Vers une Fonction de Récompense non-supervisée pour un Système de Détection d'Intrusion Basé sur l'Apprentissage par Renforcement Profond

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

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

***Résumé*—Les systèmes de détection d'intrusion (IDS - Intrusion Detection Systems) basés sur l'apprentissage profond ont prouvé leur efficacité, mais ont des difficultés à apprendre de manière continue et à détecter de nouvelles attaques au fil du temps en raison d'une fonction de récompense supervisée basée sur des étiquettes. Dans cet article, nous présentons une méthode d'apprentissage par renforcement appelé *Double Deep Q-Learning* (DDQL) utilisant une fonction de récompense non-supervisée permettant de détecter des motifs d'attaques inconnus. Pour fonctionner, la fonction de récompense s'appuie sur un score de normalité inspiré de la détection des anomalies du trafic automobile.**

Mots clés : Système de Détection d'Intrusion, Apprentissage par Renforcement Profond, Clustering, Non Supervisé

I. Introduction

Les systèmes de détection d'intrusion (IDS - Intrusion Detection Systems) sont désormais considérés comme des outils essentiels. Ils analysent le trafic pour détecter des comportements malveillants. Au fil des décennies, de nombreuses approches ont été adoptées pour le développement d'IDS, en commençant par des méthodes basées sur des signatures. Bien qu'elles soient efficaces et largement utilisées pendant des années, des attaques légèrement différents des signatures connues n'étaient pas détectés. Cela implique également la gestion du stockage, du partage et de la mise à jour régulière de ces signatures. Les IDS peuvent également tirer profit de l'apprentissage automatique (ML - Machine Learning) et de l'apprentissage profond (DL - Deep Learning) pour détecter des attaques plus complexes au sein du trafic [9] [7] [13]. Ils ont démontré un réel potentiel à discerner les variations comportementales et à identifier les attaques. Néanmoins, leur capacité à détecter des modèles inconnus reste limitée [18], et les complexités de calcul et de stockage associées à l'apprentissage régulier posent des difficultés considérables.

L'Apprentissage par Renforcement (RL - Reinforcement Learning) offre une solution pour répondre à ces contraintes et permet de détecter et de classer des attaques [1] [14] [10]. Dans le RL, un agent affine sa politique en prenant des décisions et en recevant des récompenses ou des pénalités en fonction des réponses de l'environnement délivrées à travers une fonction de récompense (Figure 1). Cependant, le RL rencontre des défis dans des scénarios réels avec de volumes de données importants et de grands espaces d'états [10]. Des travaux récents ont montré que les méthodes d'Apprentissage par Renforcement Profond (DRL - Deep Reinforcement Learning), utilisant des réseaux de neurones, sont capables d'améliorer la robustesse et la sécurité d'un réseau tout en fonctionnant de manière autonome et sans intervention humaine [1]. Malheureusement, très peu d'attention est accordée au mécanisme de récompense et de nombreux travaux sur les IDS basés sur des algorithmes de DRL utilisent une fonction de récompense supervisée qui est souvent basée sur des étiquettes, empêchant ainsi l'agent de détecter de nouveaux motifs d'attaques et de s'entraîner continuellement dans un environnement réel où les étiquettes sont inconnues [15] [10].

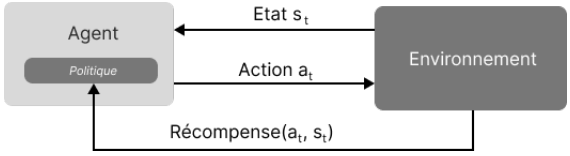

FIGURE 1 – Processus d'Apprentissage par Renforcement

L'objectif principal de notre travail est de tirer pleinement parti des capacités du RL et de mettre en œuvre un IDS non supervisé pouvant être utilisé dans des conditions réelles et détecter des motifs inconnus. Dans cet article, nous présentons une analyse de deux datasets de détection d'intrusion bien connus, en nous concentrant sur les caractéristiques qui définissent les flux réseau. De plus, nous introduisons une méthode basée sur le clustering qui s'appuie sur les propriétés des flux pour développer une fonction de récompense non-supervisée.

