Un système d'apprentissage ensembliste explicable pour détecter des attaques réseau inconnues

Céline Minh, Kevin Vermeulen, Cédric Lefebvre, Philippe Owezarski, William Ritchie

Un système d'apprentissage ensembliste explicable pour détecter des attaques réseau inconnues

Céline Minh*†, Kevin Vermeulen†, Cédric Lefebvre*, Philippe Owezarski†, William Ritchie*
*Custocy, Toulouse, France. Email: {cminh, clefebvre, writchie}@custocy.com
†LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France. Email: {celine.minh, kevin.vermeulen, owe}@laas.fr

***Résumé*—L'apprentissage automatique est une technologie prometteuse pour les systèmes de détection d'intrusions. Il existe une grande variété d'algorithmes d'apprentissage automatique dont les résultats semblent complémentaires, mais déterminer quel résultat est correct dans un cas spécifique est difficile car les algorithmes ne sont pas tous explicables. Cet article introduit un système conçu pour être explicable qui reconstruit des schémas d'attaques à partir des sorties d'un ensemble de détecteurs d'anomalies non supervisés et les présente aux analystes de sécurité pour les aider à interpréter chacune des détections.**

***Mots-clés*—Sécurité réseau, détection d'anomalies non supervisée, IA explicable, apprentissage ensembliste, CNN**

I. Introduction

L'apprentissage automatique (ML) est une technologie prometteuse pour les systèmes de détection d'intrusion réseau (NIDSs) car il permet de détecter des attaques sur de grandes quantités de données. Un inconvénient des modèles de ML est leur tendance à ne pas s'accorder sur les détections : certains modèles vont qualifier une entrée comme une attaque tandis que d'autres vont la qualifier comme étant bénigne. Identifier le modèle qui trouve la bonne réponse est compliqué car les modèles de ML sont souvent considérés comme des boîtes noires et n'apportent pas d'arguments pour justifier leurs résultats. L'apprentissage ensembliste [1], [2] est une approche qui combine plusieurs modèles d'apprentissage automatique pour obtenir un système plus performant. En particulier, l'empilement de modèles consiste à combiner plusieurs modèles de base qui effectuent la même tâche (e.g., vote majoritaire pondéré). Une méthode d'empilement plus sophistiquée est le méta-apprentissage, où les sorties de plusieurs modèles de base servent d'entrée à un modèle de niveau supérieur, appelé méta-modèle. Nous proposons une nouvelle approche ensembliste qui présente systématiquement les résultats de chaque modèle de base aux analystes de sécurité, de façon à faciliter une prise de décision éclairée. Cette décision serait prise grâce à une combinaison de représentations visuelles concises des anomalies réseau et d'un niveau élevé d'explicabilité de chaque modèle de base.

L'explicabilité [3], [4] est la propriété d'un système qui rend son raisonnement et ses résultats compréhensibles par des humains. Dans les NIDSs actuels, les analystes de sécurité prennent des décisions pour résoudre les problèmes de sécurité en se basant sur l'analyse du système. Par conséquent, fournir des preuves intelligibles des résultats des modèles d'apprentissage automatique est crucial pour que les analystes fassent confiance au système. L'explicabilité n'est pas seulement importante pour améliorer la collaboration entre les analystes de sécurité et les systèmes à base d'intelligence artificielle (IA), mais aussi pour assister les ingénieurs et les chercheurs dans la conception de systèmes plus performants, en les aidant à comprendre où et pourquoi un modèle fait des erreurs.

Nous proposons une méthode qui permet aux utilisateurs de tirer parti d'un apprentissage ensembliste pour simultanément améliorer la performance de la détection et visualiser les résultats de tous les modèles de base afin de mieux comprendre comment ces détections sont effectuées. Nous introduisons une représentation visuelle des anomalies levées par des détecteurs non supervisés (UL) au fil du temps pour à la fois aider les analystes de sécurité à comprendre ce qu'il se passe sur le réseau et permettre à notre méta-modèle, un réseau de neurones convolutif (CNN), d'identifier des schémas d'attaque [5], [6]. Notre système est censé préserver les propriétés de l'apprentissage non supervisé, y compris la détection des attaques inconnues, car la couche supervisée, le méta-modèle, analyse les sorties des modèles de base, qui sont des méta-données et non des données réseau brutes.

Nous introduisons ainsi un système, conçu pour être explicable, qui analyse et combine un ensemble de modèles non supervisés pour détecter des attaques réseau. Nos contributions sont les suivantes :

1) Un NIDS à base d'IA plus transparent, dont les résultats sont détaillés,
2) Des représentations visuelles des anomalies réseau interprétables par les utilisateurs,
3) Une méthode d'apprentissage ensembliste qui utilise un CNN comme méta-modèle pour combiner des modèles de base.

La structure de l'article sera organisée de la manière suivante :

— Dans la Section II, nous examinerons les travaux connexes sur l'apprentissage ensembliste, la détection d'attaques et l'IA explicable et nous les mettrons en perspective avec notre contribution.
— Dans la Section III, nous détaillerons la conception de notre système de détection d'anomaliess. Nous expliquerons en détail les différentes étapes du processus, c'est-à-dire, l'agrégation de flux réseau, l'attribution de scores d'anomalie avec les modèles de base, la

génération de représentations visuelles des anomalies, et la reconnaissance de schémas d'attaque.
— La Section IV présentera les résultats de chaque composante de notre système. Nous présenterons les performances de chaque modèle de base, ainsi que du modèle final.
— Dans la Section V, nous interpréterons les résultats de notre système et engagerons une discussion approfondie sur les limitations de notre approche. Nous aborderons également les perspectives d'amélioration et les directions futures de recherche.
— Enfin, dans la Section VI, nous conclurons l'article en récapitulant nos principales contributions, en soulignant les avantages de notre système de détection d'attaques réseau, et en proposant des pistes pour nos futures recherches.

