République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche

Université de Tébessa

Faculté des Sciences Exactes et
des Sciences de la Nature et de la Vie

Département des mathématiques et informatique

Mémoire
Présenté en vue de l'obtention du diplôme de MASTER

Filière : (Mathématiques/Informatique)

Option : **Réseaux & Sécurité informatique**

Par
AMARA SAKINA

UNE APPROCHE INTELLIGENTE DEEP LERNING POUR LA DETECTION DES ATTAQUES DDOS POUR LE RESEAU SDN

Date de soutenance : 13/06/2022

Devant le jury

Mr. Abbas faycel	MCB	Univérsité Larbi Tébessi	Président
Mme.Bouakkaz Fatima	MAA	Univérsité Larbi Tébessi	Examinateur
Dr.Merzoug Soltan	MCB	Univérsité Larbi Tébessi	Encadreur

Année Universitaire: **2021/2022**


بسم الله الرحمن الرحيم

Remerciement

Nous remercions **ALLAH** tout puissant pour nous avoir donné la foi et éclairé notre chemin vers la réussite durant toutes nos années d'étude.

Nous voudrions à remercier notre enseignant et encadreur de mémoire :

- **Dr.Merzoug Soltan** qui a accepté la direction de ce travail ainsi que pour ses conseil judicieux et précieux, ses compétences scientifiques, sa confiance qu'elle nous accordée surtout pour ses très grands qualités humaines et sa gentillesse.

Aussi nous voudrions à remercier les membres de jury :

- **Mme.Bouakkaz Fatima** , nous sommes très honorés de vous accepter pour présider le jury de cette thèse, vous trouvez ici l'expression de nos sincères remerciements et la confirmation de notre profonde gratitude.
- **Mr. Abbas faicel** , merci d'avoir accepté de faire partie du jury de cette thèse, pour l'intérêt que vous portez à nos travaux.

Dédicace

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quelque soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affectionne couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles : mon adorable mère Wahiba.

A l'homme, mon précieux offre du dieu, vous avez été à mes côtés pour me soutenir et m'encourager. Que ce travail reflète ma gratitude et mon amour pour toi : mon cher père Layachi.

A mes frères Taki el eddine , Mohamed Baha eddine que dieu les protégés.

A la joie de ma vie, ma petite sœur Nour el yakine, tu connais ta place dans mon cœur… Je t'aime.

A tous mes amies en particulier Rabeb, Zaineb, Manel et Meriam.

À tous ceux qui, par un mot, m'ont donné la force de continuer...

À tous ceux qui m'aiment et que j'aime...

Sommaires

Listes des figures

Listes des tableaux

Liste des abréviations

Mot	Anglaise	Français
SDN	Software Defined Networking	La réseautique définie par logiciel
SVM	Support Vector Machine	Support Vecteur Machine
Ddos	Distributed Denial of Service attack	Attaque par déni de service distribuée
IA	Intelligence Artificial	Intelligence Artificielle
ML	Machine Learning	Apprentissage automatique
DL	Deep Learning	L'apprentissage en profondeur
KNN	K Nearst Neighbors	K plus proches voisins
RFE	Recursive feature Eliminator	Éliminateur Récursive des fonctionnalités
NSL-KDD	Network Security Laboratory Knowledge - Discovery and Data Mining	Laboratoire de sécurité des réseaux - Découverte de connaissances et exploration de données

Résumé

Le réseau défini par logiciel (SDN) est un paradigme de mise en réseau prometteur qui offre une commodité, une évolutivité, une contrôlabilité et une flexibilité exceptionnelles. Malgré ces fonctionnalités prometteuses, le SDN n'est pas intrinsèquement sécurisé. Par exemple, il souffre toujours d'attaques par déni de service (DDoS), qui est l'une des principales menaces qui compromettent la disponibilité du réseau. Un type d'attaques DDoS, considéré comme l'un des plus difficiles à détecter, sont les attaques DDoS lentes. Ces dernières années, des algorithmes d'apprentissage automatique ont été appliqués pour une détection fiable et très précise des anomalies de trafic. Certains algorithmes de machine Learning sur la détection DDoS sont évalués en thèse. Les meilleures caractéristiques sont choisies en fonction de la précision de la classification et des performances du contrôleur SDN. Pour identifier les attaques SDN, Une comparaison est faite entre les modèles d'apprentissage en profondeur et les classificateurs d'apprentissage automatique en sélectionnant ensemble des fonctionnalités. Les résultats expérimentaux ont révélé qu'en utilisant la méthode d'exclusion de fonctionnalités récursives (RFE), le classificateur SVM (RFE) forme le modèle le plus précis avec une précision de 99,99 % tout en utilisant des modèles d'apprentissage en profondeur (CNN-1D), nous obtenons une précision de 63 %.

Mots clés : SDN, Ddos, Apprentissage automatique, Apprentissage profond , élimination des caractéristiques récursives (RFE) , détection automatique des attaques, NSL-KDD.

Abstract

Software-defined networking (SDN) is a promising networking paradigm that offers exceptional convenience, scalability, controllability, and flexibility. Despite these promising features, SDN is not inherently secure. For example, it still suffers from denial of service (DDoS) attacks, which is one of the main threats that compromise network availability. One type of DDoS attacks, considered one of the most difficult to detect, are slow DDoS attacks. In recent years, machine learning algorithms have been applied for reliable and highly accurate detection of traffic anomalies. Some algorithms machine learning on DDoS detection are assessed in thesis. The best characteristics are chosen based on classification accuracy and the SDN controller's performance. To identify SDN assaults, A comparison of deep learning models and machine learning classifiers is done together selecting features. Experimental results revealed that using the Recursive Feature Exclusion (RFE) method, the SVM classifier (RFE) trains the most accurate model with 99.99 percent accuracy while using deep learning models (CNN-1D) we get 63 percent accuracy.

Keywords: SDN, Ddos, machine learning, Deep learning ,automatic attack detections, NSL-KDD,RFE.

الملخص

تعد الشبكات المعرفة بالبرمجيات (SDN) نموذجًا واعدًا للشبكات يوفر راحة استثنائية وإمكانية التوسع وقابلية التحكم والمرونة. على الرغم من هذه الميزات الواعدة ، فإن SDN ليست آمنة بطبيعتها. على سبيل المثال ، لا يزال يعاني من هجمات رفض الخدمة (DDoS) ، والتي تعد أحد التهديدات الرئيسية التي تهدد توفر الشبكة. تعتبر هجمات DDoS البطيئة أحد أنواع هجمات DDoS ، والتي تعتبر من أصعب الهجمات التي يمكن اكتشافها. في السنوات الأخيرة ، تم تطبيق خوارزميات التعلم الآلي والتعلم العميق للكشف الموثوق والدقيق للغاية عن الانحرافات المرورية. يتم تقييم بعض خوارزميات التعلم الآلي والتعلم العميق على اكتشاف DDoS في أطروحة. يتم اختيار أفضل الخصائص بناءً على دقة التصنيف وأداء وحدة التحكم في SDN. لتحديد هجمات SDN ، يتم إجراء مقارنة بين نماذج التعلم العميق ومصنفات التعلم الآلي معا اختيار الميزات. كشفت النتائج التجريبية أن استخدام أسلوب استبعاد الميزات العودية (RFE) ، يقوم المصنف SVM (RFE) بتدريب النموذج الأكثر دقة بدقة 99.99 بالمائة بينما عند استخدام نماذج التعلم العميق (CNN-1D)تحصلنا على دقة 63 بالمئة .

الكلمات المفتاحية :الشبكات المعرفة بالبرمجيات، هجمات رفض الخدمة، التعلم الالي ،التعلم العميق، قاعدة البيانات(NSL-KDD) ، الكشف الالي عن الهجمات , استبعاد الميزات العودية (RFE) .

Introduction Générale

1. Introduction

Dans les réseaux traditionnels, le réseau se composait des matériels (commutateur, routeur, serveur, etc.) avec le passage du temps et les sauts dans le monde, comme l'émergence du cloud comptine et de l'Internet des objets, il est devenu nécessaire pour les réseaux de faire une révolution qui suit le rythme des changements et des exigences, et c'est ce qui s'est réellement passé là où le commutateur a virtuel , est apparu Le réseau est passé des appareils uniquement aux appareils et les logiciel cette architecture nomes Sdn.

SDN (Software Defined Networking) (La réseautique définie par logiciel) résout les problèmes des réseaux existants, le SDN est un réseau programmable et virtualisé qui vous aide à insérer de nouvelles idées dans votre recherche. SDN supprime le plan de contrôle du plan de données. Le plan de contrôle est responsable du traitement des informations, tandis que le plan de données est responsable du transfert de données [1], Le SDN peut être déployé dans de nombreux réseaux différents, tels que les réseaux privés, les réseaux d'entreprise et les réseaux étendus. Malheureusement, le SDN présente de nombreux défis qui doivent être relevés. L'évolutivité, les performances et la sécurité sont quelques-uns des défis auxquels le SDN est confronté.

La structure de la console centrale peut entraîner plusieurs problèmes de sécurité. L'un de ces défis critiques est l'impact des attaques par déni de service distribué (DDoS) sur les SDN, car de telles attaques peuvent rapidement faire tomber l'ensemble du réseau en laissant tomber la console. Étant donné que les paquets d'attaque sont envoyés avec de nombreuses adresses IP source usurpées, une attaque DDoS peut causer des problèmes à la fois aux commutateurs et au contrôleur [2]. De plus, dans le flux d'attaques DDoS, les attaquants utilisent de nombreuses adresses IP source d'usurpation, ce qui rend impossible l'arrêt des attaques en bloquant le trafic basé uniquement sur l'adresse IP source. La mise en œuvre pratique des méthodes de détection et de réponse DDoS était la clé du fonctionnement régulier du réseau. Ce problème est plus présent dans les SDN en raison du point de défaillance unique (le contrôleur).

2. La problématique

Avec l'explosion des données issues de diverses infrastructures informatiques (Réseaux, Systèmes, Internet des Objets, etc.). L'architecture SDN est confrontée à des défis importants face à de nombreuses formes de cyberattaques telles que les attaques Ddos. Comment assurer la sécurité de cette architecture en termes de taux de détection et d'identification de ce type d’attaque malveillante avec un fable taux de fausses alarmes. Comment assurer que les méthodologies de détection sont robustes et évolutives.

3. Objectif du project

L'objectif principal de ce travail est de concevoir et de développer nouvelle solution qui permettrait de protéger le réseau SDN contre les attaques de Ddos.

Dans cette étude j'ai implémenté des méthodes de détection basant sur les approches d'intelligence artificiel et évaluer les performances des systèmes élaborés. Nous utilisons le jeu de donnée NSL-KDD extrait du jeu de données KDD99 est le plus largement utilisé dans la littérature pour aider les administrateurs des réseaux et des systèmes à détecter et identifier toute violation de la sécurité réseaux.

4. Structure de la these

Dans ce mémoire, nous visons à étudier l'architecture SDN, les attaques DDoS dans les SDN les attaques DDoS dans les réseaux SDN, les techniques disponibles pour détecter cette attaque, étudier l'impact des DDoS sur différentes topologies, et on implémenter une approche qui détecte les DDoS dans les réseaux SDN.

Ce travail s'organise ainsi :

- ☑ **Chapitre 1 :** Donne un aperçu du concept des réseaux SDN, et de protocole OpenFlow après nous allons exposer les notions des attaques de DDoS, leur impact sur les réseaux SDN et les systèmes de détection DDoS dans SDN.
- ☑ **Chapitre 2 :** Présente un état de l'art sur les Datasets et les travaux connexes ainsi que notre synthèse.
- ☑ **Chapitre 3 :** Detaille notre contribution
- ☑ **Chapitre 4 :** Présente l'implémentation du mode les proposées avec une discussion sur les résultats obtenus et comparaison avec les travaux connexes.

Chapitre 1

Le réseau SDN & Les attaques Ddos

1. Introduction

Le développement rapide est l'une des caractéristiques les plus importantes de la science des technologies de l'information, et la science des réseaux, comme les autres sciences des technologies de l'information, a connu de nombreuses innovations dans les dispositifs, les applications, les services et les outils au cours des dernières décennies. Une science de la technologie qui est restée ferme sur ses fondations et son architecture anciennes, il est temps de la révolutionner avec la technologie SDN.

2. La réseautique définie par logiciel – SDN

2.1.Définition

L'Open Network Fondation (ONF) définit le réseau SDN comme suit : « Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services» [4].
La réseautique définie par logiciel (SDN) : c'est un nouveau modèle qui décrit une architecture réseau dans laquelle le plan de contrôle est complètement séparé du plan de données, L'idée principale de ce nouveau paradigme est de sortir la partie intelligente des équipements d'interconnexions, et la placer vers un seul point de contrôle appelé contrôleur, ce dernier fournit une vue centrale de réseau, ce qui simplifie d'une part, la gestion et la configuration de réseau.[3]

2.2.Architecture de SDN

Une SDN se compose de trois couches de base et d'interfaces de communication, que nous classons ci-dessous:

1) La couche de transmission (plan de données)

Il est constitué d'équipements de routage tels que des commutateurs ou des routeurs, et son rôle principal est de transmettre des données et de collecter des statistiques.

2) La couche de contrôle (plan de contrôle)

Il est principalement constitué d'un ou plusieurs contrôleurs SDN, dont le rôle est de contrôler et de gérer les équipements de l'infrastructure au travers d'une interface appelée "south-bound API".

3) La couche de applications

Ils représentent des applications qui permettent le déploiement de nouvelles fonctionnalités réseau, telles que l'ingénierie du trafic, la qualité de service, la sécurité, etc. Ces applications sont créées via une interface de programmation appelée " north-bound API" [3].