L'article est organisé comme suit : La section 2 détaille les deux datasets utilisés pour nos expériences, y compris les étapes de prétraitement. Dans la section 3, nous plongeons dans l'application du DRL à la détection d'intrusions et décrivons les composants principaux de l'algorithme. La section 4 est dédiée à l'analyse des caractéristiques des flux, qui sont des éléments d'information essentiels sur lesquels l'agent base ses décisions. Nous introduisons notre fonction de récompense non-supervisée dans la section 5 suivie d'une dis-

cussion sur les pistes potentielles pour les futures recherches dans la section 6.

II. Datasets

Pour nos expériences, nous avons choisi les datasets NSL-KDD [12] et CICIDS17 [11], tous deux bien connus et couvrant un large éventail d'attaques. Ce choix a été fait pour évaluer notre méthode à travers un large spectre d'attaques, plutôt que de la restreindre à un sous-ensemble spécifique, tout en permettant une comparaison avec des études précédentes.

Ensemble de données	**NSL-KDD**	**CICIDS17**
Modèle de données	Flux réseau	Flux réseau
Caractéristiques	41 (38 continues, 3 catégorielles)	81 (68 continues, 13 catégorielles)
Total des enregistrements	125,973 (entraînement), 22,543 (test)	2,830,743
Étiquettes	23 entraînement, 38 test	15 regroupées en 7 catégories
Classes principales	Normal, DOS, Probe, R2L, U2R	Normal, Brute Force, DoS, DDoS, Web, Infiltration, Botnet
Protocoles	TCP, UDP, ICMP	HTTP, HTTPS, FTP, SSH, Email, etc.
Séparation	Entraînement/test prédéfini	70% entraînement, 30% test

TABLE I – Comparaison des ensembles de données NSL-KDD et CICIDS17

Évidemment, avant d'utiliser NSL-KDD et CICIDS17, nous devons les préparer pour adapter les données à notre modèle et optimiser la phase d'entraînement. Conformément aux préparations de données courantes dans les études de DRL [17] [1] [10], nous appliquons les opérations suivantes :

- **Encodage :** Les caractéristiques catégorielles sont transformées en données numériques en utilisant *Label Encoder* et *One Hot Encoding*.
- **Normalisation :** Chaque caractéristique est normalisée en soustrayant la moyenne et en divisant par l'écart type.

III. Apprentissage par Renforcement Profond pour la détection d'intrusions

Bien que des algorithmes de ML et de DL soient fréquemment utilisés pour les tâches de classification, tirer parti du DRL pour aborder ces problèmes présente des défis. Néanmoins, cette approche offre des avantages clés. La fonction de récompense contribue à un meilleur contrôle et une meilleure interprétabilité durant la phase d'entraînement. Ensuite, l'agent acquiert la capacité de prendre des décisions et de s'adapter dans des environnements dynamiques, ce qui est un aspect essentiel des scénarios impliquant la détection d'attaques.

L'article se concentre sur les principaux composants du DRL et explique comment notre fonction de récompense non-supervisée prend part au processus d'entraînement.

A. Q-Network : réseau de neurones

Dans un algorithme d'apprentissage par renforcement comme le *Q-Learning* [20], le *Q-network* a un objectif important puisqu'il vise à approximer la fonction de valeur Q. Dans notre proposition, nous utilisons un simple réseau de neurones *Feed Forward (FFNN)* de deux couches cachées avec la fonction d'activation ReLU pour toutes les couches. La décision d'utiliser un FFNN a été guidée par sa popularité dans les études de DRL pour la détection d'intrusion et des problèmes similaires [8] [3], sa simplicité et la nature de l'environnement. Pour obtenir une phase d'apprentissage plus stable et une convergence plus rapide, nous utilisons deux *Q-Network* à travers l'algorithme *Double Deep Q-Learning* (DDQL) [19].