II. Travaux connexes

Notre recherche explore des mécanismes pour reconstruire des schémas d'attaque en utilisant plusieurs techniques de détection d'anomalies non supervisées. Nos travaux abordent des problématiques liées à (1) l'apprentissage ensembliste, (2) la détection de schémas d'attaque et (3) l'IA explicable. Cette section aborde ces thèmes dans cet ordre, et positionne nos travaux par rapport à des travaux connexes.

En ce qui concerne les systèmes utilisant de l'apprentissage ensembliste pour la détection d'anomalies réseau, Vanerio et Casas [1] ont comparé plusieurs méta-modèles pour détecter des attaques. Les auteurs ont étudié des algorithmes de vote majoritaire pondéré où les poids ont été définis par rapport à la précision des modèles de base. Nous avons adopté une approche différente où le méta-modèle est une autre couche de ML, un CNN, qui prend en entrée une représentation visuelle des détections des modèles de base et détecte des scénarios d'attaque sur ces représentations. Mirsky et al. [7] ont proposé Kitsune, un ensemble d'autoencodeurs pour détecter des anomalies réseau. Ce système est basé sur des autoencodeurs, qui sont souvent considérés comme des techniques non supervisées car ils ne nécessitent pas de données labellisées. En revanche, ils nécessitent une phase d'entraînement sur un ensemble de données normales pour pouvoir caractériser le comportement normal du système. Kitsune empile des autoencodeurs en utilisant un autre autoencodeur comme méta-modèle pour traiter les scores d'anomalie générés par les autoencodeurs de base. Leur approche pour combiner des modèles de base est proche de la notre, à la différence que nous traitons des algorithmes et des représentations de données hétérogènes.

Sur un sujet différent, Zhou et al. [8] ont proposé un système qui utilise LSTM pour détecter des attaques en plusieurs étapes. Leur modèle traite les séquences générées par le NIDS Snort [9] et traite le problème des relations à long terme entre les alertes. Ghafir et al. [10] ont proposé un système de détection de menaces persistantes avancées (APT). Le système utilise un modèle de Markov caché (HMM) pour détecter le scénario d'APT le plus probable étant données les alarmes. Ensuite, il prévoit la prochaine étape de l'APT en cours. Une différence significative avec notre travail est que le système a été entraîné par rapport à un cycle de vie d'attaque donné [11], [12], alors que notre système apprend les schémas d'attaque directement à partir de la donnée. Wang et al. [6] ont converti des données de trafic brutes (fichiers pcap) en images et ont aussi identifié des schémas d'attaques.

Wei et al. [13] ont observé que les méthodes d'explications généralistes actuelles, comme SHAP [14] et LIME [3] ne sont pas exploitables pour les NIDSs car ces méthodes ne considèrent pas les dépendances entre les caractéristiques des flux réseaux. Han et al. [4] se sont orientés vers solutions pour de l'IA explicable en proposant un système pour interpréter des NIDS utilisant de l'apprentissage profond non supervisé. Ce système analyse chaque détection d'un modèle en retournant les caractéristiques les plus importantes et en décrivant leur signification pour qu'un analyste de sécurité puisse les comprendre. Plutôt que d'investiguer des explications *ad hoc*, nous avons cherché à concevoir un système plus transparent qui fournit un ensemble de représentations intermédiaires pour aider les analystes de sécurité à comprendre les réactions du système et à mieux identifier de potentielles erreurs.

III. Conception du système

A. Aperçu du système

Nous avons développé un NIDS basé sur l'IA qui utilise un ensemble de modèles de base non supervisés pour détecter des attaques réseau (Figure 1). Un aspect clé de notre étude est de combiner différents algorithmes de détection d'anomalies en les appliquant sur les mêmes données. Les termes "modèle" et "algorithme" sont utilisés de manière interchangeable pour décrire les différentes méthodes d'apprentissage automatique utilisées dans notre système. Ainsi, nous cherchons à caractériser les résultats des modèles de base de manière indépendante de la définition des algorithmes utilisés. Pour cela, nous avons pris en compte uniquement les entrées et les sorties des modèles de base pour les comparer.

En ce qui concerne les entrées des modèles, nous avions besoin d'une représentation qui soit sémantiquement intéressante pour que les modèles puissent identifier des attaques (Section III-B). De façon similaire à UNADA [15], [16], nous avons agrégé des flux réseau par adresse IP source et destination, puis défini des caractéristiques statistiques des agrégats (Tableau I).

Chaque modèle de base attribue un score d'anomalie à un agrégat. Pour cela, on applique un même algorithme sur des sous-espaces de caractéristiques différents. Ensuite, on combine leurs sorties en les additionnant [15], [16]. La définition de nos scores est donc ainsi indépendante de l'algorithme utilisé, ce qui nous permet de comparer des détecteurs d'anomalies utilisant des algorithmes différents.

