الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Kasdi Merbah Ouargla

Faculté des Nouvelles Technologies de l'information et de Communication

Département d'informatique et des Technologies de l'information

جامعة قاصدي مرباح ورقلة

كلية التكنولوجيات الحديثة للمعلومات والاتصال

قسم للاعلام الالي وتكنولوجيات المعلومات

Mémoire en vue de l'obtention du diplôme de Master

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Administration et Sécurité des Réseaux

THEME :

Un système de détection d'intrusion basés sur l'apprentissage profond pour la cybersécurité

Présenté par : ZITOUNI Dounia et BARKA Aicha.

Les Membres du Jury :

Président :	**Zga Adel**		
Examinateur :	**Ben Said Khalid**		**U.K.M Ouargla**
Encadreur :	**Mr. BENKADDOUR Mohammed Kamel**	**MCA**	**U.K.M Ouargla**

Année Universitaire 2023/2024

Remerciements

Tout d'abord, je remercie Allah, le Tout-Puissant, de m'avoir accordé le courage, la patience et la santé nécessaires pour mener à bien ce travail.

La réalisation de ce mémoire a été rendue possible grâce au soutien de plusieurs personnes, auxquelles je souhaite exprimer toute ma gratitude.

Je tiens à adresser mes remerciements les plus sincères à mon directeur de recherche, Monsieur ***Dr BENKADDOUR mohammed kamel ,*** *pour ses conseils avisés, son soutien indéfectible et ses orientations précieuses. Je lui suis profondément reconnaissante pour le temps et l'aide qu'il m'a généreusement accordés tout au long de l'élaboration de ce travail.*

Je souhaite également remercier tout particulièrement ***Doctorante ABID Malika****, qui a été la première à me faire découvrir le sujet de mon mémoire.*

Je tiens à exprimer ma reconnaissance envers mes amis et collègues, pour leur soutien moral et intellectuel tout au long de ma démarche.

Enfin, j'aimerais exprimer ma gratitude à les chercheurs et spécialistes, qui ont pris le temps de discuter mon sujet.

Dédicace

Je dédie ce modeste travail :

*À ma **mémre** qui m'a soutenu et encouragé durant ces années d'études. Qu'elle trouve ici le témoignage de ma profonde reconnaissance. Que Dieu lui procure bonne santé et longue vie.*

*À mes **frères** et **sœurs**, pour leur amour, leur soutien moral, et leur patience tout au long de ce parcours.*

*À ma collègue et amie **Aïcha**, qui a partagé avec moi ce travail et cette fatigue. Nous avons surmonté ensemble de nombreuses étapes pour atteindre cet objectif.*

*À mes **amis** et c**ollègues**, pour leur aide, leur coopération et les moments partagés, qui ont rendu cette expérience enrichissante.*

*À tous les **enseignants** et **professeurs** qui m'ont transmis leur savoir et ont guidé mes pas dans ce chemin d'apprentissage.*

Enfin, à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail. Que Dieu vous bénisse et vous accorde prospérité et bonheur.

Dounia

Dédicace

Je dédie ce modeste travail :

*À mes **parents** qui m'ont apporté leur soutien et leur encouragement tout au long de mes années d'études. Je leur exprime ici ma profonde reconnaissance. Que Dieu leur accorde une santé exceptionnelle et une vie longue.*

*À ma sœur **Amel**, qui a toujours été ma motivation et mon encouragement pour terminer mon parcours académique.*

*Je tiens à remercier mes **frères** et **sœurs** pour leur amour, leur soutien moral et leur patience tout au long de cette expérience.*

*À ma collègue et amie **Dounia**, qui m'a fait part de cette tâche et de cette monotonie. Ensemble, nous avons franchi de nombreuses étapes afin d'atteindre cet objectif.*

Aicha

Résumé

Le domaine des réseaux d'information s'est considérablement élargi et ouvert grâce aux progrès technologiques, ce qui a conduit à l'émergence de nouvelles techniques pour accéder aux réseaux et aux systèmes d'information. Ces techniques exposent également le réseau à de nouvelles menaces cybernétiques, ce qui a conduit au développement de techniques de protection des réseaux, telles que les systèmes de détection d'intrusions (IDS). Pour améliorer les taux de détection des attaques au sein des IDS, des techniques d'intelligence artificielle sont utilisées.

Cette étude propose d'intégrer les systèmes IDS dans les réseaux informatiques en utilisant des approches d'apprentissage profond et d'apprentissage automatique. Trois techniques de détection basées sur des approches d'apprentissage profond (DNN, CNN, RNN) et d'apprentissage automatique (Random Forest, KNN et Naive Bayes) sont appliquées pour détecter les intrusions dans les connexions réseau. Les performances de ces approches d'apprentissage profond surpassent les algorithmes d'apprentissage automatique traditionnels, offrant une précision élevée, un taux de détection idéal et un taux de fausses alarmes non négligeable.

Les performances des algorithmes d'apprentissage machine et d'apprentissage profond sur l'ensemble de données NSL-KDD surpassent celles de l'ensemble de données CIC_DDoS_2019.

Mots-clés : Système de Détection d'Intrusion (IDS), Apprentissage Profond (DL), Apprentissage Machine (ML), CIC_DDoS_2019, NSL-KDD.

Abstract

The field of information networks has become more vast and open due to technological advancements, leading to the emergence of new techniques for accessing networks and information systems. These techniques also expose the network to new cyber threats, leading to the development of network protection techniques, such as intrusion detection systems (IDS). To enhance the detection rates of attacks within IDS, artificial intelligence techniques are used.

This study proposes integrating IDS systems into computer networks using deep learning and automatic learning approaches. Three detection techniques based on deep learning approaches (DNN, CNN, RNN) and automatic learning approaches (Random Forest, KNN, and Naive Bayes) are applied to detect intrusions in network connections. The performance of these deep learning approaches surpasses traditional automatic learning algorithms, offering high precision, an ideal detection rate, and a non-ignorable false alarm rate.

The performance of machine learning and deep learning algorithms on the NSL-KDD dataset surpasses that of the CIC_DDoS_2019 dataset.

Keywords : Intrusion Detection System (IDS), Deep Learning (DL), Machine Learning (ML), CIC_DDoS_2019, NSL-KDD.

ملخص

إن مجال الشبكات المعلوماتية قد توسع بشكل كبير وانفتح بفضل التقدم التكنولوجي، مما أدى إلى ظهور تقنيات جديدة للوصول إلى الشبكات وأنظمة المعلومات. هذه التقنيات تعرض الشبكة أيضًا لتهديدات إلكترونية جديدة، مما أدى إلى تطوير تقنيات حماية الشبكات، مثل أنظمة كشف التسلل (IDS). ولتحسين معدلات كشف الهجمات داخل أنظمة IDS، تُستخدم تقنيات الذكاء الاصطناعي.

