UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR

KOSAYLA MAMMERI

DÉTECTION ET CLASSIFICATION DES ATTAQUES DDOS DANS LES RÉSEAUX SDN PAR APPRENTISSAGE MACHINE

JUIN 2024

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

CE MÉMOIRE A ÉTÉ ÉVALUÉ PAR

UN JURY COMPOSÉ DE

▪ **Boucif Amar Bensaber,** directeur de recherche.
Professeur au département de Mathématiques et Informatique
Université du Québec à Trois-Rivières.

▪ **Ismail Biskri,** évaluateur.
Professeur au département de Mathématiques et Informatique
Université du Québec à Trois-Rivières.

▪ **Mhamed Mesfioui,** évaluateur.
Professeur au département de Mathématiques et Informatique
Université du Québec à Trois-Rivières.

À Dieu,
À ma chère mère,
À la mémoire de mon père,
À ma famille,
À ma chère femme,
À toute la famille Mammeri,
À tous ceux que j’aime.

Kosayla

REMERCIMENTS

Je tiens tout d'abord à exprimer ma reconnaissance envers Dieu, le Tout-Puissant, pour m'avoir accordé la force et le courage nécessaires pour mener à bien ce travail.

Ensuite, je souhaite exprimer ma profonde gratitude envers le professeur Amar Bensaber Boucif, mon directeur de recherche, pour son soutien constant, sa disponibilité et ses encouragements tout au long de l'élaboration de ce mémoire. Sa patience et son écoute attentive ont été des atouts précieux, et j'ai eu la chance d'apprendre de lui non seulement sur le plan scientifique, mais aussi sur le plan humain et relationnel. Il est difficile de mesurer la valeur de ce que j'ai reçu de Monsieur Boucif, mais je lui suis infiniment reconnaissant pour tout ce qu'il a fait pour moi.

La réalisation de ce mémoire a été une expérience extrêmement enrichissante pour moi, suscitant en moi le désir de poursuivre mes recherches. Cette expérience restera gravée dans ma mémoire pour toujours. Bien que je pourrais continuer à exprimer ma gratitude de diverses manières, je souhaite simplement dire : Monsieur Boucif, merci infiniment pour votre soutien inestimable.

Je tiens également à exprimer ma reconnaissance envers les professeurs qui ont minutieusement évalué ce mémoire, contribuant ainsi à son amélioration.

Mes remerciements vont également à tous les enseignants du département de mathématiques et informatique, dont les connaissances précieuses ont enrichi ma formation académique et resteront gravées dans ma mémoire.

Enfin, je souhaite exprimer ma gratitude envers ma famille et tous ceux qui m'ont soutenu et encouragé tout au long de mon parcours académique, qu'ils soient proches ou éloignés.

RÉSUMÉ

Le Réseau Défini par Logiciel (SDN) représente un nouveau paradigme de réseau qui offre une architecture dynamique et facilement gérable pour les réseaux de nouvelle génération. Il sépare la logique de contrôle du réseau des équipements d'interconnexion, favorisant ainsi la centralisation du contrôle et la programmabilité du réseau. Malgré les avantages significatifs en termes de flexibilité et de gestion des réseaux qu'il offre, le SDN s'expose également à de nouvelles vulnérabilités ainsi qu'à divers vecteurs d'attaques, notamment les attaques par déni de service (DoS) et déni de service distribué (DDoS). Le contrôleur, en tant que centre névralgique du réseau SDN, représente une cible potentielle pour ces attaques en raison de la surcharge de requêtes malveillantes, perturbant ainsi le fonctionnement normal du réseau.

Au cours de notre étude, nous proposons une analyse approfondie du Software-Defined Networking (SDN), en examinant son architecture ainsi que ses interfaces de communication. Nous offrons également une explication détaillée du protocole OpenFlow, en mettant en lumière son fonctionnement, les systèmes de détection d'intrusions (IDS), ainsi que les différentes approches pour la détection des attaques par déni de service distribué. Notre étude se concentre principalement sur les méthodes basées sur l'apprentissage machine pour atteindre ces objectifs. Notre étude a pour objectif de détecter et classifier le trafic SDN en trafic normal ou en trafic d'attaque en utilisant des algorithmes d'apprentissage machine et profond avec et sans sélection de caractéristiques à l'aide des algorithmes suivants : Régression Logistique (LR), k-Nearest Neighbor (kNN), Decision Tree (DT), Random Forest (RF) et Support Vector Machine (SVM). Nous explorons également l'application de l'algorithme éliminateur récursive des fonctionnalités (RFE) pour sélectionner les caractéristiques les plus pertinentes du jeu de données "DDoS attack SDN Dataset". Ce dernier comprend un ensemble de 23 caractéristiques, couvrant à la fois les flux normaux et les attaques utilisant les protocoles TCP, UDP et ICMP.

ABSTRACT

Software-Defined Networking (SDN) represents a new network paradigm that provides a dynamic and easily manageable architecture for next-generation networks. It separates the network control logic from the interconnection equipment, thereby promoting centralized control and network programmability. Despite the significant advantages in terms of network flexibility and management it offers, SDN also exposes new vulnerabilities and various attack vectors, including Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. The controller, as the central nervous system of the SDN network, represents a potential target for these attacks due to the overload of malicious requests, thus disrupting the normal operation of the network.

During our study, we provide an in-depth analysis of Software-Defined Networking (SDN), examining its architecture and communication interfaces. We also offer a detailed explanation of the OpenFlow protocol, highlighting its functioning, intrusion detection systems (IDS), and various approaches to detecting distributed denial-of-service (DDoS) attacks. Our study primarily focuses on machine learning methods to achieve these objectives. Our study aims to detect and classify SDN traffic into normal or attack traffic using machine learning and deep learning algorithms with and without feature selection. We utilize Logistic Regression (LR), k-Nearest Neighbor (kNN), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM) algorithms. Additionally, we explore the application of Recursive Feature Elimination (RFE) algorithm to select the most relevant features from the "DDoS attack SDN Dataset." This dataset comprises 23 features, encompassing both normal flows and attacks using TCP, UDP, and ICMP protocols.

TABLE DES MATI ÈRES

LISTE DE FIGURES

LISTE DES TABLEAUX

LISTE DES ABREVIATIONS

SDN: Software-defined network.
DDoS / DOS: Distributed /Denial of Service.
OVS: Open vSwitch.
IDS: Intrusion Detection system.
HIDS/ NIDS: Host/Network Intrusion Detection system.
DNS: Domain Name System.
NIB: Network Information Base.
OVSDB: Open vSwitch Database.
OF-CONFIG: OpenFlow Configuration and Management Protocol.
VoD: Video on Demand.
VSDN/ CSDN: Virtual/ Cellular Software-Defined Network.
QoS: Quality of Service.
SLA: Service Level Agreement.
TCP / UDP: Transmission Control Protocol. / User Datagram Protocol.
IOT: Internet of things.
SCTP: Stream Control Transmission Protocol.
SYN: Synchronization segment in the TCP.
ACK: Acknowledge in the context of the TCP.
MAC: Media Access Control.
MPLS: Multiprotocol Label Switching.
TLS: Transport Layer Security.
VLAN: Virtual Local Area Network.
DL : Deep learning.
ML : Machine learning.
DT : Arbres de décision.
SVM : Machines à vecteurs de support.
LR : régression logistique.
KNN : k plus proches voisins.
ANN : Réseau de Neurones Artificiels.
RF : Forêt d'Arbres Aléatoires.
NN : Réseaux neuronaux.
ONOS : Open Network Operating System.
KPCA : Kernel Principal Component Analysis.
GA : Algorithme Génétique.
RMSE : Root Mean Square Error.
CART : Classification and Regression Trees.
ID3 : Iterative Dichotomiser 3.
RFE : Recursive Feature Elimination.
NCA : Neighborhood Component Analysis.

CHAPITRE1. INTRODUCTION GÉNÉRALE

Ces dernières années, le développement et la mise en œuvre des réseaux définis par logiciel ou Software Defined Networking (SDN) ont connu une croissance significative offrant une approche programmable et adaptable par rapport aux systèmes de réseau traditionnels qui reposent sur des appareils fixes dédiés tels que les commutateurs et les routeurs pour contrôler le trafic réseau. Les limitations des fonctionnalités non programmables, la sécurité médiocre des réseaux, et les problèmes de performance des systèmes traditionnels ont créé de nouveaux défis pour les futurs systèmes d'information et de communication basés sur Internet. De plus, la complexité des réseaux traditionnels rend difficile la reconfiguration efficace du réseau pour résoudre les problèmes de pannes, d'équilibrage de charge, et d'erreurs. C'est là que le SDN émerge comme une solution, en prenant le contrôle des réseaux configurés manuellement de manière traditionnelle, et permettant une utilisation plus efficace de l'infrastructure physique du réseau.

La séparation du plan de contrôle et du plan de données permet la virtualisation du réseau et des configurations dynamiques programmables et améliore les performances. Les paquets de données sont acheminés par le plan de données, également appelé plan de transfert, tandis que le plan de contrôle est responsable du routage des paquets de la source à la destination. Les opérateurs réseau et les fournisseurs de services peuvent gérer et contrôler directement leurs ressources et leurs réseaux grâce au SDN, qui est basé sur des topologies de réseaux centralisées facilitant la gestion des ressources. Ceci est possible sans nécessiter de connaissances approfondies en matière de technologie matérielle [1].

Le SDN offre une grande flexibilité, une automatisation et une intégration de services pour prendre en charge des conceptions de réseau innovantes. Cependant, il soulève également des questions de sécurité, notamment en ce qui concerne la protection des réseaux contre les menaces et les attaques potentielles, notamment les attaques par dénis de service DoS et les attaques par dénis de service distribuées (DDos). Ces dernières visent les ressources réseau et les services afin de les rendre indisponibles ou inaccessibles aux utilisateurs. Les attaques DDoS se produisent en envoyant du trafic malveillant ou en submergeant les systèmes avec la participation de deux utilisateurs ou plus, de nombreux utilisateurs ou même des robots, entraînant l'exploitation de la capacité de la table de flux, de la mémoire des commutateurs, des paquets IP falsifiés reçus, et la liaison entre le contrôleur et le commutateur devient congestionnée en raison du trafic malveillant intense sur le contrôleur. Tandis que les attaques DoS surviennent en submergeant le trafic par une seule personne ou un seul système [2].

Avec le développement des connaissances expertes et le fonctionnement progressif du réseau SDN mis en œuvre, le trafic normal et anormal augmente progressivement. De nouvelles attaques réseau apparaissent plus fréquemment. Par conséquent, des systèmes de détection d'intrusions (IDS) sont conçus et des technologies de sécurité qui surveillent le trafic réseau ou les événements système à la recherche de signes d'activité malveillante ou de violations de la politique de sécurité, et qui alertent le personnel de sécurité dès que de telles activités sont découvertes [3].

Dans ce contexte, notre travail a pour objectif d'étudier les réseaux basés sur le modèle Software-Defined Networking (SDN), leurs architectures ainsi que la sécurité en abordant les Systèmes de détection d'intrusions IDS, les attaques par dénis de service distribuées (DDoS). Nous allons présenter dans le cadre de ce travail de recherche une approche de détection et de classification du réseau malveillant en exploitant des méthodes d'apprentissage machine.

Dans le deuxième chapitre, nous présenterons les principes fondamentaux du paradigme SDN, Open Flow offrant une vue d'ensemble de son fonctionnement, sa vulnérabilité, les systèmes de détection d'intrusions ainsi que les attaques par déni de services distribuées (DDoS), leurs impacts sur les réseaux SDN, ainsi que les différentes approches de détection de ces attaques.

Dans le troisième chapitre, nous présenterons un état de l'art de plusieures recherches menées sur le sujet dans différents contextes, notamment plusieurs approches de détection qui ont été abordées. Nous présentons leurs caractéristiques ainsi que leurs limitations.

Dans le quatrième chapitre, nous discuterons de la méthodologie appliquée dans notre recherche, notamment les approches d'apprentissage machine et profond abordées, l'analyse de l'ensemble des données utilisées et l'algorithme de sélection de caractéristiques.

Dans le cinquième chapitre, nous procéderons à une analyse détaillée de nos diverses expérimentations, mettant particulièrement l'accent sur les résultats obtenus. Parallèlement, une comparaison sera établie avec les travaux connexes issus de la littérature.

Dans le sixième chapitre, nous conclurons notre travail en abordant les perspectives potentielles pour faire évoluer cette recherche et l'enrichir davantage.

CHAPITRE 2. PRÉSENTATION DU RÉSEAU DÉFINI PAR LOGICIEL

1 Introduction

Les systèmes de réseau et les centres de données actuels deviennent de plus en plus sophistiqués, complexes et dépendants des données en raison de l'évolution des réseaux informatiques. Par conséquent, les concepteurs de systèmes doivent fréquemment modifier les logiciels réseau et coordonner les ressources informatiques et réseau selon des exigences particulières. Cependant, les entreprises, les fournisseurs de services et les utilisateurs finaux ne peuvent pas répondre aux exigences des architectures réseau traditionnelles. Par exemple, l'ajout de nouveaux dispositifs ou services réseau est fastidieux car la capacité de prise de décision des réseaux hérités est répartie entre différents composants du réseau [5]. Ainsi, la gestion et la configuration des réseaux deviennent de plus en plus ardues et propices aux erreurs.

Afin de surmonter ces limitations, la communauté de recherche en réseau a développé le réseau défini par logiciel (SDN), dans lequel le contrôle du réseau est séparé du mécanisme de transfert et peut être directement programmable [6]. Le SDN permet de rendre davantage de composants du système réseau programmables en dissociant la logique de contrôle des dispositifs réseau individuels et en la transférant vers des dispositifs informatiques accessibles. Cette approche offre des contrôles unifiés aux applications, ce qui facilite et rend plus flexible la conception de nouvelles fonctions et protocoles réseau par les chercheurs, les concepteurs de systèmes et les administrateurs.

Au cours de ce chapitre, nous allons revenir aux origines de cette évolution et analyser les facteurs qui ont conduit à l'émergence de ce paradigme. Ensuite, nous examinerons de manière approfondie les principes de fonctionnement du SDN, OpenFlow, des IDS, ainsi que les attaques DDoS et leurs applications les plus importantes.

2 Le Software Defined Networking

2.1 Définition du SDN

Selon la définition officielle de l'Open Networking Foundation (ONF), le Réseau Défini par Logiciel (SDN) est une architecture réseau émergente qui sépare le plan de contrôle du plan de transfert et le rend directement programmable. Ce déplacement du contrôle, autrefois étroitement lié aux dispositifs individuels du réseau, vers des dispositifs informatiques accessibles permet de séparer l'infrastructure sous-jacente des applications et des services réseau. Cela permet aux applications et aux services de traiter le réseau comme une entité logique ou virtuelle [6]. Un réseau SDN se compose des caractéristiques suivantes :

- Un plan de contrôle et un plan de données séparés.
- Un Réseau Open Source avec un contrôle centralisé.
- Une Virtualisation et une automatisation.

2.2 Architecture globale du SDN

Un réseau SDN se compose de trois couches interconnectées via des interfaces API :

2.2.1 La couche application

Elle représente le sommet de la structure SDN, étant responsable de définir le fonctionnement idéal du réseau. Les applications intégrées à cette couche peuvent englober des outils d'optimisation de trafic, des politiques de sécurité et des superpositions de réseaux virtuels.

2.2.2 La couche contrôle

La couche de contrôle a pour fonction d'appliquer les politiques et les règles définies au niveau de la couche application. Elle se présente généralement sous forme d'un contrôleur central qui communique avec les périphériques du réseau et qui gère le plan de données de manière réactive ou proactive en intégrant diverses politiques de transfert. Elle représente en quelque sorte le cerveau du réseau, où sont effectuées la plupart des opérations de calcul. Le Tableau 1 résume les caractéristiques des contrôleurs SDN les plus connus.

Contrôleur	**OpenFlow**	**Langage**	**Multitâches**	**Distribué**	**Débit K flux/s**	**Complexité de développement**
RYU	1.0, 1.1, 1.3, 1.4, 1.5	Python	Non	Non	33	Moyen
NOX	1.0, 1.1, 1.2, 1.3	C++	Non	Non	30	Moyen
POX	1.0	Python	Non	Non	30	Facile
Floodlight	1.0, 1.1, 1.2, 1.3, 1.4	Java	Oui	Oui	1500	Difficile
OpenDayLight	1.0, 1.3	Java	Oui	Oui	NA	Difficile

Tableau 1 Caractéristiques des contrôleurs SDN

2.2.3 La couche infrastructure

Elle englobe les équipements réseau physiques, tels que les commutateurs et les routeurs, qui constituent le plan de données. Ces dispositifs sont chargés de véhiculer le trafic réseau à travers le réseau en assurant sa transmission.

2.3 Les interfaces de communication

Il est possible d'identifier trois types d'interfaces principales (voir Fig.1) qui permettent aux contrôleurs de réseau de communiquer avec leur environnement : l'interface Sud, l'interface Nord et l'interface Est/Ouest.

2.3.1 L’interface SUD (Southbound)

L'API Sud a pour fonction de définir l'ensemble des instructions et des protocoles de communication dédiés aux dispositifs de transfert. Elle constitue une couche d'interface inférieure qui définit les spécifications de protocole et permet de segmenter le concept de réseau en descriptions techniques concises. Par ailleurs, elle assure la gestion et la communication entre le contrôleur centralisé et les équipements du réseau tels que les commutateurs ou les routeurs. Elle permet aux dispositifs du réseau du plan de données de découvrir les différentes topologies du réseau et d'identifier les flux de données dans le réseau [1].

2.3.2 L’interface Nord (Northbound)

Les interfaces de programmation de réseau (NBI) basées sur le SDN entre les contrôleurs et les applications offrent des vues du réseau et permettent une expression directe du comportement et des exigences du réseau [1]. Elles sont utilisées pour programmer les éléments de la transmission en tirant parti de l'abstraction du réseau fournie par le plan de contrôle. Autrement dit, elles facilitent la communication entre le contrôleur et la couche applicative [7].

2.3.3 L’interface EST/OUEST (Est/West)

Ces interfaces sont utilisées pour la communication inter-contrôleurs dans les architectures distribuées (multi-contrôleurs). Elles permettent la synchronisation des états du réseau entre les différents contrôleurs [8]. Actuellement, il n'existe aucun standard disponible pour ce type d'interfaces.

2.4 Les contrôleurs SDN

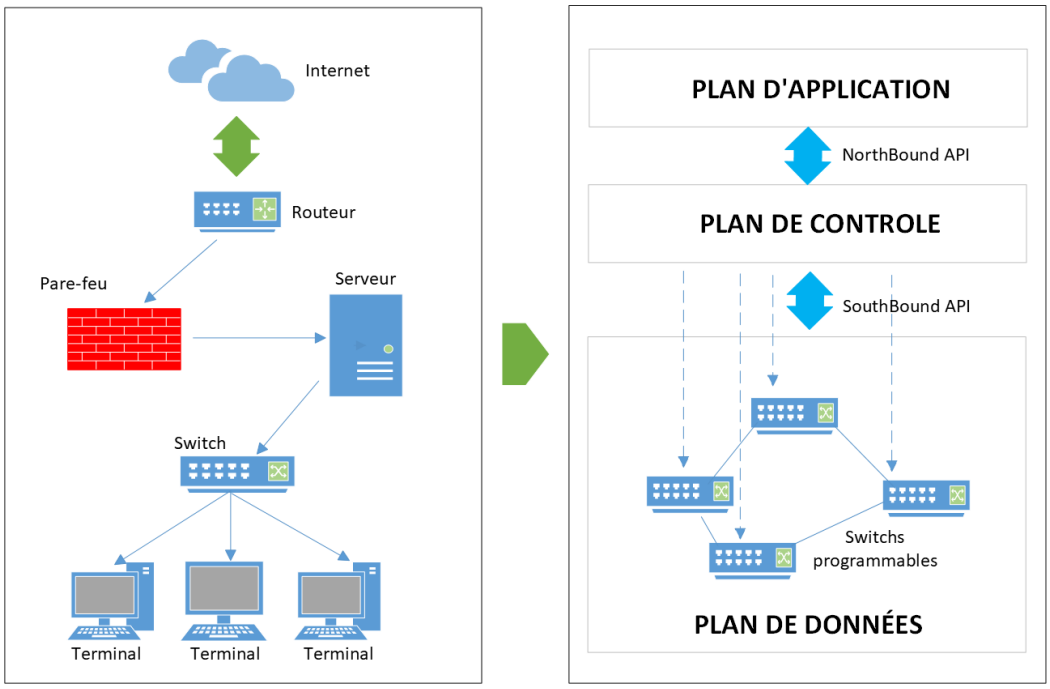
Le plan de contrôle a pour rôle de gérer et de contrôler les équipements de l'infrastructure, ainsi que de les connecter aux applications. Il est composé d'un ou de plusieurs contrôleurs (voir Fig.2) et il est considéré comme le système d'exploitation du réseau. Dans les premières versions du SDN, le plan de contrôle était constitué d'un seul contrôleur centralisé. Cependant, des architectures distribuées utilisant plusieurs contrôleurs ont été proposées par la suite pour améliorer les performances et la scalabilité du réseau. Les performances des contrôleurs SDN sont caractérisées par le débit, qui correspond à la quantité de flux traités par seconde et la latence, qui est le temps nécessaire à l'installation d'une nouvelle règle [8].

Pour interagir avec le réseau, le contrôleur doit disposer d'une vue précise de ce dernier. C'est ainsi que le concept de base d'informations de réseau (NIB - Network Information Base) a été développé afin de permettre le stockage de l'état global du réseau, comprenant sa topologie et les flux en cours. Une NIB est hiérarchisée en plusieurs couches [9]. Si un flux est destiné à un autre hôte dans le même domaine, on parle de flux intra-domaine et il est facile d'utiliser l'ingénierie du trafic au niveau du contrôleur local pour trouver les chemins en lisant la NIB locale.

2.5 Comparaison entre les réseaux traditionnels et SDN

Dans le réseau informatique traditionnel, les équipements de réseau tels que les commutateurs et les routeurs contrôlent le flux de données et sont divisés en trois plans :

1. Le plan de transfert de données, qui gère le flux de paquets.
2. Le plan de contrôle, qui s'occupe de l'algorithme de routage.
3. Le plan de gestion, qui configure les activités de base du réseau.



Réseau traditionnel Réseau SDN

Figure 1 Comparaison entre les réseaux traditionnels et SDN

Les réseaux traditionnels étaient basés sur le matériel, ce qui les rendait historiquement stables et fiables. Dans ce type de réseau, la capacité opérationnelle peut être rapidement récupérée après une perte de puissance, et il peut être exploité dans différents environnements [1]. Cependant, ces réseaux ne sont pas capables de répondre aux demandes croissantes, car tous les composants du réseau sont intégrés verticalement pour former une structure complexe difficile à gérer. Les réseaux traditionnels ne peuvent supporter que des politiques spécifiques aux fournisseurs et n'offrent aucune flexibilité pour un environnement réseau dynamique [10].