Pour aller plus loin dans la caractérisation des sorties des modèles, nous avons introduit une représentation visuelle des anomalies réseau (Section III-D) qui met en évidence à la fois un aspect spatial (e.g., est-ce que les anomalies affectent ou proviennent de la même adresse IP ?) et un aspect temporel

TABLEAU I – Caractéristiques des agrégats

Caractéristique	Clé d'agrégation	Description
n_dst_ip	IPsrc	Nombre d'adresses IP de destination
n_src_ip	IPdst	Nombre d'adresses IP source
n_dst_ports	IPsrc & IPdst	Nombre de ports de destination
n_src_ports	IPsrc & IPdst	Nombre de ports source
n_fwd_pkts	IPsrc & IPdst	Nombre de paquets forward
n_bwd_pkts	IPsrc & IPdst	Nombre de paquets backward
sum_flx_dur	IPsrc & IPdst	Somme de la durée des flux
tot_flx	IPsrc & IPdst	Nombre de flux
sum_pkts_size	IPsrc & IPdst	Somme de la taille des paquets
std_pkt_size	IPsrc & IPdst	Ecart-type de la taille des paquets

(e.g., est-ce que les anomalies sont répétées ?). Pour finir, notre méta-modèle, un CNN, analyse ces représentations des anomalies réseau pour identifier des schémas d'attaque et lever des alertes.

B. Agrégation de flux réseau

Notre système calcule les caractéristiques décrites dans le Tableau I à partir d'agrégats de flux réseau. Nous avons choisi d'analyser à la fois des agrégats par adresse IP source et destination car ces perspectives pourraient être complémentaires. En effet, certaines attaques (e.g., attaques par déni de service distribuées) impliquent plusieurs adresses IP source à destination d'une même adresse IP, tandis que d'autres attaques (e.g., scans de réseau) impliquent généralement une seule adresse IP source et plusieurs destinataires. L'utilisation de ces perspectives est propice à la détection de schémas d'attaque mais est également très lisible pour les analystes de sécurité. En effet, nous utilisons un nombre réduit de caractéristiques ainsi qu'une représentation par adresses IP, ce qui permet aux analystes de sécurité d'identifier les machines ayant un comportement anormal.

De plus, pour éviter de capturer des anomalies liées à des changements brutaux, mais légitimes, du trafic (e.g., jours ouvrés, week-ends), nous avons choisi d'appliquer les algorithmes de base, non supervisés, sur de petits intervalles de temps Δ_t que nous avons fixé empiriquement à 2 minutes, plutôt que sur l'ensemble du trafic.

Enfin, afin de réduire davantage la dimensionnalité de nos données, nous calculons les agrégats uniquement sur les adresses IP internes au réseau considéré, c'est-à-dire les machines appartenant au réseau de l'entreprise. Il n'y a donc pas d'agrégats par destination pour les destinations publiques sur Internet.

C. Attribution de scores d'anomalie non supervisée

Le composant décrit Figure 2 consiste en un ensemble de détecteurs d'anomalies non supervisés qui quantifient le degré d'anomalie de chaque agrégat d'entrée. Il prend en entrée les caractéristiques des agrégats (Tableau I), par adresse IP source ou destination, durant une même période de temps Δ_T, que nous avons fixée empiriquement à 30 minutes. Le composant retourne des représentations visuelles des anomalies qui se présentent sous la forme d'un ensemble de matrices. Chaque matrice contient les scores d'anomalie de chaque agrégat attribués par les différents détecteurs d'anomalies.

Afin d'être capable de détecter de nouvelles attaques, nous avons utilisé des détecteurs d'anomalies non supervisés comme modèles de base. Nous avons étudié plusieurs algorithmes de détection d'anomalies non supervisés implémentés dans scikit-learn [17] et PyOD [18] que nous avons appliqués sur des agrégats générés à partir du jeu de données CIC-CSE-IDS2018 [19]. Nous avons observé des résultats différents générés avec ces algorithmes de détection d'anomalies pour de mêmes agrégats d'entrée (Section IV-C).

Nous avons sélectionné un ensemble d'algorithmes de détection d'anomalies de façon à maximiser la détection d'attaques tout en minimisant les fausses alarmes (c'est-à-dire la détection erronée de trafic légitime comme étant une attaque). Nous avons défini des modèles de base en utilisant à chaque fois les mêmes données d'entrée, mais avec des algorithmes de détection d'anomalies différents. Ensuite, nous avons considéré un modèle *empilé* qui cumule les alarmes des modèles de base en effectuant la somme booléenne de leurs résultats. Nous avons étudié des algorithmes de base très différents pour construire notre modèle empilé, et ce afin de maximiser leur complémentarité :

- *Isolation Forest (IF)* [20] est un algorithme basé sur des arbres.
- *Local Outlier Factor* [21] est un algorithme basé sur une notion de densité.
- *One-Class SVM (OCSVM)* [22] utilise un hyperplan.
- *KNN non supervisé* [23] est un algorithme basé sur une notion de distance.
- *COPOD* [24] est une algorithme probabiliste.

Nous avons évalué la performance de toutes les combinaisons possibles de trois algorithmes parmi les algorithmes étudiés sur le jeu de données CIC-CSE-IDS2018 [19], et avons sélectionné la combinaison qui a le plus grand taux de vrais positifs (TPR) et avec le plus petit taux de faux positifs (FPR).