تقترح هذه الدراسة دمج أنظمة IDS في الشبكات المعلوماتية باستخدام منهجيات التعلم العميق و والتعلم الآلي. تم تطبيق ثلاث تقنيات كشف تعتمد على منهجيات التعلم العميق (DNN، CNN، RNN) والتعلم الآلي (KNN، Random Forest، Bayes Naive) للكشف عن التسلل في الاتصالات الشبكية. تتفوق أداءات هذه منهجيات للتعلم العميق على خوارزميات التعلم الآلي التقليدية، حيث توفر دقة عالية، ومعدل كشف مثالي، ومعدل إنذارات كاذبة غير مهمل.

تفوقت أداءات خوارزميات التعلم الآلي والتعلم العميق على مجموعة البيانات NSL-KDD مقارنة بمجموعة البيانات CIC_DDoS_2019.

الكلمات المفتاحية : نظام كشف التسلل (IDS)، التعلم العميق (DL)، التعلم الالي (ML)، CIC_DDoS_2019، NSL-KDD.

Table des Matières

Liste des Figures

Liste des Tableaux

Liste des Abréviations

IDS	Intrusion Détection System (Système de Détection d'Intrusion)
IDWG	Intrusion Détection exchange format Working Group
IETF	Internet Engineering Task Force
HIDS	Host-based IDS (IDS basé sur l'hôte)
NIDS	Network-based IDS (IDS basé sur le réseau)
WIDS	Wireless-based IDS (IDS basé sur le sans fil)
NBA	Network Behavior Analysis (Analyse du Comportement du Réseau)
TP	True Positive (Vrai Positif)
TN	True Negative (Vrai Négatif)
FP	False Positive (Faux Positif)
FN	False Negative (Faux Négatif)
IA	Intelligence Artificielle
ML	Machine Learning (Apprentissage Machine)
DL	Deep Learning
SVM	Support Vector Machines
KNN	K-Nearest Neighbors (k-plus proches voisins)
CNN	Convolutional Neural Networks (Réseaux Neuronaux Convolutifs)
DNN	Deep Neural Networks (Réseaux Neuronaux Profonds)
RNN	Recurrent Neural Networks (Réseaux Neuronaux Récursifs)
DBN	Deep Belief Networks (Réseaux de Croyances Profondes)
RBM	Restricted Boltzmann Machine (Machine de Boltzmann Restreinte)
TCP	Transmission Control Protocol (Protocole de Contrôle de Transmission)
UDP	Protocole de Datagramme Utilisateur (User Datagram Protocol)
FTP	File Transfer Protocol (Protocole de Transfert de Fichiers)

HTTPS	Hypertext Transfer Protocol Secure (Protocole de Transfert Hypertexte Sécurisé)
HTTP	Hypertext Transfer Protocol (Protocole de Transfert Hypertexte)
SSH	Secure Shell (Shell Sécurisé)
DDoS	Distributed Denial of Service (Déni de Service Distribué)
DoS	Denial of Service (Déni de Service)
R2L	Remote-to-Local (Accès à Distance vers Local)
U2R	User-to-Root (Utilisateur vers Administrateur)
SMOTE	Synthetic Minority Oversampling Technique

Introduction générale

Introduction générale

Grâce à la croissance des réseaux, notamment les réseaux internet, les technologies de l'information et de la communication nous offrent actuellement des avantages incontournables en ce qui concerne l'apprentissage à distance, les achats et les paiements en ligne, la communication via les messageries instantanées et les vidéoconférences, ainsi que d'autres technologies émergentes telles que les distributeurs automatiques, les véhicules autonomes, etc.Toutefois, de nouvelles vulnérabilités de sécurité ont été identifiées en raison de l'utilisation massive de ces outils informatiques.

De nos jours, les réseaux et les systèmes d'information connectés font face à de véritables menaces, qu'elles soient intentionnelles ou accidentelles. Il semble que chaque jour, une nouvelle histoire sur les défis de la cybersécurité soit publiée. Que ce soit le piratage de la vie privée sur les réseaux sociaux, les fraudes par carte de crédit, l'espionnage économique, l'infection de système informatique, les attaques de déni de service, et bien d'autres cybermenaces qui représentent des problèmes et des défis majeurs pour les années à venir. Ces cybers menaces de la cybersécurité sont considérés aujourd'hui comme l'une des principales préoccupations et un des domaine les plus actifs en recherche scientifiques .

La sécurité informatique englobe toutes les méthodes et les pratiques visant à garantir la protection de tous les aspects liés aux technologies de l'information, tels que l'accès aux données, le stockage, le traitement et la protection des données, ainsi que la transmission et la liaison). Elle est également appelée sécurité de l'information électronique.

L'invention du système de détection d'intrusion (IDS) répond à ces exigences. Son utilisation vise à prévenir toutes les risques potentiels de violation des politiques de sécurité, des réseaux et des systèmes informatiques. Le cœur de ce système est un module de détection des intrusions performant, solide et flexible. Il joue également un rôle essentiel dans tout produit de cybersécurité. Les modules de détection d'intrusion évaluent si un objet représente une menace ou une utilisation légitime en se basant sur les informations qu'ils ont collectées sur lui.

L'évolution des cyberattaques en complexité ,en volume et en fréquence, rend l'utilisation des méthodes classiques de détection moin pratique et obsolète ,alors de nouvelles technologies de protection avancées étaient nécessaires. De ce fait,les entreprises de cybersécurité sont tournées vers l'utilisation de l'apprentissage machine(ML), un domaine de l'intelligence artificiel (AI) qui avait été utilisé avec succès dans la reconnaissance d'image, la recherche et la prise de décision, afin de renforcer l'efficacité de leurs produits face à divers problèmes de détection des cyberattaques comme la détection des intrusion, la détection des logiciels malveillants, et surtout aux problèmes de sécurité liés aux infrastructures critiques comme la sécurité du système électrique, les systèmes de contrôle industriel, la détection des intrusions dans les systèmes SCADA . . .etc.

L'apprentissage profond (Deep Learning) qui fait partie de l'apprentissage machine est un domaine très prometteur pour la cybersécurité avec la disponibilité des grandes quantités de données des cyberspaces. Les méthodologies traditionnelles d'apprentissage machine reposent

sur l'ingénierie et la sélection des caractéristiques de domaine étudié, ce qui nécessite une bonne expertise afin d'accomplir ces taches.