B. Environnement et état

Comme dans de nombreux IDS basés sur le RL, l'environnement est simulé en échantillonnant un ensemble de données contenant des flux [1] [17] [14]. Chaque flux est caractérisé par un nombre fixe de caractéristiques (22 pour NSL-KDD, 27 pour CICIDS17) représentant divers aspects de l'environnement à un instant donné t. Les caractéristiques de plusieurs flux constitueront un état qui est un sous-groupe de flux.

C. Actions

Les actions visent à classer le trafic réseau. Dans notre approche, nous avons opté pour une classification binaire avec des étiquettes "attaque" et "normal". Ce choix est motivé par la volonté d'utiliser une architecture de réseau de neurones fixe. Si plusieurs classes d'attaque étaient considérées, l'introduction d'une nouvelle attaque nécessiterait d'ajouter un nouveau neurone de sortie, modifiant ainsi l'architecture du réseau de neurones. Cela ne rentre pas dans le cadre de cet article.

D. Fonction de récompense

La récompense est le retour de l'environnement à une action prise par l'agent. C'est un composant majeur car elle détermine la qualité des actions de l'agent et influence l'évolution de la politique de décision. Dans les IDS utilisant le DRL, la plupart des fonctions de récompense sont supervisées (Équation 1) et offrent à l'agent une bonne récompense si sa prédiction correspond à l'étiquette, et une mauvaise récompense sinon [1] [10] [14].

$$\text{Reward}(s_t, a_t) = \begin{cases} -1, & \text{si } a_t \neq l_t \\ 1, & \text{si } a_t = l_t \end{cases} \quad \text{avec } l_t \text{ l'étiquette pour } s_t \tag{1}$$

Cette méthode impose des limites aux circonstances dans lesquelles l'agent peut opérer et réduit ses capacités. Sans étiquettes associées à chaque flux, l'agent n'est pas capable d'apprendre davantage et il devient incapable de détecter des attaques et des comportements inconnus. Cela motive notre recherche sur une fonction de récompense non-supervisée.

IV. Sélection et analyse des caractéristiques

Pour permettre à l'agent de DRL de converger et de prendre des bonnes décisions, les états qu'il observe doivent fournir les informations les plus pertinentes. Une approche naïve pourrait être d'incorporer chaque caractéristique d'un flux