Pour caractériser et comparer les résultats des modèles de base, que nous avons obtenus avec ces différents algorithmes non supervisés, nous avons utilisé une quantification des anomalies indépendante des algorithmes, définie précédemment dans UNADA [15], [16]. Pour attribuer un score d'anomalie avec un modèle de base, nous avons considéré tous les sous-espaces de $k = 2$ caractéristiques parmi les $n = 9$ caractéristiques des agrégats et avons appliqué un même algorithme de détection d'anomalies sur chaque sous-espace. Le score d'anomalie d'un agrégat généré par un modèle de base est défini comme le nombre de fois qu'un agrégat a été détecté comme anormal. Ces scores vont de 0, pour un agrégat complètement normal (d'après le modèle), à 36 pour un agrégat complètement anormal, car cela correspond au nombre de combinaisons de $k = 2$ caractéristiques parmi $n = 9$:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

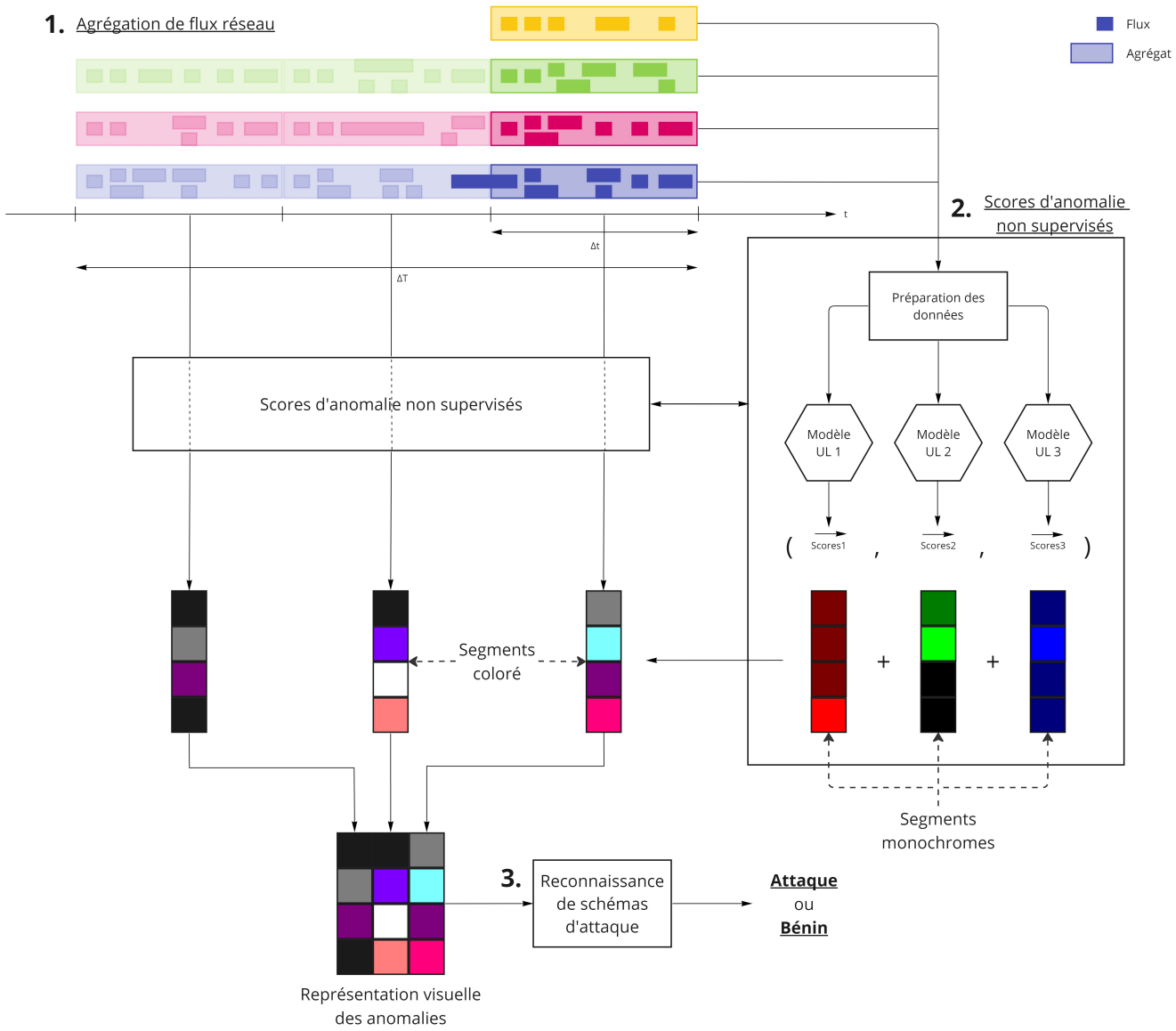

FIGURE 1 – Aperçu du système. Le système analyse une capture de trafic pendant une période Δ_T. Tout d'abord, il agrège les flux réseaux de la capture sur des intervalles de temps Δ_t et calcule leurs caractéristiques. Ensuite, il applique un ensemble de $N = 3$ modèles de base non supervisés pour attribuer des scores d'anomalie aux agrégats obtenus. Après cela, le système génère une représentation visuelle des anomalies détectées, permettant une visualisation claire des schémas d'attaque potentiellement présents dans la capture. Enfin, ces représentations sont analysées par un module de reconnaissance de schémas d'attaque, qui détermine si le réseau est attaqué pendant la période Δ_T.

D. Reconnaissance de schémas d'attaque

Dans la section précédente, nous avons sélectionné un ensemble de modèles non supervisés pour détecter des agrégats anormaux. Nous avons introduit une dimension temporelle à notre représentation des anomalies (Figure 4) pour mettre en évidence les séquences d'anomalies détectées par chaque modèle de base pendant une période de temps Δ_T de 30 minutes. Nous avons représenté les sorties de chaque modèle de base comme une matrice de scores d'anomalie où chaque ligne correspond à une adresse IP, source ou destination selon la clé d'agrégation, et chaque colonne correspond à un intervalle de temps.