Problématique

En raison de l'augmentation des données provenant des diverses infrastructures informatiques (réseaux, systèmes, internet des objets, etc.). Les systèmes de détection d'intrusion font face à de nombreux défis importants en raison de diverses formes de cyberattaques et d'utilisations légitimes de ces infrastructures. Comment garantir la performance de ces systèmes en ce qui concerne la détection et l'identification de toutes sortes d'activités malveillantes, tout en réduisant le taux de fausses alarmes. Comment garantir la robustesse et l'évolutivité des méthodologies de détection.

Motivation

La motivation principale de cette étude réside donc dans l'exploration et l'application des techniques d'apprentissage machine et d'apprentissage profond pour améliorer la détection des cybermenaces. En particulier, l'étude se concentrera sur l'analyse comparative des bases de données CIC_DDoS_2019 et NSL-KDD pour évaluer l'efficacité des modèles développés. Cette recherche vise à contribuer de manière significative à l'avancement des solutions de cybersécurité, en proposant des approches innovantes et efficaces pour protéger les infrastructures critiques et les systèmes d'information contre les cyberattaques.

Objectif et Structure du mémoire

Notre objectif dans cette étude est de mettre en œuvre diverses techniques de détection d'intrusions en utilisant les méthodes de Deep Learning et de Machine Learning, puis d'évaluer les performances des systèmes développés. Nous utilisons les bases de données CIC_DDoS_2019 et NSL-KDD, qui représentent un trafic réseau réel et incluent les types d'attaques malveillantes les plus courantes, telles que les attaques DoS et DDoS. Ces données permettent aux administrateurs de réseaux et de systèmes d'identifier et de détecter toute violation de la sécurité réseau. L'objectif final est d'améliorer les systèmes de détection d'intrusions.

Nous examinons dans cette étude les différentes méthodes d'apprentissage profond et d'apprentissage automatique qui ont déjà été utilisées dans le domaine de la détection des intrusions, en réponse aux nouveaux défis de la cybersécurité. La structure de notre mémoire est la suivante :

Le premier chapitre expose les principes de base de la sécurité informatique, puis aborde les systèmes de détection d'intrusion (IDS) en définissant ces derniers, en les modélisant et

les classifiant, et en explorant les différentes stratégies de détection ainsi que les mécanismes d'évaluation.

Le deuxième chapitre comprend L'apprentissage automatique(machine learning,ML) pour la détection d'intrusion, quelques méthodes de ML et ses principes de fonctionnement,et une synthèse des travaux reliés à la détection d'intrusion.

Le troisième chapitre présente Le système de détection d'intrusion par d'apprentissage profond,Quelques méthodes de DL et L'Architecturede chaque modèle,, ainsi que Base de données que nous avons utilisée dans ce travail

Le quatrième chapitre présente L'expérimentation , les résultats et les discussions de l'étude.

Chapitre 1

Sécurité informatique et Système de détection d'intrusion

1.1 Introduction

À l'ère numérique actuelle, la sécurité de l'information est de la plus haute importance, car les systèmes d'information sont aujourd'hui de plus en plus ouverts sur Internet. Cette ouverture, pose néanmoins un problème majeur : il en découle un nombre croissant d'attaques. La mise en place d'une politique de sécurité autour de ces systèmes est donc primordiale.

Dans ce chapitre, nous introduisons les principales notions de base de sécurité informatique, y compris sa définition, ses objectifs, les problèmes et les attaques informatiques, ainsi que les mécanismes permettant d'améliorer la sécurité. Ensuite, nous présentons les bases d'un système de détection d'intrusion (IDS) en commençant par la présentation de la notion de système de détection d'intrusion ainsi que son architecture, où nous allons citer les différents composants de ce système et la tâche de chacun. Nous présentons également la classification des IDS. Dans ce contexte, plusieurs critères sont pris en compte. Nous commençons par la classification selon la méthode d'analyse, qui découpe les IDS en deux approches (comportementale et par signatures). Enfin, les mesures d'évaluation de l'IDS seront abordées.

1.2 Sécurité informatique

La sécurité informatique est l'ensemble de technologies utilisées pour réduire les vulnérabilités du système d'information contre les attaques accidentelles ou intentionnelles. Le but est de protéger les systèmes d'information, qu'ils soient externes ou internes, afin de permettre au système de fonctionner normalement et d'assurer la disponibilité des services attendus [1].

Elle englobe toutes les mesures et pratiques visant à protéger les systèmes informatiques, les réseaux et les données contre les menaces et les attaques.

1.3 Objectifs de la sécurité informatique

Les systèmes d'information représentent l'ensemble des données d'une organisation d'une entreprise, les infrastructures matérielles et logicielles. Par rapport à la sécurité informatique, d'une manière générale, elle couvre l'ensemble des moyens, outils, techniques et méthodes pour garantir que ces ressources soient uniquement utilisées dans le cadre prévu : seules les personnes ou autres systèmes autorisés interviennent sur le système et ont accès aux données, sensibles ou non. La sécurité des systèmes d'informations vise donc à atteindre certains objectifs, dont les principaux sont les suivants[2] :

- **Disponibilité :** qui assure que les services du système sont accessibles à tout moment.
- **Confidentialité :** qui assure que les informations confidentielles des utilisateurs restent secrètes. Autrement dit, cela consiste à rendre l'information inintelligible à d'autres personnes que son propriétaire .
- **Intégrité :** qui assure que les données ne sont pas modifiées ou effacées par des utilisateurs non autorisés. Le système doit être capable de vérifier que les données n'ont pas été altérées durant une communication.
- **Non-répudiation :** qui assure que lorsqu'un utilisateur agit sur les services mis en place, le système est capable de garantir que l'utilisateur est bien celui qu'il prétend être .
- **L'authentification :** qui assure que lorsqu'un utilisateur agit sur les services, il ne lui est pas possible de nier d'avoir fait cette action.

1.4 Problèmes de la sécurité informatique

il existe trois problèmes qui affectent la sécurité informatique : les vulnérabilités, les menaces et les attaques.

- **Les vulnérabilité :** Ce sont des failles ou des faiblesses dans la spécification, conception, implémentation ou bien configuration des systèmes informatiques dont l'exploitation peut créer une intrusion[3].
- **Les menace :** Une menace c'est la possibilité d'une violation d'une propriété de la sécurité en exploitant un ou plusieurs vulnérabilités d'une façon intentionnelle ou accidentelle[4].
- **Les attaques :** Une attaque c'est une action malveillante qui tente d'exploiter une faiblesse dans le système et de violer un ou plusieurs besoins de sécurité[5].