4) Interfaces de communications

Il y a trois types d'interfaces permettent aux contrôleurs de communiquer avec leur environnement :

- **Interface Sud :** qui permettent au contrôleur SDN d'interagir avec les équipements de la couche d'infrastructure Le protocole le plus utilisé OpenFlow.
- **Interface Nord :** qui programmer les équipements de transmission, en exploitant l'abstraction du réseau fourni par le plan de contrôle.
- **Interface Est /Ouest :** sont des interfaces de communication qui permettent la communication entre les contrôleurs dans une architecture multi-contrôleurs pour synchroniser l'état du réseau.[3]

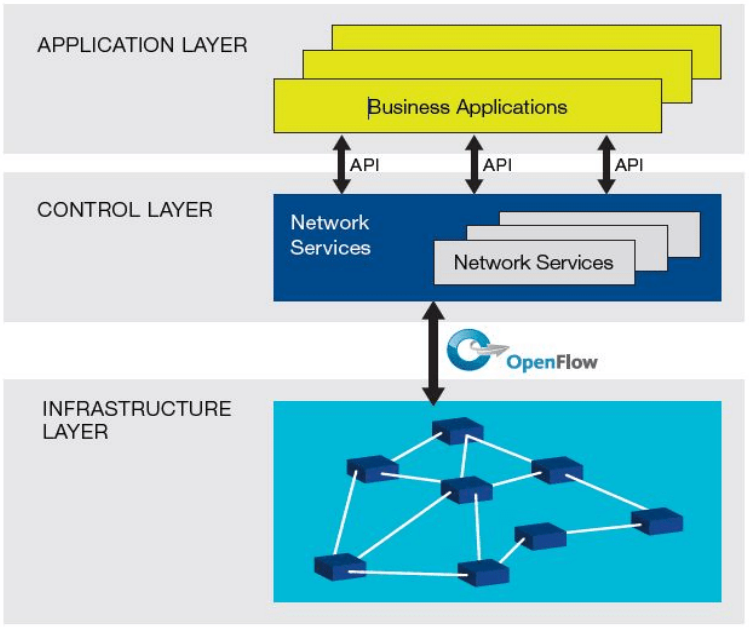

Figure 1:Architecture SDN [18]

3. Les réseaux SDN et les réseaux traditionnels

L'architecture des réseaux Internet et informatiques se compose généralement de divers périphériques réseau tels qu'un routeur, un commutateur, divers types de boîtiers intermédiaires combinés verticalement et conçus par des puces, et des ASIC (Application Spécifique Integrated Circuits) à haut débit et fonctionnalités spécifiques [11]. Pour gérer et configurer ces périphériques réseau, un ensemble de commandes de ligne prédéfinies et définies est utilisé sur

la base d'un système d'exploitation embarqué. Par conséquent, on peut dire que la gestion d'un grand nombre de périphériques réseau est très difficile et sujette à de nombreuses erreurs. Ainsi, les réseaux traditionnels souffrent d'importantes lacunes en matière de recherche, d'innovation, de fiabilité, d'évolutivité, de résilience et de gestion. Depuis la naissance d'Internet, les réseaux se sont développés et de nouvelles technologies telles que le cloud, les réseaux sociaux et la virtualisation sont apparues, et le besoin de réseaux avec une bande passante plus élevée, un meilleur accès et une plus grande agréabilité. Une dynamique plus élevée est devenue un problème critique [12]. Pour résoudre les problèmes et les limitations des réseaux traditionnels, une structure a été proposée, connue sous le nom de SDN, dans laquelle le contrôle du réseau est séparé du mécanisme de transmission et peut être directement programmé et contrôlé [11].

La figure 2 montre la différence architecturale entre Internet traditionnel et SDN. Il montre clairement comment le contrôle est géré de manière centralisée (logiquement) et le plan de données est simplifié en éléments de transmission simples. Les commutateurs de plan de données programmables peuvent être implémentés dans le matériel ou le logiciel tant qu'ils prennent en charge le protocole OpenFlow [13] pour la communication et la configuration avec le contrôleur.

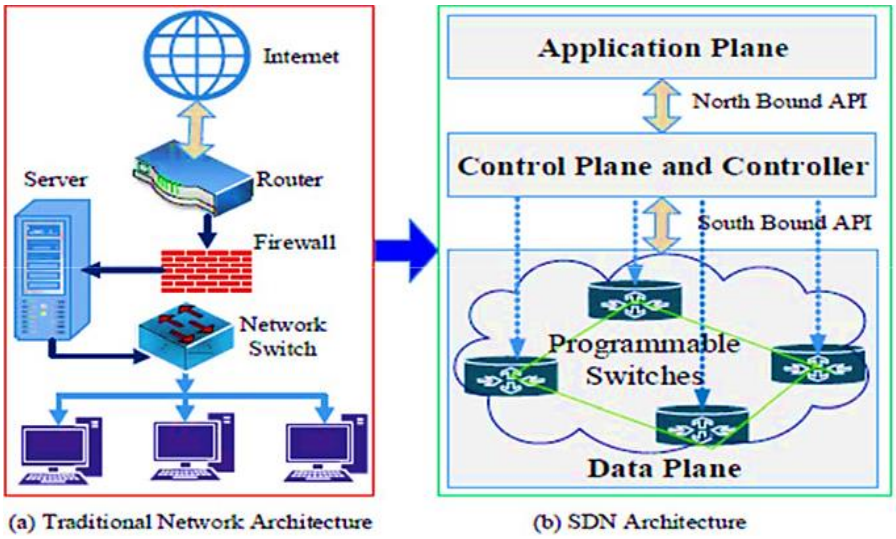

Figure 2:Comparaison entre architecture SDN et réseau traditionnel [13]

4. Protocole de communication OpenFlow

Il existe de nombreux protocoles de communication entre le plan de contrôle et le plan de données tels que ForCES, IRS et NetConf. Cependant, le protocole de communication OpenFlow reste le plus utilisé sur les équipements SDN.

OpenFlow est une norme multifournisseur définie par l'ONF qui agit comme un relais entre l'unité de contrôle et l'équipement réseau de l'avion de transport. Il fournit un protocole ouvert pour les tables de diffusion de programmes sur différents commutateurs et routeurs. Il s'agit d'un protocole standard utilisé par le contrôleur SDN pour envoyer des instructions aux

commutateurs qui programment leur niveau de données et obtenir des informations de ces commutateurs afin que le contrôleur puisse obtenir une vue d'ensemble logique du réseau physique. Cette vue est utilisée pour toutes les décisions que le niveau de contrôle doit prendre (par exemple routage, filtrage du trafic, partage de charge, traduction d'adresse). En fait, la console peut ajouter, supprimer ou modifier des flux en obtenant des statistiques sur les ports, les flux et d'autres informations à l'aide du protocole OpenFlow.[5]

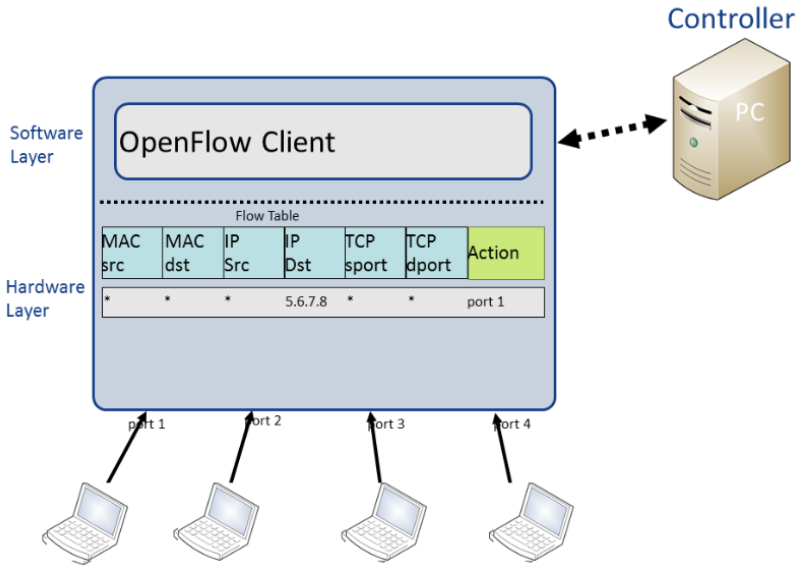

Figure 3:Le protocole OpenFlow [17]

4.1.Le commutateur OpenFlow

La plupart des commutateurs et routeurs Ethernet modernes ont des tables de flux qui sont utilisées pour exécuter les fonctions de transfert des couches 2, 3 et 4, indiquées dans les en-têtes de paquet. Bien que chaque fournisseur ait des horaires de diffusion différents, il existe un ensemble commun de fonctions pour une grande variété de commutateurs et de routeurs. Cet ensemble commun de fonctions est exploité par OpenFlow, un protocole entre la console centrale OpenFlow et le commutateur OpenFlow qui, comme indiqué, peut être utilisé pour programmer un commutateur de routage ou une logique de routage.

Un commutateur OpenFlow (OF) est un périphérique réseau qui prend en charge le protocole OpenFlow, tel qu'un commutateur, un routeur ou un point d'accès. Chaque appareil OF maintient une table de flux (autrement dit, une table de commutation ou une table de routage), construite à partir de TCAM (Triple Content Adressable Memory), qui indique Traitement appliqué à tout paquet d'un flux donné.

OpenFlow (OF) est conçu comme un moyen de tester de nouvelles méthodes de routage ou de redirection pour la construction de ces tables de flux. Pour ce faire, créez un tunnel SSH sécurisé entre la clé OpenFlow et la console. La console fonctionnera indépendamment de ce nouveau processus de routage. Lorsqu'un nouveau flux arrive, le commutateur OpenFlow envoie les

premiers paquets à la console. Le contrôleur construit Ensuite, une entrée dans la table de flux (une règle de flux ou de commutation) pour gérer le reste de cette connexion.[7]

5. Les avantages du SDN

Les avantages du SDN pour les organisations sont nombreux. Une liste des principaux avantages est présentée dans les sous-sections suivantes :

- ☑ **Programmabilité du réseau : Le** SDN peut gérer l'ensemble du réseau par programmation. Le SDN permet d'éviter plus facilement la publication de plans et de protocoles personnalisés sur chaque appareil d'un réseau individuellement. La programmabilité est véritablement possible sur le seul plan de commande, permettant de modifier le comportement d'une seule unité ou de l'ensemble du réseau. En conséquence, le contrôleur peut rapidement améliorer la fonctionnalité de conception du trafic tout en réduisant la congestion du réseau [4]
- ☑ **Prix réduit : La** majorité des produits SDN sont gratuits. Certains systèmes, tels que VMware NSX10 et la virtualisation de réseau Hyper V de Microsoft11, ne nécessitaient que les frais de licence pour que le service SDN soit payé [4].
- ☑ **Protection enrichie : Le** financement d'une machine virtuelle dans un environnement virtualisé est une tâche ardue. SDN, d'autre part, fournit une surveillance sensible sur tous les appareils [4].
- ☑ **Gestion efficace du réseau : Le** SDN permet au gestionnaire de réseau de modifier la qualité du réseau à distance. En modifiant les caractéristiques du réseau en fonction de l'arrivée de la tâche dans le réseau, un contrôle du réseau simple et fiable est possible [4].

6. Les attaques par déni de service -Ddos

6.1.Définition

L'attaque par déni de service distribué, ou DDoS (en anglais Distributed Denial of Service), vise à perturber ou paralyser totalement le fonctionnement d'un serveur informatique en le bombardant à outrance de requêtes erronées.

L'objectif peut être d'affecter les services en ligne ou le réseau de l'entreprise en saturant l'une des ressources du système : bande passante, espace de stockage, puissance de traitement de la base de données, calculs du processeur, RAM, l'ouverture d'un grand nombre de nouvelles sessions TCP dans un intervalle du temps très court, ou encore d'un nombre trop important de traitements concurrents effectués par une base de données.[8]

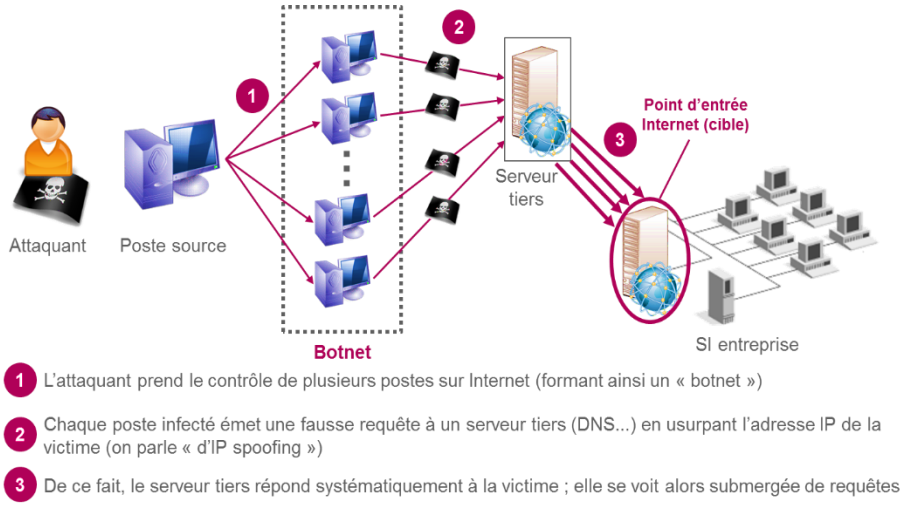

Figure 4:Exemple d'attaque DDoS [16]

6.2.Les attaques DDoS et leurs modes

6.2.1 Consommez des ressources limitées

Les machines informatiques nécessitent de nombreuses ressources pour fonctionner efficacement, telles que la mémoire, la bande passante, la puissance de traitement, etc., et si ces ressources sont utilisées par des facteurs externes, les programmes informatiques peuvent rencontrer un accès plus lent à ces ressources pour les raisons suivantes[9].

- **Connectivité réseau**

L'attaquant empêche les utilisateurs authentiques d'utiliser le service des manières suivantes dans ce type d'attaque :

L'attaque "SYN Flood" est un exemple d'une telle attaque. L'attaquant envoie le paquet de données au serveur en demandant un nouveau "connecter" avec lui. L'attaquant, quant à lui, n'envoie pas de paquet de données de « confirmation » pour vérifier la connexion.

Jusqu'à l'expiration du délai, la connexion semi-ouverte reste active. Lorsqu'un attaquant quitte un serveur cible avec un grand nombre de tels paquets, le serveur doit conserver la structure de données en mémoire jusqu'à ce que le temporisateur s'épuise. Par conséquent, le serveur est toujours occupé et incapable de répondre à d'autres demandes légitimes pour le moment.[19]

- **Utiliser les ressources des clients contre eux-mêmes**

Un attaquant peut lancer une attaque en forçant deux machines victimes à communiquer constamment entre elles. Cette forme d'agression est expliquée en détail dans. Elle connecte le service d'écho d'une machine du réseau de la victime à une autre machine du réseau de la victime à l'aide de paquets UDP falsifiés, forçant les deux machines à consommer toute leur bande passante réseau entre elles.[19]

- **Consommation de bande passante**

L'attaquant peut envoyer un grand nombre de paquets sur le réseau de la victime, absorbant toute la capacité entrante de cette dernière. Ces paquets peuvent être n'importe quoi, mais les paquets ICMP Echo sont couramment utilisés. Un attaquant peut utiliser de nombreuses machines pour transmettre une énorme quantité de paquets à la victime afin de rendre cette attaque efficace.[19]

- **Consommation d'autres ressources**

En plus de la bande passante du réseau, de nombreux attaquants peuvent essayer de drainer d'autres ressources telles que la puissance de traitement ou la mémoire.

Le serveur peut également stocker temporairement des informations dans ses structures de données, dont un attaquant peut profiter en demandant au serveur d'allouer des informations à plusieurs requêtes en même temps. Le stockage du serveur déborde d'informations sur les attaquants, rendant le système inutilisable pour d'autres requêtes.

L'attaquant peut également faire une requête en engendrant plusieurs processus CPU qui doivent attendre l'exécution. Les attentats à la bombe par e-mail sont un exemple de génération d'erreurs délibérées qui doivent être signalées.[19]

6.2.2 Les informations sur la configuration sont détruites ou modifiées :

Si certaines informations de configuration sur l'ordinateur sont modifiées ou supprimées, la machine ne pourra pas fonctionner correctement et ne pourra peut-être pas servir les autres utilisateurs qui s'y sont connectés.
L'attaquant tente d'accéder à la machine en modifiant ses paramètres et en la rendant inaccessible aux utilisateurs ordinaires. Par exemple, si un attaquant parvient à modifier les informations de la table de routage sur un routeur faiblement sécurisé, les autres utilisateurs peuvent ne pas être en mesure de se connecter à la machine liée. [19]

6.2.3 Destruction et modification des composants physiques du réseau

Dans cette forme d'attaque, l'agresseur peut physiquement nuire à l'environnement de la victime. L'attaquant peut alors désactiver ou modifier les composants du réseau.[19]

6.3 Les types des attaques DdoS

Les attaques Ddos contiennent de nombreux types d'attaques que nous divisons en catégories comme suit:

1) Attaques volumétriques

L’attaque basée sur le volume inondera la bande passante du site attaqué et elle est mesurée en bits par seconde. Ce type d'attaque comprend l'inondation UDP, l'inondation ICMP et d'autres.