Nous pouvons analyser ces représentations en utilisant des modèles d'apprentissage profonds, tels que des CNNs. Nous utilisons ainsi un CNN comme méta-modèle pour identifier des schémas d'attaque sur ces représentations du trafic. Le CNN prend en entrée les deux ensembles de matrices, qui représentent les anomalies générées à partir des sorties des modèles de base sur les agrégats par adresse IP source et destination pendant une période de temps Δ_T. Le CNN donne

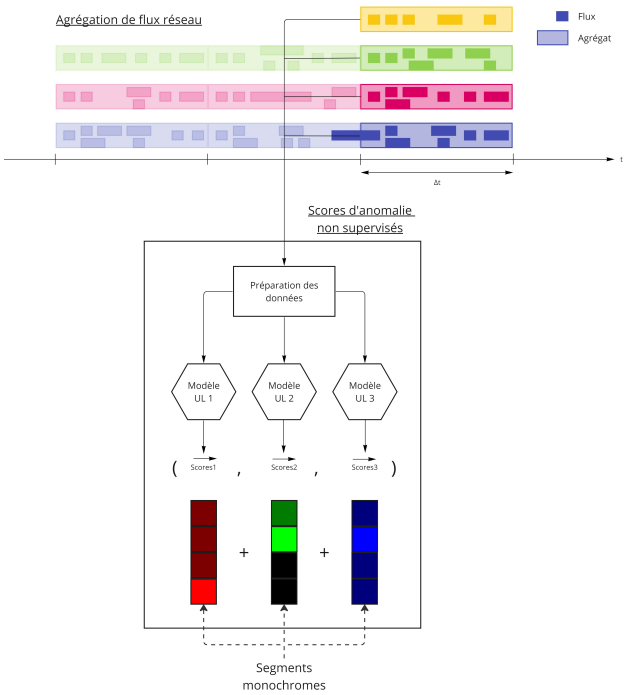

FIGURE 2 – Attribution de scores d'anomalie par 3 modèles de base non supervisés pendant un intervalle de temps Δ_t. Cette analyse génère un segment vertical qui fait partie de la représentation visuelle des anomalies générée dans la Figure 3. Les flux réseau sont regroupés en agrégats (4 dans cet exemple) pendant un intervalle de temps Δ_t. Ensuite, les caractéristiques des agrégats sont extraites et préparées avant d'être évaluées par N modèles de base non supervisés ($N = 3$ dans cet exemple). Chaque modèle de base attribue un score d'anomalie à chaque agrégat. Une couleur est assignée à chaque modèle, ce qui permet de représenter le scores d'anomalie sur un segment monochrome. En superposant les segments monochromes des N modèles de base, nous obtenons un segment coloré qui représente les anomalies détectées par l'ensemble des modèles pendant l'intervalle de temps Δ_t.

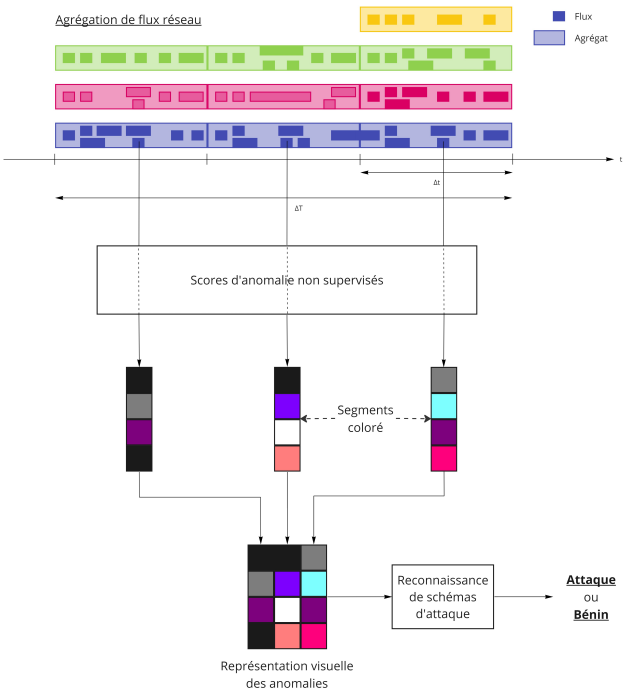

FIGURE 3 – Reconnaissance de schémas d'attaque sur une représentation visuelle des anomalies détectées pendant une période de temps Δ_T. La période Δ_T est constituée, dans cet exemple, de 3 intervalles de temps consécutifs Δ_t. Les flux réseau de chaque intervalle de temps Δ_t sont analysés comme dans la Figure 2. Chaque intervalle de temps Δ_t est représenté par un segment coloré. En disposant les 3 segments côte à côte, nous obtenons une représentation visuelle des anomalies réseau détectées par les $N = 3$ modèles de base pendant la période Δ_T.

le résultat final du système de détection, en indiquant si le réseau est attaqué durant la période Δ_T.

Le CNN nécessite un jeu de données labellisé pour son entraînement et sa validation. Nous avons choisi de labelliser une représentation comme étant une attaque si elle contient au moins un agrégat d'attaque (ce qui veut dire que le réseau subit une attaque pendant la capture), et bénigne sinon.

Pour visualiser les analyses de chaque modèle de base, nous leur avons assigné une couleur (rouge pour Local Outlier Factor, bleu pour KNN, et vert pour COPOD). L'intensité de la couleur est proportionnelle au score d'anomalie généré par le modèle. Ainsi, un pixel blanc signifie que tous les modèles ont détecté l'agrégat comme très anormal et un pixel noir signifie qu'aucun des modèles n'a levé une alarme (Figure 4).