Les réseaux traditionnels présentent plusieurs difficultés dans la réalisation d'expériences à grande échelle. En effet, ils étaient équipés de matériels et de réseaux coûteux

qui étaient installés par les chercheurs eux-mêmes. De plus, le taux d'innovation est lent car les protocoles sont définis avec un manque d'abstraction de haut niveau.

La technologie SDN sépare le plan de contrôle et de gestion du plan de données. (Voir Figure 1) Cette séparation permet une programmabilité réseau flexible. Le trafic peut être géré de manière conceptuellement centralisée sans altérer les données elles-mêmes. Ainsi, la complexité des commutateurs SDN est significativement réduite par rapport à d'autres types de réseaux [11], permettant à l'infrastructure sous-jacente d'être abstraite pour les applications et les services de réseau, c'est-à-dire la virtualisation des fonctions réseau telles que les commutateurs virtuels.

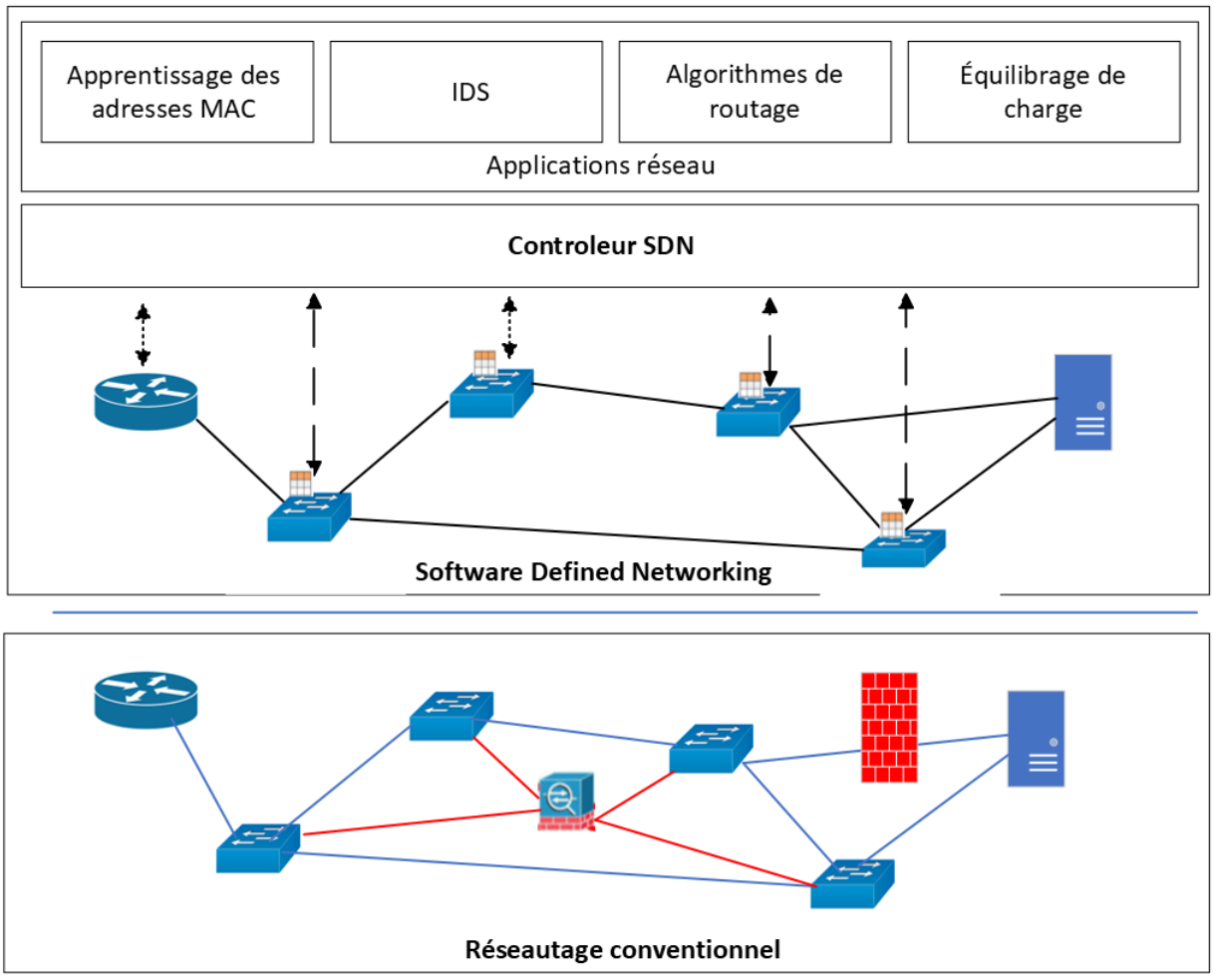


Figure 2 SDN vs Réseaux conventionnels

2.6 Les avantages du SDN

SDN adopte la séparation du plan de données et du plan de contrôle pour les réseaux. Cette séparation renforce le contrôleur réseau. De plus, dans le concept SDN, les applications réseau ne sont pas contraintes de gérer les configurations de bas niveau des dispositifs du plan de données et bénéficient d'une vue abstraite du réseau fournie par les contrôleurs [12]. Les avantages du SDN résident dans la programmabilité du trafic, la gestion du réseau basée sur des stratégies, la virtualisation et l'automatisation du réseau. Ce réseau offre l'assurance aux organisations de pouvoir s'adapter aux évolutions rapides. Il permet une conception de réseau adaptée aux applications gourmandes en données telles que les machines virtuelles et les réseaux de données volumineuses, comme le Big Data et la virtualisation [1].

Le transfert basé sur le flux dans les réseaux offre une flexibilité dans la gestion des priorités en routant différents flux d'applications vers des traitements spécifiques. Les règles de flux peuvent être mises à jour en temps réel par les opérateurs réseau sans perturber les opérations en cours des dispositifs réseau. Grâce au SDN, il est également possible d'analyser les flux et les paquets pour extraire les informations d'en-tête correspondantes, permettant au SDN d'avoir une vue d'ensemble du réseau. Cette approche facilite le suivi des états associés à un flux tout au long de son parcours, ainsi que la surveillance des statistiques du réseau à différents niveaux tels que par flux, par port, par protocole, et plus encore. Dans les réseaux SDN compatibles avec OpenFlow, la gestion des files d'attente et les opérations de planification peuvent également être réalisées grâce à des plugins supplémentaires orientés sud, tels que les protocoles OF-CONFIG et OVSDB [12].

2.7 Cas d'utilisation du SDN

2.7.1 QoS sur Internet.

Les services de nouvelle génération tels que le streaming vidéo, la vidéo à la demande (VoD) ou la visioconférence sont très sensibles aux retards et aux erreurs de transmission et nécessitent donc un réseau stable pour acheminer les paquets. Cependant, Internet a toujours été une architecture qui ne garantit pas cette stabilité. En utilisant la vue centralisée du réseau offerte par SDN, il est possible de sélectionner différents chemins pour les différents flux de trafic en fonction du débit. C'est ainsi qu'est né le concept de VSDN (Vidéo sur SDN), une architecture qui détermine le chemin optimal en se basant sur la vue globale fournie par le contrôleur [8].

2.7.2 Réseaux mobiles.

Le SDN permet aux opérateurs d'exploiter pleinement le potentiel de leur infrastructure et d'optimiser les débits au maximum, tout en conservant une visibilité essentielle sur le comportement de leurs équipements réseau. L'architecture CSDN (Cellular SDN) utilise les concepts SDN pour une exploitation dynamique et centralisée des ressources en collectant les données des utilisateurs et les conditions du réseau, puis les transmet au contrôleur. Ceci permet d'optimiser la consommation d'énergie, l'utilisation des ressources et la personnalisation des services pour les utilisateurs [15].

2.7.3 Réseaux d'entreprise.

Au-delà de la virtualisation, l'une des applications les plus intéressantes pour les entreprises est la programmation d'un routage sensible aux applications (Application aware routing). Avec le grand nombre d'applications fonctionnant sur les équipements des utilisateurs, telles que les réseaux sociaux et le streaming audio et vidéo, il en résulte souvent une surcharge du réseau pouvant entraîner une latence pour les applications déployées à des fins professionnelles ou académiques. Le SDN permet de gérer ces applications en leur attribuant des priorités différentes et en installant un équilibreur de charge (Load Balancer), ce qui élimine tout type de congestion. Cela offre ainsi aux professionnels un environnement de travail fluide.

2.7.4 Centre de données.

Dans les centres de données équipés de SDN, les ressources informatiques sont contrôlées par une plateforme de gestion de cloud telle que OpenStack, tandis que les ressources réseau sont gérées par un contrôleur SDN. Le module de gestion, également appelé orchestrateur, assure la coordination entre le contrôleur SDN et les services OpenStack en utilisant des interfaces API RESTfull orientées vers le nord. Cela permet d'effectuer la planification de la migration des machines virtuelles dans l'algorithme de gestion des ressources, tel que la stratégie d'économie d'énergie conforme aux accords de niveau de service (SLA) [13].

2.7.5 Sécurité.

Dans l'architecture SDN à trois couches, la couche de contrôle présente la plus grande vulnérabilité aux attaques réseau. Les attaques DDoS ciblant cette couche peuvent entraîner un point de défaillance unique pour l'ensemble du réseau SDN. Les applications tierces représentent une menace potentielle, car elles peuvent initier des attaques contre d'autres hôtes, telles que l'infiltration de données, la manipulation du réseau, les attaques de type DoS/DDoS, et bien d'autres. De plus, ces applications sont susceptibles de contenir un code défectueux qui pourrait permettre l'installation de portes dérobées sur le serveur réseau, donnant ainsi un accès non autorisé au réseau [14].

3 Le protocole OpenFlow

OpenFlow a été développé à l'Université de Stanford, grâce à un groupe de chercheurs connu sous le nom de Clean Slate Program. Leur objectif était d'explorer comment concevoir Internet si l'on partait de zéro, en tirant parti de 20 à 30 années de connaissances accumulées. En 2008, l'Open Networking Foundation (ONF) a publié OpenFlow, un protocole réseau qui permet la mise en œuvre de l'architecture des réseaux définis par logiciel (Software Defined Networking). Ce protocole offre la possibilité d'accéder directement au plan de données des périphériques réseau tels que les routeurs et les commutateurs [16]. OpenFlow est devenu la norme de facto pour les API sud utilisées dans les réseaux SDN. Il s'agit d'une approche dynamique, centralisée et programmable pour interagir avec les différents éléments de l'infrastructure.

3.1 Architecture du protocole OpenFlow

Le protocole OpenFlow établit la connexion entre un commutateur OpenFlow et un contrôleur (voir fig.3). Sa puissance réside dans un ensemble de messages qui sont échangés entre le contrôleur et le commutateur dans les deux directions. Ces messages permettent au contrôleur de maintenir le contrôle sur les commutateurs et le trafic des utilisateurs [17]. OpenFlow est caractérisé en tant que protocole de communication expérimental, émergent et ouvert, et il représente la première interface de communication standardisée établie entre le plan de contrôle et l'architecture de données des réseaux définis par logiciel (SDN). OpenFlow offre la possibilité d'accéder directement et de manipuler le plan de transfert des dispositifs réseau, qu'ils soient des commutateurs physiques ou des routeurs virtuels [18].

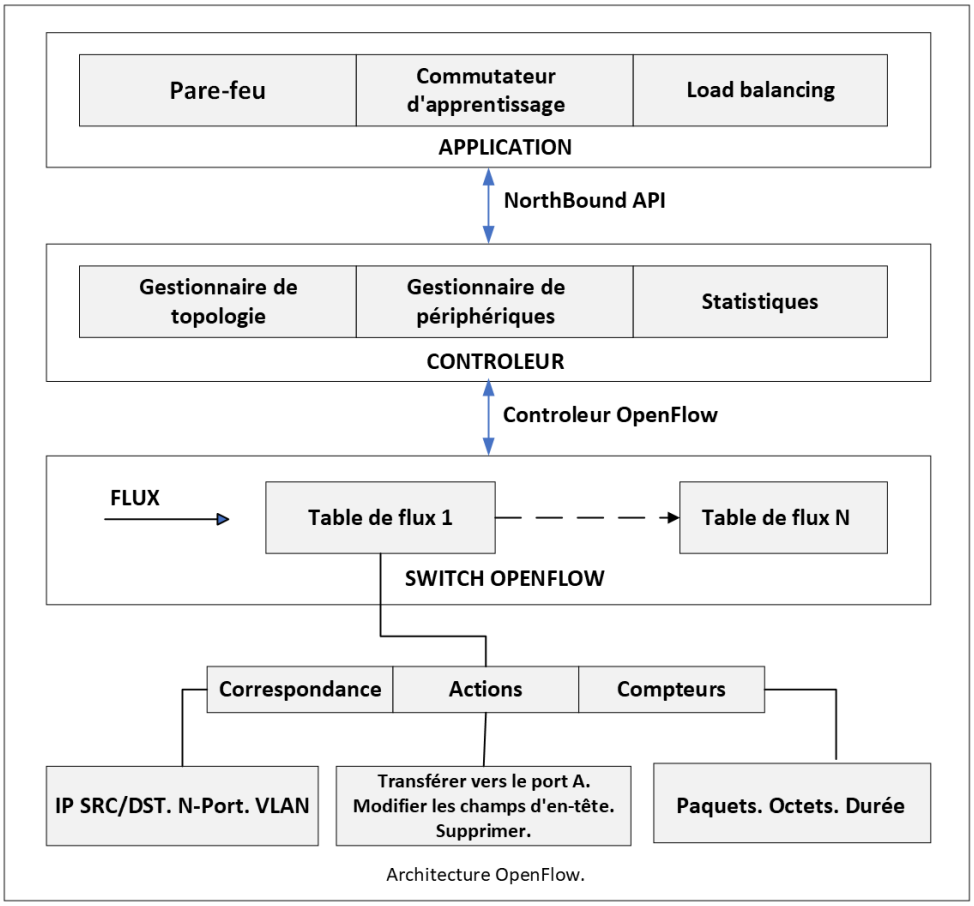


Figure 3 Architecture Openflow

3.2 Le Commutateur OpenFlow

Les commutateurs Ethernet et les routeurs les plus récents sont équipés de tables de flux qui permettent d'effectuer des fonctions de transfert basées sur les deuxièmes, troisièmes et quatrièmes couches, comme indiqué dans les en-têtes des paquets. OpenFlow utilise ces

fonctionnalités communes en tant que protocole entre un commutateur OpenFlow et un contrôleur central OpenFlow. Il est capable de programmer la logique de routage ou de transfert du commutateur. Les équipements du plan de données du réseau, également appelés commutateurs de manière générale, effectuent diverses tâches. Bien que chaque fournisseur ait sa propre table de flux, un certain nombre de fonctions communes sont présentes dans une variété de commutateurs et de routeurs.

On distingue deux principales fonctions des commutateurs :

La fonction de support du contrôle

Elle interagit avec la couche de contrôle SDN pour permettre la programmabilité à travers les interfaces de gestion des ressources. Le commutateur établit une communication avec le contrôleur, et celui-ci gère le commutateur en utilisant le protocole OpenFlow. Il peut être utilisé à la fois pour le contrôle et la gestion.

Fonction d'acheminement des données

Cette fonction permet d'accepter les flux de données entrants provenant d'autres équipements de réseau et des systèmes de terminaison, puis de les transmettre sur un chemin de commutation préalablement calculé et établi. Ce chemin est déterminé à partir des règles définies par les applications SDN, qui sont transmises au contrôleur et renvoyées vers le commutateur. Ainsi, la fonction d'acheminement des données assure le transfert efficace des données conformément aux directives établies par le contrôleur SDN.

Les règles d'acheminement des données sont stockées dans les tables de routage. Ces règles déterminent le prochain saut sur le chemin à suivre pour des catégories spécifiques de paquets de données. Le commutateur a la capacité de modifier l'en-tête du paquet avant de le faire passer ou de le rejeter. Comme illustré dans la figure 3, les paquets entrants sont placés dans une file d'attente en entrée, en attendant leur traitement par le commutateur. Une fois acheminés, les paquets sont placés dans une file d'attente en sortie avant d'être transmis.

Le commutateur est équipé de trois ports d'entrée/sortie : un port dédié à la communication de contrôle avec un contrôleur SDN et deux autres ports utilisés pour l'entrée et la sortie des paquets de données. Cet exemple est simplifié. En réalité, le commutateur peut comporter plusieurs ports permettant la communication avec plusieurs contrôleurs SDN, en plus des ports destinés aux paquets de données entrants et sortants du commutateur.

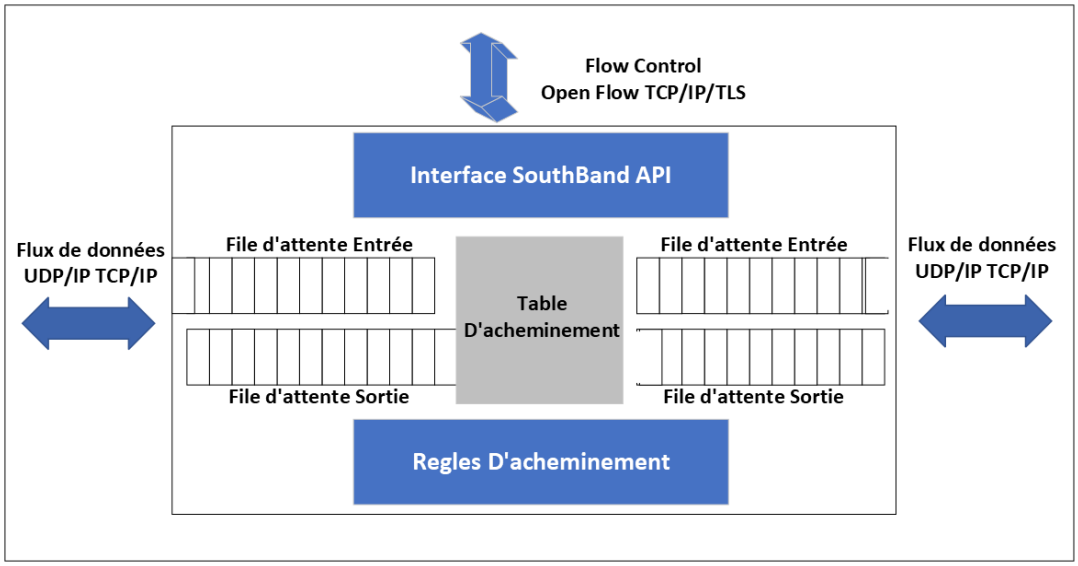


Figure 4 Commutateur OpenFlow

Les flux de données sont constitués de flux de paquets IP. Dans certaines situations, il peut être nécessaire pour la table de routage de configurer des entrées en fonction des champs d'en-tête des protocoles de couche supérieure tels que TCP, UDP, SCTP ou les protocoles d'application. Le commutateur examine l'en-tête IP, ainsi que d'autres en-têtes si nécessaire, de chaque paquet et prend une décision concernant son acheminement.

L'autre flux de données crucial se produit via l'interface sud, qui utilise le protocole OpenFlow ou tout autre protocole équivalent, bien que OpenFlow soit le protocole de référence. Un commutateur activé par OpenFlow joue le rôle de composant de transfert essentiel en acheminant les paquets en fonction de sa table de flux. Cette table de flux, similaire aux tables d'acheminement traditionnelles, n'est ni gérée, ni entretenue par le commutateur lui-même. Le commutateur est connecté au contrôleur via un canal sécurisé, par lequel des messages OpenFlow sont échangés entre le commutateur et le contrôleur. Il existe différentes versions des spécifications du protocole OpenFlow disponibles sur l'OpenFlow Switch.

Les fonctions principales du commutateur OpenFlow version 1.0 et sa relation avec le contrôleur sont les suivantes :

1. Recevoir les paquets entrants sur un port et les transmettre vers un autre port en effectuant les modifications nécessaires sur les paquets en cours de route.

2. La fonction de correspondance des paquets dans le commutateur OpenFlow est cruciale. Une table de flux est représentée sur la table adjacente. La flèche en pointillés sur le chemin commence dans la logique de décision, indique une

correspondance avec une entrée spécifique dans la table de flux et dirige le paquet correspondant vers une action spécifique dans la case à droite.

3. Les actions de traitement proposent trois options principales pour le paquet entrant : le transférer vers un port de sortie avec la possibilité de modifier certains champs d'en-tête du paquet, le supprimer complètement, ou l'envoyer encapsulé dans un message OpenFlow PACKET_IN vers le contrôleur.

4. Lorsque le chemin C est suivi, le paquet est transmis au contrôleur via un canal sécurisé. Si le contrôleur reçoit un message de contrôle ou un paquet de données à envoyer au commutateur, il utilise le même canal sécurisé dans la direction inverse. Lorsqu'un contrôleur doit transmettre un paquet de données via le commutateur, il utilise le message OpenFlow PACKET_OUT. Grâce à la logique OpenFlow, un paquet de données du contrôleur peut emprunter deux chemins distincts.

5. Un commutateur OpenFlow peut être pur ou hybride. Dans ce dernier cas, le commutateur peut traiter les paquets à la fois en mode OpenFlow et en mode conventionnel, ce qui le rend similaire à un commutateur Ethernet ou à un routeur IP typique. Le mode hybride fonctionne comme un commutateur OpenFlow qui fonctionne avec un commutateur traditionnel indépendant. Un mécanisme de classification est nécessaire pour ce type de commutateur hybride. Il permet de sélectionner et de diriger les paquets vers le contrôleur OpenFlow ou de les traiter de manière conventionnelle.