1.5 Les différents types d'attaques informatique

1.5.1 Quelques attaques courantes

Il existe plusieurs attaques, pour cela nous avons presenté quelques attaques[6] :

- **Déni de service (DoS) / Déni de service distribué (DDoS) :** Les attaques par déni de service (DoS) et les attaques par déni de service distribué (DDoS) visent à submerger les

ressources d'un système, rendant ainsi le service inaccessible aux utilisateurs légitimes. Contrairement à d'autres attaques visant à obtenir un accès ou à augmenter un accès existant, les attaques DoS/DDoS ont pour seul objectif de perturber le fonctionnement du service ciblé. Elles peuvent également ouvrir la voie à d'autres types d'attaques en créant des vulnérabilités dans le système cible.

Les dénis de service (DoS) sont des attaques où un hacker lance seul une attaque contre une victime, souvent en cachant son identité. Les dénis de service distribués (DDoS) sont plus complexes, impliquant plusieurs agents contrôlés par des maîtres, chaque agent menant une attaque DoS simple la figure 1.1 [7].

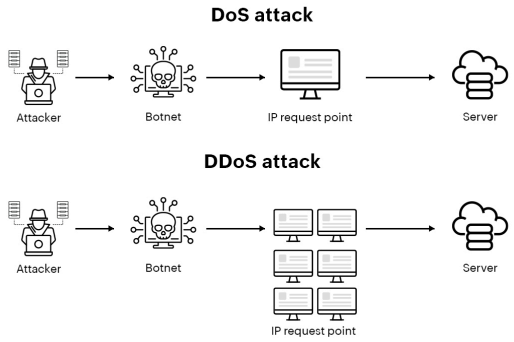


Figure 1.1 – Attaque par déni de service (DoS) et l' attaque par déni de service distribué (DDoS).

- **Attaque de l'homme au milieu (MitM) :** Les attaques de type Man-in-the-middle (MITM) permettent à un attaquant d'intercepter et d'écouter les données échangées entre deux parties, en se positionnant entre elles de manière invisible. Les deux parties pensent communiquer normalement, mais l'attaquant peut modifier ou accéder aux messages de manière illicite.

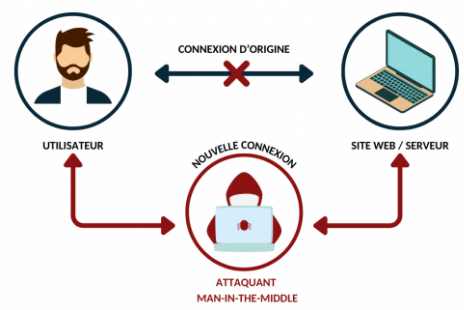


Figure 1.2 – Attaque de l'homme au milieu .

- **Attaque hameçonnage (phishing) et harponnage (spear phishing) :** L'hameçonnage consiste à envoyer des e-mails frauduleux provenant de sources apparemment fiables pour obtenir des informations personnelles ou inciter à des actions dangereuses. Le harponnage est une forme ciblée d'hameçonnage, où les attaquants créent des messages personnalisés pour tromper leurs cibles. Les techniques incluent l'usurpation d'adresses e-mail et le clonage de sites Web légitimes.

- **Attaque par mot de passe :** Cette technique consiste à essayer plusieurs mots de passe afin de trouver le bon. Elle peut s'effectuer à l'aide d'un dictionnaire des mots de passe les plus courants (et de leur variantes), ou par la méthode de brute force (toutes les combinaisons sont essayées jusqu'à trouver la bonne). Cette technique longue et fastidieuse, souvent peu utilisée à moins de bénéficier de l'appui d'un très grand nombre de machines [8].

1.6 Techniques et mécanismes de sécurisation

La variété et la disponibilité des outils d'attaques augmentent le risque des intrusions. Par conséquent les administrateurs s'appuient sur diverses solutions comme les antivirus, les pare-feux, les DMZ, les VPN..... dans le but de maintenir la protection du réseau informatique. Nous détaillons dans la suite chacune de ces méthodes.

- **Un antivirus :** Les antivirus sont des programmes qui permettent de détecter la présence de virus sur un ordinateur et de les supprimer. L'éradication d'un virus est le terme utilisé pour le nettoyage d'un ordinateur. Il y a plusieurs méthodes d'éradication :
  - Nettoyez le fichier infecté en supprimant le code malveillant.
  - Retrait du fichier infecté entièrement.
  - La mise en quarantaine le fichier infecté, qui consiste à le déplacer à un emplacement où il ne peut être exécuté [9].
- **Les Pare-feu (firewalls) :** Un pare-feu (Firewall) est un système physique ou logique qui inspecte les paquets entrants et sortants du réseau afin d'autoriser ou d'interdire leur passage en se basant sur un ensemble de règles appelées ACL. Il enregistre également les tentatives d'intrusions dans un journal transmis aux administrateurs du réseau et permet de contrôler l'accès aux applications et d'empêcher le détournement d'usage [10].

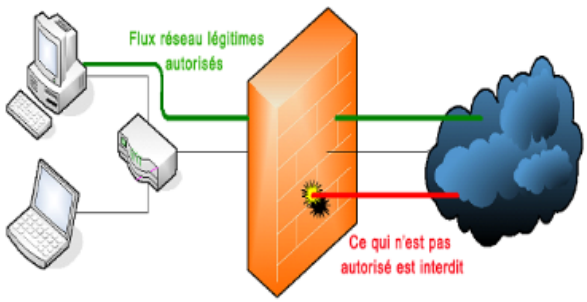


Figure 1.3 – L'emplacement de pare-feu.

- **DMZ (Demilitarized zone) :** Une DMZ (Demilitarized zone) est une zone tampon d'un réseau d'entreprise, située entre le réseau local et Internet, derrière le pare-feu. Il s'agit d'un réseau intermédiaire regroupant des serveurs publics (HTTP, DHCP, mails, DNS, etc.). Ces serveurs devront être accessibles depuis le réseau interne de l'entreprise et,

pour certains, depuis les réseaux externes. Le but est ainsi d'éviter toute connexion directe au réseau interne [11].

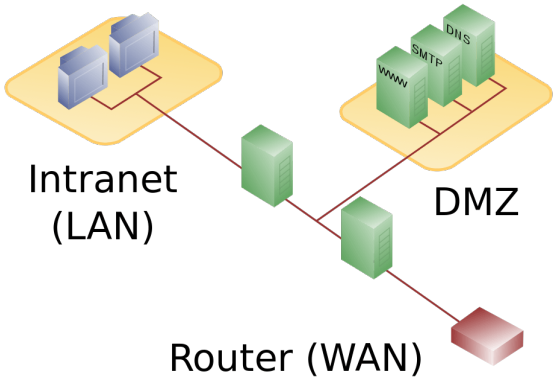


Figure 1.4 – la DMZ entre LAN et WAN.