IV. RÉSULTATS

A. *Jeu de données*

Nous avons évalué notre système sur le jeu de données CIC-CSE-IDS2018 [19]. Sharafaldin et al. [19] ont implémenté un réseau d'entreprise réaliste avec 420 machines et 30 serveurs, et une infrastructure d'attaque constituée de 50 machines. Le jeu de données consiste en dix jours de captures réseau pendant les horaires de travail. Les auteurs ont simulé plusieurs scénarios d'attaque, notamment des attaques par force brute, des attaques par déni de service distribuées, des injections SQL, etc. Le jeu de données fournit les heures des attaques ainsi que les adresses IP de l'attaquant et des victimes, nous avons donc labellisé les flux réseau en conséquence. Un agrégat par source ou par destination est labellisé comme une attaque si la source ou la destination émet ou reçoit du trafic depuis ou vers une adresse IP attaquante. Finalement, une image est labellisée comme une attaque si un agrégat dans l'image est étiqueté comme une attaque.

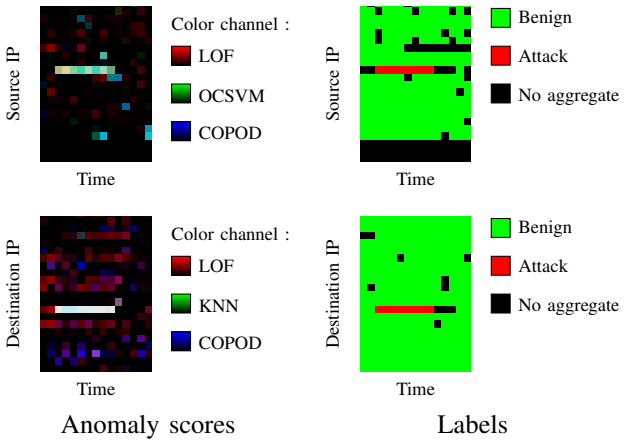

FIGURE 4 – Représentation du trafic pendant une attaque par déni de service. Les images à droite affichent les labels, où chaque pixel rouge représente un agrégat d'attaque et chaque pixel vert représente un agrégat bénin. Nous observons une ligne rouge sur chacune des images, ce qui signifie que l'attaque provient et touche une seule IP et que les agrégats sont rapprochés dans le temps. Les images à gauche ont été générées en utilisant les modèles décrits Section III-C. Sur les images de gauche, nous observons que les ensembles de modèles non supervisés ont détecté la ligne d'attaque entière.

B. Agrégation de flux réseau

Notre système agrège d'abord les flux réseaux par adresse IP source et destination (Section III-B). Le Tableau II décrit la répartition des agrégats ainsi obtenus pour chaque catégorie de trafic. On constate qu'il y a moins d'agrégats par IP destination que par IP source, ce qui indique qu'il y a davantage de machines émettrices de trafic que de machines réceptrices dans cette capture de trafic.

C. Attribution de scores d'anomalie non supervisée

Notre système génère ensuite des représentations intermédiaires visuelles des anomalies réseau (Figure 4). Pour cela, nous avons sélectionné un ensemble d'algorithmes qui génèrent des résultats complémentaires (Section III-C). Le tableau III montre la performance des cinq meilleures combinaisons de modèles de base, que l'on appelle *modèles empilés*, sur les agrégats par adresse IP source et destination. Il indique leurs taux de vrais positifs (TPR) et de faux positifs (TFP) sur les agrégats par source et destination. Nous considérons que le modèle empilé détecte un agrégat comme une attaque si l'un des modèles de base détecte l'agrégat comme une attaque.

Les meilleures combinaisons sont (LOF, OCSVM, COPOD) pour analyser les agrégats par adresse IP source, et (LOF, KNN, COPOD) pour analyser les agrégats par adresse IP de destination. Bien que nous ayons sélectionné la combinaison avec les meilleurs résultats de manière empirique, notre intuition est que cette performance provient de l'utilisation d'algorithmes avec des méthodes très différentes pour détecter les anomalies.

Ensuite, nous avons appliqué ces différents algorithmes sur chaque sous-espace de $k = 2$ caractéristiques des agrégats d'un même intervalle de temps Δ_t et cumulé les alarmes levées sur chaque sous-espace pour définir un score d'anomalie associé à chaque modèle de base.

Le Tableau IV décrit les taux d'attaques détectées, ou vrais positifs (TVP) et de fausses alarmes, ou faux positifs (TFP) pour chaque modèle de base ainsi que pour le modèle empilé, obtenu en cumulant les alarmes levées par chacun des modèles de base. On observe que le modèle le plus performant en terme de TVP ne détecte pas certaines attaques que des modèles moins performants ont détectées. Le modèle empilé a un meilleur TVP mais lève plus de fausses alarmes. Le traitement de l'ensemble des alarmes levées par ce modèle empilé sera effectué par le méta-modèle Section IV-D.

D. Reconnaissance de schémas d'attaque

Dans la section précédente, nous avons attribué un score d'anomalie par modèle à chaque agrégat. Cette étape nous permet de générer un segment monochrome par modèle de base et par intervalle de temps. Nous superposons ensuite les segments de chaque couleur correspondant au même intervalle de temps Δ_t et les disposons côte à côte pour représenter le trafic sur une période de temps plus longue Δ_T.

Enfin, nous avons obtenu un jeu de données de 4214 paires d'images, qui représentent les scores d'anomalie des agrégats sur une période Δ_T. Ce jeu de données sera utilisé pour entraîner et évaluer le module de reconnaissance de schémas d'attaque (Section III-D).