2) Attaques de protocole

Elles consomment les tables d'état de connexion présentes dans les composants de l'infrastructure réseau tels que les équilibreurs de charge, les pare – feu et les serveurs d'applications. L'attaque est mesurée en paquets par seconde Exemple : SYN, ACK, TCP, attaque de fragmentation, etc.

3) Attaques de la couche d'application

Elles consomment les ressources ou le service de l'application, les rendant ainsi indisponibles pour les autres utilisateurs légitimes. Elle est mesurée en requêtes par seconde Exemple : Attaque HTTP GET / POST.

7. Les impacts des attaques de DdoS sur le réseau SDN

Due à la centralisation du contrôle dans l'architecture SDN, les attaques de DdoS peuvent avoir des graves répercussions sur les performances du réseau conduisant à des cas où tout le réseau devient incapable à répondre aux besoins des utilisateurs. Ces attaques affectent la performance des trois éléments principaux dans le réseau SDN : le contrôleur SDN, la liaison entre le contrôleur et les commutateurs et le plan de transmission (les commutateurs et les liens du réseau).

7.1. Impact de l'attaque Ddos sur le plan de contrôle

L'attaquant va envoyer une grande quantité de flux à travers le réseau SDN. Lorsque les commutateurs dans la couche infrastructure reçoivent ces nouveaux flux entrants, ils envoient des demandes au contrôleur pour obtenir des règles de commutation afin de les envoyer à la destination demandée. Par conséquent, le contrôleur SDN sera surchargé à cause de la quantité énorme de demandes provenant du plan de donnée du réseau, menant à des cas où le contrôleur devient totalement paralysé et ne puisse pas prendre aucune décision du routage.

7.2. Impact de l'attaque Ddos sur le plan de données

Généralement, les commutateurs doivent enregistrer les règles de commutation dans leur stable de commutation et les utiliser jusqu'à l'expiration des temporisateurs, l'idole time out et le hard timeout. Dans une situation d'attaque de Ddos, où l'attaquant inonde le commutateur avec une quantité du flux importante, tout le trafic de données reçu par les commutateurs, se Traduit en règles de commutation fournit par le contrôleur, afin de les acheminer vers la destination. En effet, la mémoire TCAM du commutateur sera remplie par ces règles envoyées de contrôleur jusqu'à sa saturation. Lorsque cela se produit, les commutateurs sont forcés d'ajouter et de supprimer continuellement les règles de flux et d'envoyer plus des demandes vers le contrôleur;

cela engendre la congestion du plan de transmission ainsi qu'un retard dans le temps de

transmission de données

7.3.Impact de l'attaque Ddos sur la liaison contrôleur-commutateur

Due à la communication agressive entre le contrôleur SDN et les commutateurs demandant des décisions de routage, la liaison entre le contrôleur et le commutateur (appelé aussi la bande passante du plan de contrôle) sera exténuée et congestionnée, cela cause la perte de plusieurs messages « paquet-in » ainsi que le retard dans le temps de réponse des messages échangés entre le contrôleur et les commutateurs.

8. Les méthodes de Détection les attaque Ddos dans réseau SDN

Ces dernières années, la détection et l'atténuation des attaques DDoS dans le SDN ont été une grande préoccupation pour les chercheurs.

Cette section présente l'analyse systématique des travaux récents réalisés dans la détection et l'atténuation des attaques DDoS dans le contexte SDN, toutes ces solutions sont divisées en six catégories différentes selon le type de métrique de détection et de mécanisme de détection utilisé, à savoir :

8.1.Solutions basées sur les entrées de table

Le principe de cette solution repose sur contrôler l'installation des règles de flux et des politiques de remplacement des entrées de table, dans ce cas il existe plusieurs propositions par exemple le PacketChecker, l'idée centrale derrière PacketChecker est de donner aux contrôles la possibilité de distinguer les paquets malveillants par crée une table de mappage pour l'adresse MAC et le port du commutateur [14].

8.2.Solutions basées sur la planification

Prioriser et planifier les demandes de flux à traiter par le contrôleur est l'une des stratégies proposées pour vaincre les attaques DoS/DDoS sur le SDN. Les solutions examinées ont implémenté un module de planificateur dans la couche de contrôle pour gérer le traitement des demandes de flux. Yan et al. [34] ont présenté une méthode d'ordonnancement multifiles d'attente en fonction de la stratégie d'allocation des tranches de temps. Leur solution se compose de deux modules : les modules de détection DDoS et d'algorithme MultiSlot. Ils ont dû modifier le mécanisme de traitement de flux du contrôleur en mettant en file d'attente les demandes de flux de chaque commutateur dans sa file d'attente logique correspondante. Le module de détection DDoS détecte l'attaque DDoS sur le SDN et l'étendue d'un commutateur attaqué. [15]

8.3.Solutions basées sur l'architecture

L'ensemble des solutions pour vaincre les attaques DoS/DDoS liées aux hiérarchies et aux rôles des composants SDN sont appelés solutions architecturales. Cette catégorie comprend le moins

de travaux de recherche évalués. [15] ont proposé une solution pour considérer le pool de contrôleurs de secours à utiliser au cas où un contrôleur de travail aurait été attaqué. Leur solution est conçue pour gérer les attaques Ddos contre le SDN des contrôleurs distribués. Chaque contrôleur arrière surveille au moins un contrôleur en ligne principal. La communication entre le contrôleur arrière et les contrôleurs actifs dépend de protocoles comprenant un algorithme de mappage, des messages de pulsation, des messages de synchronisation, un processus de reprise et un mode de protection.

8.4.Solutions basées sur les statistiques de flux

Ont développé des mécanismes de neutralisation DoS/DDoS basés sur des modèles statistiques qui dépendent de la fréquence des différentes caractéristiques capturées, de la matrice de corrélation, de l'entropie et du test de qualité de l'ajustement du chi carré. Boite et al. [36] ont proposé une solution pour détecter et atténuer les attaques DDoS en utilisant des capacités de traitement dans le commutateur. Le trafic est surveillé pour capturer les fonctionnalités pertinentes (par exemple, IP src, IP dst, port src, port dst). Plus tard, ces caractéristiques sont utilisées comme entrées dans un algorithme de détection d'anomalies basé sur l'entropie. D'autres solutions basées sur l'entropie ont classé les paquets entrants en fonction de l'adresse IP de destination et de la taille de la fenêtre, ont exploité l'IP de destination collectée pour chaque fenêtre d'observation pour calculer la probabilité d'occurrence de l'IP de destination pour chaque paquet dans la fenêtre d'observation, et ont analysé la distribution de la fréquence d'occurrence des IP source et destination. [15]

8.5.Solutions basées sur l'intelligence artificiel

Les méthodes ML se concentrent sur la création et la formation automatiques d'un modèle à utiliser ultérieurement pour détecter et atténuer les attaques DoS/DDoS dans les SDN. Ce modèle est entraîné à l'aide d'un jeu de données d'entraînement. L'ensemble de données comprend une collection d'instances de données qui sont définies à l'aide d'un ensemble d'entités et des étiquettes associées. Plusieurs méthodes ML telles que les machines à vecteurs de support (SVM), ANN, basées sur les graphes, K-NearestNeighbor (KNN) et les méthodes de clustering ont été utilisées afin de détecter et/ou d'atténuer les attaques DoS/DDoS dans les SDN. [15]

8.6.Solutions hybrides

Les solutions hybrides combinent au moins deux des types de solutions susmentionnés. Les mécanismes basés sur les statistiques de flux sont largement utilisés dans les solutions hybrides. FlowSec et Blackbox sont deux méthodes de défense DdoS qui s'appuient sur les statistiques de flux pour détecter ou atténuer les attaques DdoS contre le SDN. . .Une solution hybride combinant les statistiques de flux et ML est proposée par Jankowski et Amanowicz [14]. Leur méthode consiste en quatre modules ; bundle de flux, intégrateur, générateur de fonctionnalités

et classificateur basé sur ML. Le module Flow Bundle capture et extrait les statistiques de flux du trafic entrant. L'intégrateur analyse et traite les statistiques de flux extraites afin de générer des fonctionnalités supplémentaires. Le bundle de flux et les modules Integrator forment le module générateur de fonctionnalités. La sortie du module générateur de fonctionnalités est utilisée comme entrée du classificateur SOM qui est chargé de détecter le flux malveillant. [15]

9. Conclusion

Dans ce chapitre, nous avons présenté Une définition du réseau SDN a été fournie, son architecture, ses avantages. Nous avons également fait une brève comparaison entre les réseaux traditionnels et les réseaux SDN. Après il parle à les attaque ddos et leur impact dans le SDN et nous avons cite les différentes solutions qui propose, dans le chapitre suivant on va parler à la solution qui choisir et leur travaille connexe.

Chapitre 2

Etat de l'art
« L'intelligence artificiel pour la détection des attaques DDoS »

1. Introduction

Plusieurs approches intelligentes basée sur l'apprentissage automatique ont été étudié récemment pour la détection des attaques. L'objectif d'un systeme basé sur ML est d'analyser les données collectées pour détecter des modèles qui pourraient refléter d'éventuelles attaques sur la machine hôte cible et/ou le réseau. L'ensemble de données d'apprentissage peut être étiqueté ou non étiqueté. En conséquence, l'algorithme d'apprentissage exploité peut être respectivement supervisé ou non supervisé. Bien que l'apprentissage supervisé permette généralement d'obtenir de meilleures performances, il nécessite un ensemble de données étiqueté contenant un ensemble d'exemples représentatifs de données bénignes et malveillantes. Cela peut nécessiter un effort humain considérable pour collecter des exemples malveillants et étiqueter l'ensemble de données.

Dans ce chapitre, nous avons étudié et analysé divers travaux de détection des attaques DDoS basés sur l'apprentissage automatique.

1. Les enjeux de l'IA dans la cybercuirité

Avec la disponibilité de grandes quantités de données provenant des réseaux, des systèmes d'exploitation et des systèmes d'information, des méthodes et des techniques de l'intelligence artificielle (IA) telles que l'apprentissage automatique et l'apprentissage profond, l'exploration de données, les statistiques et d'autres capacités interdisciplinaires ont été utilisées pour relever les défis de la cybersécurité. [15].

L'apprentissage automatique, ainsi que l'apprentissage en profondeur, pourraient être utilisés pour le réseau SDN basé sur la détection d'anomalies.

Ces méthodes de classification/détection/prévention peuvent être utilisées pour découvrir des modèles et des comportements communs dans diverses cyberattaques, permettant une cyber réponse en temps réel. Ils ont la capacité de détecter les attaques dès qu'elles se produisent, ainsi que la capacité de prédire les futures attaques potentielles [35].

Les méthodes basées sur un apprentissage en profondeur peuvent être utiles pour surmonter les défis liés au développement d'un IDS efficace [17, 41].

D'autre part, la collecte de données et le trafic réseau ont entraîné un problème de mégadonnées. Les spécialistes de la sécurité s'efforcent constamment d'obtenir de meilleurs résultats avec le taux de détection le plus élevé et le taux de fausses alarmes le plus bas. En conséquence, les approches d'apprentissage en profondeur qui s'adaptent bien à de grandes quantités de données deviennent de plus en plus populaires. Ceux-ci ont été introduits pour détecter les anomalies du réseau afin de faire la distinction entre un comportement normal et anormal afin de détecter les activités malveillantes ou suspectées [37].

2. L'intelligence artificielle

L'intelligence artificielle (IA) vise à imiter le fonctionnement du cerveau humain, ou du moins sa logique lorsqu'il s'agit de prendre des décisions. L'IA implique la mise en œuvre d'un certain nombre de techniques pour permettre aux machines d'imiter une certaine forme d'intelligence réelle.

L'IA englobe différents sous-domaines tels que les règles métier, le Machine Learning (ML), le Deep Learning (DL), etc. (voir la figure 2.1).

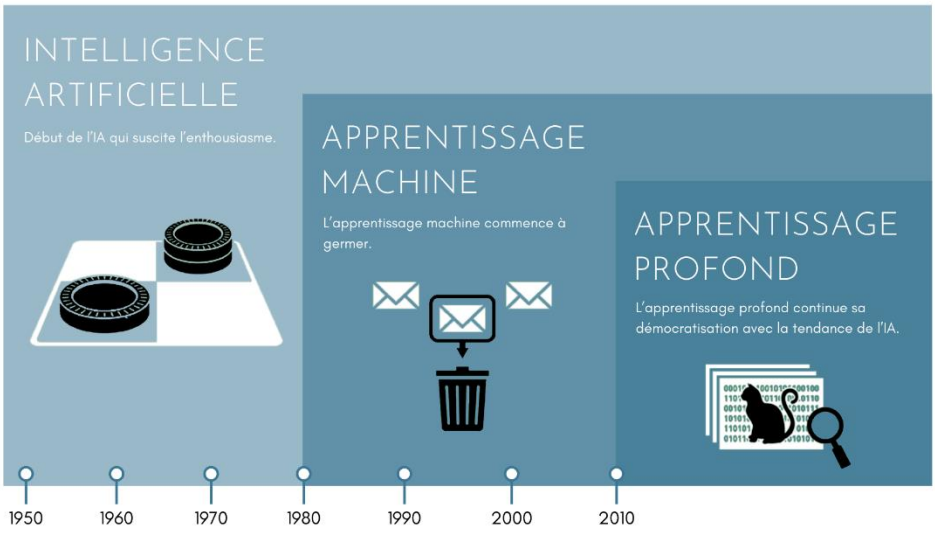

Figure 2. 1:Relation entre IA, ML et DL[34]

3. Apprentissage automatique

« Le Machine Learning est le domaine de l'IA qui permet à une machine d'apprendre. C'est-à-dire améliorer progressivement les performances d'une tâche spécifique basée sur des données sans être explicitement planifiée. » [14].

L'apprentissage automatique s'effectue via des réseaux de neurones conçus pour imiter les capacités de prise de décision humaines. Nous devons l'appliquer pour résoudre tout problème nécessitant une réflexion, qu'il soit humain ou artificiel.

3.1.Les types de l'apprentissage automatique

L'apprentissage automatique est généralement divisé en deux classes principales, à savoir "l'apprentissage supervisé" et "l'apprentissage non supervisé".

1) Apprentissage supervisé

La forme la plus courante d'apprentissage automatique est l'apprentissage supervisé. L'apprentissage supervisé est une méthode de transformation d'un ensemble de données en un

autre, le programme est entraîné sur un ensemble prédéfini d'exemples d'entraînement, ce qui facilite ensuite sa capacité à parvenir à une conclusion précise lorsque de nouvelles données sont fournies [15] [16]. Les algorithmes de classification supervisée de ML sont : Forest Random, Decision Trees, Logistic Regression, et le plus connu est SVM Support Vector Machines.