- **VPN (Virtual Private Network) :** Dans les réseaux informatiques, le réseau privé virtuel (Virtual Private Network en anglais, abrégé en VPN) est une technique permettant aux postes distants de communiquer de manière sûre, tout en empruntant des infrastructures publiques (internet). Un VPN repose sur un protocole, appelé protocole de tunnelisation, c'est-à-dire un protocole permettant aux données passant d'une extrémité à l'autre du VPN d'être sécurisées par des algorithmes de cryptographie [12].
- **La détection d'intrusions :** La détection des intrusions est un mécanisme de cybersécurité essentiel dont le rôle est d'identifier les activités malveillantes au sein des environnements hôtes et/ou réseaux. En repérant ces activités nuisibles, il est possible de réagir rapidement pour, par exemple, interrompre une attaque en cours. En raison de l'importance cruciale de la détection des intrusions, les communautés de recherche et les industries ont élaboré et mis au point divers systèmes de détection des intrusions (IDS) [13].Nous allons focaliser notre étude sur le dernier mécanisme.

1.7 Système de détection d'intrusion

1.7.1 Notion d'un Système de Détection d'Intrusion IDS

Dans le domaine de sécurité de système d'information, Il existe tant des problèmes qui affectent et considéré comme souci nous avons les vulnérabilités, les menaces, les attaques et les intrusions.

La notion de l'intrusion informatique est définie comme toute tentative pouvant nuire à l'intégralité, la confidentialité ou la disponibilité dans le réseau ainsi que toute tentative visant à contourner les dispositifs de sécurité mis en place sur le réseau ou une machine[14].

Le terme est définie aussi, comme étant un accès intentionnel à un ordinateur sans autorisation ou en abusant des accès autorisés (United States (2010)). De façon plus générale, une intrusion correspond à la violation d'une politique de sécurité de l'organisme Bejtlich (2004))[15].

L'acte de l'intrusion est causé par des attaquants qui accèdent au système via l'Internet mais selon des motivations et des intentions diversifiées. Pour dévoiler toute intrusion pouvant nuire à la politique de sécurité de l'organisme, il existe plusieurs techniques et mécanismes de sécurité parmi eux nous avons la détection d'intrusion.

La détection des intrusions est le processus de surveillance des événements se trouvant dans un système des ordinateurs ou du réseau et les analysants pour détecter les signes des intrusions[16].

La détection d'intrusions consiste à analyser les informations collectées par les mécanismes d'audit de sécurité en utilisant un système qui effectue la détection d'intrusion d'une manière automatique, ce système est appelé « IDS : Intrusion Détection System »[17].

IDS signifie Intrusion Détection System. Il s'agit d'un équipement permettant de surveiller l'activité d'un réseau ou d'un hôte donné, afin de détecter toute tentative d'intrusion et éventuellement de réagir à cette tentative[8].

Un système de détection des intrusions est un logiciel ou un matériel qui automatise des surveillances et les processus analysés[16].

Ces systèmes de surveillance du réseau sont devenus pratiquement indispensables.

1.8 Modélisation des systèmes de détection des intrusions

D'après ce que nous avons mentionné précédemment concernant la définition de système de détection d'intrusion que ce système est un équipement permettant de surveiller l'activité d'un réseau ou d'un hôte donné. Cet équipement sont généralement de logiciel et éventuellement de matériel.

Ces derniers sont composés automatiquement des outils qui fonctionnent ensemble et chaque élément a sa propre tache mais pour aboutir le même objectif qui est «la détection d'intrusion» Pour décrire les déférents éléments constituants ce système, plusieurs schémas ont été proposés.

Parmi eux, nous avons retenu celui qui issu des travaux de l'Intrusion Détection exchange format Working Group (IDWG) de l'Internet Engineering Task Force (IETF)[18].

La figure suivante illustre le schéma fonctionnel du détecteur d'intrusion et les différents composants qui entrent en jeux au niveau de ce système voir figure 1.5.

Selon l'architecture IDWG, ce model générique contient des capteurs qui envoient des évènements à un analyseur. Un ou des capteurs couplés avec un analyseur forment une sonde. Une sonde envoie des alertes vers un manager qui la notifie à un opérateur humain.

- **L'administrateur :** Concrétisé par une personne chargée de mettre en place la politique de sécurité, par conséquent, de déployer et configurer les p les différents composants d'IDS : capteur(s), analyseur(s), manager(s) .
- **La source de données :** dispositif générant de l'information sur les activités des entités du système d'information.
- **Sonde :** Un équipement qui regroupe un ou des capteurs couplés avec un analyseur pour constituer une sonde .Elle permet de gérer la qualité des flux réseau.

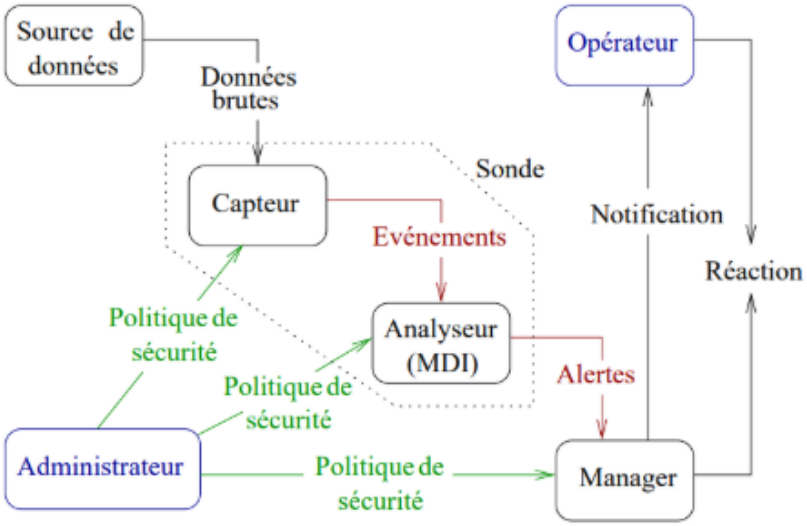


Figure 1.5 – Modèle générique de la détection d'intrusions proposé par l'IDWG [19].

a- Le capteur : C'est un logiciel qui collecte les évènements provenant de sources multiples(Le réseau, Le système, Les applications…. etc. Ce logiciel accède aux données brutesIl les filtre et les formate pour ne renvoyer que les événements intéressants à l'analyseur C'est-à-dire que le résultat de traitement est un message formaté appelé événement.

b -L'analyseur : c'est un outil logiciel qui permet d'analyser les événements engendrés par le capteur en détectant toute activité suspecte ou inhabituelle où il met en œuvre l'approche convenable pour la détection qu'elle soit comportementale ou scénarios, pour décider de la présence ou non d'une intrusion, il envoie une alerte au manager (qui notifie l'opérateur humain).