Le Tableau V montre le F-score et la matrice de confusion de notre module de reconnaissance de schémas d'attaque, i.e., notre CNN, qui analyse des représentations d'agrégats par adresse IP source et destination (le modèle combiné). Pour déterminer si les deux représentations sont complémentaires pour le modèle, nous avons comparé la performance du modèle combiné avec celle des modèles qui traitent uniquement les représentations des agrégats avec une seule clé d'agrégation (adresse IP source ou destination). Nous observons dans le Tableau V que le modèle combiné a un meilleurs F-score que les deux autres modèles, il n'a levé aucun faux positif, cependant, il a détecté moins d'attaques réelles que le modèle qui ne regarde que les adresses IP de destination.

V. DISCUSSION

Nous avons proposé un système qui combine plusieurs détecteurs d'anomalies non supervisés en utilisant un CNN sur une représentation visuelle du trafic. Cette méthode de combinaison a permis d'améliorer significativement la performance de détection du système. Ce système génère plusieurs représentations intermédiaires qui peuvent aider à identifier les erreurs restantes.

Dans cet article, nous avons détecté des anomalies sur des agrégats par adresse IP source et destination pendant des intervalles de temps de 2 minutes. Cependant, ce niveau de granularité peut ne pas être adapté pour détecter certaines attaques. Par exemple, sur la Figure 4, on observe que nos

TABLEAU II – Répartition des agrégats par adresse IP source et destination pour chaque catégorie de trafic

	IP source		IP de destination	
Catégorie de trafic	**Nombre**	**Proportion (%)**	**Nombre**	**Proportion (%)**
Attaque par force brute	41	0.004455	97	0.009547
Déni de service	13	0.001412	70	0.004231
Attaque web	0	0	133	0.013089
Infiltration	32	0.003477	76	0.007480
Bot	1500	0.162978	0	0
Déni de service distribué	58	0.006302	62	0.006102
Bénin	918723	99.821376	1015651	99.956894
Total	**6516**	**100**	**3486**	**100**

TABLEAU III – 5 meilleures combinaisons de 3 modèles de base pour les agrégats par source et destination. Les combinaisons sont classées selon deux critères : le TPR, représentant le pourcentage d'attaques correctement détectées, et le TFP, indiquant la proportion de fausses alarmes.

IP source			IP de destination		
Sous-ensemble de modèles de base	**TVP**	**TFP**	**Sous-ensemble de modèles de base**	**TVP**	**TFP**
(LOF, OCSVM, COPOD)	0.9878	0.6763	(LOF, KNN, COPOD)	0.9384	0.3994
(IF, LOF, OCSVM)	0.9811	0.6704	(LOF, OCSVM, COPOD)	0.9384	0.4437
(LOF, OCSVM, KNN)	0.9732	0.6762	(LOF, OCSVM, KNN)	0.9315	0.4312
(LOF, DBSCAN, OCSVM)	0.9726	0.6620	(LOF, DBSCAN, COPOD)	0.9292	0.3917
(IF, OCSVM, COPOD)	0.9690	0.4937	(IF, LOF, COPOD)	0.9292	0.3939

TABLEAU IV – Taux d'attaques détectées (TVP) et de fausses alarmes (TFP) pour les modèles de base

IP source			IP de destination		
Modèle de base	**TVP**	**TFP**	**Modèle de base**	**TVP**	**TFP**
LOF	0.8923	0.3754	LOF	0.9041	0.3366
OCSVM	0.8394	0.4634	KNN	0.8904	0.1862
COPOD	0.7652	0.1467	COPOD	0.8744	0.1795
Modèle empilé	**0.9878**	**0.6763**	**Modèle empilé**	**0.9384**	**0.3994**

TABLEAU V – Résultats des CNNs

	TVP	**TFP**	**F-score**	**Matrice de confusion** [**VP FP** / **VN TN**]
CNN Source	0.9796	0.0062	0.9905	[96 2 / 2 322]
CNN Destination	0.9796	0.0093	0.9882	[96 3 / 2 321]
Combined CNN	**0.9898**	**0.0093**	**0.9906**	[**97 3** / **1 321**]

détecteurs d'anomalies n'ont pas détecté la totalité de l'attaque par force brute sur les agrégats par IP de destination. Cela peut s'expliquer par le fait que la machine victime reçoit des paquets malveillants agrégés avec des paquets légitimes, ce qui rend cette attaque moins identifiable au niveau des agrégats par adresse IP de destination.

De plus, nous avons sélectionné la combinaison de modèles de base qui détecte le plus grand nombre d'attaques tout en minimisant les fausses alarmes. Cependant, il est important de souligner que cette combinaison est statique et que le critère de sélection des modèles est conservateur, ce qui peut entraîner un grand nombre de fausses alarmes dans les résultats. Dans certains cas très spécifiques, un détecteur ayant une performance inférieure peut être le seul à détecter correctement une attaque. Il pourrait être intéressant d'explorer une sélection dynamique des modèles, qui interrogerait ce détecteur uniquement lorsque cela est nécessaire pour obtenir une détection précise.

Pour finir, nous avons observé la présence d'un à la fin d'une attaque. Notre système pourrait bénéficier des techniques d'analyse de vidéos pour détecter les attaques dans leur intégralité.

Dans nos travaux futurs, nous évaluerons notre système sur un jeu de données plus volumineux et évaluerons sa capacité à détecter de nouvelles attaques.