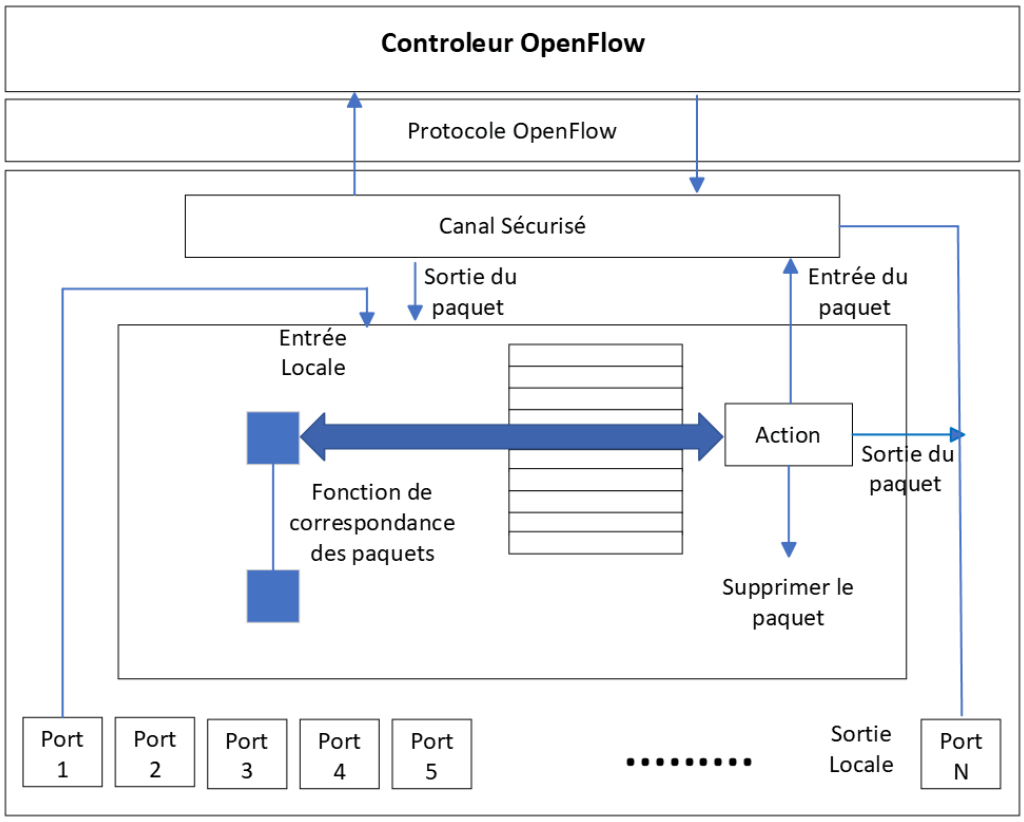


Figure 5 Les fonctions du Commutateur Openflow V1.0

3.2.1 Table de flux

La table de flux dans un commutateur OpenFlow contient un ensemble d'entrées de flux, qui sont composées de trois parties principales : les champs d'en-tête, les compteurs d'activité et un ensemble d'actions.

Champs d'en-têtes

Ces champs proviennent des en-têtes de paquets pour comparer les paquets entrants. Les informations des en-têtes du paquet sont utilisées pour identifier les flux et effectuer des correspondances avec les entrées dans la table de flux. Les champs d'en-tête peuvent inclure des informations telles que les adresses source et destination, les numéros de port, les étiquettes MPLS, etc.

Compteurs d'activité

Chaque entrée de flux peut avoir un compteur associé pour enregistrer le nombre de paquets et d'octets qui ont été traités en utilisant cette entrée de flux. Les compteurs d'activité permettent de surveiller le trafic et les performances du réseau.

Actions

Chaque entrée de flux peut spécifier un ensemble d'actions à appliquer aux paquets qui correspondent à cette entrée. Les actions peuvent inclure le transfert du paquet vers un port de sortie spécifique, la modification des champs d'en-tête du paquet, le rejet du paquet, ou l'envoi du paquet au contrôleur pour traitement.

Lorsque le commutateur reçoit un paquet, il le compare aux entrées de flux dans la table de flux pour trouver une correspondance. Si le paquet correspond à une entrée de flux, les actions associées à cette entrée sont exécutées sur le paquet. Si le paquet ne correspond à aucune entrée de flux, il peut être envoyé au contrôleur via le canal sécurisé, selon la configuration du commutateur.

3.2.2 Champs de correspondance.

Pour effectuer la recherche d'entrée de flux, diverses informations de l'en-tête du paquet sont utilisées, couvrant les couches 1 à 4 du modèle OSI. Le protocole OpenFlow propose des champs d'en-tête standardisés pour effectuer la correspondance, tels que les adresses MAC, les adresses IP, les numéros de port TCP/UDP, les étiquettes MPLS, etc. Les versions les plus récentes de OpenFlow peuvent également prendre en charge des champs d'en-tête pour IPv6 et d'autres protocoles réseau émergents.

Champs de correspondance	Actions	Compteurs

Tableau 2 Table de flux pour Openflow v1.0

3.2.3 Compteurs.

Dans un commutateur OpenFlow, le compteur est utilisé pour enregistrer le nombre de paquets qui ont été traités (c'est-à-dire ceux qui ont satisfait au prédicat) ainsi que la durée de vie de cette règle. Chaque règle dans la table de flux est associée à une priorité, et seule la règle avec la priorité la plus élevée qui correspond au paquet est prise en compte pour l'exécution des actions sur celui-ci. Cela permet de définir des actions par défaut qui ne s'appliquent aux paquets que si aucune autre règle ne peut être utilisée.

3.2.4 Actions.

Les actions dans un commutateur OpenFlow sont des opérations que le commutateur peut effectuer sur les paquets correspondants. Les actions les plus courantes comprennent le transfert du paquet vers un port de sortie spécifique, la suppression du paquet, la modification des champs d'en-tête du paquet, ou encore l'envoi du paquet au contrôleur pour traitement.

Le traitement en pipeline est également pris en charge dans les versions les plus récentes d'OpenFlow. Cela permet à un paquet d'être traité par plusieurs règles provenant de différentes tables de flux. Le commutateur suit alors le chemin défini par ces règles pour effectuer les actions appropriées sur le paquet. Il est important de noter que certains

commutateurs OpenFlow peuvent ne pas prendre en charge tous les types d'actions. Cependant, ils doivent obligatoirement supporter les actions nécessaires pour assurer le fonctionnement minimal d'OpenFlow.

3.3 Les messages openflow

En ce qui concerne les messages OpenFlow, la communication entre le contrôleur et le commutateur s'effectue via un canal sécurisé établi grâce à une connexion TLS sur TCP. Les messages sont regroupés en trois catégories : contrôleur-commutateur, symétriques et asynchrones.

3.3.1 Les messages contrôleur-commutateur

Ils sont les plus essentiels et comprennent des messages de configuration du commutateur, des commandes du contrôleur, des statistiques, des configurations de files d'attente et des barrières. Ils sont utilisés pour configurer le commutateur, envoyer des paquets de données au commutateur, modifier les entrées de flux, obtenir des statistiques du commutateur, etc.

3.3.2 Les messages symétriques

Ils peuvent être émis indifféremment par le contrôleur ou le commutateur sans avoir été sollicités par l'autre entité. Ils incluent des messages HELLO, ECHO, VENDOR et Error.

3.3.3 Les messages asynchrones

Ils sont envoyés du commutateur vers le contrôleur pour notifier un changement d'état ou l'arrivée d'un paquet. Ils incluent les messages Packet-in, Flow-removed, Port-status et Role-status.

Ces messages permettent au contrôleur de configurer le commutateur, de surveiller l'état du réseau, de prendre des décisions de transfert de paquets, etc., dans le cadre de l'architecture SDN basée sur OpenFlow.

3.4 Les spécifications Openflow

Version Openflow	Modifications apportées par la spécification
Version 1.0	- Les paquets Ethernet et IP sont distingués en fonction des adresses source et de destination, ainsi que des ports source et destination UDP et TCP. - Les champs de protocole, de DS (DiffServ) et d'ECN (Explicit Congestion Notification) sont utilisés au niveau de la couche 3. - Champs Ethernet-type et VLAN pour la couche 2.
Version 1.1	- Inclusion des champs d'identification MPLS. - Inclusion du pipeline, de la table de groupe et des métadonnées.
Version 1.2	- Maintenant, un commutateur peut être connecté à plusieurs contrôleurs en mode Maître/Esclave. - Le modèle OXM (OpenFlow Extensible Match) qui offre une prise en charge de l'IPv6 et une structure de correspondance plus souple.
Version 1.3	- Un meilleur support des contrôleurs multiples et l'ajout d'une table de mesure (Meter table) qui permet de mesurer le taux de paquets attribués à une entrée.
Version 1.4	- L'amélioration de l'OXM (OpenFlow Extensible Match) et de la structure TLV (Type-Length-Value) permettant d'obtenir une meilleure synchronisation et d'optimiser les performances.
Version 1.5.1	- Tables de sortie qui permettent de traiter les paquets à la fois sur le port de sortie et sur les ports d'entrée.

Tableau 3 Comparaison entre les versions d'OpenFlow

4 Les vulnérabilités dans les réseaux SDN

La mise en œuvre du SDN offre de nombreux avantages en termes de programmabilité, de gestion basée sur des stratégies, de virtualisation et d'automatisation du réseau. Cela permet aux organisations de s'adapter rapidement aux évolutions technologiques et aux besoins des applications gourmandes en données telles que le Big Data et la virtualisation.

Cependant, comme toute technologie émergente, le SDN présente également des vulnérabilités et des risques pour la sécurité. La séparation des plans de contrôle et de données ainsi que la centralisation logique de l'intelligence réseau créent un point de défaillance unique, susceptible d'être exploité pour compromettre l'ensemble du réseau SDN. Des erreurs de configuration ou des déploiements incorrects des composants SDN peuvent également créer des surfaces d'attaque potentielles.

Les vulnérabilités du SDN peuvent être classées en fonction de leurs objectifs, comme l'écoute, l'accès non autorisé, la modification non autorisée des informations du réseau, la destruction des informations du réseau et la perturbation du service. Ces vulnérabilités peuvent être exploitées par des attaquants pour compromettre la sécurité du réseau, accéder à des données sensibles, altérer les règles de flux, ou perturber les services réseau.

Pour atténuer ces risques, il est essentiel de mettre en place des mécanismes de sécurité solides, tels que des mécanismes d'authentification et de contrôle d'accès robustes, des communications chiffrées, et des politiques de sécurité bien définies. De plus, les fournisseurs de solutions SDN et les organisations doivent rester à jour sur les dernières vulnérabilités et correctifs de sécurité, et mettre en œuvre des mises à jour régulières pour réduire les risques liés aux failles de sécurité connues.

5 Les Systèmes De Détection Des Intrusions (IDS)

Un système de détection d'intrusion est une nouvelle génération de technologie de sécurité qui surveille un système d'une façon automatisé et chargé de reconnaître pour éviter les activités malveillantes. Il s'agit d'un système de sécurité qui détecte l'accès non autorisé ou les activités malveillantes dans un réseau ou système informatique [25].

5.1 Types de systèmes de détection d'intrusions

Deux catégories principales de systèmes de détection d'intrusions sont généralement distinguées : les IDS basés sur les signatures et les IDS basés sur les anomalies.

5.1.1 IDS à base de signature

Les approches basées sur les signatures examinent les flux réseau portant des caractéristiques d'attaques précédentes, similaire aux logiciels antivirus. Une séquence d'événements et de conditions associés à une tentative d'intrusion représentent ce flux. Ensuite, elles utilisent la correspondance de modèles pour identifier les concepts (voir fig.6). Si l'IDS est en mode actif, une alarme peut être déclenchée lorsqu'une menace est détectée, sinon, l'IDS se contentera de documenter l'attaque.

5.1.2 IDS à base d'anomalie

Les IDS basés sur des anomalies nécessitent une instruction initiale qui implique l'acquisition par l'outil d'une connaissance du comportement typique des flux applicatifs dans le réseau. Chaque flux et le comportement par défaut qu'il présente doivent être documentés. L'IDS alerte en cas de flux anormal, mais il n'est pas en mesure de déterminer la gravité potentielle de la menace. Les identifiants spécifiques au comportement ont été développés plus tard que les identifiants spécifiques à la signature, mais n'ont pas encore atteint leur plein développement (voir fig.6). Par conséquent, l'utilisation de cet IDS peut s'avérer difficile, les alertes étant susceptibles de contenir un grand nombre de fausses alertes [50].

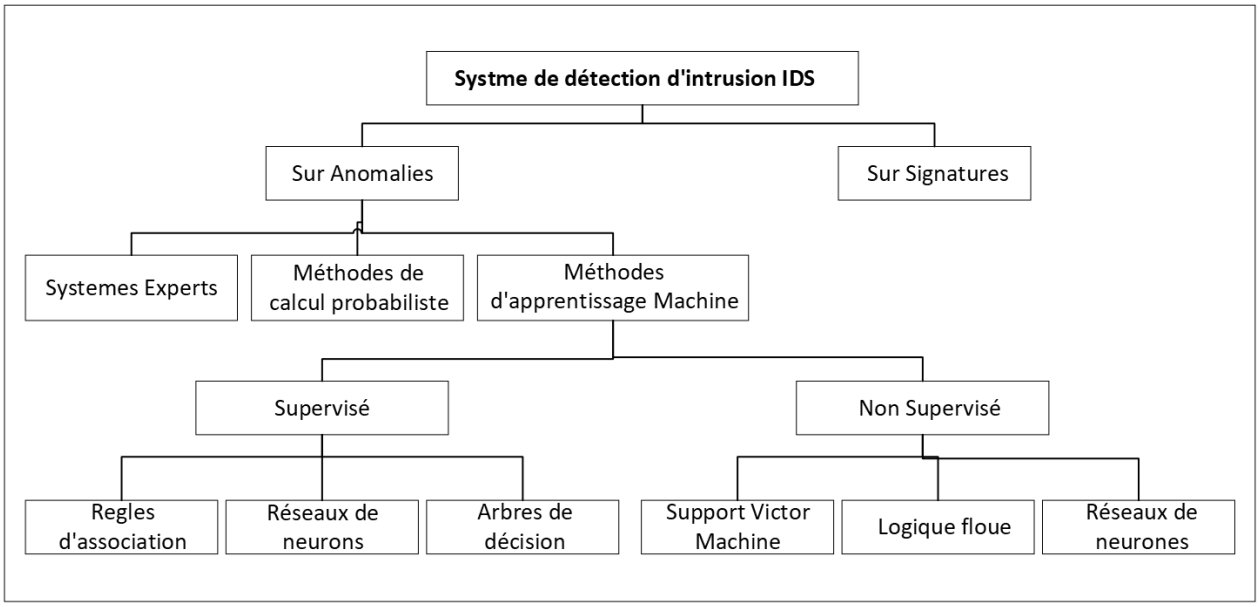


Figure 6 Techniques de détection d'intrusions

5.2 Architecture et fonctionnement des IDS

L'architecture des IDS fait référence à la conception globale et à la structure d'un système de détection d'intrusion (IDS) conçu pour un environnement réseau particulier, tel qu'un réseau ad hoc mobile (MANET) ou les réseaux SDN. Elle englobe les composants, modules et algorithmes utilisés pour détecter et réagir aux menaces de sécurité, ainsi que les méthodes de collecte, d'analyse et de corrélation des données liées à la sécurité provenant de diverses sources. Cette architecture peut adopter une approche hiérarchique, en grappes, ou s'appuyer sur des systèmes multi-agents, et elle doit être soigneusement conçue pour relever les défis spécifiques de sécurité et répondre aux exigences de l'environnement réseau concerné [26].

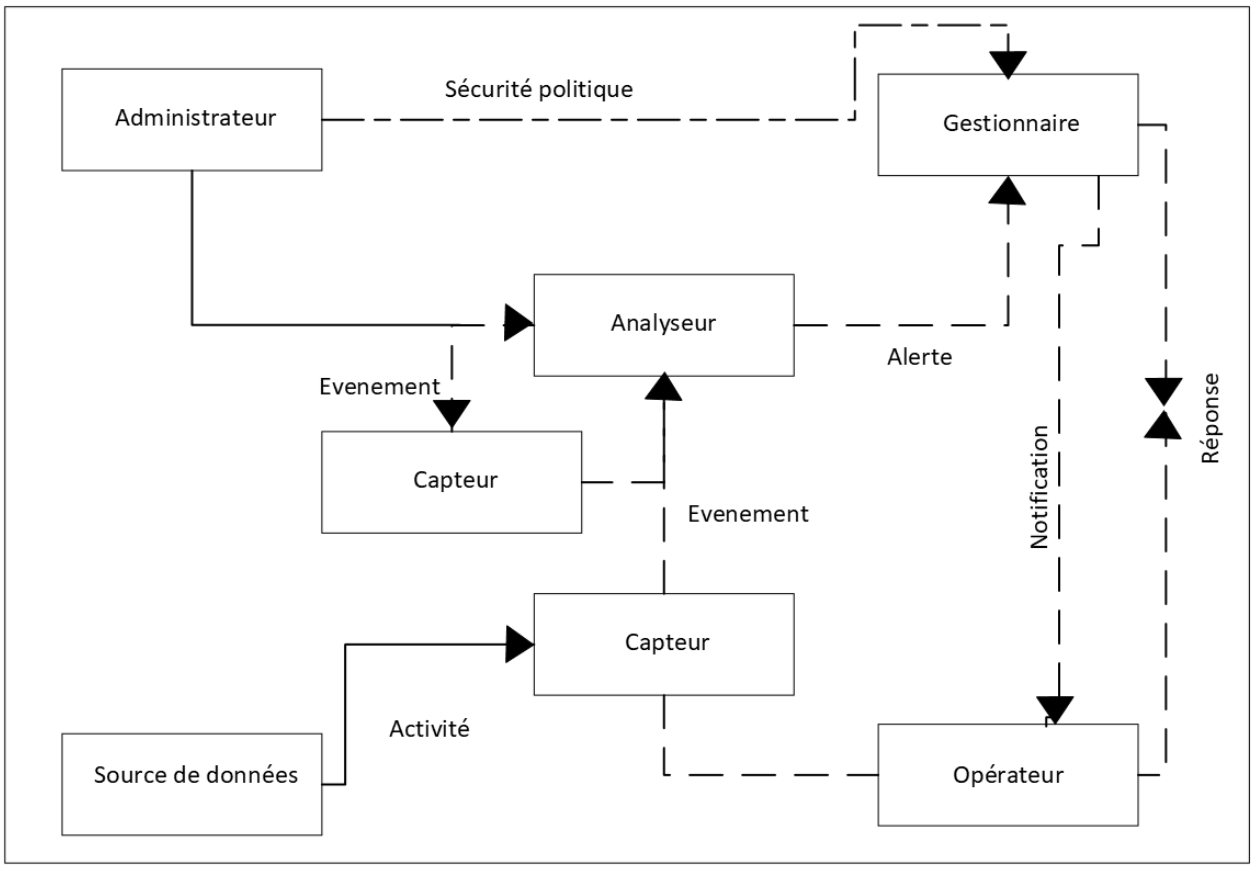


Figure 7 Modèle fonctionnel IDS proposé par l'IDWG [54]

- La source de données comprend divers types d'informations issues de multiples sources telles que le réseau, le système, les applications et les alertes. Le système IDS utilise des capteurs appropriés pour analyser les informations provenant de ces différentes sources afin de détecter les activités non autorisées ou indésirables.
- Le capteur acquiert les données brutes, qui incluent toutes les informations concernant l'activité en cours, celles-ci sont ensuite transmises à l'analyseur sous forme d'événements, définis comme des ensembles d'activités.
- L'analyseur est un composant essentiel du système dont l'administrateur de sécurité analyse ces événements en provenance du capteur en détail pour déterminer s'ils sont autorisés ou indésirables, ce qui suscitera l'intérêt de l'administrateur de sécurité. Dans la majorité des IDS existants, le capteur et l'analyseur sont combinés en une seule pièce.
- Le gestionnaire joue un rôle crucial pour fournir aux opérateurs une interface qui leur permet de superviser et de gérer les différentes parties du système. Les tâches du gestionnaire incluent généralement le paramétrage des capteurs, la configuration des

analyseurs, la notification des événements, l'intégration des données et la gestion des rapports.

- Réponse : Il s'agit de mesures prises en réponse à un événement. Ces actions peuvent être respectivement automatisées ou manuelles. Les réponses courantes incluent l'envoi d'un message à l'opérateur. D'autres mesures de réponse incluent l'enregistrement de l'activité, le stockage de données brutes provenant de sources de données qui décrivent des événements, la gestion du réseau ou des temps d'arrêt des utilisateurs, la gestion des sessions d'application, ainsi que la modification des droits d'accès au réseau ou système [27].

5.3 Emplacement IDS

L'emplacement des systèmes de détection d'intrusions (IDS) est adaptable aux exigences de sécurité spécifiques d'une organisation. Traditionnellement, ces systèmes sont déployés en périphérie du réseau pour surveiller les flux de données entrants et sortants. Toutefois, leur implantation peut également se faire au niveau des hôtes pour une surveillance plus ciblée, voire au sein du réseau lui-même pour une analyse approfondie des activités internes. Le choix de l'emplacement est souvent guidé par la topologie du réseau et les objectifs de sécurité de l'entreprise.

5.3.1 Les Host-Base IDS.

Un IDS basé sur l'hôte (HIDS) surveille le trafic sur un seul hôte. Il effectue une vérification de l'intégrité des systèmes de fichiers, une analyse des appels et journaux système. Les HIDS dépendent du système sur lequel ils sont installés qui peut s'appuyer sur le système d'exploitation lui-même pour vérifier son intégrité et générer des alertes. Il peut également intercepter les paquets réseau entrants ou sortants de l'hôte pour détecter les signaux d'intrusion.

5.3.2 Les IDS hybrides.

Les IDS hybrides généralement utilisés combinent les propriétés du NIDS et du HIDS. Ils permettent d'observer à la fois les réseaux et les appareils au sein d'un seul instrument. Les sondes sont positionnées à des points stratégiques et fonctionnent comme NIDS ou HIDS selon leur position. Ces sondes communiquent ensuite les alertes à un centralisateur qui regroupe, combine et relie les informations provenant de plusieurs sources. En conséquence, ils sont issus d'une conception distribuée dans laquelle chaque composant s'appuie sur un protocole commun pour notifier les personnes.

5.3.3 Les IDS réseaux.

Un système de détection d'intrusion réseau (NIDS) est une technologie de sécurité qui surveille le trafic réseau à la recherche de signes d'activité malveillante et alerte les administrateurs réseau lorsqu'une telle activité est détectée. Il est conçu pour détecter divers

types d’attaques et de tentatives d’accès non autorisés pouvant compromettre la sécurité du réseau. Il est positionné sur un parcours réseau où il écoute activement le trafic afin de reconnaître des signatures d'attaques ou des différences par rapport à la procédure de fonctionnement standard. La protection croissante du trafic sur les réseaux commutés augmente la difficulté d'écoute du réseau, et par conséquent, l'analyse des segments est compliquée car le contenu des paquets est encapsulé. L'IDS réseau peut écouter tout le trafic arrivant à l'interface tout en restant invisible [30].

6 Les attaques par déni de service distribué DDoS.

Les attaques DDoS sont une menace courante visant à saturer les ressources d'un système cible, comme les serveurs ou les réseaux, afin d'entraver leur fonctionnement normal. Cette exploration des attaques DDoS examine leurs mécanismes, leurs impacts dévastateurs sur les infrastructures en ligne, et les stratégies défensives pour contrer ces assauts numériques.