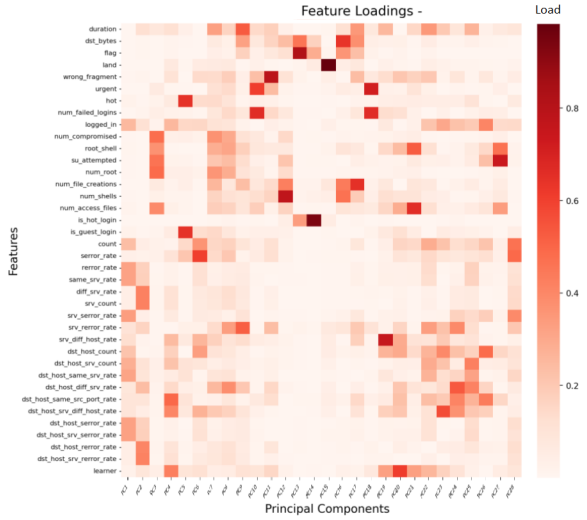

FIGURE 2 – CP de NSL-KDD expliquée

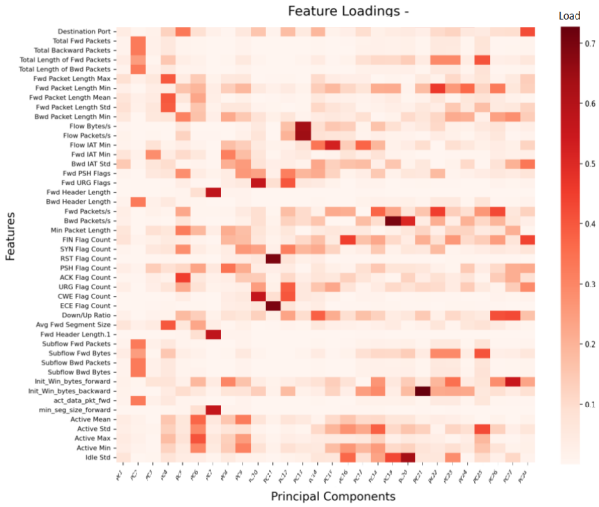

FIGURE 3 – CP de CICIDS17 expliquée

(Tableau I) en tant que caractéristique d'état. Cependant, cela pourrait conduire à des états de haute dimension peuplés de caractéristiques non pertinentes et pourraient augmenter les temps d'apprentissage. Pour surmonter ce problème, une fois que les données des flux sont formatées de manière appropriée, nous appliquons une Analyse en Composantes Principales (ACP) comme suggéré dans [16] en conservant les dimensions représentant 97% de la variance. Nous obtenons alors 22 composantes principales pour le dataset NSL-KDD et 27 pour le dataset CICIDS17. Ces composantes principales (CP) résultent de combinaisons linéaires des caractéristiques de l'ensemble de données d'origine. Les états sont maintenant décrits en utilisant ces nouvelles composantes. Dans les images 2 et 3, nous détaillons comment les CP sont construites et quelles caractéristiques elles utilisent. Pour le dataset NSL-KDD (Figure 2), il est indiqué que les caractéristiques de taux d'erreur sont critiques pour la détection d'anomalies. Dans CICIDS17 (Figure 3), les caractéristiques basées sur les propriétés des paquets — telles que la longueur, la taille et les propriétés temporelles — émergent comme les indicateurs les plus révélateurs d'attaques. Cette analyse nous aide à mieux comprendre quelles caractéristiques fournissent le plus d'informations sur un flux et ce que l'agent observe pour prendre des décisions.

V. VERS UNE FONCTION DE RÉCOMPENSE NON-SUPERVISÉE

La fonction de récompense est un composant clé de l'algorithme DRL car elle oriente l'apprentissage et les actions futures. La manière dont nous la concevons peut avoir un impact significatif sur la performance de l'agent, que ce soit en termes de précision ou de temps de convergence [6].

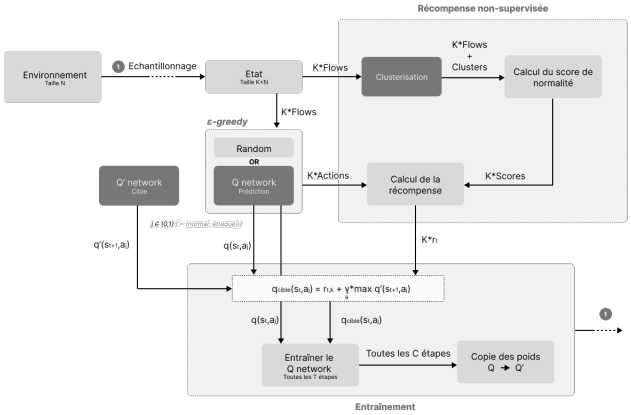

FIGURE 4 – Architecture de la fonction de récompense non-supervisée dans l'algorithme de DDQL

Dans le contexte de la détection d'intrusions basée sur l'apprentissage par renforcement, la récompense est souvent déterminée par la simple comparaison entre les étiquettes et les actions de l'agent. Le véritable défi consiste à maintenir ce mécanisme lorsque les étiquettes ne sont pas disponibles. Des recherches sur la détection d'anomalies dans les flux de trafic routier ont conduit au développement d'une fonction de récompense basée sur le regroupement en clusters [4]. Actuellement, nos travaux se concentrent sur l'utilisation d'algorithmes de clustering pour calculer ces récompenses. À chaque itération, les flux de réseau sont regroupés en clusters afin de déterminer un score de normalité, calculé à partir des attributs de leurs clusters respectifs. Ce score permet d'évaluer la "normalité" de chaque flux réseau et est ensuite intégré dans la fonction de récompense. Cette approche oriente ainsi le comportement de l'agent d'apprentissage, en le guidant vers une prise de décision précise et non-supervisée.