- **L'alerte :** C'est un message formaté émis par un analyseur au manager s'il détecte des activités intrusives.
- **Le Manager :** Parmi les composants d'un IDS qui permet de notifier et de gérer les alertes reçues par l'analyseur à l'opérateur humain. Eventuellement, le manager est chargé de la réaction. Cette dernière peut être menée automatiquement par le manager ou manuellement par l'opérateur.
- **La réaction :** mesures passives ou actives prises en réponse à la détection d'une attaque, pour la stopper ou pour corriger ses effets.

Dans ce modèle qui représente le processus complet de la détection ainsi que l'acheminement des données au sein d'un IDS. L'administrateur configure les différents composants (capteur(s), analyseurs(s), manager(s)) selon une politique de sécurité bien définie. Les capteurs accèdent aux données brutes, les filtrent et les formatent pour ne renvoyer que les événements intéressants à un analyseur. Les analyseurs utilisent ces événements pour décider de la présence ou non d'une intrusion et envoient dans le cas échéant une alerte au manager qui notifie l'opérateur humain, une réaction éventuelle peut être menée automatiquement par le manager ou

manuellement par l'opérateur[20].

1.9 Taxonomie des IDSs

Actuellement, il existe différentes types d'IDS qui se caractérisent par des multiples approches au niveau de la surveillance et de l'analyse. Les approches peuvent être décrites en termes d'un modelé d'IDS. La meilleure façon pour la classification des IDS est de les regrouper en se basant sur leur emplacement dans le système informatique (l'environnement de déploiement), les méthodes de détection, les types de réponses et les fréquences utilisées.

Nous pouvons classifier les systèmes de détection d'intrusions selon des critères voir figure 1.6 :

1.9.1 Les sources des données

- **Type de données :** Parmi les caractéristiques essentielles des systèmes de détection d'intrusions, les sources de données à analyser constituent la matière première du processus de détection. Ces données proviennent soit de logs (journaux) générés par le système d'exploitation sont appelés Host-based IDS (HIDS) , soit de logs des applications, soit d'informations provenant du réseau sont appelés Network-based IDS (NIDS) , un autre type d'IDS.
- **Collection de données :** Deux architectures de systèmes différentes :

  • *Centralisé :* Toutes les données sont collectées et analysées à partir d'un seul système surveillé.

  • *Distribué :* Les données sont collectées à partir de plusieurs systèmes surveillés.
- **L'outil de collection :** Le rôle des agents et des capteurs dans un système IDS est d'accéder aux données brutes, de les filtrer pour ne renvoyer que les informations pertinentes à un analyseur d'IDS.

1.9.2 La stratégie de détection

- **L'analyse et le traitement :** c'est l'architecture de traitement utilisée "centralisé" ou "distribués".
- **Discipline de détection :** Les outils de détection peut baser sur :

  •*Basée sur l'État :* vise à reconnaître si le système est dans un état sécurisé ou non sécurisé afin de détecter les intrusions. Cela implique de surveiller en continu l'état du système pour identifier tout changement qui pourrait indiquer une intrusion ou une menace potentielle.

  •*Basée sur les Transitions :*se concentre sur la reconnaissance de certaines transitions spécifiques qui conduisent à un état non sécurisé. Cela signifie identifier les changements ou