6.1 Définition d’attaques DDoS.

Une attaque par déni de service distribué (DDoS) est une forme de cyberattaque impliquant plusieurs ordinateurs compromis, chacun étant infecté par un logiciel malveillant. Ces ordinateurs sont utilisés pour submerger un système ou un réseau unique avec un volume important de trafic ou de requêtes. Le but d'une attaque DDoS est de paralyser le système ou le réseau cible, l'empêchant d'être accessible aux utilisateurs légitimes. Ces agressions peuvent être perpétrées n’importe où dans le monde et sont souvent difficiles à reconnaître et à prévenir [21].

Divers types d'attaques par déni de service distribué (DDoS) sont provoquées par les agents, notamment TCP, UDP et ICMP. Lors d'une attaque DDoS, une usurpation d’identité d’adresse est effectuée. De plus, ils attribuent les numéros de port de destination et source à un générateur de nombres aléatoires en fonction du type d’attaque (voir Fig.8). Lors d’une attaque DDoS, il est extrêmement difficile de différencier le véritable attaquant en raison des adresses IP falsifiées basées sur une approche hiérarchique [22].

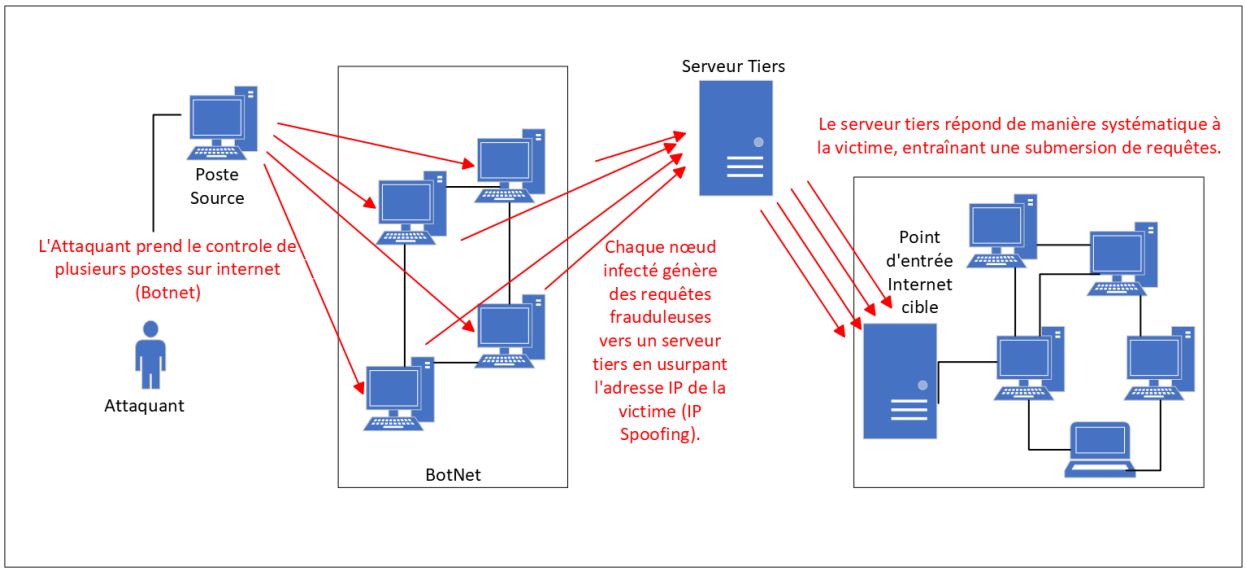


Figure 8 Illustration d'une attaque DDoS

Les attaques DDoS peuvent être divisées en deux catégories en fonction du niveau de protocole ciblé : les attaques au niveau réseau/transport et les attaques au niveau de l'application. Les attaques de type inondation DDoS au niveau réseau/transport sont principalement déclenchées à l'aide de paquets des protocoles TCP, UDP, ICMP et DNS (voir Fig.9). Cette catégorie comprend quatre types d'attaques : les attaques d'inondation et les attaques d'exploitation de protocole par inondation [30].

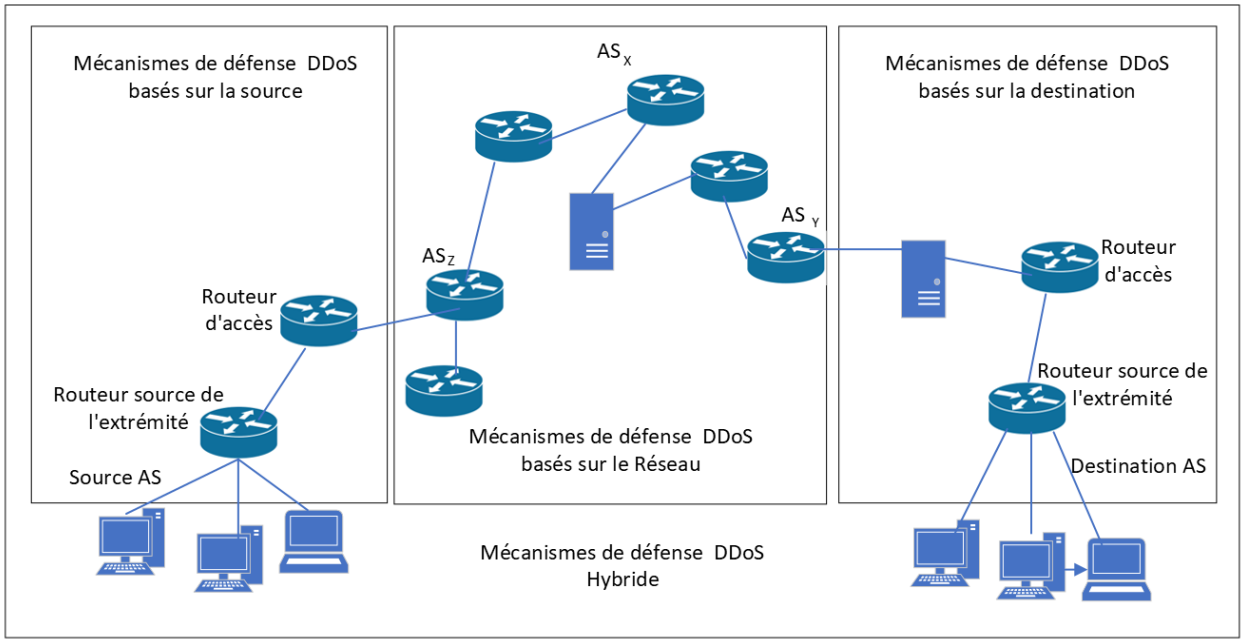


Figure 9 Catégorisation des mécanismes de défense contre les attaques DDoS au niveau réseau/transport

6.2 Les types d’attaques DDoS.

On distingue trois principales catégories d'attaques DDoS: les attaques de la couche application, les attaques de protocole et les attaques volumétriques. Chacune de ces catégories cible des éléments spécifiques des infrastructures en ligne, menaçant leur disponibilité et leur fonctionnement.

6.2.1 Attaques de la couche application.

Ils utilisent des ressources ou des services de l'application, auxquels d'autres utilisateurs légitimes ne peuvent pas accéder. C'est quantifié en requêtes par seconde comme les attaques déni de service HTTP_GET, HTTP_POST.

6.2.2 Attaques de protocole.

Ils consomment des tables d'état présentes dans les composants réseaux dédiés à la communication, tels que les équilibreurs de charge, les pare-feu et les serveurs dédiés aux applications. L'attaque se mesure en paquets par seconde : TCP, SYN, ACK, et d’autres attaques qui se produisent pendant le protocole. Tout ceci est pris en compte.

6.2.3 Attaques volumétriques.

L'attaque volumétrique vise à saturer la bande passante du site attaqué et se mesure en bits par seconde. Tout en englobant des méthodes : l'inondation ICMP, l'inondation UDP, et d'autres méthodes également.

6.3 Méthodes d’attaques DDoS.

Les attaques DDoS sont menées selon diverses méthodes, notamment l'altération ou la modération des ressources, ainsi que la destruction et la modification des composants physiques du réseau SDN.

6.3.1 Altération des ressources.

Le fonctionnement du réseau est compromis lorsque des données de configuration cruciales sur l'ordinateur sont altérées, potentiellement entravant la fourniture de services aux autres utilisateurs connectés. L'attaquant cherche à accéder à la machine en modifiant ses paramètres, les informations de la table de routage sur un routeur peu sécurisé. Un assaillant pourrait empêcher d'autres utilisateurs de se connecter à la machine concernée [28].

6.3.2 Modération des ressources limitées.

Pour garantir leur efficacité, les ordinateurs nécessitent un éventail de ressources, comprenant la mémoire, la bande passante, la puissance de traitement, etc. Lorsque ces ressources sont monopolisées par des éléments externes, les programmes informatiques peuvent faire face à une diminution de l'accès à ces ressources.

6.3.3 Destruction et modification de composants physiques.

L’assaillant a la capacité de causer des dommages physiques à l'environnement de la cible. Ceci lui permet de désactiver ou d’altérer les composants du réseau.

6.4 Approches de Détection d’attaques DDoS SDN.

On distingue différentes approches de détection d’attaques DDoS sur les réseaux SDN comprennant les approches basées sur les entrées de table, les statistiques de flux, sur l’architecture, sur la planification et sur l’intelligence artificielle.

6.4.1 Approche basée sur les entrées de table.

La base de cette solution est le contrôle de l'installation des règles de flux et la politique de remplacement des entrées de table. Dans ce cas, il existe plusieurs propositions illustrées par « Packet-Checker ». Le concept central de cette solution est de donner aux contrôles la capacité de se différencier des paquets nuisibles en créant une table de mappage pour l’adresse MAC et le port du switch [24].

6.4.2 Approche basée sur les statistiques de flux.

Les mécanismes d'atténuation DDoS sont développés sur la base de modèles statistiques qui s'appuient sur des tests de fréquence, de matrice de corrélation, d'entropie et d'adéquation du khi-carré des différentes caractéristiques capturées. Des solutions visant à atténuer les attaques DDoS en exploitant la puissance de traitement des commutateurs, surveiller le trafic pour capturer les caractéristiques pertinentes telles que les IP source/destination, les ports source/destination. Ces caractéristiques sont ensuite utilisées comme entrée dans un algorithme de détection d’anomalies basé sur l’entropie. D'autres approches basées sur l'entropie ont classifié les paquets entrants en fonction de l'adresse IP de destination et de la taille de la fenêtre. Ils utilisent les adresses IP de destination recueillies pour chaque fenêtre d'observation afin de calculer la probabilité d'occurrence de chaque adresse IP de destination pour chaque paquet dans la fenêtre d'observation, et ils analysent la distribution de la fréquence d'occurrence des adresses IP source et destination [29].

6.4.3 Approche basée sur l'architecture.

Une approche basée sur l'architecture pour la détection des attaques DDoS implique la conception d'une structure ou d'un cadre complet pour identifier et atténuer efficacement les attaques DDoS. Cette approche se concentre généralement sur l'organisation et l'intégration de divers éléments de la sécurité réseau afin d'améliorer la résilience contre les attaques DDoS. Les principaux composants et stratégies sont :

- Diviser le trafic réseau en segments en fonction de la source, la destination ou le type de service et analyser chaque segment de manière indépendante.
- Construire une matrice de corrélation pour identifier les relations entre différentes variables dans le trafic réseau.

- Analyser les corrélations pour repérer des déviations qui pourraient indiquer une attaque DDoS.
- Utiliser des tests statistiques, tels que le test du khi-carré, pour évaluer la qualité de l'ajustement entre les données observées et les schémas attendus.

6.4.4 Approche basée sur la planification.

La planification et la priorisation des demandes de flux sont proposées pour contrer les DDoS sur les réseaux SDN. Ces demandes sont destinées à être traitées par le contrôleur et les solutions examinées utilisent un composant de planification dans la couche supérieure qui gérerait le traitement des demandes de flux.

6.4.5 Approche basée sur l'intelligence artificielle.

Les approches de l'apprentissage machine se base sur la création et l'entrainement d'un modèle à l'aide d'un ensemble de données d'entraînement pour détecter et atténuer les attaques DDoS au sein des réseaux SDN. On cite quelques modèles connus :

- **Support Vector Machines (SVM):** un algorithme populaire d'apprentissage machine supervisé utilisé pour des tâches de classification binaire. Il peut être appliqué pour détecter des anomalies dans le trafic réseau qui peuvent indiquer une attaque DDoS.
- **Random Forest**: une technique d'apprentissage par ensemble qui combine plusieurs arbres de décision pour améliorer la précision de la classification. Elle peut être utilisée pour identifier des motifs inhabituels ou des valeurs aberrantes dans le trafic SDN (Software-Defined Networking).
- **Deep Learning:** Les techniques d'apprentissage profond, telles que les Réseaux Neuronaux Convolutifs (CNN) et les Réseaux Neuronaux Récursifs (RNN), peuvent être utilisées pour l'extraction de caractéristiques et la détection d'anomalies dans les données de trafic SDN (Software-Defined Networking).
- **K-Means Clustering:** Les techniques d'apprentissage non supervisées telles que le regroupement K-Means peuvent être utilisées pour regrouper les données de trafic réseau en clusters, ce qui facilite l'identification des anomalies ou des schémas d'attaque DDoS.
- **Naive Bayes:** Les classificateurs Naive Bayes sont des modèles probabilistes qui peuvent être utilisés pour détecter des anomalies en se basant sur la probabilité d'observer certains schémas dans le trafic réseau.

6.4.6 Approches hybrides

Une approche hybride fait référence à la combinaison de plusieurs techniques de détection pour améliorer la précision et l'efficacité de la détection des attaques DDoS dans les réseaux SDN. Ces techniques peuvent inclure l'analyse statistique, l'apprentissage machine, les méthodes basées sur des règles, entre autres. En combinant ces techniques, une approche hybride peut exploiter les forces de chaque méthode pour fournir un système de détection plus complet et robuste. [24]

6.5 Impacts d’attaques DDoS sur le réseau SDN

Les attaques DDoS peuvent exercer une influence substantielle sur le réseau SDN. Dans ce contexte, les attaques DDoS peuvent se manifester sous la forme de flux de trafic manipulés ou contrefaits au sein du plan de données, visant à cibler les dispositifs de transfert et les contrôleurs. La communication intensive entre le contrôleur SDN et les commutateurs sollicitant des décisions de routage, peut entraîner l'épuisement et la congestion de la bande passante du plan de contrôle entre le contrôleur et le commutateur. Cela émerge la perte de nombreux messages "Paquet-In" et provoque des retards dans le temps de réponse des échanges entre le contrôleur et les commutateurs.

6.5.1 Plan de contrôle.

Lorsqu'une attaque DDoS se produit dans le plan de contrôle d'un réseau SDN, elle peut rendre le contrôleur injoignable, entraînant la perte de l'architecture SDN. Même si un contrôleur de secours est déployé, il devra faire face au même défi. Par conséquent, un déluge de paquets légitimes et de paquets falsifiés par DDoS sollicite collectivement les ressources du contrôleur, entraînant leur épuisement [23].

6.5.2 Plan de données.

Dans le cas d'une attaque DDoS, l'attaquant injecte une grande quantité de flux dans le Switch. Le trafic de données reçu par le Switch est converti en règles de commutation afin de les acheminer vers leur destination. Ces règles sont attribuées par le contrôleur. La mémoire (Content-Addressable Memory) du Switch sera remplie de ces règles envoyées par le contrôleur jusqu'à sa saturation. Lorsque cela se produit, le commutateur est obligé d'ajouter et de supprimer continuellement des règles de flux et d'envoyer davantage de requêtes au contrôleur. Ceci provoque une congestion dans le plan de transport et des retards dans les temps de transfert de données.

7 Conclusion

En conclusion, le SDN offre des avantages significatifs en matière de gestion et de flexibilité du réseau, mais il introduit également de nouvelles vulnérabilités de sécurité qui doivent être gérées de manière proactive. Une approche sécurisée et bien pensée dans la conception, le déploiement et la gestion des réseaux SDN est essentielle pour profiter pleinement de cette technologie tout en minimisant les risques potentiels.

Au chapitre suivant, nous allons présenter une revue de littérature approfondie, dans laquelle nous synthétisons plusieurs travaux scientifiques traitant de la problématique liée à la détection d’attaques DDoS sur les réseaux SDN. Cette revue a pour objectif d'explorer les approches, les méthodes et les résultats des recherches antérieures pertinentes, dans le but de mieux appréhender les défis et les solutions avancées par d'autres chercheurs dans ce domaine.

CHAPITRE 3. REVUE DE LA LITTÉRATURE

1 Introduction

Dans ce chapitre, nous présentons les objectifs en matière de sécurité des réseaux SDN et les solutions proposées par les chercheurs pour détecter et atténuer les attaques DDoS dans plusieurs réseaux.

2 Détection des attaques DDoS dans les environnements SDN.

La sécurité est le problème majeur pour implémenter les réseaux SDN. Les auteurs dans [31] abordent une analyse méthodique des stratégies de détection des attaques DDoS reposant sur l'apprentissage machine et profond dans les réseaux SDN. Ils évaluent l'efficacité de différentes approches d'apprentissage machine avec une liste des avantages et des inconvénients de chaque étude, telles que Arbres de décision (DT), Machines à vecteurs de support (SVM), (KNN), Réseaux neuronaux (NN), Random forest (RF), Réseaux neuronaux convolutionnels (CNN), Mémoire à court terme longue (LSTM), et Auto-encodeur (AE) utilisant les datasets, KDD Cup99[55], Kyoto 2006+[56], NSL-KDD[57], UNSW-NB15[58], CIC-IDS2017[59], CSE-CIC-IDS2018[60], SCX2012[61], et CICDDoS2019[62]. Pour l'ensemble de données ISCX2012, les techniques LSTM, CNN et LSTM-Bayes ont toutes affiché une précision inférieure à 98,8 %. En revanche, sur l'ensemble de données NSL-KDD, seule la technique CNN a atteint une précision supérieure à 99 %, nécessitant des calculs complexes pour y parvenir.

Dans [21], [36], les auteurs exploitent des approches de détection des attaques DDoS dans les réseaux SDN basée sur le Deep Learning, en utilisant la dataset CICIDS2017 [36]. Ils ont démontré l'efficacité de leur approche pour détecter avec précision le trafic d'attaque DDoS au milieu du trafic légitime en direction de la victime. L'utilisation d'un ensemble de réseaux neuronaux convolutionnels (CNN) aboutit à une impressionnante précision de détection de 99,48%, avec une intégration aisée dans les architectures SDN existantes, bien que sa performance puisse être moindre pour des attaques DDoS plus complexes. D'autres méthodes telles que les approches statistiques, les chaines de blocs, l'apprentissage machine et le Deep Learning ont été abordés [21]. Les auteurs soulignent le potentiel des systèmes de détection DDoS basés sur le Deep Learning pour fournir des solutions dynamiques et efficaces, permettant une gestion, une sécurité et une optimisation intelligentes des réseaux SDN. Ces systèmes, en traitant les flux de données SDN, offrent la possibilité au réseau de développer une capacité d'apprentissage autonome et de réagir de manière appropriée.

Les auteurs dans [35] étudient les attaques DDoS à faible taux dans les environnements SDN en utilisant l'apprentissage machine. Ils abordent la création d'une architecture de sécurité modulaire. Le module IDS conçu pour détecter les flux en exploitant divers modèles d'apprentissage machine préalablement entrainés. Ils ont obtenu un taux de précision de 95 % lors des tests de six algorithmes d'apprentissage machine effectués en utilisant la Dataset CIC-DoS en déployant l'architecture dans un environnement virtualisé réel à l'aide de la machine virtuelle Mininet sur VirtualBox et du contrôleur ONOS.

Dans le cadre de la détection des attaques DDoS, un modèle basé sur la régression linéaire multiple a été présenté dans [33]. L'approche permet de construire un modèle de prédiction. Elle se base sur la sélection de caractéristiques en utilisant la méthode du « Gain d'Information » afin de déterminer les attributs importants qui sont les meilleurs indicateurs pour le modèle de prédiction. Les caractéristiques sélectionnées sont ensuite prises en compte pour effectuer une analyse. Les auteurs ont utilisé la dataset CICIDS_2017 en analysant les graphiques d'ajustement et les graphiques des résidus. Ils rapportent que leur modèle a atteint une précision de prédiction de 97,86 %.

Dans [34], les auteurs ont proposé une approche de mitigation des attaques DDoS en SDN qui utilise des algorithmes d'apprentissage machine (ML). L'approche proposée implique l'entrainement d'un modèle formé à l'aide d'une Dataset de caractéristiques du trafic réseau et utilisé pour classer le trafic entrant comme normal ou malveillant. Les auteurs présentent un prototype de système de preuve de concept qui démontre l'efficacité de l'approche proposée avec des résultats qui indiquent une précision de 98,77 %. Elle est efficace pour identifier rapidement et correctement les menaces et les contrecarrer sans perturber le trafic régulier, en revanche la manipulation de vastes quantités de données en temps réel peut présenter des défis de performance dans les environnements SDN de grande envergure, pouvant entraîner des retards dans la détection et la neutralisation des attaques.

Les auteurs dans [37] étudient la classification des attaques DDoS en SDN. Ils abordent plusieurs modèles d'apprentissage machine tels que le support Vector Machine (SVM), Naive Bayes (NB), Réseau de Neurones Artificiels (ANN), et K-Nearest Neighbors (KNN). Les résultats des tests ont montré que l'utilisation de la sélection de caractéristiques avec un classifieur KNN a atteint le taux de précision le plus élevé 98,3 % dans la détection des attaques DDoS. D'après les auteurs, l'utilisation de la sélection de caractéristiques avec un classifieur KNN est le meilleur model pour la détection des attaques DDoS.

Dans [42], Les auteurs ont analysé plusieurs classificateurs d'apprentissage machine, notamment XGBoost, Régression Logistique (LR), Machine à Vecteurs de Support (SVM), Naive Bayes (NB) et K-Nearest Neighbors (KNN) en utilisant l'émulateur Mininet pour générer une Dataset spécifique à SDN, comprenant à la fois des flux légitimes et malveillants. Ils ont utilisé une matrice de confusion pour évaluer la performance des modèles et leur capacité de classification. L'analyse révèle que XGBoost, LR, SVM et KNN présentent de solides performances de classification. L'algorithme NB présente des limites en termes de précision de classification, avec un nombre plus élevé de faux positifs et de faux négatifs. L'étude présente certaines limitations, la dataset utilisée est trop limitée et peut ne pas donner des résultats pour certains types d'attaques DDoS évoluées et plus complexes.