2) Apprentissage non supervisé

L'apprentissage non supervisé, également connu sous le nom d'apprentissage à partir d'observations, partage une propriété commune avec l'apprentissage supervisé : il transforme un ensemble de données en un autre. Mais l'ensemble de données dans lequel il se transforme n'est pas connu ou compris auparavant. Contrairement à l'apprentissage supervisé, il ne se nourrira quant à lui que d'exemples, et créera lui-même les classes qu'il jugera les plus judicieuses (clustering) ou des règles d'association (algorithmes Apriori). L'algorithme K-mean (Kmeans) permet de comprendre facilement le concept de classification non supervisée [17].

4. Apprentissage profond

" L'apprentissage profond est une classe de techniques d'apprentissage automatique dans lequel plusieurs couches de traitement informatique itératif dans des architectures hiérarchiques et des algorithmes supervisés sont exploitées pour des algorithmes d'apprentissage non supervisés pour des tâches d'analyse et de classification" [18]

L'apprentissage profond consiste essentiellement à calculer les caractéristiques hiérarchiques des paramètres des réseaux de neurones artificiels pour les représentations vectorielles des données d'observation ou d'entrée. La famille des méthodes d'apprentissage en profondeur s'enrichit de plus en plus, englobant celles des réseaux de neurones, des modèles probabilistes hiérarchiques, ainsi que de nombreux algorithmes d'apprentissage de fonctionnalités supervisés et non supervisés.

5.Apprentissage profond VS Apprentissage automatique

Il existe deux caractéristiques principales qui distinguent le deep learning du machine learning qui sont :

	Extraction des caractéristiques	Fonctionnalité
Apprentissage automatique	La plupart des fonctionnalités de l'application sont requises par un expert, puis codées manuellement par domaine et type de données.	Les résultats prévalent à mesure que la quantité de données augmente.
Apprentissage profond	Les algorithmes essaient de connaître la fonctionnalité de haut niveau des données	Lorsque les données sont petites, les performances des algorithmes d'apprentissage en profondeur donnent de mauvais résultats car il faut une grande quantité de données pour bien les comprendre.

Tableau 2. 1: Apprentissage profond VS Apprentissage automatique

5.Apprentissage automatique pour la détection des attaque DDoS

Des recherches récentes ont démontré l'efficacité des applications d'apprentissage automatique dans la détection des attaques DDoS.

Pour les systèmes traditionnels tel que :

1) ***Les systèmes basés sur la signature :*** les intrusions sont détectées en comparant les comportements surveillés avec des motifs d'intrusion prédéfinis
2) ***Les systèmes basés sur les anomalies :*** se concentrent sur la connaissance du comportement normal afin d'identifier toute déviation et toutes activités suspectes.

Les méthodes de d'apprentissage automatique sont applicables pour les 2 types de détection grâce à ces capacités qui permettent d'extraire des niveaux plus élevés de relations non linéaires entre les données, afin d'identifier toute déviation d'une activité bénigne.

Dans cette section nous présentons un état de l'Art sur l'ensembles de dataset disponible pour la détection/classification des attaques DDos ainsi qu'une partie des travaux connexes on se concentrent fortement sur les attaques des DDos dans les domaines SDN [21]

6.Les datasets

La validation de tous les algorithmes d'apprentissage automatique suggérés dépend des ensembles de données utilisés dans les études publiées pour l'application de l'apprentissage automatique à la cybersécurité. En raison de problèmes de confidentialité, certains de ces ensembles de données ne sont pas disponibles gratuitement. Voici des exemples d'ensembles de données dédiés à la détection des différents attaques :

6.1KDD Cup'99

Les ensembles de données de découverte d'interruption KDD Cup'99 qui dépendent absolument de l'ensemble de données DARPA '98 donnent un ensemble de données nommé à l'analyste s'exécutant à l'intérieur de la zone d'identification d'interruption et parlent à l'ensemble de données nommé librement accessible. Le jeu de données KDD'99 est fait de l'utilisation d'une reproduction d'un système militaire. Enfin, il existe un renifleur qui enregistre toutes les informations d'activité du système transmises en utilisant l'arrangement Tcp dump. L'ensemble de données de préparation KDD contient environ 4 900 000 vecteurs d'association uniques, qui intègrent tous 41 qualités et sont classés comme agression ou typique, avec correctement un type d'agression indiqué. Les agressions imitées sont réparties dans les quatre classes suivantes : agressions par déni de service (Dos), sonde, distant vers local (r2l) et utilisateur vers racine (u2r).[41]

6.2NSL-KDD

Pour résoudre les problèmes de l'ensemble de données de la KDD Cup, ils ont proposé un nouvel ensemble de données, à savoir NSL-KDD, qui consiste en des enregistrements sélectionnés de l'ensemble de données complet de la KDD Cup'99. Voici les avantages de l'ensemble de données NSL-KDD par rapport à l'ensemble de données KDD Cup'99 :

- ✓ Il n'inclut pas les enregistrements non pertinents dans la rame, de sorte que les classificateurs ne seront pas partisans d'enregistrements plus répétés a partir de chaque enregistrement de difficulté, le nombre d'enregistrements choisis est inversement proportionnel au pourcentage d'enregistrements dans l'ensemble de données KDD. Ainsi, il en résulte que le pourcentage de classification des différentes techniques ML (Machine Learning) diffère dans une large gamme. Cela rend l'évaluation complète et structurée des approches de ML .
- ✓ Dans l'ensemble de données d'apprentissage et de test, le nombre d'enregistrements est logique, ce qui facilite la réalisation des expériences sur l'ensemble de données sans avoir à choisir de petits segments aléatoires. Par conséquent, les résultats d'évaluation des différents travaux seront stables.[42]

6.3MAWI

L'ensemble de données MAWI contient des traces de trafic quotidiennes de 15 minutes avec des en-têtes de transport couvrant la dernière décennie. Bien que l'ensemble de données soit disponible pour la communauté au sens large depuis un certain temps, la courte durée de chaque trace l'a prêté à une étude plus approfondie dans les zones. La présence de traces de flux complètes est moins importante, comme les anomalies Internet ou lorsque la caractérisation du trafic est basée sur les paquets, reposant uniquement sur l'inspection de l'en-tête IP et des numéros de port.[43]

6.4ISCX

L'ensemble de données ISCX a été généré à l'aide de paramètres réseau réels par capturer des paquets en temps réel pendant une période de sept jours.

Les données contiennent environ 85 Go de données de trafic réseau ainsi que des profils qui décrivent le flux des données et l'at-coups de fil qui se sont produits au cours de cette semaine. Le jeu de données ISCX est caractérisé par le fait qu'il a été recueilli en temps réel et que les attaques n'ont pas été simulées, mais au lieu de cela les attaques ont été lancées pendant le processus de capture et d'enregistrement des paquets circulant au sein du réseau. Nous choisissons le jeu de données ISCX dans notre analyse pour les raisons suivantes :

- ✓ Il représente un jeu de données réaliste sans aucune trace post-capture insertions sur les données, offrant ainsi au chercheur une mesure plus réaliste des effets de certaines attaques sur réseau.
- ✓ L'ensemble de données a des profils qui décrivent les attaques avec quelques informations supplémentaires qui décrivent l'heure et approche utilisée pour lancer l'attaque.[44]

6.5.CICIDS2017

Il s'agit d'un ensemble de données public accessible gratuitement à (http://www.unb.ca/cic/datasets/IDS2017.html). Il se compose de données réelles qui ont été collectées sur la base du comportement d'un réseau de 25 utilisateurs basé sur les protocoles FTP, HTTPS, HTTP, e-mail et SSH. CICIDS2017 inclut le trafic d'attaques malveillantes bénignes et différentes telles que l'infiltration, l'attaque Web, le botnet et le saignement cardiaque..., etc. Nous avons sélectionné le fichier qui contient une attaque de botnet (Friday-WorkingHours-Morning.pcap_ISCX) et l'avons testé dans notre proposition maquette. Ce fichier comprend 191033 trafic bénin et 1966 trafic Botnet.[45]

6.6.CIC-DDoS-2019

Il s'agit de véritables données de flux réseau avec plusieurs formes d'attaques DDOS les plus récentes et les plus courantes.) regroupe tous les paquets dans une fenêtre temporelle qui partagent des attributs spécifiques mais ne transportent pas de charge utile. Il existe deux types de données dans l'ensemble de données : les données PCAP brutes et les données CSV. Les auteurs ont extrait plus de 80 caractéristiques des fichiers PCAP à l'aide de l'analyseur de trafic CICFlowMeter-V3, et les résultats ont été enregistrés dans des fichiers CSV formatés et étiquetés par l'Université du Nouveau-Brunswick.[46]

Tableau 1:DDoS Datasets

Data-set public	Type	Étiqueté	Nombre de classe	Année	[Réf]
KDD99	Trafic Du réseau	Oui	5 classes (normal, DoS, R2L (accès non autorisé depuis une machine distante), U2R (accès non autorisé à Root), Probe.)	1999	[41]
NSL-KDD	Trafic Du réseau	Oui	5classe (,normal,DoS, R2L (accès non autorisé depuis une machine distante), U2R (accès non autorisé à Root), Probe.)	2009	[42]
MAWI	Trafic Du réseau	Oui	Multi classe	2011	[43]
ISCX dataset	Trafic Du réseau	Oui	Multi classe	2012	[44]
CIC DoS 2017	Trafic du réseau	Oui	7 classes (dos,portscan,Bot,brute-force,webattack,infiltration)	2017	[45]
CIC DoS 2019	Trafic Du réseau	Oui	13classe (Syn,DrDoS_SNMP, DrDoS_LDAP,DrDoS_SSDP, DrDoS_NetBIOS, DrDoS_MSSQL ,TFTP,DrDoS_UDP,DrDoS_DNS,UDP-lag,DrDoS_NTP,BENIGN,WebDDoS)	2019	[46]

7.Les travails connexes

Les auteurs de [22] ont développé un modèle d'apprentissage automatique pour prédire les attaques DDoS et botnet en utilisant un algorithme d'apprentissage automatique avec régression linéaire multiple. Ils ont utilisé l'ensemble de données de référence CICIDS 2017 le plus largement utilisé avec des charges utiles de paquets entiers au format pcap, qui est largement utilisé dans les flux de réseau étiquetés. Ils ont également démontré que leur modèle d'apprentissage automatique pouvait détecter les attaques DDoS en utilisant la technique d'analyse de régression.

Dans [23.] les auteurs ont montré que Recurrent Neural Network surpasse Random Forest en termes de généralisation. L'efficacité de l'apprentissage en profondeur, où le taux d'erreur a été réduit de 7,517 % à 2,103 % à l'aide d'un ensemble de données ISCX2012, par rapport aux méthodes traditionnelles d'apprentissage automatique. Leur expérience utilise l'ensemble de données ISCX2012, mis à disposition par l'Université du Nouveau-Brunswick en 2012.

Article	Annotation	Algorithm	Mesures	Extraction features	Dataset	Domaine	Commentaires
[23] (2017)	Détection	LSTM RF	Acc=97.6% taux d'erreur = (7,517 % à 2,103 %)		ISCX2012	SDN Controler	**LSTM** réduire le taux d'erreur de 7,517 % à 2,103 % par rapport à la méthode d'apprentissage automatique conventionnelle
[24] (2019)	Détection	KNN RF	Acc= 99.97%	SBFE[1] SFFS	Bot-Iot Dataset	Iot	L'application de la technique de sélection de caractéristiques hybrides peut être considérée comme le compromis entre la simplicité de la sélection de caractéristiques basée sur des modèles de filtre et les techniques d'encapsulation plus exigeantes en termes de calcul
[22] (2020)	Détection	RL	Acc=73.79%	gain-based	CICIDS 2017	Cloud	-
[25] (2020)	Détection	RF SVM, KNN	Acc = 98.97%	Optimisation bayésienne	KDD-Cup99, Digiturk, Labris	Systèmes de détection d'intrusion (IDS)	les hyperparamètres optimaux pour les méthodes de classification trouvés à l'aide de la méthode d'optimisation bayésienne.
[26] (2020)	Détection	SVM, LSTM.	Acc= 90.59٪	MinMax[2] SVD[3]	UNSW-NB15 NSL-KDD	IDS	L'ensemble de formation et de test de ces ensembles de données a été traité à l'aide de la méthode de transformation de caractéristiques arbitraires minmax.
[27] (2020)	Détection	SVM	Acc= 99.1%	AE[4]	CICIDS	IDS	-

[28] (2020)	Classification	KNN, SVM	Acc= 99%	feature scalling[5]	CICDDoS2019	Cloud	-
[29] (2022)	Classification	SVM RF	Acc=99, 97 %	RFE	NSL-KDD	SDN controller	-Les résultats montrent que le classificateur de forêt aléatoire (RF) entraîne le modèle le plus précis avec une précision de 99,97 % en utilisant la méthode (RFE).

Figure 5:Travaux connexes

Les auteurs [24] ont proposé le concept d'utiliser des modèles de sélection de caractéristiques hybrides pour réduire la taille de la caractéristique afin d'obtenir des résultats plus précis. L'ensemble de données contenait 115 entités. Pour limiter le nombre de fonctionnalités, des modèles hybrides ont été utilisés pour choisir la fonctionnalité potentielle. Ces caractéristiques ont ensuite été chargées dans un modèle K Nearest Neighbor (KNN) et Random Forest, qui avaient tous deux une précision élevée de 99%.

Dans le travail de [25.]les auteurs démontrent qu'en utilisant des algorithmes d'apprentissage automatique traditionnels, des ensembles et des méthodes d'extraction de caractéristiques profondes, l'optimisation bayésienne est plus rapide que l'optimisation de recherche de grille traditionnelle. Cependant, elle nécessite plus de ressources informatiques que l'étape train-test. La bibliothèque scikit de Python est utilisée pour implémenter les méthodes de classification des expériences. Les données des paquets de trafic réseau ont été capturées et converties en enregistrements de connexion. Ils ont utilisé trois types de fonctionnalités : les fonctionnalités de base, les fonctionnalités basées sur le temps et les fonctionnalités basées sur la connexion. Les fonctionnalités de base sont des caractéristiques qui peuvent être facilement dérivées des en-têtes de paquet en comptant les propriétés de paquet spécifiques pour la connexion. Avant d'évaluer les performances d'un modèle, l'optimisation des hyperparamètres permet aux

[1] **SBFE** : la sélection séquentielle des fonctionnalités avant (SFFS) et l'élimination séquentielle des fonctionnalités en arrière (SBFE). Les deux algorithmes tentent d'affiner un ensemble actuel de fonctionnalités en ajoutant/supprimant des fonctionnalités de manière itérative, respectivement, en fonction des performances d'un classificateur

[2] **min-max**: méthode pour étendre les ensembles de données à une taille similaire

[3] **SVD** : décomposition en valeurs singulières(SVD) : est une factorisation d'une matrice réelle ou complexe. Il généralise la décomposition propre d'une matrice normale carrée avec une base propre orthonormée.