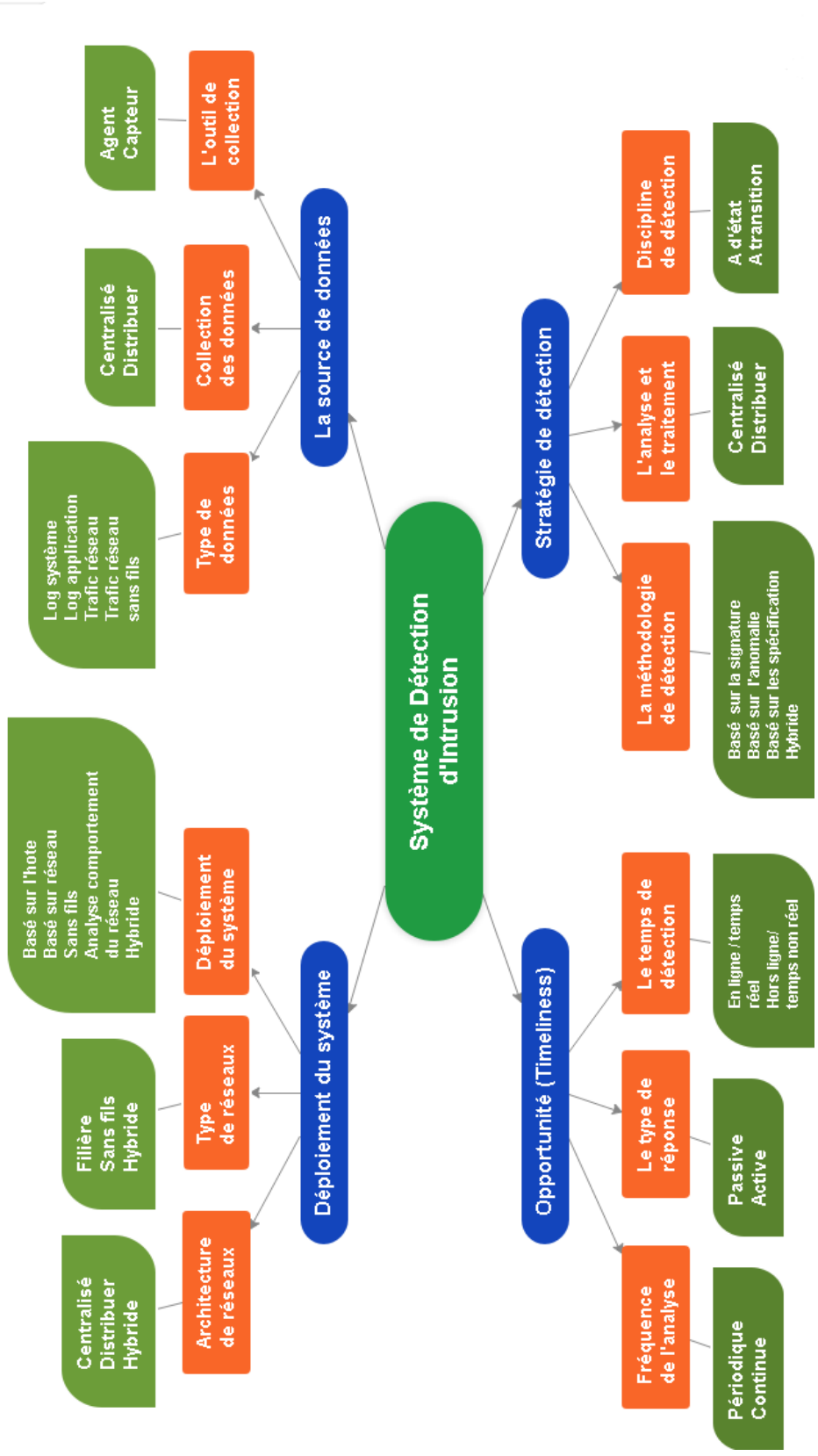


Figure 1.6 – Taxonomie des systèmes de détection d'intrusion [21].

les événements qui peuvent entraîner une dégradation de la sécurité du système, comme des tentatives d'accès non autorisées ou des activités suspectes qui pourraient mener à une vulnérabilité.

- **La Méthodologie de détection**

  •Détection d'signature (Signature based detection) : est le processus de comparaison des motifs avec les événements capturés pour reconnaître les intrusions possibles. En utilisant les connaissances accumulées par des attaques spécifiques et des vulnérabilités système.

  •Détection d'anomalies (Anomaly based détection) : Recherche des déviations par rapport aux comportements normaux. Utilise des profils de comportement pour identifier des activités anormales.

  •Détection basé sur les spécifications (Specification-based detection) : Trace les états des protocoles pour détecter les anomalies. Utilise des modèles de protocole spécifiques pour identifier les comportements non conformes.

  SD et AD sont des méthodes complémentaires, car le premier concerne certains types d'attaques/menaces (Faux alarme) et le second se concentre sur les attaques inconnues (n'existent pas dans la base de données), et SPA incapable de détecter les attaques qui se ressemblent à des utilisations bénins de protocole. Pour cela, un système IDS Hybride utilisent plusieurs méthodologies pour fournir une détection plus étendue et précise.

1.9.3 L'opportunité (Timeliness)

- **Le temps de détection :** C'est le temps entre l'événement analysé et la détection, elle peut être une détection en temps réel (On-line Detection), ou bien une détection en temps non-réel (Off-line Detection).

- **Le type de réponse :**

  • *Passive :* L'IDS n'applique aucune contre-mesure directe mais se contente de générer des alertes ou des notifications pour informer les administrateurs ou les responsables de la sécurité lorsqu'une attaque est détectée.

  • *Active :* L'IDS prend des mesures correctives ou préventives en réponse à une activité suspecte ou à une intrusion détectée. Ces mesures peuvent inclure le blocage des adresses IP.

- **La fréquence d'analyse :**

  •*Périodique :* Ce mode implique la sauvegarde d'une quantité de données pendant une période de temps définie, puis le démarrage des traitements de détection sur ces données sauvegardées. En d'autres termes, l'IDS collecte des données pendant une période donnée, par exemple toutes les heures ou tous les jours, puis effectue l'analyse et la détection sur ces données sauvegardées à des intervalles réguliers.

  •*Continu :* Dans ce mode, la détection des intrusions s'effectue de manière continue et en temps réel sur toutes les activités et événements qui se produisent. L'IDS surveille

en permanence les flux de données pour identifier les signes d'attaques dès qu'ils se produisent, sans interruption.

1.9.4 Déploiement du système

- **Architecture de réseaux :**Liés au nombre de systèmes IDSs et à la corrélation des données entre eux, il existe 3 Architectures :

  •*Centralisée :* Dans ce cas, le système collecte et analyse les informations à partir d'un seul système surveillé. Cela peut être efficace pour des environnements où la surveillance est concentrée sur un point spécifique.

  • *Distribuée :* Cette architecture collecte des données à partir de plusieurs systèmes surveillés. Elle est particulièrement utile pour détecter des attaques distribuées qui proviennent de plusieurs sources ou pour surveiller un réseau étendu.

  •*Hybride :* Cette approche combine à la fois des éléments centralisés et distribués. Elle permet de détecter une gamme plus large d'attaques, y compris les attaques distribuées et coopératives, tout en offrant une certaine concentration de la surveillance à des points clés.

- **Type de réseaux :** Dépend de l'interconnexion de l'IDS avec les systèmes de surveillance. Il peut être une connexion filière, sans fils, hybride.

- **Type de technologie :** L'adoption de plusieurs types de technologies IDS peut atteindre l'objectif d'une détection plus complète et plus précise.

  • *HIDS (Host-based IDS) :* surveille et collecte les caractéristiques des hôtes contenant des informations sensibles, des serveurs exécutant des services publics et des activités suspectes.

  •*NIDS (Network-based IDS) :* apture le trafic réseau à des segments spécifiques du réseau via des capteurs, puis analyse les activités des applications et des protocoles pour reconnaître les incidents suspects.

  • *WIDS (Wireless-based IDS) :* Similaire au NIDS, mais capture le trafic du réseau sans fil, tel que les réseaux ad hoc, les réseaux de capteurs sans fil et les réseaux maillés sans fil.

  • *NBA (Network Behavior Analysis) :* Inspecte le trafic réseau pour reconnaître les attaques avec des flux de trafic inattendus.

  •*Hybride (Mixed IDS) :* peut permettre d'atteindre l'objectif d'une détection plus complète et précise.

1.10 Les techniques de la détection d'intrusion

1.10.1 Les Méthodes de détection

Afin de détecter un intrus, nous devons employer un modèle de détection d'intrusion Nous avons l'IDS. Ce système doit pouvoir distinguer le comportement normal et anormal,dans le but de découvrir les tentatives malveillantes. Dans ce contexte, il existe deux grandes catégories de méthodes de détection les plus utilisables par le système de détection d'intrusion sont celles basées sur une approche comportementale (par exemple l'analyse statistique, l'analyse bayésienne, les réseaux neuronaux) et celles basées sur une approche par scénarios (par exemple la recherche de signatures, le pattern matching, ou la simulation de réseaux de Pétri).