Une approche qui utilise des modèles d'apprentissage profond a été proposée dans [49]. Elle surpasse les méthodes traditionnelles d'apprentissage machine telles que Random Forest, en termes de précision, de rappel, de score F1 et d'Aire sous la Courbe (AUC). Les auteurs révèlent que les quatre modèles de réseaux neuronaux récurrents atteignent des

performances presque équivalentes, bien que le modèle LSTM (Long Short-Term Memory) affiche légèrement une précision supérieure à 97.83%, un score de rappel plus élevé de 97.378%, et un taux d'erreur moindre sur un Dataset donné de 2.394%. LSTM excelle dans la capture des dépendances à long terme au sein de données séquentielles et utilise une cellule de mémoire pour conserver les informations au fil du temps.

Dans un autre travail, K. S. Sahoo et al. [40] ont proposé un classificateur SVM pour prédire le trafic malveillant. L'approche de détection combine SVM avec KPCA et GA. L'extraction des caractéristiques est effectuée par KPCA, et le classificateur SVM est utilisé pour la classification des attaques. De plus, une fonction de noyau de base radiale améliorée a été utilisée pour réduire le temps d'entraînement. L'algorithme génétique a également été utilisé pour optimiser divers paramètres du classificateur. Ils ont exécuté le module sur le contrôleur et validé dans un environnement simulé comprenant le contrôleur POX, OVS et l'émulateur Mininet. Les auteurs ont comparé les résultats de détection d'attaque à d'autres classificateurs, montrant que le modèle SVM réalise une classification efficace et précise par rapport aux autres avec une précision de 98,907%. L'approche proposée a été validée dans un environnement simulé comprenant le contrôleur POX, OVS et l'émulateur Mininet. D'où des tests supplémentaires et une validation dans des environnements SDN réels sont nécessaires afin d'évaluer son efficacité et sa faisabilité.

Dans [38], A. T. Kyaw, M. Zin Oo et C. S. Khin décrivent l'utilisation du SVM polynomial dans un système proposé pour classifier les attaques par inondation DDOS au sein des réseaux SDN. Ils ont effectué une comparaison des performances entre le classificateur SVM polynomial proposé et un classificateur SVM linéaire. Les auteurs ont évalué également les performances du système en utilisant l'ensemble de données KDD Cup99 [55]. Les résultats ont montré que le classificateur SVM polynomial a obtenu une précision supérieure et des taux de fausses alarmes réduits par rapport au classificateur SVM linéaire. D'après les auteurs, le classificateur SVM polynomial est mieux adapté. Mais il n'est pas clair comment le système se comporterait avec d'autres ensembles de données ou dans des scénarios réels.

Les auteurs dans [39] proposent une méthode d'évaluation des performances basée sur la sensibilité. Les indicateurs de performance incluent les vrais positifs, TP, qui indique le nombre de paquets correctement classés et transmis, les vrais négatifs, TN, qui indique le nombre de paquets correctement classés et rejetés. Les faux positifs, FP, indique le nombre de paquets incorrectement classés et transmis.et les faux négatifs, FN, qui indique le nombre de paquets incorrectement classés et rejetés. Ils ont évalué les performances par une analyse des temps d'entraînement et de test pour calculer le taux de fausses alarmes des attaques DDoS et le taux de détection des flux de paquets. Ensuite, ils ont mesuré les performances du SVM et du SVM adaptatif avec une Dataset générée en temps réel. Leur approche basée sur la sensibilité avec une structure d'arbre AVL permet une détection en temps réel hautement précise en réduisant les faux positifs. Cette approche nécessite une mise à jour continue pour s'adapter aux évolutions des attaques DDoS.

En 2020, un nouveau modèle en utilisant un réseau neuronal convolutif profond est proposé par S. Haider et al [48] afin contribuer à une détection plus précise des attaques DDoS. Les auteurs ont vérifié l'infrastructure à l'aide de la Dataset CICIDS2017 connue sous le nom de ISCX Dataset. Ils ont comparé leur nouvelle approche aux approches actuelles présentées par divers auteurs, telles que l'approche RNN, hybride (RBM+SVM), SVM et LSTM. Le modèle nouvellement construit a donné une précision de 99,45 %, ce qui est supérieur à la précision des autres modèles.

Deux autres approches pour détecter les attaques DDoS dans les réseaux SDN ont été proposé par S. Dong et M. Sarem. [41]. La première approche est l'algorithme basé sur le degré d'attaque (DDADA), utilisant un nouveau concept appelé le degré d'attaque pour détecter les attaques DDoS. La deuxième est l'algorithme de détection DDoS basé sur l'apprentissage machine (DAMDL), utilisant une version améliorée de l'algorithme KNN basé sur l'apprentissage machine (ML). Les auteurs ont fait cette étude avec un trafic réseau collecté, composé de protocoles UDP, TCP et ICMP. Ils ont évalué et comparé les algorithmes proposés à d'autres algorithmes tels que NB, KNN, SVM et CIC_SVM. Leurs résultats ont montré que les valeurs de TPR pour les algorithmes DDADA et DDAML étaient respectivement de 0,987 et 0,994, dépassant les valeurs de TPR des autres algorithmes comparés, les valeurs de FPR étaient respectivement de 0,016 et 0,009, inférieures aux valeurs de FPR des autres algorithmes comparés. La Précision, le Rappel et la F-mesure des algorithmes DDADA et DDAML étaient supérieurs à ceux des autres algorithmes.

Dans [43], cinq classificateurs d'apprentissage supervisé ont été élaborés afin d'entraîner et tester les sous-ensembles de caractéristiques sélectionnés d'un ensemble de donné CSE-CIC-IDS2018. Notamment les algorithmes de machine à vecteurs de support (SVM), forêt aléatoire (RF), d'arbre de décision, XGBoost et de k plus proches voisins (KNN). L'algorithme XGBoost a obtenu la meilleure précision (0,969) sur l'ensemble de données original, bien que moins précis que RF, il a affiché la meilleure performance globale. Après l'extraction des caractéristiques, la précision de chaque classificateur a progressé, l'algorithme de l'arbre de décision ayant montré la plus grande amélioration 0,0341, suivi de RF 0,0278 et SVM 0,0202. RF s'est révélé le meilleur, avec des améliorations dans toutes les autres métriques. L'étude effectué par les auteurs ne se limite pas à un type spécifique d'attaque DDoS et peut être appliqué à divers types d'attaques DDoS. Cependant, elle pourrait ne pas être s'adaptée au facteur d'échelle pour les grands réseaux SDN en raison de la complexité computationnelle de la sélection des caractéristiques et des algorithmes d'apprentissage machine utilisés.

Dans [32], les auteurs ont abordé la problématique de la détection des attaques par déni de service distribué (DDoS) dans le contexte des réseaux définis par logiciel (SDN) en utilisant des techniques d'apprentissage machine. L'article se concentre sur le développement d'une approche novatrice qui exploite l'analyse des composants de voisinage (NCA) pour sélectionner les caractéristiques pertinentes et améliorer la détection des attaques DDoS dans les environnements SDN. L'ensemble de données utilisé est le "DDOS attack SDN Dataset"

public, spécifiquement créé dans l'environnement SDN pour la recherche en apprentissage machine et profond. L'étude a rapporté des taux de précision élevés pour différents algorithmes d'apprentissage machine, la méthode de l'arbre de décision (DT) ayant atteint le meilleur taux de précision de 100% dans une étude expérimentale. De plus, d'autres algorithmes d'apprentissage machine tels que le k-Nearest Neighbor (kNN), le Réseau de Neurones Artificiels (ANN) et la Machine à Vecteurs de Support (SVM) ont également démontré des taux de précision élevés allant de 97,75% à 99,78%. Ces résultats suggèrent l'efficacité de l'approche proposée dans la détection précise des attaques DDoS dans les environnements SDN.

Les auteurs dans [44] ont fait une comparaison entre les techniques basées sur l'apprentissage machine pour atténuer les intrusions et les attaques DDoS. Ils ont mis en œuvre des IDS (systèmes de détection d'intrusions) basés sur les anomalies capables d'apprendre à partir de données et de prendre des décisions pour des données de test ou inconnues. Ces techniques peuvent classifier les comportements intrusifs et normaux/non intrusifs dans les réseaux SDN en utilisant les machines à vecteurs de support, les algorithmes génétiques, la logique floue, les réseaux bayésiens et les arbres de décision. Les SVM offrent une capacité d'apprentissage efficaces pour les problèmes de classification. Les arbres de décision (DT) sont également largement utilisés comme une méthode pratique pour apprendre des expressions disjonctives.

Une autre approche dans cette étude a été élaboré par R. Doshi, N. Apthorpe and N. Feamster. [45], utilisant cinq classificateurs différents d'apprentissage machine : K-Nearest Neighbors (KN), Linear Support Vector Machine (LSVM), Decision Tree (DT), Random Forest (RF) et Neural Network (NN). Ces classificateurs ont été entraînés sur un ensemble de données comprenant du trafic normal et des attaques DDoS collectés à partir d'un réseau expérimental d'appareils IoT grand public. Les auteurs rapportent que les cinq algorithmes avaient une précision sur l'ensemble de test supérieure à 0,99, indiquant que l'utilisation de comportements réseau spécifiques aux IoT peut conduire à une détection précise des attaques DDoS dans le trafic réseau IoT avec divers algorithmes d'apprentissage machine, y compris les réseaux neuronaux. Les auteurs suggèrent que leur approche pourrait être étendue pour détecter d'autres types d'attaques plus subtiles que les inondations DDoS, telles que celles impliquant l'exfiltration de données ou le trafic de commande et de contrôle.

Dans [47], N. Ravi et S. M. Shalinie proposent un mécanisme LEDEM (Learning-Driven Detection Mitigation) afin d'étudier la détection et l'atténuation des attaques DDoS dans les réseaux IoT à l'aide d'un algorithme d'apprentissage machine semi-supervisé. Les auteurs déploient le mécanisme dans un banc d'essai du monde réel et un émulateur SDN, avec plusieurs agents répartis dans le réseau pour détecter les attaques DDoS. LEDEM a atteint un taux de précision amélioré de 96,28 % dans la détection des attaques DDoS. Ils ont comparé la précision de LEDEM avec d'autres solutions de pointe et ont démontré que LEDEM a une meilleure précision de détection que les autres modèles. Cependant, Les

auteurs utilisent un seul ensemble de données UNB-ISCX qui pourrait ne pas être suffisamment diversifié pour représenter tous les scénarios possibles dans les réseaux IoT.

Les auteurs dans [46] proposent un modèle d'apprentissage Deep Learning appelé DDoSNet pour détecter les attaques DDoS sur les réseaux SDN. Ils ont évalué le modèle à l'aide d'un nouvel ensemble de données appelé CIC-DDoS2019, contenant des types d'attaques DDoS complets et récents. Les auteurs ont démontré que DDoSNet donne les meilleures mesures d'évaluation en termes de rappel, de précision, de score F et d'exactitude de 99% par rapport aux techniques classiques bien connues d'apprentissage machine existantes, concluant que DDoSNet est une méthode efficace pour détecter les attaques DDoS SDN. L'aptitude du modèle proposé à se généraliser à d'autres ensembles de données et à des scénarios du monde réel demeure incertaine.

Tous ces travaux convergent sur le même objectif dans différents contextes, obtenir une méthode optimale pour la prédiction des différentes attaques DDoS. Le tableau 2 résume les principales caractéristiques de ces recherches.

Approche	**Caractéristiques**	**Référence**
Deep Learning for DDoS Detection in SDN	- Différentes approches de détection et d'atténuation des attaques basées sur des approches statistiques, blockchain, l'apprentissage machine et le Deep Learning. - Suggère que les systèmes basés sur le Deep Learning offrent des solutions dynamiques, efficaces et intelligentes pour la gestion, la sécurité et l'optimisation des réseaux SDN. - Atteinte d'une haute précision de détection de 99,48 % avec un calcul propice.	[21], [36]
Ensembles de réseaux neuronaux convolutifs profonds (Deep CNN)	- Analyse méthodique des stratégies de détection des attaques DDoS dans les réseaux SDN. - Évalue l'efficacité de différentes approches d'apprentissage machine, dont Arbres de décision (DT), Machines à vecteurs de support (SVM), K-Nearest Neighbors (KNN), Réseaux neuronaux (NN), Random Forest (RF), Réseaux neuronaux convolutionnels (CNN), Mémoire à court terme longue (LSTM), et Auto-encodeur (AE). - Taux de précision de plus de 99 % obtenu dans une grande partie de la littérature.	[31]

Attaques DDoS de la couche transport et d'application.	- Evaluation de trois méthodes d'apprentissage machine : SVM, Random Forest (RF), K-NN. - Quatre mécanismes d'apprentissage profond (DL) : perceptron multicouche (MLP), réseau neuronal convolutif (CNN), unités récurrentes à porte (GRU) et réseau neuronal à mémoire à court terme longue (LSTM). - Datasets : CICDoS2017 et CICDDoS2019. - Précision supérieure à 99 % lors de la phase de test. - Atteinte de taux élevés de détection pour les attaques DDoS à volume élevé et à débit lent. - Solution proposée non évaluée par rapport à d'autres systèmes de détection DDoS de pointe.	[48]
Modular Security Architecture with Machine Learning	- Architecture de sécurité modulaire pour détecter les attaques DDoS à faible taux dans les SDN. - Utilise un module IDS avec six algorithmes d'apprentissage machine pour détecter les flux malveillants. - Atteinte d'une précision de 95 % avec une architecture déployée dans un environnement virtualisé réel.	[35]
Linear Regression Multiple Classifier	- Modèle de prédiction basé sur la régression linéaire multiple pour détecter les attaques DDoS. - Sélectionne les caractéristiques par la méthode du Gain d'Information. - Atteinte d'une précision de prédiction de 97,86 % avec la dataset CICIDS 2017.	[33]
Machine Learning-Based Mitigation	- Approche de mitigation des attaques DDoS en SDN en utilisant un modèle formé par des caractéristiques du trafic réseau. - Présente un prototype avec une précision de 98,77 %. - La manipulation de vastes quantités de données en temps réel peut présenter des défis de performance dans les réseaux SDN à grande échelle.	[34]

Classification des attaques DDoS en SDN	- Étude de la classification des attaques DDoS en SDN avec plusieurs modèles d'apprentissage machine : - Support Vector Machine (SVM), Naive Bayes (NB), Réseau de Neurones Artificiels (ANN), et K-Nearest Neighbors (KNN). - KNN avec sélection de caractéristiques atteint un taux de précision de 98,3 %.	[37]
Performance Evaluation Based on Sensitivity	- Méthode d'évaluation des performances basée sur la sensibilité avec des indicateurs tels que TP, TN, FP, et FN. - Utilise un classificateur SVM et un SVM adaptatif avec une structure d'arbre AVL. - Fournit une détection en temps réel hautement précise avec une réduction des faux positifs.	[39]
SVM Classifier with KPCA and GA	- Propose un classificateur SVM pour prédire le trafic malveillant. - Combinaison du SVM avec KPCA et GA. - Validé dans un environnement simulé avec le contrôleur POX, OVS, et l'émulateur Mininet. - Atteinte d'une précision de 98,907 %.	[40]
Polynomial SVM for DDoS Classification	- Utilise un classificateur SVM polynomial pour classifier les attaques DDoS dans les réseaux SDN. - Obtient une précision supérieure et des taux de fausses alarmes réduits par rapport au classificateur SVM linéaire.	[38]
Supervised Classifiers	- Élabore cinq classificateurs d'apprentissage supervisé avec des sous-ensembles de caractéristiques sélectionnées - L'algorithme XGBoost obtient la meilleure précision (0,969) sur l'ensemble de données original. - La précision de chaque classificateur a progressé après l'extraction des caractéristiques, - L'algorithme de l'arbre de décision ayant montré la plus grande amélioration 0,0341, suivi de RF 0,0278 et SVM 0,0202. RF s'est révélé le meilleur.	[43]

Machine and deep Learning Classifiers for SDN DDoS Detection	- Utilisation de l'analyse des composants de voisinage (NCA). - Utilisation du "DDOS attack SDN Dataset" public. - Taux de précision élevés pour différents algorithmes : DT (100%), kNN, ANN et SVM (97,75% à 99,78%).	[32]
DDADA and DAMDL Algorithms	- Propose deux approches pour détecter les attaques DDoS avec les algorithmes DDADA et DAMDL. - Obtient des valeurs de TPR élevées et des valeurs de FPR faibles, surpassant les autres algorithmes comparés.	[41]
Performance Analysis of Machine Learning Classifiers	- Analyse plusieurs classificateurs d'apprentissage machine, XGBoost, Régression Logistique (LR), SVM, Naive Bayes (NB) et KNN. - L'utilsation de l'émulateur Mininet pour générer une Dataset spécifique à SDN. - XGBoost, LR, SVM et KNN présentent de solides performances. - NB présente des limites en termes de précision.	[42]
Deep Learning-Based DDoS Detection System	- Utilise des modèles d'apprentissage profond et surpasse les méthodes traditionnelles d'apprentissage machine comme Random Forest. - Les modèles de réseau neuronal récurrent, notamment le LSTM (Long Short-Term Memory), atteignent des performances équivalentes. - LSTM excelle dans la capture des dépendances à long terme dans les données séquentielles.	[49]
Comparison of Machine Learning Techniques for Intrusion Detection	- Compare les techniques basées sur l'apprentissage machine, SVM, les algorithmes génétiques, la logique floue, les réseaux bayésiens et les arbres de décision. - SVM offre une capacité d'apprentissage efficace. - Les arbres de décision sont largement utilisés.	[44]

Machine Learning Classifiers for IoT DDoS Detection	- Utilise cinq classificateurs d'apprentissage machine, dont K-Nearest Neighbors (KN), Linear Support Vector Machine (LSVM), Decision Tree (DT), Random Forest (RF), Neural Network (NN). - L'ensemble des algorithmes montrent une précision sur l'ensemble de test supérieure à 0,99, indiquant une détection précise des attaques.	[45]
LEDEM: Semi-Supervised Machine Learning for IoT DDoS	- Propose un mécanisme LEDEM utilisant un algorithme d'apprentissage machine semi-supervisé pour étudier la détection et l'atténuation des attaques DDoS dans les réseaux IoT. - Atteint un taux de précision amélioré de 96,28 % dans la détection des attaques DDoS. - Utilise un seul ensemble de données qui pourrait ne pas être suffisamment diversifié.	[47]
DDoSNet: Deep Learning Model for SDN DDoS Detection	- Propose un modèle d'apprentissage Deep Learning appelé DDoSNet pour détecter les attaques DDoS sur les réseaux SDN. - Modèle évalué avec l'ensemble de données CIC-DDoS2019. - Démontre de meilleures mesures d'évaluation en termes de rappel, de précision, de score F et d'exactitude (99%). - L'aptitude du modèle à se généraliser à d'autres ensembles de données et à des scénarios du monde réel reste incertaine.	[46]

Tableau 4 Synthèse des méthodes pour la détection des différentes attaques DDoS.

3 Conclusion

Afin d'atténuer les attaques, diverses approches de recherche proposent plusieurs solutions visant à améliorer les protocoles et l'architecture sécuritaire des réseaux définis par logiciel (SDN). Cependant, pour garantir la sécurité, il ne suffit pas seulement de disposer de cadres de communication sécurisés, mais également d'algorithmes et de méthodes robustes capables de faciliter la détection d'intrusions malveillantes au sein des réseaux.

Comme mis en évidence dans notre analyse bibliographique, diverses approches ont été proposées pour élaborer des systèmes de détection d'attaques de type DDoS, englobant

des modèles fondés sur les réseaux de neurones, l'apprentissage profond, et l'apprentissage machine. Tandis que certaines de ces approches ont démontré des performances prometteuses, d'autres ont affiché des résultats moins concluants. Au chapitre suivant, nous exposerons notre propre étude axée sur la détection et la classification des attaques DDoS dans les réseaux SDN. Nous détaillerons la méthodologie spécifique que nous avons utilisée pour atteindre les objectifs définis dans ce mémoire. Nous présenterons plusieurs approches de détection des attaques DDoS sur les réseaux SDN.

CHAPITRE 4. IMPLEMENTATION

1 Introduction

Dans ce chapitre, nous exposons les étapes de la méthodologie employée pour parvenir aux résultats. Nous détaillons les caractéristiques et les classes de l'ensemble de données public utilisé dans cette étude. Nous expliquons également l'algorithme de sélection des caractéristiques, utilisé pour déterminer les attributs qui renforceront la précision de la classification dans cet ensemble de données. De plus, nous expliquons en détail les caractéristiques des classificateurs employés après la phase de sélection des attributs. Ensuite, nous expliquons les résultats de notre travail. La figure 10 représente un schéma explicatif de la méthodologie de notre travail.

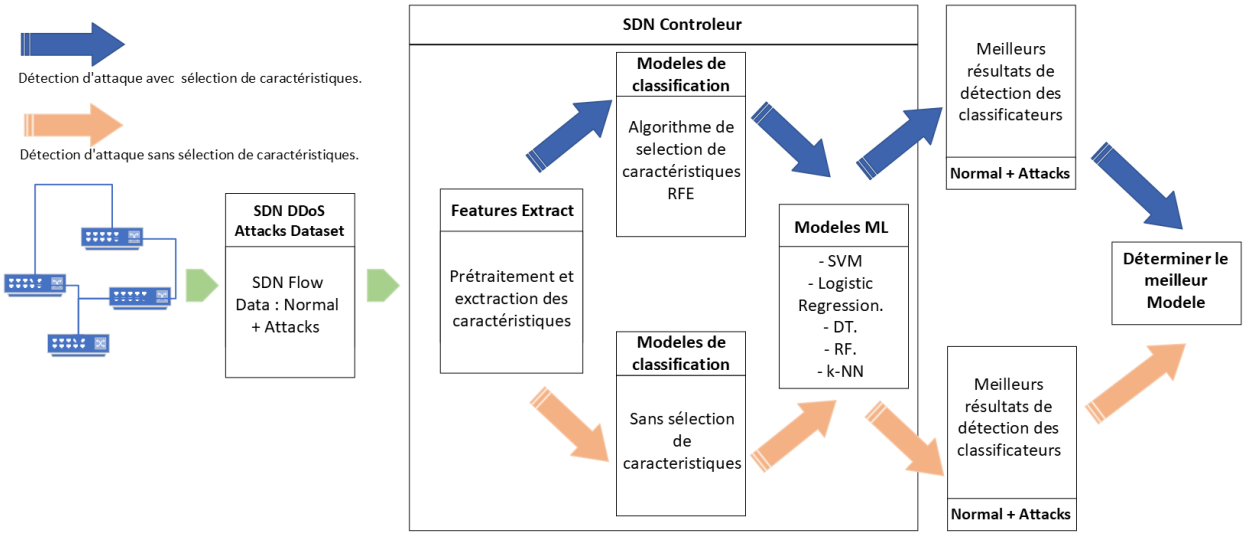


Figure 10 Méthodologie de travail

2 Langage et environnement

Python

Python est un langage de programmation interprété et polyvalent. Sa première version a été publiée en 1991 par Guido van Rossum. Python est simple, lisible et facile à apprendre pour les débutants en programmation tout en restant puissant et flexible pour les développeurs expérimentés.