[4] **AE** : Aggregate Expenditure : est un algorithme d'apprentissage non supervisé composé d'une ou plusieurs couches cachées. Il est utilisé dans l'apprentissage des caractéristiques et la réduction dimensionnelle pour obtenir une généralisation non linéaire

[5] **feature scalling** :est une méthode utilisée pour normaliser la plage de variables indépendantes ou les caractéristiques des données

chercheurs d'affiner ses hyperparamètres. Le processus d'optimisation bayésien est utilisé pour créer des échantillons de valeurs d'hyperparamètres afin de localiser les optimaux.

Dans Un autre travaille [26] les auteurs ont comparé les résultats d'algorithmes d'apprentissage automatique traditionnels tels que Naive Bayes (NB), Decision Tree (DT) et Support Vector Machine (SVM). Les algorithmes d'apprentissage en profondeur suggérés LSTM et Singular Value Decomposition (SVD) démontrent une amélioration significative. Le prétraitement des données est effectué sur les données du réseau, ce qui comprend des méthodes de normalisation des données et de conversion des caractéristiques. La méthode de normalisation nécessite de limiter les valeurs des caractéristiques du réseau à une plage étroite de valeurs et la méthode de conversion des caractéristiques nécessite de transformer les valeurs des caractéristiques non numériques en valeurs numériques.

Dans [27] l'auteur a utilisé des fonctionnalités de réduction dimensionnelle dans le modèle d'auto-encodeur (AE) et le classificateur Support Vector Machine (SVM) pour classer les données encodées comme DDoS ou normales. AE-SVM distingue avec succès le trafic d'attaque normal et DDoS. La méthode min-max a été utilisée pour normaliser leurs données entre 0 et 1, et les vecteurs d'entraînement pour le modèle AE ont été créés. Avec le processus d'encodage, le modèle entraîné a fourni l'apprentissage et la réduction des fonctionnalités. Les résultats ont montré que la méthode AE-SVM fonctionnait bien en termes de faibles taux de détection de faux positifs DDoS et de détection rapide des anomalies

Dans [28],les auteurs appliquent six algorithmes de classification (l'arbre de décision, l'algorithme des K plus proches voisins, les machines à vecteurs de support,le classificateur de forêt aléatoire, régression Logistique, gaussienne NB), avec la méthode Analyse en Composantes principales (PCA) sur ensembles de dataset CICDDoS 2019 pour détecter les diffèrent attaques dans un enivrements cloud computing. Les caractéristiques donne les meilleurs résultat avec randomForest qui avait une précision de 97 %.

Finalement dans le travail de [29], certaines méthodes de sélection de fonctionnalités importantes pour l'apprentissage automatique sur la détection DDoS sont évaluées.

La sélection des fonctionnalités optimales reflète la précision de la classification des techniques d'apprentissage automatique et les performances du contrôleur SDN.

Une analyse comparative de la sélection de fonctionnalités et des classificateurs d'apprentissage automatique est également dérivée pour détecter les attaques SDN.

Les résultats expérimentaux montrent que le classificateur de forêt aléatoire (RF) entraîne le modèle le plus précis avec une précision de 99,97 % en utilisant un sous-ensemble de caractéristiques par la méthode d'élimination de caractéristiques récursives (RFE).

8.Synthèse

Tous les travaux mentionnés ci-dessus sont du bon travail visant à détecter/classifier les attaques DDoS dans différents environnements (CLOUD, SDN, etc.) ou identification dans un réseaux générale.

Les résultats de ces travaux sont stupéfiants, et leurs valeurs varient en fonction du type d'algorithmes utilisés et du jeu de données sélectionné.

Le travail 29 représente un travail spécifique qui reprend tous les points de notre thèse.

Cet article présente une analyse comparative de différentes classes d'apprentissage automatique basée sur le sous-ensemble optimal de fonctionnalités pour une détection précoce et précise des attaques DDoS via SDN.

Ce travail a également démontré que la combinaison des fabricants d'apprentissage automatique et des fonctionnalités SDN protège le contrôleur SDN des attaques DDoS.

Ce dernier représente notre domaine de travail, et donc en tant que synthèse, nous avons choisi de travailler sur l'ensemble de données utilisé dans cet article, d'appliquer une variété d'algorithmes d'apprentissage automatique afin de détecter les attaques DdoS dans l'environnement SDN, et enfin de comparer les résultats avec les résultats de ce travail.

Conclusion

Dans ce chapitre, nous avons présenté en premier point le rôle d'apprentissage automatique dans la détection des attaques DDoS, en plus nous avons cité l'ensemble de dataset disponibles pour les attaques DDoS. Pour rendant les systèmes fiables, nous avons établir un état de l'art sur les méthodes intelligentes détection de DDoS, finalement une synthèse sur les travaux base sur l'approche d'apprentissage automatique pour la détection des attaques DDos dans un environnement SDN est présenter à la fin de ce chapitre.

Le chapitre qui suit, on va présenter notre contribution ainsi que notre modèles les algorithmes utilisés dans notre architecture.

Chapitre 3

Contribution

1. Introduction

L'architecture SDN est divisée en trois couches. Toutes les couches peuvent être ciblées avec une sécurité différente des menaces. Cependant, la console et la bande passante de contrôle sont les points cibles les plus sensibles pour les attaques DDoS.

Dans ce chapitre, nous présenterons les emplacements des attaques, puis expliquerons l'architecture de l'approche proposée, et nous introduirons une approche intelligent .

2. Les emplacements des attaques DDoS sur SDN

Les emplacements des attaques ddos sont divisés en SDN Comme suit :

2.1.Commutateur SDN

les commutateurs SDN sont utilisés pour le transfert de données et le traitement des nouveaux paquets. Ils ont une taille très limitée de tables de flux. C'est une grande préoccupation pour la sécurité.

2.2.Liaisons entre switchs SDN

Les paquets de flux sont transférés pour être acheminés depuis un passer à un autre commutateur. La plupart des paquets transférés ne sont pas codés et peuvent contenir informations sensibles. Ces paquets peuvent être facilement interceptés par les attaquants, en particulier lorsque les liaisons entre commutateurs sont sans fil. [38]

2.3.Contrôleur SDN

Lorsque le contrôleur ("cerveau du réseau SDN") effectue des activités cruciales pour SDN, toute anomalie dans celui-ci peut paralyser l'ensemble du réseau. La fonctionnalité complète d'un réseau dépend du contrôleur. De ce fait, c'est la cible la plus attrayante pour les assaillants. Il peut souffrir d'un point de défaillance unique si le réseau n'a qu'un seul contrôleur. [38]

2.4.Le lien entre le contrôleur et le switch

Si le paquet accède à un switch et un switch la clé est incapable de gérer, donc le paquet est transmis à la console pour plus d'informations en cours de traitement. Par conséquent, de nouvelles règles de transfert de paquets ont été ajoutées au flux clés. Les règles du paquet sont envoyées via l'interface sud lors de la commutation. Ces paquets de données peuvent être désactivés par un attaquant vers le sud interface, ce qui entraîne l'ajout de certaines règles nuisibles ou la modification de l'existant Grammaire. Placer ces mauvaises règles dans la table de commutation conduit à un mauvais routage des paquets. [38]

2.5.Liaisons entre deux contrôleurs

Dans le scénario basé sur plusieurs contrôleurs, la communication est partagé entre les contrôleurs via des API orientées est-ouest. Les paquets entre les contrôleurs peuvent être gênés par un attaquant pour obtenir des informations essentielles pour compromettre les contrôleurs. Ainsi, la communication entre les contrôleurs doit être sécurisée et authentique. Les contrôleurs distribués peuvent également souffrir de pannes en cascade en raison de demandes inondées. [38]

2.6.Applications

Les applications telles que le routage, la surveillance du trafic et la virtualisation sont mis en œuvre sur la couche de contrôle SDN. La plupart des demandes sont établies par des tiers. [38]

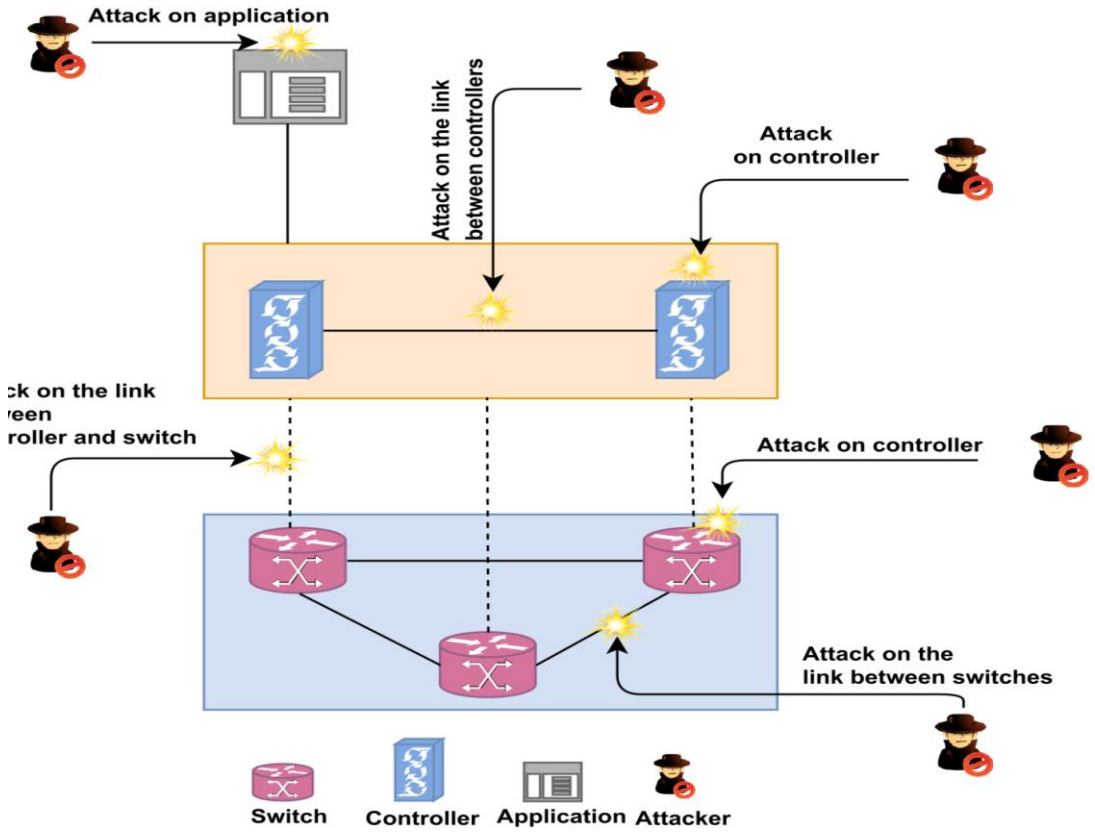

Figure 6:localisations d'attaque [38]

3. Approche proposée

3.1.L'architecture de modèle

Comme notre solution propos de la détection des attaques DDOS basé sur l'intelligence artificiel, on a proposé deux model différent :

3.1.1L'architecture de modèle d'apprentissage automatique :

Nous devons tout d'abord passer par les étapes de machine Learning Nous avons implémenté type d'approche d'apprentissage automatique, le schéma conceptuel de notre méthode d'implémentation des méthodes machine Learning proposes comme la figure 12.1 au début on télécharge le jeu de donnée qui contient jeu de donne de test et jeu de donnée de train après on fait le prétraitement (la normalisation et encodage de donnée) ensuit on crée des modèles d'apprentissage automatique à la fin on teste les modèles avec le jeu de donnée de test.

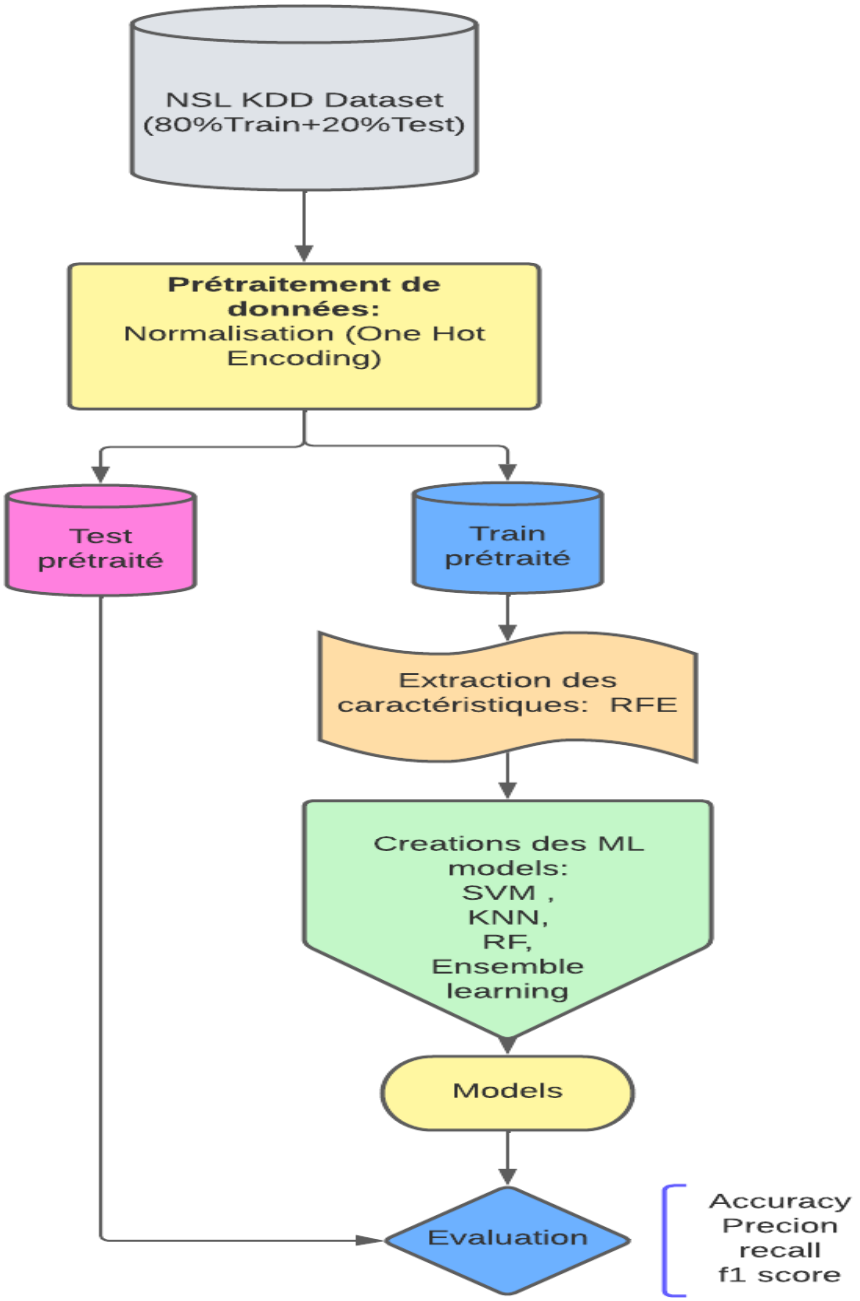

Figure 7.1 : L'architecture de modèle d'apprentissage automatique

3.1.2L’architecture de modèle d’apprentissage profond :

Notre tâche principale ici est de créer un modèle de détection d'attaque Ddos, comme le montre la figure 12.2, et de l'entraîner avec un réseau neuronal convolutif (CNN 1D), afin d'aider à l'identification du plus grand nombre d'attaques DDoS probables sur notre réseau.