VI. CONCLUSION

Nous avons proposé un système de détection d'attaques réseau conçu pour être explicable. Dans notre approche, nous avons utilisé des techniques non supervisées pour détecter des anomalies sur des agrégats basés sur les adresses IP source et destination. Les résultats produits par notre système sont interprétables et compréhensibles pour les analystes de sécurité. Les sorties incluent les adresses IP associées aux agrégats identifiés comme anormaux, ce qui permet aux analystes de les relier à des machines spécifiques dans le

réseau. De plus, les couples de caractéristiques anormaux sont également rapportés, ce qui fournit ainsi un niveau de détail supplémentaire sur les anomalies détectées. Pour représenter ces anomalies, nous avons utilisé des images générées à partir d'un ensemble de modèles de base non supervisés. Ces images permettent aux analystes de sécurité d'observer visuellement les schémas d'attaque identifiés par le système. Enfin, nous avons analysé ces représentations du trafic en utilisant un CNN pour détecter des schémas d'attaque.

Notre approche vise donc à concevoir un système de détection d'attaques plus transparent, qui permet aux analystes de sécurité de suivre les décisions prises par le système. L'évaluation de notre système sur une partie du jeu de données CIC-CSE-IDS2018 [19] a démontré une précision correcte, mais surtout, les erreurs commises par le système sont facilement identifiables et peuvent être analysées par les analystes de sécurité.

Une perspective prometteuse est la détection de nouvelles attaques. Comme les attaquants développent constamment de nouvelles attaques, il est essentiel de développer des systèmes capables de détecter et de caractériser ces attaques inconnues. Dans notre système, nous avons utilisé des modèles non supervisés pour détecter les anomalies. Il serait désormais intéressant d'évaluer la capacité de notre système à identifier des schémas d'attaques inconnus.

RÉFÉRENCES

[1] J. Vanerio and P. Casas, "Ensemble-learning Approaches for Network Security and Anomaly Detection," in *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. Los Angeles CA USA : ACM, Aug. 2017, pp. 1–6.

[2] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," *IEEE Access*, vol. 7, pp. 82 512–82 521, 2019.

[3] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You ?" : Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA : Association for Computing Machinery, Aug. 2016, pp. 1135–1144.

[4] D. Han, Z. Wang, W. Chen, Y. Zhong, S. Wang, H. Zhang, J. Yang, X. Shi, and X. Yin, "DeepAID : Interpreting and Improving Deep Learning-based Anomaly Detection in Security Applications," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. Virtual Event Republic of Korea : ACM, Nov. 2021, pp. 3197–3217.

[5] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, and F. J. Aparicio-Navarro, "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems*, vol. 89, pp. 349–359, Dec. 2018.

[6] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, Jan. 2017, pp. 712–717.

[7] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune : An Ensemble of Autoencoders for Online Network Intrusion Detection," in *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA : Internet Society, Feb. 2018.

[8] P. Zhou, G. Zhou, D. Wu, and M. Fei, "Detecting multi-stage attacks using sequence-to-sequence model," *Computers & Security*, vol. 105, p. 102203, Jun. 2021.

[9] "Snort - Network Intrusion Detection & Prevention System," https ://www.snort.org/.

[10] I. Ghafir, K. G. Kyriakopoulos, S. Lambotharan, F. J. Aparicio-Navarro, B. Assadhan, H. Binsalleeh, and D. M. Diab, "Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats," *IEEE Access*, vol. 7, pp. 99 508–99 520, 2019.

[11] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," in *ICIW2011-Proceedings of the 6th International Conference on Information Warfare and Secuirty : ICIW*. Academic Conferences Limited, 2011, pp. 113–125.

[12] D. McWhorter, "Mandiant Exposes APT1 — One of China's Cyber Espionage Units & Releases 3,000 Indicators," *Mandiant, February*, 2013.

[13] F. Wei, H. Li, Z. Zhao, and H. Hu, "xNIDS : Explaining deep learning-based network intrusion detection systems for active intrusion responses," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA : USENIX Association, Aug. 2023, pp. 4337–4354.

[14] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[15] P. Casas, J. Mazel, and P. Owezarski, "UNADA : Unsupervised Network Anomaly Detection Using Sub-space Outliers Ranking," in *10th IFIP Networking Conference (NETWORKING)*, vol. LNCS-6640. Springer, May 2011, pp. 40–51.

[16] J. Dromard, G. Roudière, and P. Owezarski, "Online and Scalable Unsupervised Network Anomaly Detection Method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, Mar. 2017.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, "Scikit-learn : Machine Learning in Python," *MACHINE LEARNING IN PYTHON*, p. 6, 2011.

[18] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD : A Python Toolbox for Scalable Outlier Detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019.

[19] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization :," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. Funchal, Madeira, Portugal : SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116.

[20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *2008 Eighth IEEE International Conference on Data Mining*, Dec. 2008, pp. 413–422.

[21] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF : Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA : Association for Computing Machinery, May 2000, pp. 93–104.

[22] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support Vector Method for Novelty Detection," in *Advances in Neural Information Processing Systems*, vol. 12. MIT Press, 1999.

[23] F. Angiulli and C. Pizzuti, "Fast Outlier Detection in High Dimensional Spaces," in *European conference on principles of data mining and knowledge discovery*, T. Elomaa, H. Mannila, and H. Toivonen, Eds. Berlin, Heidelberg : Springer, 2002, pp. 15–27.

[24] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "COPOD : Copula-Based Outlier Detection," in *2020 IEEE international conference on data mining (ICDM)*. IEEE, Nov. 2020.