Google Colaboratory

Google Colaboratory « Colab », est un environnement qui est particulièrement adapté pour des applications telles que l'apprentissage machine, l'analyse de données et l'éducation, développé par Google Research. Il offre la possibilité d'écrire et d'exécuter du code Python depuis un navigateur web.

3 Analyse de l'ensemble de données.

Dans le cadre de cette étude, nous avons utilisé l'ensemble de données public "DDOS attack SDN Dataset", créé dans l'environnement SDN et mis à la disposition des chercheurs pour leurs travaux en apprentissage machine et profond [51]. Cet ensemble comprend 23 caractéristiques et englobe 104 345 flux de trafic, composé de trafic TCP, UDP et ICMP, avec des étiquettes de classe pour les flux normaux et les attaques. Les caractéristiques statistiques, telles que le nombre d'octets, la durée en secondes, le taux de paquets, et le nombre de paquets par flux, sont incluses, à l'exception des caractéristiques définissant les machines source et cible.

```
[ ] data.head()
```

	dt	switch	src	dst	pktcount	bytecount	dur	dur_nsec	tot_dur	flows	...	pktrate	Pairflow	Protocol	port_no	tx_bytes	rx_bytes	tx_kbps	rx_kbps	tot_kbps	label
0	11425	1	10.0.0.1	10.0.0.8	45304	48294064	100	716000000	1.010000e+11	3	...	451	0	UDP	3	143928631	3917	0	0.0	0.0	0
1	11605	1	10.0.0.1	10.0.0.8	126395	134737070	280	734000000	2.810000e+11	2	...	451	0	UDP	4	3842	3520	0	0.0	0.0	0
2	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	1	3795	1242	0	0.0	0.0	0
3	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	2	3688	1492	0	0.0	0.0	0
4	11425	1	10.0.0.2	10.0.0.8	90333	96294978	200	744000000	2.010000e+11	3	...	451	0	UDP	3	3413	3665	0	0.0	0.0	0

5 rows × 23 columns

```
[ ] data.shape
```

```
(104345, 23)
```

Figure 11 Vue de l'ensemble de données : DDOS attack SDN Dataset

La figure 11 représente une vue de l'ensemble de données qu'on va utiliser dans notre travail. Les caractéristiques de cet ensemble de données sont représentées dans le tableau 5 suivant :

Caractéristique	Type de données	Description
dt	int64	Indique le timestamp, auxquels les données ont été collectées.
Switch	int64	Identifie le commutateur (switch) réseau auquel les données se rapportent.
pktcount	int64	Nombre total de paquets dans l'ensemble de données.
dst	object	Adresse de destination du trafic réseau.
src	object	Adresse source du trafic réseau.
bytecount	int64	Nombre total d'octets dans l'ensemble de données.
Dur	int64	Représente la période pendant laquelle les données ont été collectées.
dur_nsec	int64	Durée en nanosecondes, unité de mesure du temps plus précise.
tot_dur	float64	Durée totale, agrégation de la durée sur plusieurs points

		dans le temps.
Flows	int64	Nombre de flux de données distincts.
packetins	int64	Nombre total de paquets d'entrée.
pktperflow	int64	Nombre moyen de paquets par flux.
byteperflow	int64	Nombre moyen d'octets par flux.
Pktrate	int64	Taux de paquets, indiquant la fréquence à laquelle les paquets sont générés.
Pairflow	int64	Nombre de paires de flux.
Protocol	object	Protocole réseau utilisé.
port_no	int64	Numéro de port réseau associé aux données.
tx_bytes	int64	Nombre d'octets transmis.
rx_bytes	int64	Nombre d'octets reçus.
tx_kbps	int64	Débit de transmission en kilobits par seconde.
rx_kbps	float64	Débit de réception en kilobits par seconde.
tot_kbps	float64	Total des débits de transmission et réception en kilobits par seconde.
label	int64	Étiquette associée à l'ensemble de données, utilisée pour la classification du trafic bénin et malveillant.

Tableau 5 Liste de caractéristiques de l'ensemble de données : DDOS Attack SDN Dataset

L'ensemble de données contient 23 caractéristiques, au total, 20 caractéristiques de type numérique (17 caractéristiques de type entiers, 3 caractéristiques de type réel), et 3 de type objet dont : src, dst, protocol.

4 Prétraitement des données.

Avant d'initier l'entraînement du modèle d'apprentissage machine, nous avons effectué une étape cruciale qui est le prétraitement des données. Les propriétés telles que les octets par flux, les paquets par flux, et le taux de paquets, ont été exclues de l'ensemble de données en raison de valeurs dupliquées. Les variables catégorielles, notamment les adresses IP source-destination et le protocole sans valeurs numériques, ont été encodées en utilisant la technique de codage one-hot [52].

Ensuite, nous avons l'étiquette nommée label, qui comprend deux valeurs booléennes :

- 0 désigne un trafique Bénin qui est de 63561 requêtes.
- 1 désigne un trafique Malveillant qui est de 40784 requêtes.

La figure 12. Représente la proportion de requêtes bénignes et malveillantes dans l'ensemble de données, le trafic bénin représente un pourcentage de 60.91%, et le trafic malveillant représente 39.09% du total du trafic.

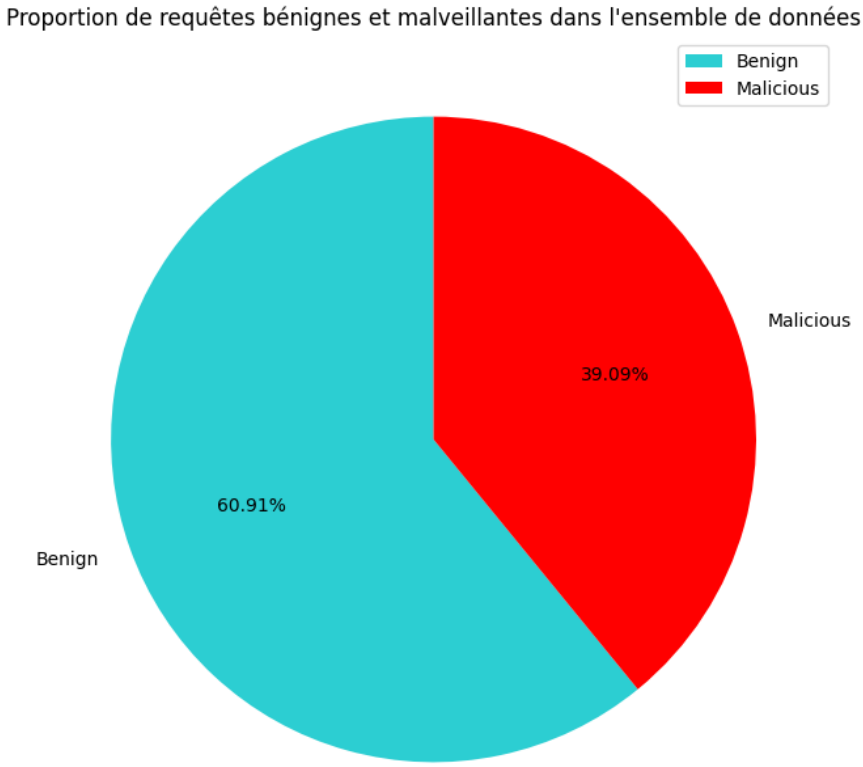


Figure 12 Proportion de requêtes bénignes et malveillantes dans l'ensemble de données

A l'étape suivante, nous allons examiner les caractéristiques de type objet dont Source, Destination et Protocole, afin de déterminer le nombre de total requêtes et d'attaques à partir de différents adresses IP (voir figure 13). On représente le nombre de requêtes provenant de différentes adresses IP sources ainsi que le nombre d'attaques DDoS pour chaque adresse.

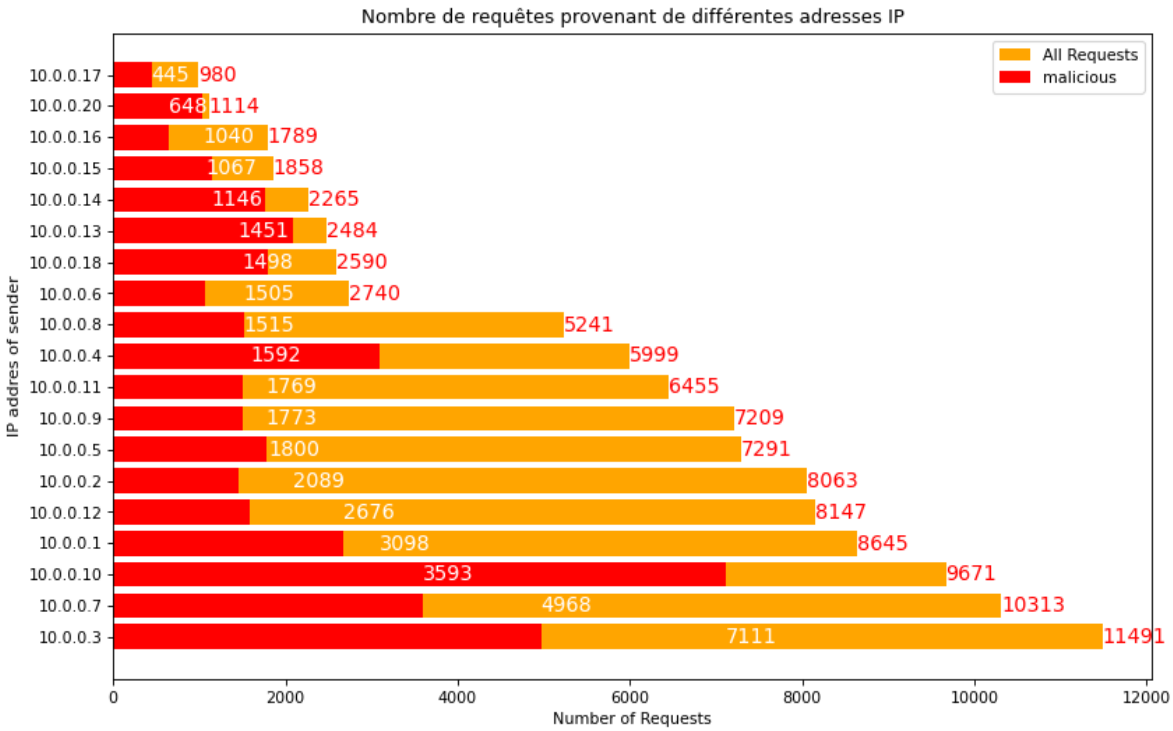


Figure 13 Nombre de requêtes provenant de différentes adresses IP

Nous allons également examiner le nombre de total requêtes et d'attaques à partir de différents protocoles tels que : TCP, UDP et ICMP. Sur la figure 14. Nous constatons :

- 29436 requêtes provenant du protocole TCP, dont 13866 attaques.
- 33588 requêtes provenant du protocole UDP, représentant ainsi le protocole le plus vulnérable avec un nombre plus élevé d'attaque dont 17499 attaques.
- 41321 requêtes provenant du protocole ICMP, dont 9419 attaques.

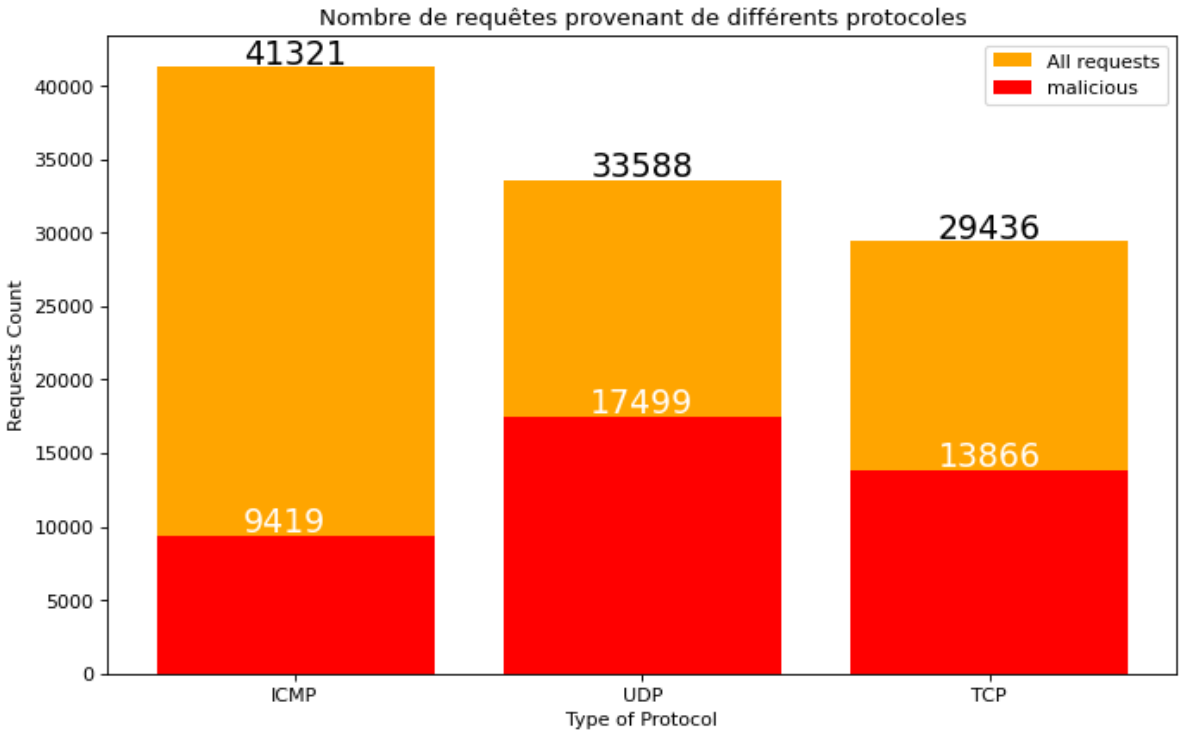


Figure 14 Nombre de requêtes provenant de différents protocoles

5 Modèles classiques d'apprentissage machine.

Avant d'implémenter les modèles d'apprentissage machine, nous avons divisé l'ensemble de données en deux sous-ensembles : un ensemble d'entrainement et un ensemble de test.

5.1 Support Vector Machine (SVM).

Le SVM vise à maximiser la marge, qui représente la distance entre les vecteurs de support les plus proches (les instances les plus proches de la frontière de décision) et l'hyperplan de séparation. Cette marge est calculée comme la distance entre l'hyperplan et les vecteurs de support.

Pour un hyperplan donné wTx + b = 0, la distance d'une instance **Xi**, à l'hyperplan est **:**

$$d(x_i) = \frac{|w^T x_i + b|}{||w||} \quad (1)$$

Où ||x|| est la norme Euclidienne du vecteur de poids w.

Une fois que les paramètres w et b sont trouvés, la fonction de décision d'un SVM pour la classification binaire est définie comme suit :

$$f(x) = sign(w^T x + b) \quad (2)$$

Où sign est la fonction de signe qui attribue une classe positive ou négative en fonction du signe de wTx + b.

SVM détermine l'hyperplan de séparation qui maximise la séparation entre les classes dans l'espace des fonctionnalités, permettant ainsi une classification efficace des données.

5.2 Régression Logistique (LR).

La régression logistique est un modèle de régression utilisé pour prédire une variable dépendante binaire basée sur un ensemble de variables indépendantes. Le modèle est déterminé comme suit :

Supposons que nous ayons **n** observations avec **p** variables indépendantes.

A. La fonction logistique (sigmoid) :

La Régression Logistique fait usage de la fonction logistique (sigmoid) afin de transformer la combinaison linéaire des variables indépendantes en une valeur située entre 0 et 1. Cette valeur représente la probabilité que la variable dépendante appartienne à la classe positive. La fonction logistique est définie comme suit :

$$P(Y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p)}} \quad (1)$$

B. La fonction logit (log-odds) :

La transformation logit est utilisée pour exprimer la relation linéaire entre les variables indépendantes et la variable dépendante. La fonction logit est définie comme suit :

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \quad (1)$$

C. La fonction de coût (log loss) :

Afin d'ajuster les paramètres du modèle, la fonction de coût "log loss" ou "cross-entropy" est utilisée. Elle mesure la performance du modèle en comparant les prédictions aux valeurs réelles. Elle est définie comme suit :

$$J(\beta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \quad (1)$$

Où **yi** est la valeur réelle de la variable dépendante pour l'observation **i**, **pi** est la probabilité prédite par le modèle pour l'observation **i**, et **β** représente les coefficients du modèle.

D. Optimisation des coefficients (Maximum Likelihood Estimation) :

L'objectif est de déterminer les coefficients β qui minimisent la fonction de coût. Cette tâche peut être réalisée en employant des techniques d'optimisation telles que l'Estimation du Maximum de Vraisemblance (MLE) ou d'autres méthodes d'optimisation numérique.

La Régression Logistique cherche ainsi à trouver les coefficients β qui minimisent la fonction de coût, ce qui permet de modéliser la relation entre les variables indépendantes et la probabilité que la variable dépendante soit dans la classe positive.

5.3 Classificateur d'arbre de décision (DT).

Le modèle de l'arbre de décision repose sur une série d'algorithmes qui cherchent à diviser les données en sous-groupes homogènes. Voici les principales étapes de sa construction :

1. **Sélection de la meilleure division :**
   L'algorithme identifie la meilleure caractéristique et la meilleure valeur seuil pour diviser les données en sous-groupes homogènes. Cette sélection se base sur des mesures telles que l'indice de Gini ou l'entropie.

2. **Construction récursive de l'arbre :**
   Une fois la division effectuée, le processus est répété de manière récursive pour chaque sous-groupe jusqu'à atteindre une profondeur maximale ou un nombre minimum d'échantillons.
   La construction de l'arbre peut être formalisée par des équations, mais celles-ci dépendent des algorithmes utilisés, tels que CART, ID3, C4.5, etc.
   Voici comment l'indice de Gini est calculé pour mesurer l'impureté d'un nœud dans un arbre de décision :
   Soit p(i|t) la probabilité que la classe i apparaisse dans le nœud t. L'indice de Gini (G(t)) pour le nœud t est calculé comme suit :

$$G(t) = 1 - \sum_{i=1}^{c} [p(i|t)]^2 \quad (1)$$

   Où **c** représente le nombre de classes. Un indice de Gini plus élevé indique un mélange plus important de classes dans le nœud, ce qui signifie une plus grande impureté.

5.4 Classificateur de Forêt d'Arbres Aléatoires (RF).

La méthode de la forêt aléatoire (RF), également désignée sous le nom de technique d'ensemble, peut aborder deux aspects en développant plusieurs arbres de décision pour résoudre des problèmes de régression ou de classification. Chaque arbre fonctionne comme un apprenant faible, et ils sont ensuite regroupés pour former un apprenant robuste. Cette méthode est efficace et rapide, surtout lorsqu'il s'agit de traiter des ensembles de données déséquilibrés, même lorsque ces ensembles de données contiennent des milliers de caractéristiques. Chaque arbre au sein de la forêt aléatoire émet un vote pour la classification d'une classe, comme illustré dans la figure 15 [53]. Un nouvel objet est créé, et il se voit attribuer le plus grand nombre de votes pour une classe donnée.

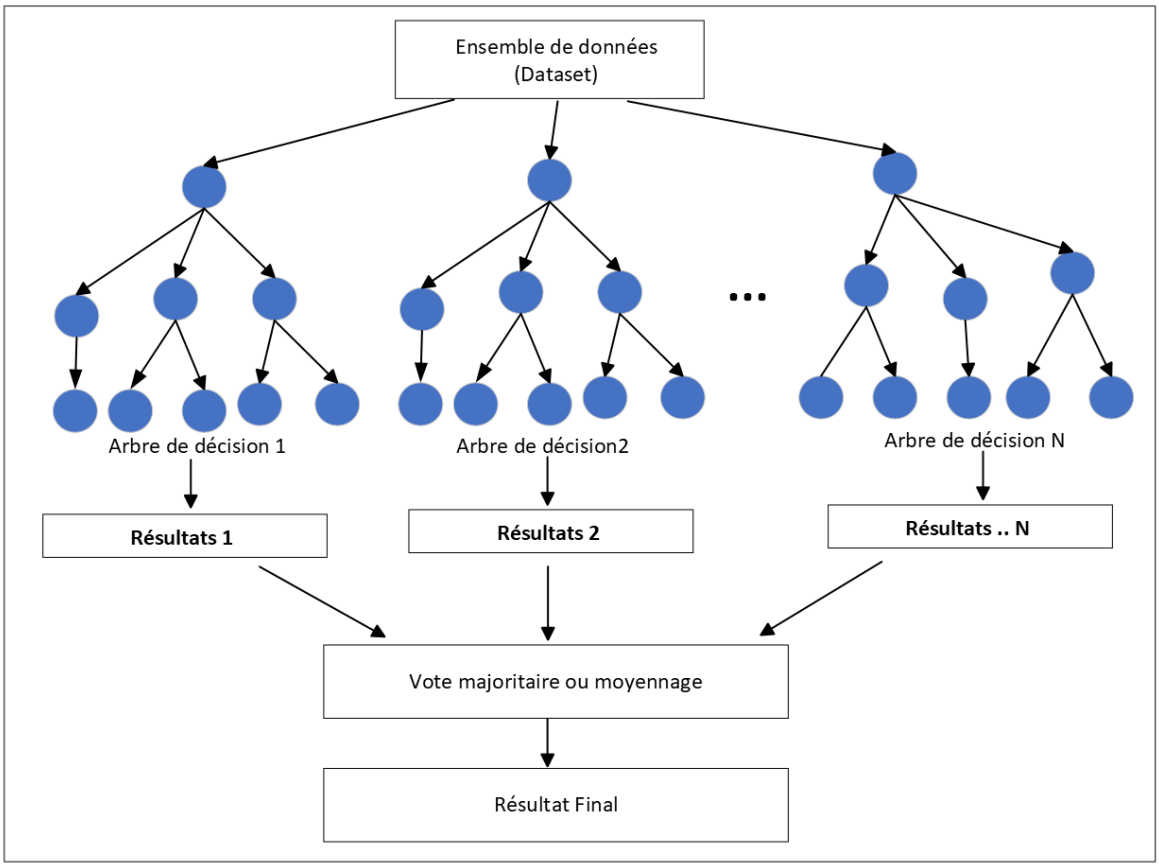


Figure 15 Forêt d'Arbres Aléatoires (RF)

5.5 K plus proches voisins (k-NN).

Une approche non paramétrique, supervisée, basée sur la distance en mesurant avec une fonction de distance les similarités dans l'ensemble de données. Les données de test sont classées en fonction des votes majoritaires de leurs k-voisins les plus proches.
Un ensemble d'entraînement est défini comme des paires X et Y :

- Soit X = {X1, X2, . . . Xn}. $x_i \in R^n$, correspond aux données d'entraînement dans l'ensemble de caractéristiques à n dimensions.
- Soit Y = {Y1, Y2, . . . Yn}, correspond aux étiquettes cibles.