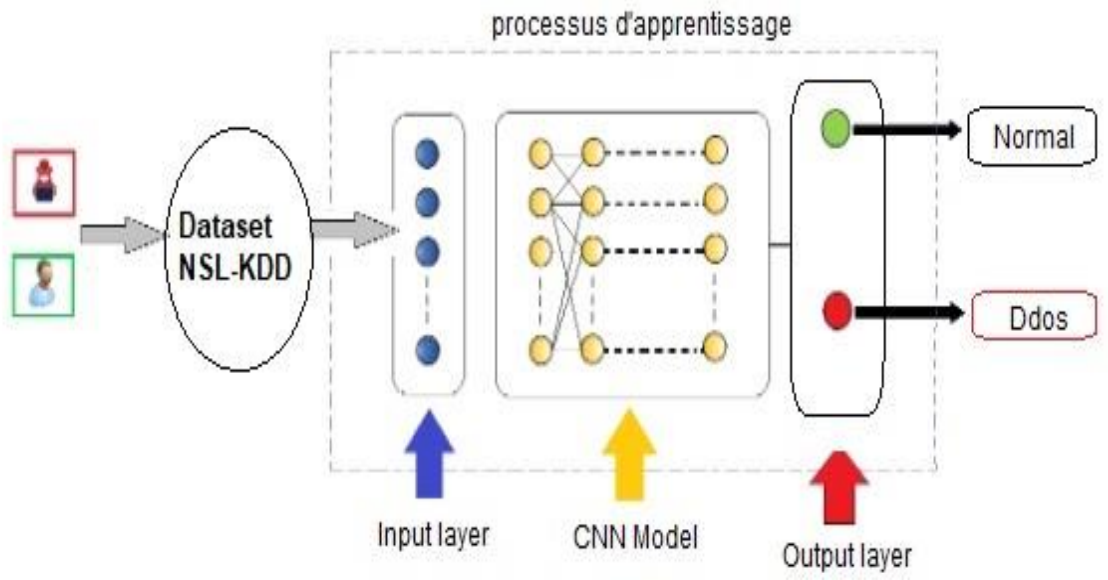

Figure 8.2 : L’architecture de modèle d’apprentissage profond

3.2Architecture et fonctionnement de l'approche proposée :

Le système fonctionne à chaque fois que le contrôleur demande aux commutateurs de renvoyer les statistiques des tables de flux. Ensuite, les commutateurs renvoient ces statistiques au contrôleur. Le contrôleur utilise ces statistiques pour prédire si les flux sont légitimes ou s'il s'agit de trafic DDoS. Si le trafic est prédit comme DDoS, le contrôleur indique la source de l'attaque et la victime.

La figure 9 explique le fonctionnement de ce mécanisme. Il montre comment les paquets sont envoyés au contrôleur une fois qu'ils arrivent au commutateur. Donc ici, le contrôleur extrait toutes les fonctionnalités de ce paquet et le sauvegarde. Ensuite, avant de l'envoyer au modèle, il sélectionne les fonctionnalités les plus importantes pour effectuer la prédiction afin de réduire le temps de prédiction. Dans la phase de détection, le contrôleur utilise le modèle pour faire la prédiction en utilisant les caractéristiques sélectionnées. Selon le résultat de la prédiction, le contrôleur prend la décision en informant comme dans les deux cas et en cas d'attaque, le contrôleur ajoute une nouvelle règle de flux au commutateur en utilisant le protocole OpenFlow pour échanger ces messages afin que le commutateur abandonne tous les paquets provenant du l'attaquant atténue et empêche ce trafic DDoS.

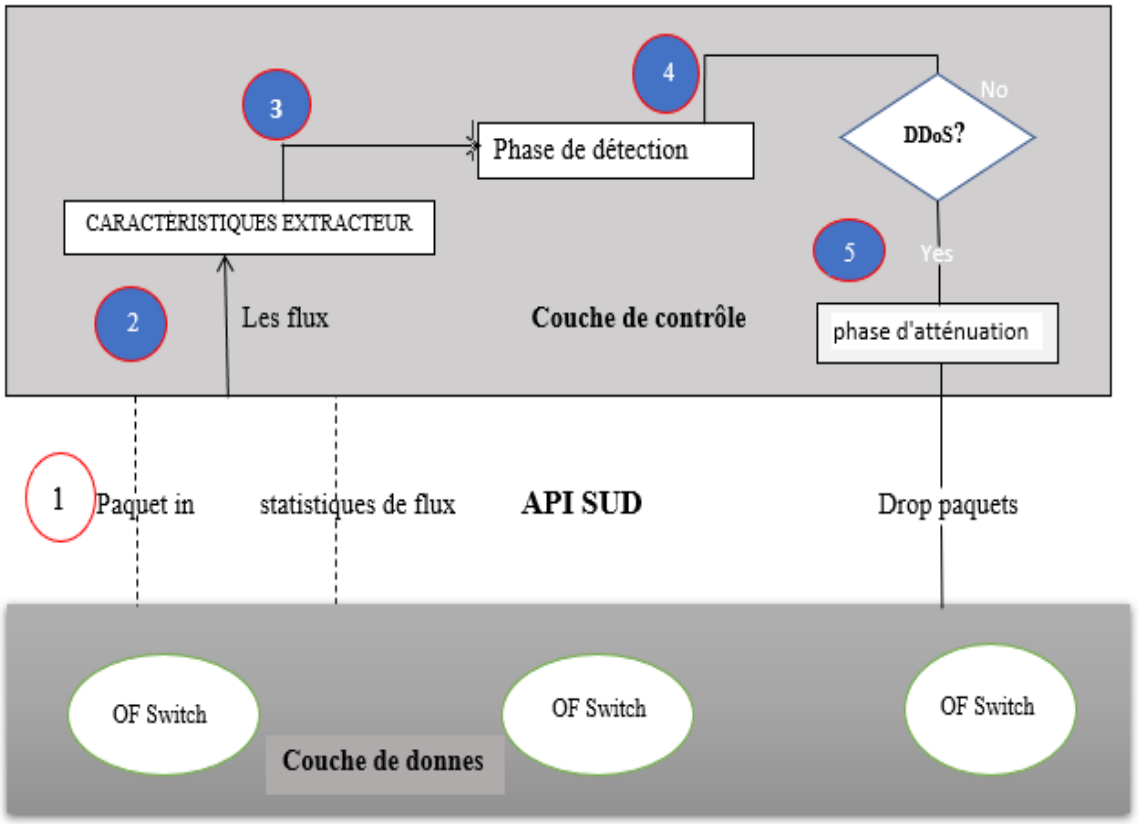

Figure 9:Proposed approach

3.3Dataset choisir

Le collecte de données choisi pour cette étude est la collecte de données NSL-KDD, Le jeu de données NSL-KDD est extrait du jeu de données KDD99 [39] est le plus largement utilisé dans la littérature. Les attaques présentées dans l'ensemble de données appartiennent à quatre grandes familles :

1) Attaque par déni de service (DoS)

Lorsqu'un attaquant rend un ordinateur ou une ressource mémoire trop occupé ou plein pour traiter des demandes valides, ou refuse aux utilisateurs légitimes l'accès à un système, cela s'appelle une attaque par déni de service (par exemple syn flood).

2) Attaque distante vers locale (R2L)

lorsqu'un attaquant ayant la capacité de transmettre des paquets à une machine sur un réseau mais sans compte sur cette machine utilise une vulnérabilité pour obtenir un accès local en se faisant passer pour l'utilisateur de cette machine (par exemple : devinez le mot de passe).

3) User to Root Attack (U2R)

il s'agit d'un type de piratage dans lequel l'attaquant accède d'abord à un compte d'utilisateur normal sur le système (peut-être par le biais d'un reniflage de mot de passe, d'une attaque par dictionnaire ou d'ingénierie sociale), puis exploite une vulnérabilité pour obtenir accès racine.

4) Probing Attack

est une tentative de recueillir des informations sur un réseau d'ordinateurs dans le but apparent de contourner ses contrôles de sécurité (par exemple: analyse des ports).[30]

Tableau 2:Distribution les fichier et les classes de NSL-KDD.[30]

Nom fichier	Description	Normal	DoS	Probe	R2L	U2R
KDDTrain+20 %	L'ensemble complet de trains NSL-KDD, y compris les étiquettes de type d'attaque et le niveau de difficulté au format CSV	13449 (53.39%)	9234 (36.65%)	2289 (9.09%)	209 (0.83%)	11 (0.04%)
KDDTrain+	Un sous-ensemble de 20% du fichier KDDTrain + .txt	67343 (53.46%)	45927 (36.46%)	11656 (9.25%)	995 (0.79%)	52 (0.04%)
KDDTest+	L'ensemble de test NSL-KDD complet, y compris le type d'attaque étiquettes et niveau de difficulté au format CSV.	9711 (43.08%)	7458 (33.08%)	2421 (10.74%)	2754 (12.22%)	200 (0.89%)
KDDTest-21	Un sous-ensemble du fichier KDD Test.txt qui n'inclut pas les enregistrements avec un niveau de difficulté de 21 sur 21	2152 (18.16%)	4342 (36.64%)	2402 (20.27%)	2754 (23.24%)	200 (1.69%)

Le tableau 3 contient une liste complète des qualités. Le terme "symbolique" est utilisé pour les variables de catégorie dans l'article original, tandis que "continu" est utilisé pour les variables numériques. Selon [29], les caractéristiques du jeu de données sont divisées en trois

catégories : basique, trafic et contenu.

1) Basique

Les aspects liés à la connexion, tels que les hôtes et les ports, les services utilisés et les protocoles, sont considérés comme fondamentaux.

2) Trafic

Les caractéristiques de trafic sont celles qui sont calculées en tant que groupe sur une période de temps. Le même agrégat basé sur l'hôte et le même agrégat de service sont deux autres catégories. La fenêtre temporelle a été remplacée dans NSLKDD par une fenêtre de connexion des 100 dernières connexions, ce qui constitue une distinction notable entre KDD'99 et NSLKDD.

3) Contenu

Les fonctionnalités de contenu sont collectées à partir des données de paquets ou de la charge utile et sont liées au contenu de l'application ou aux protocoles utilisés.[30]

Tableau 3:caractéristiques de NSL-KDD

Feature	Type	Feature	Type
duration	cont.	is_guest_login	sym.
protocol_type	sym.	count	cont.
service	sym.	srv_count	cont.
flag	sym.	serror_rate	cont.
src_bytes	cont.	rerror_rate	cont.
dest_bytes	cont.	srv_rerror_rate	cont.
land	sym.	diff_srv_rate	cont.
wrong_fragment	cont.	srv_diff_host_rate	cont.
urgent	cont.	dst_host_count	cont.
hot	cont.	dst_host_srv_count	cont.
num_failed_logins	cont.	dst_host_same_srv _rate	cont.
logged_in	sym.	dst_host_diff_srv_rate	cont.
num_compromised	cont.	dst_host_same_src_port_rate	cont.
root_shell	cont.	dst_host_srv_diff_host_rate	cont.
su_attempted	cont.	dst_host_serror_rate	cont.
num_root	cont.	dst_host_srv_serror_rate	cont.
num_file_creations	cont.	dst_host_rerror_rate	cont.
num_access_files	cont.	dst_host_srv_rerrorv rate	cont.
num_outbound_cmds	cont.	is_host_login	sym.

4. Conclusion

Dans ce chapitre, on crée le modèle intelligent pour mettre en œuvre a le chapitre suivant, nous avons choisi l'emplacement où nous avons proposé l'approche intelligente, enfin on fait un bref a le jeu de donnée qui vous a sélectionné.

Chapitre 4

Implémentation et Résultat

1. Introductions

Dans ce chapitre nous avons suivi les étapes d'un projet d'apprentissage automatique, nous présenterons les résultats de chaque algorithme que nous avons obtenus, ainsi que les différentes méthodes de sélection des attributs, Nous faisions une comparaison avec les résultats d'une autre étude. Enfin décrire les différents outils de développement.

2. Les outils de développement

2.1.Hardware

L'implémentation de notre contribution est généré dans un pc laptop Dell , Intel(R) Core(TM) i5-8550U CPU @ 1.80GHz 2.00 GHzavec un ram de 8Go .

2.2. Software

1) Google Colab

Est un produit de Google Research. Colab permet à quiconque d'écrire et d'exécuter du code Python arbitraire via le navigateur, et est particulièrement bien adapté à l'apprentissage automatique, à l'analyse de données et à l'éducation.

Cet outil nous permet de développer en un clin d'œil des applications de Machine Learning en Python. Pour commencer, tout ce que nous devons faire est d'avoir un compte Gmail.[59]

Figure 10: Logo de google colab

2) Python

Le langage Python est un langage de programmation open source multiplateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures. Il n'est donc pas, comme le langage HTML par exemple, uniquement dédié à la programmation web.[55]

Figure 11:Logo de language python

3) Sklearn

Nous avons utilisé Sklearn qui est une bibliothèque d'apprentissages statique en python, contient toutes les fonctions de l'état de l'art du Machine Learning. Ony trouve les algorithmes les plus importants ainsi que diverses fonctions de pré-processing.[40]

Les fonctionnalités fournies par scikit-learn incluent :

•Régression, (la régression linéaire et logistique).

•Classification, (les voisins les plus proches).

•Clustering, (K-Means et K-Means++).

•Sélection du modèle.

•Pré-traitement, (la normalisation Min-Max).[57]

Elle est construite à base de :

Figure 12:Logo de Sklearn

4) Pandas

Pandas est une excellente bibliothèque pour importer vos tableaux Excel (etautres formats) dans Python dans le but de tirer des statistiques et de charger votre Dataset dans Sklearn.[40]

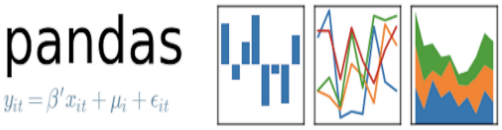

Figure 13:Logo de pandas

5) Matplotlib

Matplotlib est la bibliothèque qui permet de visualiser nos Datasets, nos fonctions, nos résultats sous forme de graphes, courbes et nuages de points.[40]

Figure 14 :Logo de matplolib

6) Seaborn

Seaborn est une bibliothèque permettant de créer des graphiques statistiques en Python. Elle est basée sur Matplotlib, et s'intègre avec les structures Pandas.[58]

Figure 15:Logo de seaborn

3. Implémentation

3.1.Etape de l'implémentation

Voici les étapes que nous avons suivies pour mener à bien notre projet :

L'analyse exploratoire des données

C'est la première étape de la définition de la stratégie de modélisation par la compréhension des variables. Il est divisé en deux sections :

Analyse de la forme

- Attribuer les noms des columens selon la description de dataset

```
col_names = ["duration","protocol_type","service","flag","src_bytes",
    "dst_bytes","land","wrong_fragment","urgent","hot","num_failed_logins",
    "logged_in","num_compromised","root_shell","su_attempted","num_root",
    "num_file_creations","num_shells","num_access_files","num_outbound_cmds",
    "is_host_login","is_guest_login","count","srv_count","serror_rate",
    "srv_serror_rate","rerror_rate","srv_rerror_rate","same_srv_rate",
    "diff_srv_rate","srv_diff_host_rate","dst_host_count","dst_host_srv_count",
    "dst_host_same_srv_rate","dst_host_diff_srv_rate","dst_host_same_src_port_rate",
    "dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_host_srv_serror_rate",
    "dst_host_rerror_rate","dst_host_srv_rerror_rate","label"]
```

Figure 16:Attribuer les noms des columens

Nombre de lignes et de colonnes

```
print('Dimensions of the Training set:',df.shape)
print('Dimensions of the Test set:',df_test.shape)
```

```
Dimensions of the Training set: (125973, 42)
Dimensions of the Test set: (22544, 42)
```

Figure 17:Le nombre de ligne et de colonne

4. L'implémentations de model d'apprentissage profond :

La conception du réseau de neurones diffère selon le modèle utilisé. Dans notre scénario, nous avons utilisé une architecture distincte basée sur CNN 1D pour tester comment le choix de différentes conceptions affecte notre précision. Dans les architectures, nous avons utilisé une fonction d'activation ReLU car elle produit systématiquement les meilleurs résultats. Le modèle peut empiler une chaîne de couches convolutives avec des couches de regroupement (MaxPooling1D), et nous avons utilisé un ReLU dans la couche de sortie et une fonction d'activation SoftMax. Étant donné que les exemples d'étiquettes sont catégoriques, il existe une

couche de sortie distincte pour chaque couche de sortie dans la conception. Le nombre de classes différentes est le même que les dimensions de la régression Softmax pour que la couche de sortie classe le paquet.

