizing clustering algorithms to calculate the rewards. At every iteration, individual network flows are aggregated into clusters to determine a normality score based on the attributes of their respective clusters. This calculated score reflects the 'normality' of each network flow and is then employed within the reward function. This approach informs the learning agent's behavior, steering it toward accurate decision-making in an unsupervised way.

Achieving our objective necessitates overcoming multiple challenges. Foremost, the selection of an appropriate clustering algorithm is crucial; it must demonstrate efficacy in distinguishing between the various classes of network flows present in IDS datasets. Clustering algorithms are diverse [20] and can be classified into five main categories: hierarchical, partitioning, density-based, grid-based and model-based [21] [2]. Each category presents distinct advantages and challenges, and our research is currently focused on identifying the clustering algorithm best suited to our specific application. This selection involves consideration of factors such as the expected shape of the clusters, the necessary parameters involved and the computational complexity of the algorithms.

In addition, we will concentrate on refining a dimensionality reduction algorithm tailored to preserve the most salient features of network flows, thereby enhancing the efficacy of clustering operations.

VI. Conclusion and future works

In this paper, we presented DDQL as a promising methodology for intrusion detection in network systems, highlighting its potential to detect and to learn previously unknown attacks. We propose that the use of an unsupervised reward function can facilitate the deployment of agents in a real-life scenario. Nevertheless, this work has highlighted several areas for further exploration.

The clustering algorithm is a key component of the reward function. We intend to rigorously evaluate various clustering algorithms for network traffic, ensuring that the chosen methodology contributes effectively to a meaningful score for the DDQL agent's reward structure. Another important point to study is feature drift: at each step, we need to provide the agent with the most relevant features to make a decision. Consequently, designing robust mechanisms to detect and adapt to feature drift will be a major area of interest. This is a important challenge and an essential effort for future research,

as it is crucial for sustained success in intrusion detection scenarios.

One of the main obstacles to the process of continuous training with dynamic data streams is the phenomenon of "catastrophic forgetting" [5], a situation in which the neural network struggles to recognize old classes as it learns new ones. To mitigate this phenomenon, we aim to investigate several strategies, such as the use of an experience replay mechanism [3], to enable the network to retain its former knowledge while assimilating new information.

Finally, to assess the effectiveness of the proposed model, we are working on the definition of a comprehensive evaluation framework. Conventional performance metrics such as accuracy and F1 score are essential; however, we also seek to use a large spectrum of scenarios to rigorously test the model's ability to detect new attack. We also focus on evaluating energy consumption during learning and inference phases. Given the growing importance attached to sustainable computing, we believe that energy efficiency is an indispensable measure when evaluating machine learning algorithms.

REFERENCES

[1] Hooman Alavizadeh, Hootan Alavizadeh, and Julian Jang-Jaccard. Deep q-learning based reinforcement learning approach for network intrusion detection. *Computers*, 11(3), 2022.

[2] Absalom E Ezugwu, Abiodun M Ikotun, Olaide O Oyelade, Laith Abualigah, Jeffery O Agushaka, Christopher I Eke, and Andronicus A Akinyelu. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743, 2022.

[3] Daniel Fährmann, Nils Jorek, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Double deep q-learning with prioritized experience replay for anomaly detection in smart environments. *IEEE Access*, 10:60836–60848, 2022.

[4] Dan He, Jiwon Kim, Hua Shi, and Boyu Ruan. Autonomous anomaly detection on traffic flow time series with reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 150:104089, 2023.

[5] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[6] Adam Laud and Gerald DeJong. The influence of reward on the speed of reinforcement learning: An analysis of shaping. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 440–447, 2003.

[7] Wei-Chao Lin, Shih-Wen Ke, and Chih-Fong Tsai. Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, 78:13–21, 2015.

[8] Manuel Lopez-Martin, Belen Carro, and Antonio Sanchez-Esguevillas. Application of deep reinforcement learning to intrusion detection for supervised problems. *Expert Systems with Applications*, 141:112963, 2020.

[9] Gerhard Münz, Sa Li, and Georg Carle. Traffic anomaly detection using k-means clustering. In *Gi/itg workshop mmbnet*, volume 7, 2007.

[10] Thanh Thi Nguyen and Vijay Janapa Reddi. Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3779–3795, August 2023.

[11] Ranjit Panigrahi and Samarjeet Borah. A detailed analysis of cicids2017 dataset for designing intrusion detection systems. *International Journal of Engineering & Technology*, 7(3.24):479–482, 2018.

[12] Sathyanarayanan Revathi and A Malathi. A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)*, 2(12):1848–1853, 2013.

[13] Shahadate Rezvy, Yuan Luo, Miltos Petridis, Aboubaker Lasebae, and Tahmina Zebin. An efficient deep learning model for intrusion classification and prediction in 5g and iot networks. In *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, 2019.

[14] Kamalakanta Sethi, Rahul Kumar, Nishant Prajapati, and Padmalochan Bera. Deep reinforcement learning based intrusion detection system for cloud infrastructure. In *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, pages 1–6, 2020.

[15] Kamalakanta Sethi, Rahul Kumar, Nishant Prajapati, and Padmalochan Bera. Deep reinforcement learning based intrusion detection system for cloud infrastructure. In *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, pages 1–6, 2020.

[16] Meng Shen, Yiting Liu, Liehuang Zhu, Ke Xu, Xiaojiang Du, and Nadra Guizani. Optimizing feature selection for efficient encrypted traffic classification: A systematic approach. *IEEE Network*, 34(4):20–27, 2020.

[17] Haonan Tan, Le Wang, Dong Zhu, and Jianyu Deng. Intrusion detection based on adaptive sample distribution dual-experience replay reinforcement learning. *Mathematics*, 12(7), 2024.

[18] Imad Tareq, Bassant Mohamed Elbagoury, Salsabil Amin El-Regaily, and El-Sayed M El-Horbaty. Deep reinforcement learning approach for cyberattack detection. *International Journal of Online & Biomedical Engineering*, 20(5), 2024.

[19] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning, 2015.

[20] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[21] Alaettin Zubaroğlu and Volkan Atalay. Data stream clustering: a review. *Artificial Intelligence Review*, 54(2):1201–1236, February 2021.