Une fonction de distance euclidienne est utilisée pour mesurer la similarité dans les données d'entraînement. Pour deux points nommés a et b ayant des coordonnées cartésiennes (a1, a2) et (b1, b2), la distance entre (a) et (b) est calculée selon l'équation (1) :

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \quad (1)$$

La catégorie des données de test X est déterminée selon des votes majoritaires de ses k-voisins les plus proches. Ces étapes permettent de réaliser une prédiction pour des données de test x appliquées en entrée au modèle K-NN [49].

6 Prédiction sans sélection de caractéristiques.

Dans cette partie de prédiction sans sélection de caractéristiques, nous procédons par plusieurs étapes :

Etape 1 :

Dans cette phase, nous procéderons à la division de notre ensemble de données en deux parties distinctes : une portion constituera 70% des données, destinée à l'entraînement du modèle (X_train et y_train), tandis que l'autre portion, représentant 30% des données, sera réservée à l'évaluation des performances des modèles (X_test et y_test). Ces ensembles seront utilisés pour évaluer l'efficacité de différents algorithmes d'apprentissage machine (ML) tels que le Support Vector Machine (SVM), la Régression Logistique (LR), la Forêt Aléatoire (RF), les Arbres de Décision (DT) et la méthode des k plus proches voisins (kNN). Diverses mesures peuvent être employées pour évaluer les performances des algorithmes d'apprentissage machine, ces mesures étant dérivées des paramètres de la matrice de confusion.

Etape 2 :

Nous procéderons à la suppression des lignes de l'ensemble de données comportant au moins une valeur manquante, garantissant ainsi l'intégrité des données et les préparant à être utilisées dans nos modèles. De plus, nous extrairons la variable cible 'Label', essentielle à la classification et à l'évaluation des performances des modèles. Cette étape est fondamentale pour les modèles d'apprentissage supervisé, permettant à ces derniers d'apprendre à partir des données d'entraînement.

Etape 3 :

Dans le processus d'encodage des caractéristiques, les valeurs des attributs catégoriels sont transformées en valeurs numériques à l'aide de la technique d'encodage "one-hot" [52]. Cette méthode attribue une valeur numérique à chaque catégorie, permettant ainsi la préparation des données pour une utilisation efficace avec divers algorithmes d'apprentissage automatique. Notre ensemble de données inclut une caractéristique catégorielle, nommément le Protocole, comme indiqué dans la figure 16. Après l'encodage de cette caractéristique, l'ensemble de données contient au total 22 caractéristiques.

	0	1	2	3	\
dt	11425	11605	11425	11425	
switch	1	1	1	1	
src	10.0.0.1	10.0.0.1	10.0.0.2	10.0.0.2	
dst	10.0.0.8	10.0.0.8	10.0.0.8	10.0.0.8	
pktcount	45304	126395	90333	90333	
bytecount	48294064	134737070	96294978	96294978	
dur	100	280	200	200	
dur_nsec	716000000	734000000	744000000	744000000	
tot_dur	101000000000.0	281000000000.0	201000000000.0	201000000000.0	
flows	3	2	3	3	
packetins	1943	1943	1943	1943	
pktperflow	13535	13531	13534	13534	
byteperflow	14428310	14424046	14427244	14427244	
pktrate	451	451	451	451	
Pairflow	0	0	0	0	
Protocol	UDP	UDP	UDP	UDP	
port_no	3	4	1	2	
tx_bytes	143928631	3842	3795	3688	
rx_bytes	3917	3520	1242	1492	
tx_kbps	0	0	0	0	
rx_kbps	0.0	0.0	0.0	0.0	
tot_kbps	0.0	0.0	0.0	0.0	
label	0	0	0	0	

Figure 16 Caractéristiques numériques et catégorielles

7 Prédiction avec sélection de caractéristiques.

7.1 L'algorithme Éliminateur récursive des fonctionnalités (RFE) :

Cette méthode est employée pour la sélection de caractéristiques spécifiques au sein d'un ensemble de données dans le dessein de simplifier sa structure. L'algorithme opère en éliminant méthodiquement les caractéristiques les moins pertinentes jusqu'à ce qu'un nombre prédéfini de caractéristiques demeure.

Le déroulement de l'algorithme RFE se décrit comme suit. Tout d'abord, nous initialisons l'ensemble de caractéristiques avec toutes ses fonctionnalités. Ensuite, nous évaluons l'ensemble initial des caractéristiques. Subséquemment, nous éliminons l'attribut jugé le moins significatif de l'ensemble des attributs. Par la suite, nous procédons à une réévaluation approfondie de l'ensemble réduit de caractéristiques. Nous répétons les étapes précédentes jusqu'à ce que le nombre désiré de caractéristiques soit atteint, tel que prédéfini. À l'aide de mesures telles que la précision, le rappel, la mesure F ou le RMSE, il est possible d'évaluer l'ensemble de caractéristiques. L'algorithme RFE représente une approche simple et efficace pour la sélection des aspects essentiels d'un ensemble de données. Il est souvent utilisé conjointement avec d'autres techniques de sélection de caractéristiques, telles que la sélection basée sur la corrélation ou la signification des variables.

7.2 Selection des caractéristiques.

Après l'utilisation de l'algorithme REF, nous avons obtenu les caractéristiques illustrées sur la figure 17 suivante :

```
Index(['dt', 'switch', 'pktcount', 'dur', 'flows', 'packetins', 'pktperflow',
       'byteperflow', 'pktrate', 'Pairflow', 'port_no', 'tx_kbps', 'rx_kbps',
       'tot_kbps'],
      dtype='object')
```

Figure 17 Ensemble de caractéristiques sélectionnées

Ensuite, nous avons complété les étapes suivantes :

Étape 4 : Nous allons réitérer les trois (3) étapes mentionnées dans la section précédente.

Étape 5 : Nous supprimons la caractéristique : *'dt'*, car nous n'avons pas besoins pour notre travail.

Étape 6 : Nous allons Calculer la matrice de corrélation absolue des caractéristiques.

7.3 Matrice de corrélation.

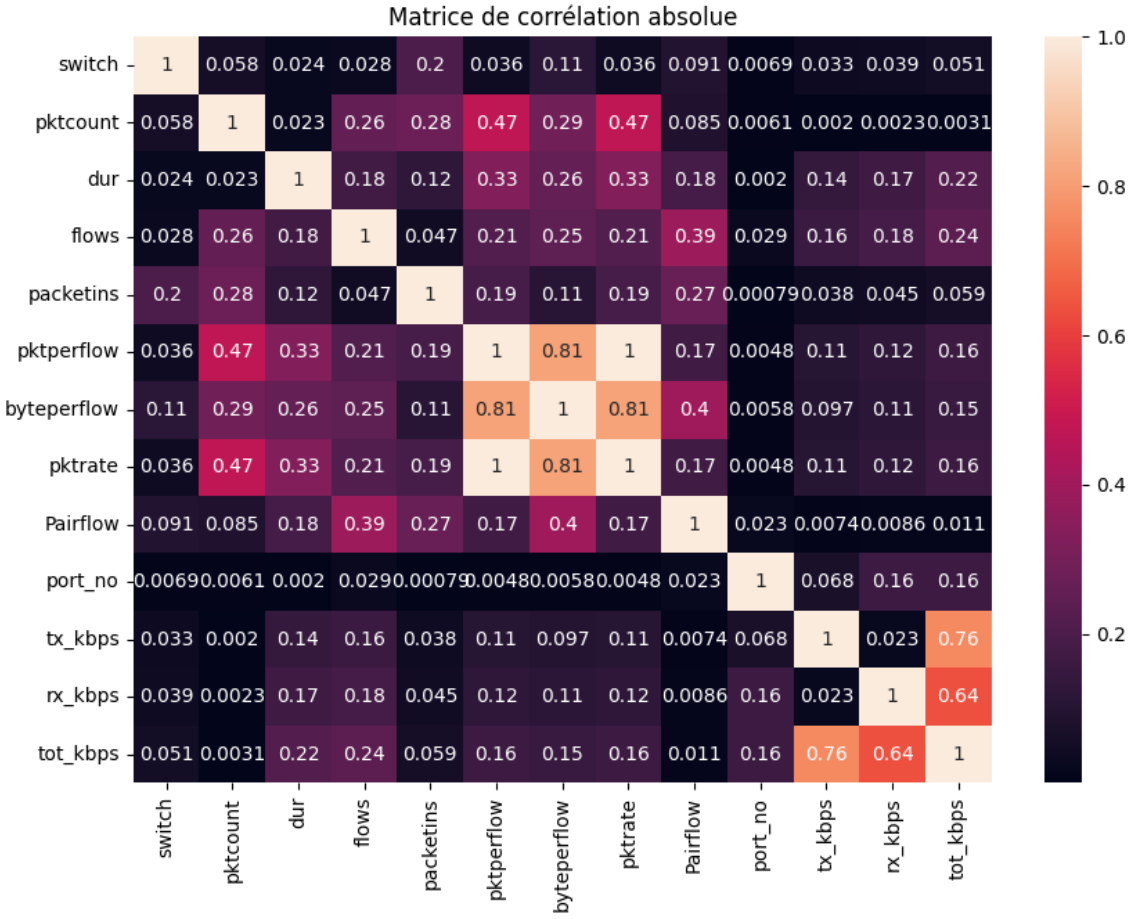


Figure 18 Matrice de corrélation absolue

Les valeurs de la matrice de corrélation indiquent le degré de corrélation entre deux caractéristiques. Une valeur de corrélation proche de 1 indique une forte corrélation positive, une valeur de corrélation proche de -1 indique une forte corrélation négative et une valeur de corrélation proche de 0 indique une faible corrélation.

D'après la matrice de corrélation, on peut voir que les caractéristiques les plus corrélées entre elles sont *"byteperflow"* et *"pktrate"* (corrélation de 0,81). Cela indique que ces deux caractéristiques sont très liées et qu'elles fournissent une information similaire, donc possible de conserver une seule de ces caractéristiques sans perdre de l'information.

Les autres caractéristiques qui sont corrélées entre elles sont *"tx_kbps"* et *"tot_kbps"* (corrélation de 0,76), *"rx_kbps"* et *"tot_kbps"* (corrélation de 0,64), *"byteperflow"* et *"pktperflow"* (corrélation de 0,81).

Les caractéristiques qui ne sont pas corrélées entre elles sont *"switch"*, *"flow"*, *"dur"* et *"port_no"*. Cela indique que ces caractéristiques fournissent des informations uniques qui ne sont pas redondantes.

Dans le chapitre 5, nous entreprendrons toutes les étapes précédentes pour chaque expérimentation. Les trois premières étapes seront réalisées pour la prédiction, tant sans qu'avec une sélection de caractéristiques. Les étapes 4 et 5 mentionnées dans la section 6.1 seront spécifiquement exécutées pour la prédiction avec sélection de caractéristiques. Nous allons également exposer et analyser les divers résultats obtenus.

CHAPITRE 5. DISCUSSION DES RESULTATS

1 Introduction

Dans ce qui suit, nous évaluerons les modèles élaborés au cours de notre étude et présenterons en détail les résultats obtenus. Nous effectuerons également une comparaison de nos résultats obtenus avec les travaux connexes.

2 Les mesures d'évaluation des modèles.

Pour notre travail, on utilise l'accuracy (ACC), la précision (P), le rappel (R), la spécificité (S), le temps d'entraînement et le temps de test comme métriques d'évaluation de détection d'anomalies. Ci-dessous, des descriptions relatives à ces métriques, accompagnées de leurs formules respectives :

- Vrai positif (TP) : Un résultat de test qui indique avec précision la présence d'une caractéristique ou condition.
- Faux positif (FP) : Un résultat de test qui indique à faux qu'une caractéristique ou une condition particulière est présente.
- Vrai négatif (TN) : Un résultat de test qui indique avec précision l'absence d'une caractéristique ou condition.
- Faux négatif (FN) : Un résultat de test qui indique à faux qu'une caractéristique ou une condition particulière est absente.

- **Accuracy** : pourcentage de flux correctement classés dans les classes normale et anormale par rapport au total des flux classés.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

- **Précision :** représente le pourcentage de flux anormaux correctement classés parmi l'ensemble des flux classés comme anormaux.

$$P = \frac{TP}{TP+FP} \times 100\%$$

- **Recall :** ou sensibilité, il représente le pourcentage de flux anormaux correctement classés parmi l'ensemble des flux effectivement anormaux.

$$R = \frac{TP}{TP+FN} \times 100\%$$

- **Score F1 :** une mesure de la précision d'un modèle d'apprentissage machine, en tenant compte à la fois de la précision et du rappel. Il est calculé en utilisant la formule :

$$F1 = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}$$

- **Support :** représente le pourcentage de flux normaux correctement classés parmi tous les flux réellement normaux.

$$S = \frac{TN}{TN+FP} \times 100\%$$

- **Temps d'entraînement :** représente le temps que le classificateur met à former le modèle.

- Temps de test : représente le temps que le classificateur met à tester le modèle.

2.1 La matrice de confusion.

La matrice de confusion fournit une représentation concise et visuelle des performances de l'algorithme, permettant l'évaluation de l'efficacité d'un algorithme de classification. Dans la matrice, chaque colonne représente des instances de la classe réelle, tandis que chaque ligne représente des instances de la classe prédite.

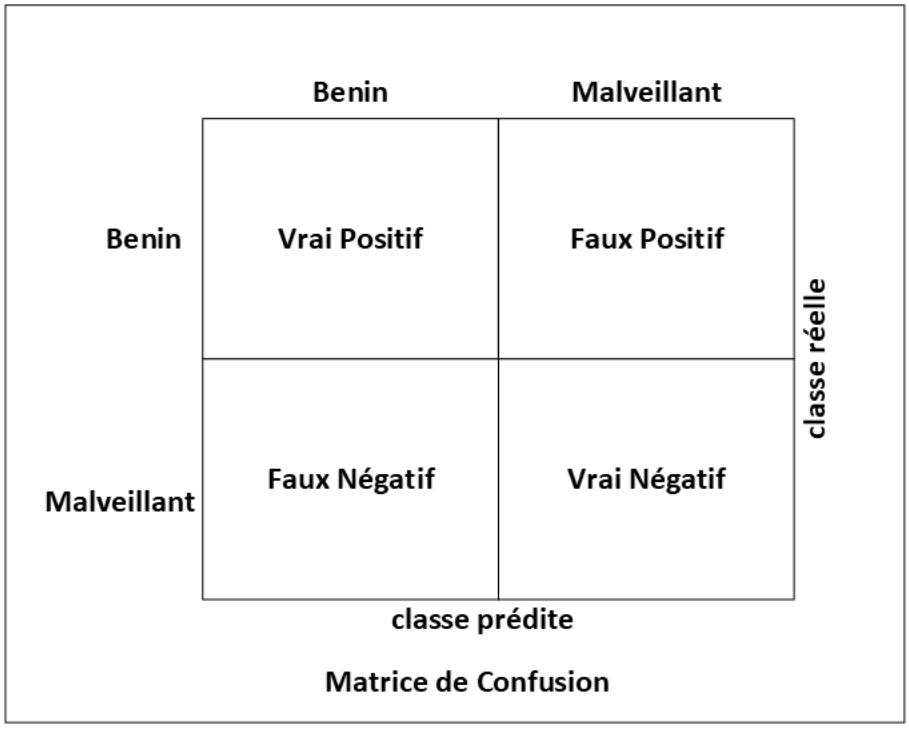


Figure 19 Matrice de confusion

3 Résultats de prédiction sans sélection de caractéristiques.

Après l'exécution des trois (3) étapes pour la prédiction sans sélection de caractéristiques citées dans le chapitre 4, nous procédons au test de cinq (5) méthodes de classification (SVM, LR, RF, DT, k-NN) sur le sous-ensemble de test de la Dataset. Le tableau 6, montre les résultats de chaque modèle pour obtenir le meilleur résultat de prédiction.

Méthode	Accuracy	Précision	Recall	F1-score	Support
SVM (RBF)	96.0 %	96.0%	97.0%	96.0%	18586
LR	73.94 %	84.0%	76.0%	80.0%	21086
RF	99.81 %	100%	100%	100%	18877
DT	98.13 %	98.0%	99.0%	98.0%	18877
K-NN	97.0 %	97.0%	98.0%	97.0%	18891

Tableau 6 Résultats de prédiction sans sélection de caractéristiques

Parmi nos méthodes étudiées, Random Forest (RF) a obtenu les meilleurs résultats avec une accuracy de 99.81%, grâce à sa capacité à reconnaître les schémas discriminatoires pour chaque catégorie, La classe minoritaire BENIGN qui représente le trafic normal avec une accuracy converge vers le 1 et un rappel de 100%. Cette précision signifie que le nombre de fausses alarmes est minimale, le rappel signifie que le modèle est efficace dans l'identification des attaques réseaux avec un taux de fausse négative (FN) réduits.

4 Résultats de prédiction avec sélection de caractéristiques.

4.1 Support vector Machine (SVM).

Dans le but de mesurer la performance du modèle SVM, nous avons expérimenté la prédiction en paramétrant le kernel en mode Linéaire ['linear'], puis en mode aléatoire : linéaire, polynomial, rbf (fonction à base radiale), sigmoid ['linear', 'poly','rbf', 'sigmoid'].

-Kernel : ['linear'], les résultats de prédiction sont représentés dans la figure 20, et la figure 21 présente la matrice de confusion. La classe Benign signifie que l'enregistrement n'est pas une attaque et la classe Malicious est une attaque.

```
------------------------------------------------------
              precision    recall  f1-score   support

           0       0.86      0.80      0.83     20543
           1       0.66      0.76      0.70     10609

    accuracy                           0.78     31152
   macro avg       0.76      0.78      0.77     31152
weighted avg       0.79      0.78      0.79     31152

------------------------------------------------------
--- 1162.4669806957245 seconds ---
```

Figure 20 Résultats de prédiction SVM avec kernel Linéaire

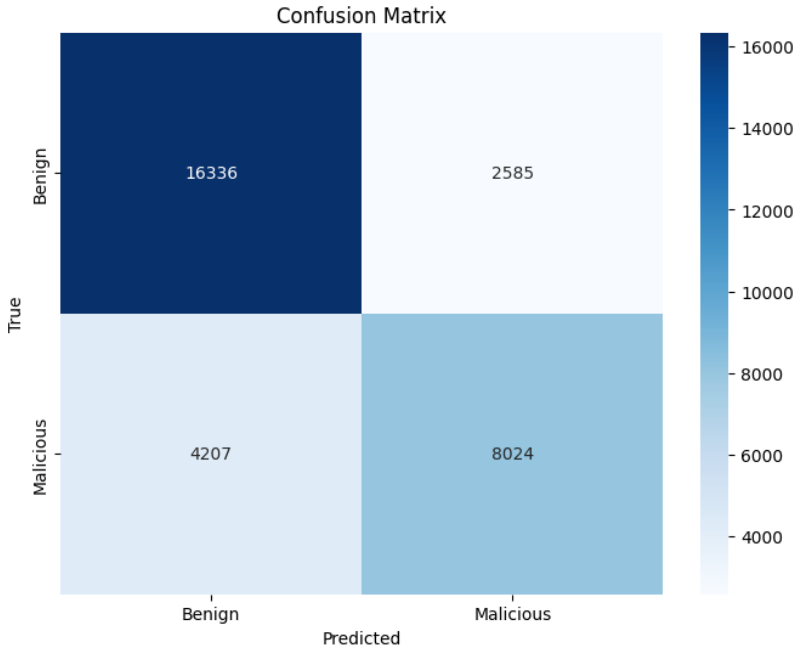


Figure 21 Matrice de confusion SVM avec kernel Linéaire

L'algorithme SVM avec le kernel Linéaire donne une Accuracy de 78.0% avec un temps d'entrainement et de prédiction du modèle égale à : 1162.47 secondes.

-Kernel : ['linear', 'poly','rbf', 'sigmoid'], la Figure 22 montre le rapport de classification des attaques, la matrice de confusion a également été extraite pour montrer les résultats (voir Figure 23).

```
------------------------------------------------------------
Accuracy of SVM model 97.0%

------------------------------------------------------------
best kernel is :  rbf
------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     18731
           1       0.97      0.95      0.96     12421

    accuracy                           0.97     31152
   macro avg       0.97      0.96      0.97     31152
weighted avg       0.97      0.97      0.97     31152

------------------------------------------------------------
--- 1543.4165852069855 seconds ---
```

Figure 22 Résultats de prédiction SVM avec kernel aléatoire

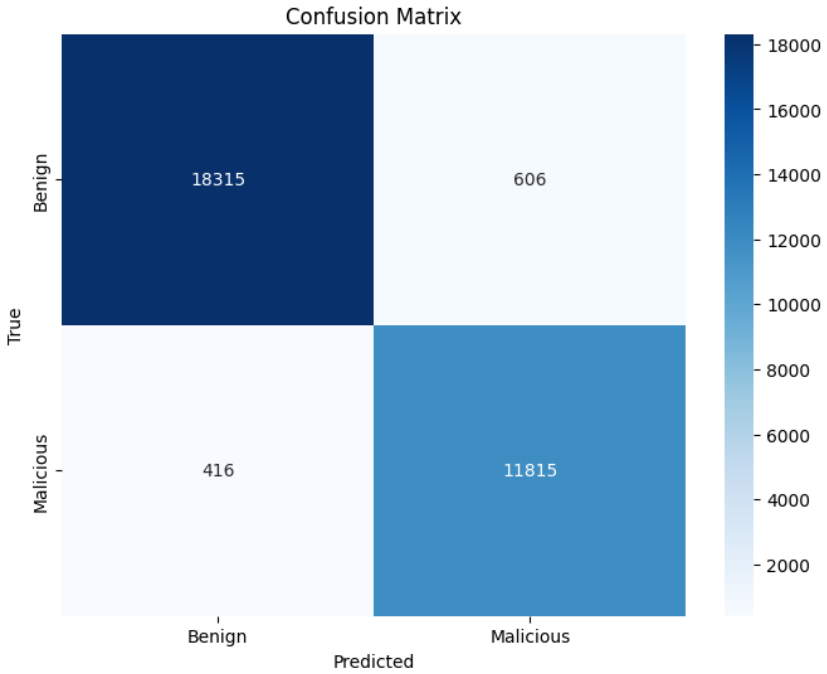


Figure 23 Matrice de confusion SVM avec kernel aléatoire

Le meilleur kernel est le rbf (fonction à base radiale) pour l'algorithme SVM qui donne une Accuracy de 97.0%, meilleure que le kernel Linear qui est de 78.0%, avec un temps d'entrainement et de prédiction du modèle égale à : 1543.42 secondes, supérieur à : 1162.47 secondes, en utilisant le kernel Linear.