Layer (type)	Output Shape
conv1d (Conv1D)	(None, 40, 64)
max_pooling1d (MaxPooling1D)	(None, 20, 64)
dropout (Dropout)	(None, 20, 64)
conv1d_1 (Conv1D)	(None, 20, 64)
conv1d_2 (Conv1D)	(None, 20, 32)
max_pooling1d_1 (MaxPooling1D)	(None, 10, 32)
dropout_1 (Dropout)	(None, 10, 32)
flatten (Flatten)	(None, 320)
dense (Dense)	(None, 128)
dropout_2 (Dropout)	(None, 128)
dense_1 (Dense)	(None, 15)
dropout_3 (Dropout)	(None, 15)
dense_2 (Dense)	(None, 2)

Figure 18.1 : Architecture CNN-1D de NSL_KDD

L'expérimentation (classification binaire) qui a été faite sur l'ensemble de données du NSL-KDD Data-set indiqué aux figure 21.2. Le modèle ont étaient évaluer directement sur l'ensemble de test. Le modèles Cnn a été formé sur 30 itérations, nous notons que l'exactitude d'apprentissage et de validation augmente constamment du début à la fin, elle atteint une valeur maximale tend vers 1. Nous notons aussi que la valeur de perte se diminue fortement durant l'entraînement et l'évaluation et atteint une valeur minimale tend vers 0. Cela signifie que ces modèles apprennent mieux et effectuent des meilleures prédictions après chaque époque d'optimisation.

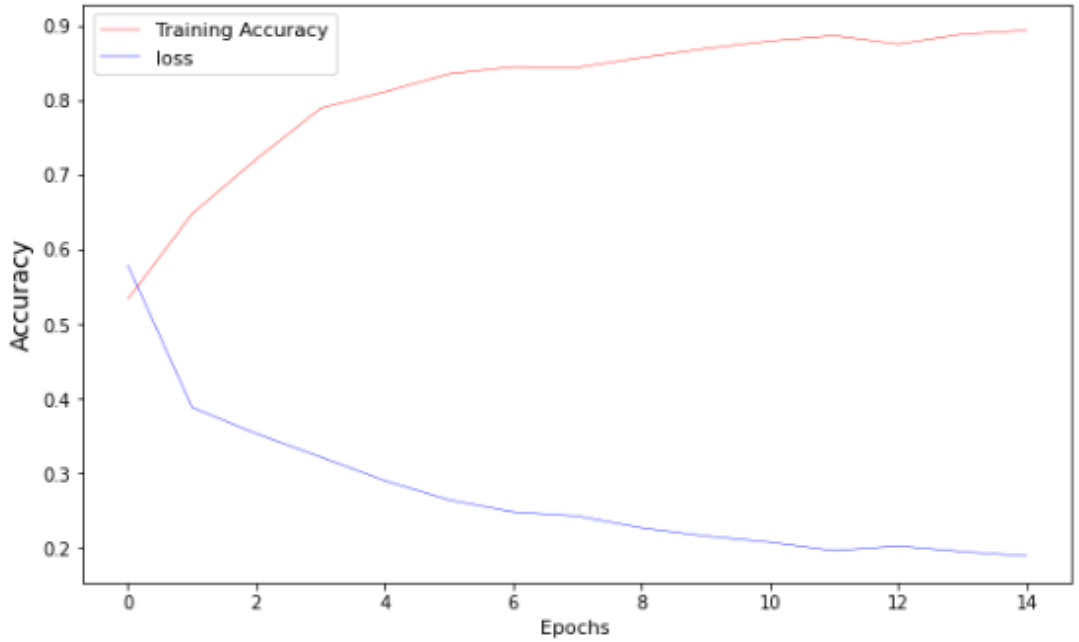

Figure 19.2 : Architecture CNN-1D de NSL_KDD de précision training et test

5. L'implémentations de model d'apprentissage automatique :

5.1Prétraitement

Le prétraitement est l'étape au cours de laquelle nous préparons nos données avant de les transmettre à la machine pour l'apprentissage, et il consiste à maitres de données dans un format adapté à l'apprentissage automatique

1) Préparation du data set

- One-Hot-Encoding

Est utilisé pour convertir toutes les propriétés catégorielles en propriétés binaires. Exigence One-Hot-Endcoding, l'entrée de ce transformateur doit être une matrice entière exprimant les valeurs reçues avec des propriétés catégorielles (discrètes). La sortie sera une matrice creuse où chaque colonne correspond à une valeur possible. Les propriétés d'entrée sont supposées prendre des valeurs dans la plage [0, n_values]. Par conséquent, pour convertir chaque catégorie en nombre, les propriétés doivent d'abord être converties avec LabelEncoder.

LabelEncoder : insérer des caractéristiques catégorielles dans un tableau numpy 2D et transformez les caractéristiques catégorielles en nombres à l'aide de LabelEncoder.

```
df_categorical_values_enc=df_categorical_values.apply(LabelEncoder().fit_transform)
```

```
  protocol_type   service flag
0           tcp  ftp_data   SF
1           udp     other   SF
2           tcp   private   S0
3           tcp      http   SF
4           tcp      http   SF
--------------------
   protocol_type  service  flag
0              1       20     9
1              2       44     9
2              1       49     5
3              1       24     9
4              1       24     9
```

Figure 20:Préparation du data set à l'aide de LabelEncoder

De nouvelles colonnes numériques sont ajoutées qui manquantes au dataframe de train et de test.

```
print(df_cat_data.shape)
print(testdf_cat_data.shape)
```

```
(125973, 84)
(22544, 84)
```

Figure 21:2.2 Nombre de colonne avant l'ajout

```
print(newdf.shape)
print(newdf_test.shape)
```

```
(125973, 123)
(22544, 123)
```

Figure 22:Nombre de colonne après l'ajout

L'ensemble de données a été divisé en ensembles de données distincts pour chaque catégorie d'attaque. Les tags d'attaque ont été renommés pour chacun. 0=Normal, 1=DDoS, 2=Sonde, 3=R2L, 4=U2R. Dans les nouveaux ensembles de données, la colonne d'étiquette est remplacée par de nouvelles valeurs.

```
labeldf=newdf['label']
labeldf_test=newdf_test['label']


# change the label column
newlabeldf=labeldf.replace({ 'normal' : 0, 'neptune' : 1 ,'back': 1, 'land': 1, 'pod': 1, 'smurf': 1, 'teardrop': 1,
                            'ipsweep' : 2,'nmap' : 2,'portsweep' : 2,'satan' : 2,'mscan' : 2,'saint' : 2
                            ,'ftp_write': 3,'guess_passwd': 3,'imap': 3,'multihop': 3,'phf': 3,'spy': 3,'warezclient'
                            'buffer_overflow': 4,'loadmodule': 4,'perl': 4,'rootkit': 4,'ps': 4,'sqlattack': 4,'xterm
newlabeldf_test=labeldf_test.replace({ 'normal' : 0, 'neptune' : 1 ,'back': 1, 'land': 1, 'pod': 1, 'smurf': 1, 'tea
                            'ipsweep' : 2,'nmap' : 2,'portsweep' : 2,'satan' : 2,'mscan' : 2,'saint' : 2
                            ,'ftp_write': 3,'guess_passwd': 3,'imap': 3,'multihop': 3,'phf': 3,'spy': 3,'warezclient'
                            'buffer_overflow': 4,'loadmodule': 4,'perl': 4,'rootkit': 4,'ps': 4,'sqlattack': 4,'xterm
```

Figure 23:Remplacée l'étiquette de la colonne avec nombre

Filtrer toutes les lignes avec une valeur d'étiquette

```
to_drop_DoS = [0,1]
DoS_df=newdf[newdf['label'].isin(to_drop_DoS)];
#test
DoS_df_test=newdf_test[newdf_test['label'].isin(to_drop_DoS)];
```

Figure 24:Filtrer toutes les lignes avec une valeur d'étiquette

2) Sélectionne les caractéristiques

Élimination des fonctionnalités récursives (RFE): est le meilleur moyen conçu jusqu'à présent pour choisir les meilleures fonctionnalités. Il fonctionne sur une sorte d'algorithme récursif. Il prend d'abord en compte toutes les fonctionnalités, construit un modèle prédit, puis trouve les fonctionnalités les moins importantes. Maintenant, il élimine ces fonctionnalités, reconstruit le modèle et vérifie à nouveau les effets de la suppression des fonctionnalités. Ce processus implique une étape de validation croisée k fois afin de supprimer tout type de problème de déséquilibre dans l'ensemble. Nous définissons l'estimateur ou le modèle que le processus utilise.

Dans cette étape on utilise Éliminateur récursive des fonctionnalités (RFE) qui choisissait 13 caractéristiques à partir de 123 caractéristiques.

```
print('Features selected for DDoS:',rfecolname_DoS)
print()
print(X_rfeDoS.shape)
```

```
Features selected for DDoS: ['src_bytes', 'wrong_fragment', 'count', 'srv_count', 'same_srv_rate', 'diff_srv_rate'

(113270, 13)
```

Figure 25:Sélectionne les caractéristiques

5.2Résultat des Algorithmes de l'apprentissage automatique

A travers le chapitre 02, nous avons implémenté trois algorithmes de classification (KNeighbors, SVM, Random Forest), à la fin vous avez collecté et appliqué les trois algorithmes.

Apprentissage de l'algorithme par les données

Nous allons enfin pouvoir appliquer nos algorithmes de classification, pour chaque algorithme nous allons montrer les différentes fonctions utilisées :

```
import pandas as pd
import numpy as np
from sklearn.feature_selection import RFE
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
from sklearn import metrics
import sys
import sklearn
import io
import random
```

Figure 26:importation des bibliographies

Nous avons utilisé quatre modèles de classification le fig.29 représente les estimateurs Les algorithmes que nous avons utilisés :

```
clf = RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_KNN_DoS=KNeighborsClassifier()

clf_SVM_DoS=SVC(kernel='linear', C=1.0, random_state=0)

clf_voting_DoS = VotingClassifier(estimators=[('rf', clf_DoS), ('knn', clf_KNN_DoS), ('svm', clf_SVM_DoS)], voting='hard')
```

Figure 27:Déclaration des algorithmes

Les tableaux suivants montrent les résultats de chacun pour obtenir le meilleur résultat et les figures présenter les matrices de confusion. La classe 0 signifie que l'enregistrement n'est pas une attaque et la classe 1 est une attaque :

1)Application de l'algorithme Random Forest :

Tableau 4:Résultats spécifique de l'algorithme Random Forest

Algorithme	Paramètre	Précision	Recall	F1-score	Accuracy
Random Forest	`n_estimators=10, n_jobs=2`	`0.99745 (+/- 0.00326)`	`0.99517 (+/- 0.00659)`	`0.99631 (+/- 0.00353)`	`0.99668 (+/- 0.00321)`

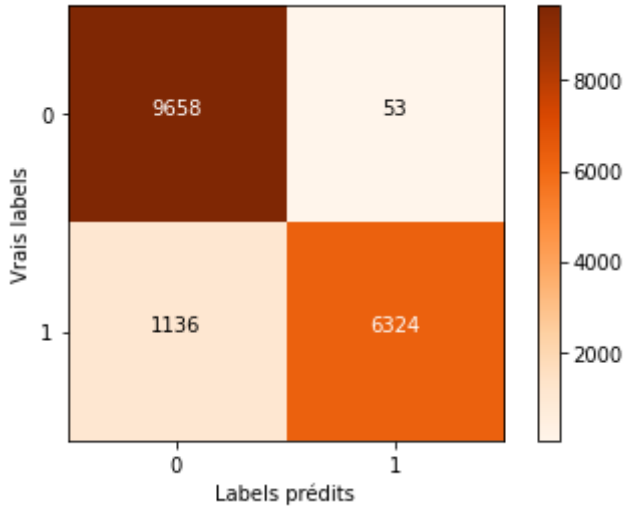

Figure 28:matrice de confusion de l'algorithme Random Forest

2)Application de l'algorithme KNN :

Tableau 5:Résultats spécifique de l'algorithme KNN

Algorithme	Précision	Recall	F1-score	Accuracy
KNN	`0.99678 (+/- 0.00383)`	`0.99665 (+/- 0.00344)`	`0.99672 (+/- 0.00320)`	`0.99715 (+/- 0.00278)`

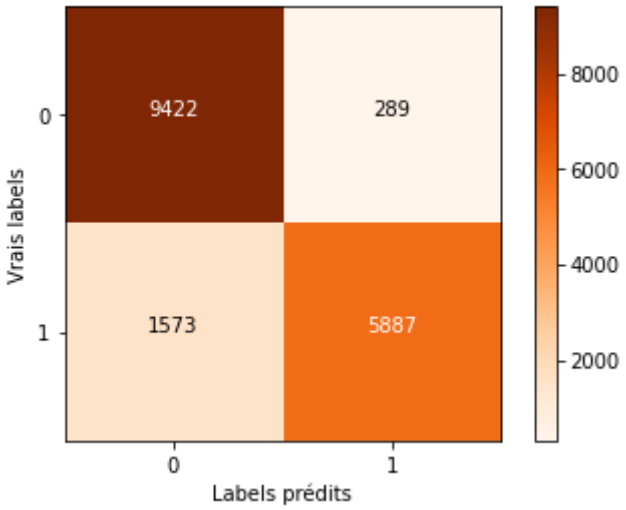

Figure 29:matrice de confusion de l'algorithme KNN

3)Application de l'algorithme SVM :