Atteindre notre objectif nécessite de surmonter plusieurs défis. Tout d'abord, la sélection d'un algorithme de clustering approprié est cruciale ; il doit démontrer son efficacité pour distinguer les différentes classes de flux réseau présents dans les datasets. Les algorithmes de clustering sont divers

[21] et peuvent être classés en cinq catégories principales : hiérarchique, partitionnement, basé sur la densité, basé sur une grille et basé sur un modèle [22] [2]. Chaque catégorie présente des avantages et des défis distincts, et notre recherche est actuellement axée sur l'identification de l'algorithme de clustering le mieux adapté à notre application spécifique. Cette sélection implique de prendre en compte des facteurs tels que la forme attendue des clusters, les paramètres nécessaires impliqués et la complexité en temps et en espace des algorithmes.

De plus, nous allons nous concentrer sur l'affinage d'un algorithme de réduction de dimension pour préserver les caractéristiques des flux réseau contenant le plus d'informations, améliorant ainsi l'efficacité des opérations de clustering.

VI. Conclusion et travaux futurs

Dans cet article, nous avons introduit le DDQL comme une approche prometteuse pour la détection d'intrusions dans les systèmes réseaux, soulignant son potentiel à identifier et à apprendre des attaques jusqu'alors méconnues. Nous suggérons qu'une fonction de récompense non supervisée pourrait grandement faciliter le déploiement d'agents dans des scenarios réels. Cependant, ce travail a mis en évidence plusieurs domaines nécessitant une exploration plus approfondie.

L'algorithme de clustering constitue un élément central de la fonction de récompense. Nous avons l'intention d'examiner rigoureusement plusieurs algorithmes de clustering pour le trafic réseau, en veillant à ce que la méthode choisie contribue efficacement à la fonction de récompense de l'agent DDQL. Un autre aspect crucial à étudier est la "dérive des caractéristiques" : il est essentiel de fournir à l'agent, à chaque étape, les caractéristiques les plus pertinentes pour une prise de décision efficace. Ainsi, la conception de mécanismes robustes pour détecter et s'adapter à la dérive des caractéristiques sera un axe de recherche majeur. Il s'agit d'un défi important et d'un effort essentiel pour les futures recherches, car cela représente un point crucial pour une détection d'intrusions réussie et pérenne.

Un des principaux obstacles à l'apprentissage continu avec des flux de données dynamiques est le phénomène d'"oubli catastrophique" [5], où le réseau de neurones peine à se souvenir des anciennes classes quand il en apprend des nouvelles. Pour atténuer ce phénomène, nous prévoyons d'explorer plusieurs stratégies, comme l'utilisation d'un mécanisme appelé "Experience Replay" [3], afin de permettre au réseau de neurones de conserver ses connaissances les plus anciennes tout en acquiesçant des nouvelles.

Enfin, pour évaluer l'efficacité du modèle proposé, nous travaillons sur l'élaboration d'un cadre d'évaluation exhaustif. Bien que les mesures de performance conventionnelles telles que l'exactitude et le score F1 soient essentielles, nous cherchons également à tester le modèle dans un large éventail de scénarios pour évaluer sa capacité à détecter de nouvelles attaques. Nous nous penchons également sur l'évaluation de la consommation d'énergie pendant les phases d'apprentissage et d'inférence. Compte tenu de l'importance croissante de l'informatique durable, nous considérons que l'efficacité énergétique est une métrique indispensable dans l'évaluation des algorithmes d'apprentissage automatique.

Références

[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] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8 :279–292, 1992.

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

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