Après l'extraction des caractéristiques les plus pertinentes en utilisant l'algorithme éliminateur récursive des fonctionnalités (RFE) et le calcul de la matrice de corrélation citée dans le chapitre 4, nous procédons à deux expérimentations afin de tester nos cinq (5) méthodes de classification : SVM avec le kernel rbf, LR, RF, DT, k-NN sur le sous-ensemble de test de la Dataset.

4.2 La première expérimentation.

Nous avons éliminé deux caractéristiques fortement corrélées, "pktrate" et "pktperflow", le tableau 7 montre les résultats de prédiction pour chaque classificateur.

Méthode	Accuracy	Précision	Recall	F1-score	Support
SVM (RBF)	97.0%	97.0%	98.0%	97.0%	18731
LR	76.23%	84.0%	78.0%	81.0%	20299
RF	99.99%	100%	100%	100%	18919
DT	98.35%	98.0%	99.0%	99.0%	18734
K-NN	98.0%	99.0%	99.0%	99.0%	18949

Tableau 7 Résultats de prédiction avec sélection de caractéristiques

Les résultats de l'expérimentation montrent une amélioration de précision pour l'ensemble des modèles, Random Forest (RF) a obtenu les meilleurs résultats avec une accuracy de 99.99%. La figure 24, montre le rapport de classification des attaques pour le modèle Random Forest (RF).

```
Accuracy of RF is : 99.99%

------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     18919
           1       1.00      1.00      1.00     12233

    accuracy                           1.00     31152
   macro avg       1.00      1.00      1.00     31152
weighted avg       1.00      1.00      1.00     31152

------------------------------------------------------
--- 38.33941888809204 seconds ---
```

Figure 24 Résultats de prédiction du modèle Random Forest (RF)

La figure 25, illustre la matrice de confusion qui a également été extraite pour montrer les résultats obtenus.

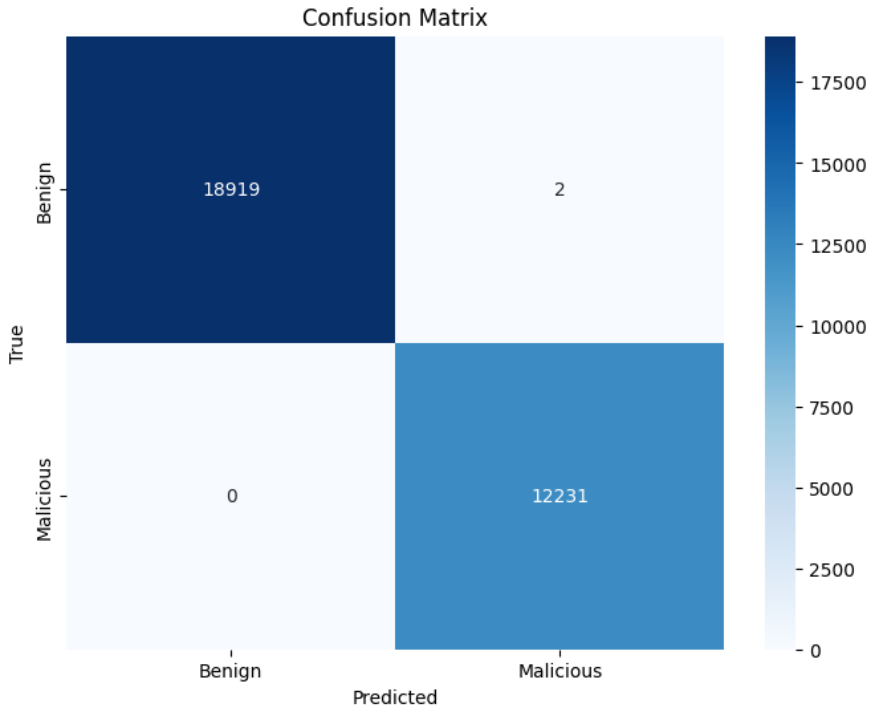


Figure 25 Matrice de confusion du modèle Random Forest (RF)

4.3 La deuxième expérimentation.

Pour cette deuxième expérimentation, nous avons éliminé quatre caractéristiques qui peuvent être considérées fortement corrélées, dont : *"tot_kbps", "pktrate", "pktperflow"*, le tableau 8 suivant, montre les résultats de prédiction pour chaque classificateur.

Méthode	Accuracy	Précision	Recall	F1-score	Support
SVM (RBF)	94.0%	95.0%	95.0%	95.0%	19002
LR	62.80%	94.0%	63.0%	75.0%	28055
RF	99.42%	99.0%	100%	100%	18890
DT	96.95%	96.0%	99.0%	97.0%	18446
K-NN	96.0%	97.0%	97.0%	97.0%	18866

Tableau 8 Résultats de prédiction avec sélection de caractéristiques

Les résultats de la deuxième expérimentation révèlent une diminution de la précision pour l'ensemble des modèles, bien que la méthode de la forêt aléatoire (Random Forest, RF) ait affiché les performances les plus élevées, avec une précision de 99,42%. Les résultats indiquent également que la sélection de caractéristiques fortement corrélées effectuée lors de

la première expérimentation a conduit à des performances supérieures. De plus, nos analyses démontrent l'efficacité du "Recursive Feature Elimination (RFE)" sur ces données, avec une amélioration supplémentaire des scores de classification après la réduction de dimensionnalité utilisant les mêmes classificateurs.

5 Comparaison entre nos résultats et les travaux issus de la littérature.

Nous avons effectué une comparaison de nos meilleurs résultats finaux avec ceux présentés dans l'article [32], basés sur le même ensemble de données mais en utilisant un algorithme de sélection de caractéristiques différent (NCA). Nos premières observations révèlent des résultats presque similaires à ceux de l'article [29], bien que les performances de notre modèle soient inférieures à celles du classificateur Decision Tree et du KNeighbors Classifier de l'article. En revanche, notre modèle Random Forest affiche d'excellents résultats avec une précision de 99,99 %, un F1-score de 100,00 %, un rappel de 100,00 % et une précision de 100,00 %. Il convient de noter que notre étude englobe une plus grande diversité de modèles étudiés. Le tableau 9 suivant, illustre les différents résultats obtenus :

	Nos résultats obtenus	**Résultats d'article [32].**
Approches	**Accuracy**	**Accuracy**
SVM	97.0%	97.20%
LR	76.23%	Non Etudié.
Random Forest (RF)	99.99%	Non Etudié.
Decision tree (DT)	98.35%	99.82%
K-NN	98.00%	99.23%

Tableau 9 Analyse comparative de nos résultats avec ceux présentés dans l'article [32]

De plus, l'application de l'algorithme RFE a donné un résultat exceptionnel avec une précision de 100 %. L'algorithme NCA s'est également révélé efficace pour cet ensemble de données. Nos résultats confirment que les modèles d'apprentissage machine étudiés sont efficaces pour détecter le trafic d'attaque, ce qui constitue l'objectif de notre recherche.

Nos expérimentations ont prouvé que l'utilisation de la méthode de sélection des caractéristiques RFE sur les données de trafic SDN améliore la précision des méthodes d'apprentissage machine dans la détection des attaques. Bien que cette méthode ne fournisse pas le nombre optimal de caractéristiques, ce point faible a été étudié dans notre recherche à travers diverses sélections de caractéristiques. Dans le contexte des attaques DDoS, qui requièrent une réponse rapide, il est crucial de détecter le trafic d'attaque de manière efficace pour une utilisation optimale des ressources système. Par conséquent, lors de la conception de modèles d'apprentissage machine, il est primordial de sélectionner les caractéristiques les plus pertinentes.

CHAPITRE 6. CONCLUSION ET PERSPECTIVES

Au cours des années, la distinction entre les attaques DDoS présentant différents taux et motifs et le trafic normal s'est avéré être une tâche particulièrement complexe. Divers chercheurs ont proposé de nombreuses méthodes efficaces d'apprentissage machine (ML) et d'apprentissage profond (DL) pour relever ce défi. Cependant, l'applicabilité de ces techniques est souvent limitée en raison de l'évolution constante des tactiques d'attaque utilisées par les assaillants. Notre revue de la littérature a synthétisé la taxonomie proposée pour la détection des attaques DDoS en utilisant des techniques d'apprentissage machine et profond, tout en identifiant les avantages et inconvénients respectifs de chaque approche. Le taux de précision rapporté dans une grande partie de la littérature dépasse les 96 %. Cependant, étant donné que la plupart de ces études ont évalué leurs modèles à l'aide d'une analyse de données hors ligne pour l'évaluation et la comparaison, certains indicateurs de performance peuvent varier dans un contexte opérationnel réel.

Au centre de cette étude réside le défi de détecter et de classifier les attaques à l'aide de méthodes d'apprentissage machine et profond, dans le but de renforcer la sécurité du système d'exploitation d'un réseau défini par logiciel (SDN), notamment son contrôleur, contre les attaques par déni de service distribué (DDoS). Notre recherche contribue à la sécurité du contrôleur dans les réseaux définis par logiciel (SDN). La nature novatrice du sujet des SDN ainsi que le manque de travaux de recherche dans ce domaine pourraient être attribués au déploiement encore restreint de cette infrastructure en tant que réseau de production. Les apports majeurs de notre étude se résument comme suit :

- Fournir un aperçu sur les réseaux définis par logiciel (SDN), y compris leurs architectures et les systèmes de détection d'intrusions (IDS) qui leur sont associés.
- Fournir une synthèse des diverses approches d'apprentissage machine et profondes présentées dans la littérature, puis évaluer leur précision et leur efficacité dans la détection des attaques DDoS.
- Proposer des orientations pour les travaux de recherche à venir, notamment l'importance d'avoir des ensembles de données plus précis et fiables, ainsi que l'exploration d'autres méthodes potentiellement plus efficaces que les réseaux neuronaux à convolution profonde.
- Diverses méthodes d'apprentissage machine et profond ont été examinées, comprenant notamment les machines à vecteurs de support (SVM), les arbres de décision (DT), la régression logistique (LR), la méthode des k plus proches voisins (kNN) et les forêts aléatoires (RF).
- Une analyse de l'efficacité de l'algorithme Recursive Feature Elimination (RFE) a été réalisée, comprenant une comparaison avec l'algorithme Neighborhood Component Analysis (NCA). L'objectif était d'évaluer la performance relative de chaque algorithme dans le contexte spécifique de l'étude.
- Notre étude a impliqué la réalisation de plusieurs expérimentations pour atteindre ses objectifs de recherche.

- Nos résultats, comparés à ceux de la littérature existante, fournissent des motifs d'encouragement, renforçant ainsi la crédibilité et l'importance de notre étude dans le contexte de la recherche actuelle.

En ce qui concerne les orientations de recherche à venir, nos conclusions soulignent la nécessité d'actualiser de manière dynamique et régulière les méthodes d'apprentissage machine et profond pour la détection des attaques DDoS. Cette nécessité découle de l'évolution constante et rapide des schémas d'attaques, qui exigent une adaptation aux nouveaux types d'attaques émergents. Cette flexibilité revêt une importance critique dans le contexte actuel où de nouvelles technologies émergent rapidement, accompagnées de menaces de plus en plus sophistiquées. Il convient de noter toutefois qu'aucun modèle d'apprentissage profond de cette nature n'est actuellement répertorié dans la littérature.

Dans les études à venir, nous envisageons d'accroître la diversité des attaques examinées et de comparer les performances de classification des modèles d'apprentissage machine en utilisant différents algorithmes de sélection de caractéristiques. De plus, nous prévoyons améliorer les méthodes de sélection de caractéristiques et de tirer parti des recherches menées sur les réseaux SDN en conditions réelles. Bien que les modèles actuels aient démontré leur efficacité dans la détection du trafic d'attaque DDoS, nos efforts futurs seront axés sur l'amélioration de ces modèles et l'intégration de nouveaux modèles afin d'optimiser davantage les résultats.

RÉFÉRENCES

[1] Anand Nayyar, Bhawna Singla, Preeti Nagrath, “Software Defined Networks: Architecture and Applications”, 2022, ISBN: 9781119857907.

[2] M. Parashar, A. Poonia and K. Satish, "A Survey of Attacks and their Mitigations in Software Defined Networks," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-8.

[3] E. Gyamfi and A. D. Jurcut, "Novel Online Network Intrusion Detection System for Industrial IoT Based on OI-SVDD and AS-ELM," in IEEE Internet of Things Journal, vol. 10, no. 5, pp. 3827-3839, 1 March1, 2023.

[4] M. Bano, S. S. A. Gilani and A. Qayyum, "A Comparative Analysis of Hybrid Routing Schemes for SDN Based Wireless Mesh Networks," 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, UK, 2018, pp. 1189-1194.

[5] Wenjuan Li, Weizhi Meng, Lam For Kwok, A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures, Journal of Network and Computer Applications, Volume 68, 2016, Pages 126-139, ISSN 1084-8045.

[6] Open Networking Foundation. Software-defined networking: The new norm for networks. ONF White Paper, 2012.

[7] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, Jan. 2015.

[8] Benamrane, Fouad & Ben Mamoun, Mouad & Redouane, Benaini. (2017). Étude des Performances des Architectures du Plan de Contrôle des Réseaux ‘Software-Defined Networks'. Doi: 10.13140/RG.2.2.33883.57126.

[9] Jingyu Hua, Laiping Zhao, Suohao Zhang, Yangyang Liu, Xin Ge, Sheng Zhong, Topology-Preserving Traffic Engineering for Hierarchical Multi-Domain SDN, Computer Networks, Volume 140, 2018, Pages 62-77, ISSN 1389-1286.

[10] D. B. Rawat and S. R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 325-346, Firstquarter 2017.

[11] Venkatraman Balasubramanian, Moayad Aloqaily, Martin Reisslein, An SDN architecture for time sensitive industrial IoT, Computer Networks, Volume 186, 2021, 107739, ISSN 1389-1286.

[12] Murat Karakus, Arjan Durresi, Quality of Service (QoS) in Software Defined Networking (SDN): A survey, Journal of Network and Computer Applications, Volume 80, 2017, Pages 200-218, ISSN 1084-8045.

[13] TianZhang He, Adel N. Toosi, Rajkumar Buyya, Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers, Journal of Parallel and Distributed Computing, Volume 131, 2019, Pages 55-68, ISSN 0743-7315.

[14] P. Ohri, S. G. Neogi and S. K. Muttoo, "Security Analysis of Open Source SDN (ODL and ONOS) Controllers," 2023 IEEE International Students' Conference on Electrical,

Electronics and Computer Science (SCEECS), Bhopal, India, 2023, pp. 1-4.
[15] A. Bradai, K. Singh, T. Ahmed and T. Rasheed, "Cellular software defined networking: a framework," in IEEE Communications Magazine, vol. 53, no. 6, pp. 36-43, June 2015.
[16] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, Jan. 2015.
[17] Steven Noble. 2017. Building Modern Networks, July 29th, 2017, ISBN: 9781786466976.
[18] C. Martinez, R. Ferro and W. Ruiz, "Next generation networks under the SDN and OpenFlow protocol architecture," 2015 Workshop on Engineering Applications - International Congress on Engineering (WEA), Bogota, Colombia, 2015, pp. 1-7.
[19] X. -N. Nguyen, D. Saucez, C. Barakat and T. Turletti, "Rules Placement Problem in OpenFlow Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1273-1286, Secondquarter 2016.
[20] Sahay R., Blanc G., Zhang Z., Toumi K., Debar H. "Adaptive policy-driven attack mitigation in SDN", XDOM0 2017 - Co-located with European Conference on Computer Systems, EuroSys 2017, art. no. 4.
[21] S. Muzafar, N. Jhanjhi, N. A. Khan and F. Ashfaq, "DDoS Attack Detection Approaches in on Software Defined Network," 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), Karachi, Pakistan, 2022, pp. 1-5.
[22] Keunsoo Lee, Juhyun Kim, Ki Hoon Kwon, Younggoo Han, Sehun Kim, DDoS attack detection method using cluster analysis, Expert Systems with Applications, Volume 34, Issue 3, 2008, Pages 1659-1665, ISSN 0957-4174.
[23] J. Ashraf and S. Latif, "Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques," *2014 National Software Engineering Conference*, Rawalpindi, Pakistan, 2014, pp. 55-60.
[24] Deng, Shuhua & Gao, Xing & Lu, Zebin & Gao, Xieping. (2017). Packet Injection Attack and Its Defense in Software-Defined Networks. IEEE Transactions on Information Forensics and Security. PP. 1-1.
[25] A. Borkar, A. Donode and A. Kumari, "A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS)," 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 2017, pp. 949-953.
[26] S. Chadli, M. Saber, M. Emharraf and A. Ziyyat, "A new model of IDS architecture based on multi-agent systems for MANET," 2014 Second World Conference on Complex Systems (WCCS), Agadir, Morocco, 2014, pp. 252-258.
[27] Wood, Mark, and Michael Erlinger. Intrusion detection message exchange requirements. No. rfc4766. 2007.
[28] Software Engineering, Journal of, Intelligent Systems, and Qanetah Ahmed. "APPLICATIONS OF MACHINE LEARNING IN EDUCATION AND HEALTH SECTOR: AN EMPIRICAL STUDY." Journal of Software Engineering & Intelligent Systems 4, no. 3 (2019): 163–68.

[29] Mbaye, Maissa & Hamdi, Omessaad. (2020). Un plan de contrôle intelligent pour le déploiement de services de sécurité dans les réseaux SDN. 10.51926/ISTE.9008.ch2.
[30] S. T. Zargar, J. Joshi and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," in IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046-2069, Fourth Quarter 2013.
[31] Ali, Tariq Emad, Yung-Wey Chong, and Selvakumar Manickam. 2023. "Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review" Applied Sciences 13, no. 5: 3183.
[32] Tonkal, Ö.; Polat, H.; Başaran, E.; Cömert, Z.; Kocaoğlu, R. Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking. Electronics 2021, 10, 1227.
[33] Sambangi, Swathi, and Lakshmeeswari Gondi. 2020. "A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression" Proceedings 63, no. 1: 51.
[34] S. Murtuza and K. Asawa, "Mitigation and Detection of DDoS Attacks in Software Defined Networks," 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2018, pp. 1-3.
[35] J. A. Pérez-Díaz, I. A. Valdovinos, K. -K. R. Choo and D. Zhu, "A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning," in IEEE Access, vol. 8, pp. 155859-155872, 2020.
[36] S. Haider, A. Akhunzada, G. Ahmed and M. Raza, "Deep Learning based Ensemble Convolutional Neural Network Solution for Distributed Denial of Service Detection in SDNs," 2019 UK/ China Emerging Technologies (UCET), Glasgow, UK, 2019, pp. 1-4.
[37] Polat, Huseyin, Onur Polat, and Aydin Cetin. 2020. "Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models" Sustainability 12, no. 3: 1035.
[38] A. T. Kyaw, M. Zin Oo and C. S. Khin, "Machine-Learning Based DDOS Attack Classifier in Software Defined Network," 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, 2020, pp. 431-434.
[39] M. M. Oo, S. Kamolphiwong and T. Kamolphiwong, "The Design of SDN Based Detection for Distributed Denial of Service (DDoS) Attack," 2017 21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand, 2017, pp. 1-5.
[40] K. S. Sahoo et al., "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," in IEEE Access, vol. 8, pp. 132502-132513, 2020.
[41] S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," in IEEE Access, vol. 8, pp. 5039-5048, 2020.
[42] K. Alhamami and S. Albermany, "DDOS attack detection using machine learning algorithm in SDN network," 2023 Al-Sadiq International Conference on Communication and Information Technology (AICCIT), Al-Muthana, Iraq, 2023, pp. 97-102.

[43] Liu, Zhenpeng, Yihang Wang, Fan Feng, Yifan Liu, Zelin Li, and Yawei Shan. 2023. "A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks" Sensors 23, no. 13: 6176.
[44] J. Ashraf and S. Latif, "Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques," 2014 National Software Engineering Conference, Rawalpindi, Pakistan, 2014, pp. 55-60.
[45] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2018, pp. 29-35.
[46] M. S. Elsayed, N. -A. Le-Khac, S. Dev and A. D. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland, 2020, pp. 391-396.
[47] N. Ravi and S. M. Shalinie, "Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture," in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3559-3570, April 2020.
[48] S. Haider et al., "A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks," in IEEE Access, vol. 8, pp. 53972-53983, 2020.
[49] X. Yuan, C. Li and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," 2017 IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, China, 2017, pp. 1-8.
[50] N. Chaid. « La sécurité des communications dans les réseaux VANET », Mémoire, Universite Elhadj Lakhder – Batna, Faculté des sciences de l'ingénieur, Département d'informatique.
[51] Ahuja, N.; Singal, G.; Mukhopadhyay, D. "DDOS attack SDN Dataset", Mendeley Data, V1; Bennett University: Greater Noida, India, 2020.
[52] Shao, E. Encoding IP Address as a Feature for Network Intrusion Detection. Ph.D. Dissertation, Purdue University Graduate School, West Lafayette, Indiana, 2019.
[53] Li, Y.; Li, L.; Fang, Y.; Peng, H.; Ling, N. Bagged Tree and ResNet-Based Joint End-to-End Fast CTU Partition Decision Algorithm for Video Intra Coding. Electronics 2022, 11, 1264. https://doi.org/10.3390/electronics11081264.
[54] "RFC 4766 - Intrusion Detection Message Exchange Requirements." https://datatracker.ietf.org/doc/rfc4766/, doi:10.17487/RFC4766. (Consulté le 21 Mai 2024).
[55] KDD Cup 1999 Data. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (Consulté le 21 Mai 2024).
[56] Kyoto 2006+ Dataset. Available online: https://www.impactcybertrust.org/dataset_view?idDataset=918 (Consulté le 21 Mai 2024).
[57] Canadian Institute for Cybersecurity. Available online: https://www.unb.ca/cic/datasets/nsl.html (Consulté le 21 Mai 2024).
[58] The UNSW-NB15 Dataset. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (Consulté le 21 Mai 2024).

[59] Canadian Institute for Cybersecurity. Available online: https://www.unb.ca/cic/datasets/ids-2017.html (Consulté le 21 Mai 2024).
[60] Canadian Institute for Cybersecurity. Available online: https://www.unb.ca/cic/datasets/ids-2018.html (Consulté le 21 Mai 2024).
[61] Canadian Institute for Cybersecurity. Available online: https://www.unb.ca/cic/datasets/ids.html (Consulté le 21 Mai 2024).
[62] Canadian Institute for Cybersecurity. Available online: https://www.unb.ca/cic/datasets/ddos-2019.html (Consulté le 21 Mai 2024).