Tableau 6:Résultats spécifique de l'algorithme SVM

Algorithme	Paramètre	Précision	Recall	F1-score	Accuracy
SVM	`kernel='linear' C=1.0, random_state=0`	`0.99107 (+/- 0.00785)`	`0.99450 (+/- 0.00388)`	`0.99278 (+/- 0.00428)`	`0.99371 (+/- 0.00375)`

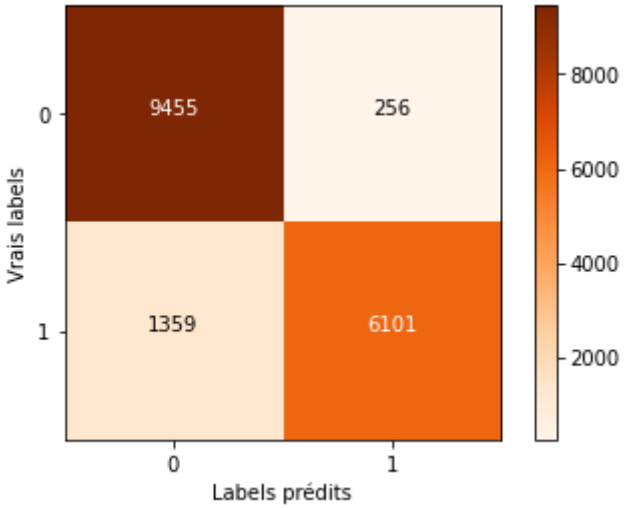

Figure 30:matrice de confusion de l'algorithme SVM

4)Application des plusieurs algorithmes :

Tableau 7:Résultats spécifique des plusieurs algorithmes

Algorithme	Paramètre	Précision	Recall	F1-score	Accuracy
Plusieurs algorithme	estimators=('rf', clf_DoS), ('knn', clf_KNN_DoS), ('svm', clf_SVM_DoS), voting='hard'	0.99812 (+/- 0.00298)	0.99718 (+/- 0.00369)	0.99765 (+/- 0.00270)	0.99802 (+/- 0.00203)

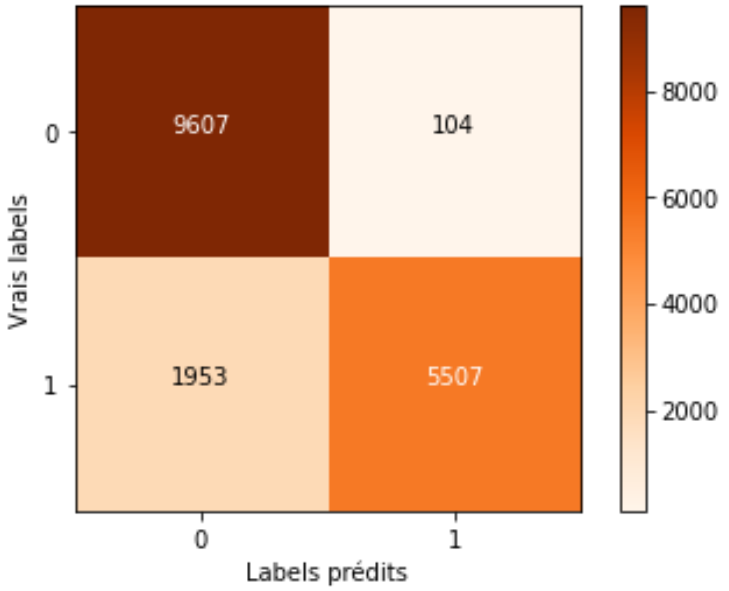

Figure 31: matrice de confusion des plusieurs algorithmes

Les résultats de cette étude montrent que le RFE fonctionne très bien avec ces données. Les scores se sont améliorés encore plus après la réduction des caractéristiques avec le même classificateur. Les résultats de Recall sont très bons.

6. Comparaison entre nos résultats et les travaux connexe

Nous avons comparé nos résultats finals avec les résultats de l'article :« DDoS Detection in SDN using Machine Learning Techniques »[29] et de l'article [26]

D'après les premières observations, on a des bons résultats en comparaison avec le [29] mais mal résulta avec le [26]

- ✓ Le fonctionnement d'algorithme SVM de [29] n'est pas bien en comparaison avec notre résultat.

- ✓ Notre modèle a un très bon résultat avec un accuracy 99.74% et F1-score de99.70%, et un recall de 99.83%et précision de 99.89%.
- ✓ Les résultats de nos modèles (Random Forest et KNeighbors) sont bons par rapport à [29].
- ✓ En arrive un excellent résultat après d'applique Ensemble Learning avec un accuracy 100%.

Le tableau suivant montre les différents résultats obtenus :

Tableau 8:Comparaison entre nos résultats et les résultats d'un autre article [29]et [26]

	Nos résultats	**Les résultats de l'article**
Algorithms	**Accuracy**	**Accuracy**
Random Forest classifier	99.72%	**99.97%**
KNeighbors Classifier	**99.99%**	98.57%
SVM	**99.74%**	89.18%
Ensemble learning	**100%**	99.80%
CNN-1D	**63%**	90%

7. Conclusion

Dans ce chapitre plusieurs systèmes de classification pour l'ensemble de donnée NSL-KDD ont été mis au point à l'aide de l'intelligence artificiel au début on applique un model de deep Learning (CNN-1D) et on a donné des résulta moyenne de 63% après on applique les algorithmes d'apprentissage automatique afin de classer les données dans catégorie attaque ddos ou dans la catégorie normale. La plupart du temps le développement a été consacré à résoudre dans Le prétraitement de donnée. L'utilisation de l'Éliminateur récursive des fonctionnalités (RFE) avec les quatre model Random Forest, SVM KNN et ensemble Learning donne de résulta très élevés et le meilleur résultat obtenu avec le classificateur SVM 99,99% .et nous avons présenté langage de programmation et la bibliothèque utilisée.

Conclusion Générale

& Perspectives

Conclusion

Bien que le SDN réalise des gains de mise en réseau significatifs, il est confronté à un certain nombre de défis de sécurité, dont les plus fréquents sont les attaques DDoS. Le contrôleur SDN est le principal point de contrôle de l'ensemble du réseau, ce qui le rend plus vulnérable aux attaques DDoS. Par conséquent, une détection efficace des attaques DDoS dans le SDN est requise. En réponse à ce problème, cette étude propose une comparaison entre les model deep Learning (CNN-1D) et les classificateurs alternatifs d'apprentissage automatique basés sur le meilleur sous-ensemble de fonctionnalités pour détecter les attaques DDoS via le SDN de manière précoce et précise.

De plus, pour identifier correctement une attaque, l'extraction et la sélection de fonctionnalités appropriées pour les modèles basés sur l'apprentissage automatique sont essentielles. Les résultats expérimentaux montrent que l'utilisation d'un classificateur RFE avec un sous-ensemble de fonctionnalités récursives pour détecter les agressions dans le contrôleur SDN produit des résultats décents.

Perspectives

Les résultats trouvés sont vraiment intrigants. Nous proposons quelque modification pour des travaux futurs à l'issue des travaux que nous avons effectués dans le cadre de ce projet de fin d'études :

- ☑ D'autres types d'attaques dans le SDN seraient détectés à l'aide de ces classificateurs d'apprentissage automatique et d'approches de sélection de fonctionnalité RFE, telles que smurf, Probe, R2L et U2R avec le jeu de donne NSL-kdd .
- ☑ Peut-être on test sur d’autres types des problèmes réels contraints et de représentations de données.

References

& Bibliographies

Références & Bibliographies

[1] Haji, S. H., Zeebaree, S. R., Saeed, R. H., Ameen, S. Y., Shukur, H. M., Omar, N., ... & Yasin, H. M. (2021). Comparison of software defined networking with traditional networking. *Asian Journal of Research in Computer Science*, 1-18.

[2]Klöti, R., Kotronis, V., & Smith, P. (2013, October). OpenFlow: A security analysis. In *2013 21st IEEE International Conference on Network Protocols (ICNP)* (pp. 1-6). IEEE.

[3] Choukri, I., Ouzzif, M., & Bouragba, K. (2019, June). Software Defined Networking (SDN): Etat de L'art. In *Colloque sur les Objets et systèmes Connectés*.

[4] Facchini, H., Perez, S., Blanchet, R., Roberti, B., & Azcarate, R. (2021, December). Experimental performance contrast between SDN and traditional networks. In *2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)* (pp. 1-6). IEEE.

[5]Mehmeri, V. D., Wang, X., & Palacharla, P. (2017). Self-Healing Services with Software-Programmed Networking. *IEEE Communications Standards Magazine*, *1*(4), 62-69.

[6] Andrianirina, Y. Z. (2021). *Développement d'un réseau défini par logiciel (SDN) programmable, transparent et ouvert* (Doctoral dissertation, Université du Québec en Outaouais).

[7] Bahnasy, M. M. (2014). OpenFlow protocol extension for optical networks.

[8] https ://www.futura-sciences.com/tech/definitions/internet-deni-service-2433. [En ligne ; Consulté le 4/5/2022,10 :30]

[9] Labidi, A. (2017). *Partage efficace des ressources de calcul dans le nuage informatique* (Doctoral dissertation, École de technologie supérieure).

[10] Dridi, L. (2017). *Mitigation des attaques de déni de service dans les réseaux définis par logiciel* (Doctoral dissertation, École de technologie supérieure).

[11] Masoudi, R., & Ghaffari, A. (2016). Software defined networks: A survey. *Journal of Network and computer Applications*, *67*, 1-25.

[12] Feamster, N., Rexford, J., & Zegura, E. (2014). The road to SDN: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, *44*(2), 87-98.

[13] Rawat, D. B., & Reddy, S. R. (2016). Software defined networking architecture, security and energy efficiency: A survey. *IEEE Communications Surveys & Tutorials*, *19*(1), 325-346.

[14] Deng, S., Gao, X., Lu, Z., & Gao, X. (2017). Packet injection attack and its defense in software-defined networks. *IEEE Transactions on Information Forensics and Security*, *13*(3), 695-705.

[15] Singh, J., & Behal, S. (2020). Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. *Computer Science Review*, *37*, 100279.

[16]https://www.riskinsight-wavestone.com/2013/06/ddos-les-attaques-se-diversifient/ddos-volumetrique/. [En ligne ; Consulté le 15/03/2022]

[17] https://hal.archives-ouvertes.fr/hal-00914181/document. [En ligne ; Consulté le 15/03/2022]

[18]https://www.researchgate.net/figure/ONF-SDN-architecture-3_fig1_287706382. [En ligne

; Consulté le 15/03/2022].

[19] Hafeez, K. O. M. A. L., & Ahmed, Q. A. N. E. T. A. H. (2019). Applications of machine learning in education and health sector: An empirical study. *Journal of Software Engineering and Intelligent Systems*, *3*.
[20] BOUDJRADA, A., & DJEKAOUA, Y. (2021). *PREDICTIONS DES ETATS DES RADIATIONS PAR RN RECURENT (TDNN)* (Doctoral dissertation, université Ghardaia).

[21] HAMOUDA, D. (2020). Un système de détection d'intrusion pour la cybersécurité.
[22] Sambangi, S., & Gondi, L. (2020, December). A machine learning approach for DDoS (distributed denial of service) attack detection using multiple linear regressioN. In *Proceedings* (Vol. 63, No. 1, p. 51). MDPI.
[23] Yuan, X., Li, C., & Li, X. (2017, May). DeepDefense: identifying DDoS attack via deep learning. In *2017 IEEE international conference on smart computing (SMARTCOMP)* (pp. 1-8). IEEE.
[24] Guerra-Manzanares, A., Bahsi, H., & Nõmm, S. (2019, October). Hybrid feature selection models for machine learning based botnet detection in IoT networks. In *2019 International Conference on Cyberworlds (CW)* (pp. 324-327). IEEE.
[25] Gormez, Y., Aydin, Z., Karademir, R., & Gungor, V. C. (2020). A deep learning approach with Bayesian optimization and ensemble classifiers for detecting denial of service attacks. *International Journal of Communication Systems*, *33*(11), e4401.
[26] Ugwu, C. C., Obe, O. O., Popoọla, O. S., & Adetunmbi, A. O. (2021, February). A distributed denial of service attack detection system using long short term memory with Singular Value Decomposition. In *2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA)* (pp. 112-118). IEEE.
[27] Kasim, Ö. (2020). An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. *Computer Networks*, *180*, 107390.
[28] Amira, F., & Yousra, L. (2021). *Détection des attaques DDOS dans le Cloud Computing* (Doctoral dissertation, university center of abdalhafid boussouf-MILA).
[29] Nadeem, M. W., Goh, H. G., Ponnusamy, V., & Aun, Y. (2022). Ddos detection in sdn usingmachine learning techniques. *Comput. Mater. Contin.*, *71*(1), 771-789.
[30] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). Ieee.

[31] https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning.

[32] https ://www.lebigdata.fr/confusion-matrix-definition.

[33] https://datascientest.com/glossary/validation-croisee-cross-validation.

[34]https://www.ledigitalab.com/2017/10/02/intelligence-artificielle-machine-learning-deep-learning kezako/.

[35]https://www.javatpoint.com/machine-learning-random-forest-algorithm.

[36] MBAYE, M. (2020). Un plan de contrôle intelligent pour le déploiement de services de sécurité dans les réseaux SDN. *Gestion et contrôle intelligents des réseaux: Sécurité intelligente, optimisation multicritères, Cloud Computing, Internet of Vehicles, radio intelligente*, 29.

[37]https://www.memoireonline.com/04/12/5750/mIdentification-et-commande-des-systemes-nonlineaires21.html. [En ligne ; Consulté le 14/04/2022].

[38] Swami, R., Dave, M., & Ranga, V. (2019). Software-defined networking-based DDoS defense mechanisms. *ACM Computing Surveys (CSUR)*, *52*(2), 1-36.

[39] Thomas, R., & Pavithran, D. (2018). A survey of intrusion detection models based on NSL-KDD data set. *2018 Fifth HCT Information Technology Trends (ITT)*, 286-291.

[40] https://www.scribd.com/document/455692323/Apprendre-le-ML-en-une-semaine-pdf

[41]Santosh Ku,Yogesh Ku, Sheo Ku,"Characteristics Categorization Dataset KDD cup'99",Srivastava, Santosh Kumar; Sharma, Yogesh Kumar; Kumar, Sheo (2019).

[42] Sarika Ch, Nishtha Ke,"Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasetsusing Deep Learning in IoT",International Conference on Computational Intelligence and Data Science (ICCIDS 2019).

[43] Araújo, J. T., & Fukuda, K. (2011, December). MALAWI: Aggregated longitudinal analysis of the MAWI dataset. In *Proceedings of The ACM CoNEXT Student Workshop* (pp. 1-2).

[44] Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *computers & security*, *31*(3), 357-374.

[45] Jabbar, A. F., & Mohammed, I. J. (2020, November). Development of an optimized botnet detection framework based on filters of features and machine learning classifiers using cicids2017 dataset. In *IOP Conference Series: Materials Science and Engineering* (Vol. 928, No. 3, p. 032027). IOP Publishing.

[46] Jabbar, A. F., & Mohammed, I. J. (2020, November). Development of an optimized botnet detection framework based on filters of features and machine learning classifiers using cicids2017 dataset. In *IOP Conference Series: Materials Science and Engineering* (Vol. 928, No. 3, p. 032027). IOP Publishing.