- **La détection basée sur les signatures :** La « détection de signatures » compare la charge

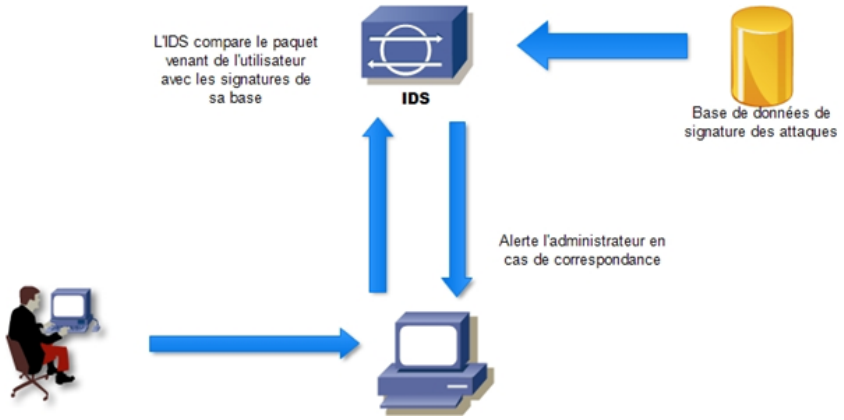


Figure 1.7 – Procédure de détection d'attaques d'un IDS à base de signature.

utile des messages à des signatures d'attaques enregistrées dans une base de données. Cette méthode ne détecte que des attaques connues et sa performance est conditionnée à une maintenance rigoureuse de la base sous peine d'obtenir beaucoup de faux négatifs. Elle souffre en outre d'un coût conséquent de stockage et de traitement, non compatible avec des systèmes contraints[22].

- **La détection basée sur les anomalies :**

  La « détection d'anomalies » compare le comportement du système à un modèle normal, le dépassement d'un seuil d'écart traduisant une attaque. Cette méthode est efficace pour détecter les nouvelles attaques, nombreuses dans l'IoT, y comprises les attaques zero-day Logiquement, cette méthode peut souffrir d'un nombre excessif de faux positifs. Dispensant de la synthèse complexe d'un modèle par un expert, les techniques d'apprentissage de l'intelligence artificielle sont souvent employées mais elles ne peuvent être hébergées que dans des nœuds avec assez de ressources. Des approches hybrides mêlant les deux méthodes existent dans la littérature[22].

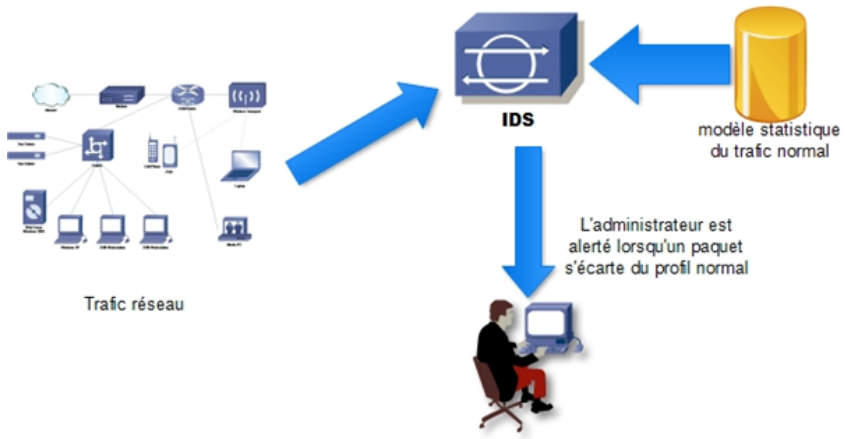


Figure 1.8 – Procédure de détection d'attaques d'un IDS à base d'anomalies.

1.10.2 Les mesures d'évaluation de l'IDS

Pour évaluer l'efficacité d'un système de détection d'intrusion, Porras et Valdes[23] ont proposé les trois paramètres suivants :

- **Précision :** La précision concerne la détection correcte des attaques et l'absence de fausses alertes.
- **Performance :** La performance d'un système de détection d'intrusion est le taux auquel les événements d'audit sont traités.
- **Exhaustivité (complétude) :** L'exhaustivité est la capacité d'un système de détection d'intrusion à détecter toutes les attaques.

De plus, introduisons deux propriétés supplémentaires :

- **Tolérance aux erreurs :** Un système de détection d'intrusion doit lui-même être résistant aux attaques, en particulier aux attaques de type déni de service, et doit être conçu dans cet objectif.
- **Rapidité :** Un système de détection d'intrusion doit effectuer et propager son analyse aussi rapidement que possible pour permettre à l'officier de sécurité de réagir avant que des dommages importants ne soient causés, et également pour empêcher l'attaquant de subvertir la source d'audit ou le système de détection d'intrusion lui-même [24].

En général, un taux de détection élevé est crucial pour les systèmes de détection d'intrusion (IDS) afin de prévenir les attaques avant qu'elles ne causent des violations de sécurité. Pour les IDS basés sur l'apprentissage automatique, une précision de détection élevée avec un faible taux de fausses alarmes est essentielle pour assurer l'efficacité de ces systèmes .[25].

Les principaux aspects à considérer lors de la mesure de détection et la précision de classification des attaques sont :

- True Positive (TP) : nombre d'intrusions correctement détectées.
- True Negative (TN) :nombre de non-intrusions correctement détectées.
- False Positive (FP) : nombre de non-intrusions mal détectées.
- False negative (FN) : nombre d'intrusions mal détectées.

Il existe plusieurs types d'erreurs venant d'un détecteur, influençant plus ou moins sa puissance. Les vrais positifs sont les cas où une alarme se déclenche quand il y a une violation des politiques de sécurité. Les vrais négatifs sont les cas où aucune alarme ne se déclenche et rien d'anormal ne se produit. Les faux positifs sont les cas où une alarme se déclenche alors qu'il ne se produit rien d'anormal. Les faux négatifs sont les cas où une alarme ne se déclenche pas alors qu'il se produit une chose anormale. À la première vue,on pourrait supposer qu'un faux positif est moins dangereux qu'un faux négatif [26].

1.11 Conclusion

Dans un monde où le progrès technologique avance à une vitesse fulgurante, où les individus, les entreprises, les organisations, les pays et même les objets sont de plus en plus interconnectés, les cyberattaques se multiplient. La question de la cybersécurité se pose à tous les niveaux et tend à devenir un enjeu crucial dans les années à venir.

Dans ce chapitre, nous avons abordé diverses notions relatives à la sécurité informatique, en présentant les différents types d'attaques, ainsi que les techniques employées pour protéger les systèmes contre ces menaces. Parmi ces mécanismes, nous avons particulièrement détaillé les systèmes de détection d'intrusions, qui représentent l'objectif principal de ce mémoire. Ces systèmes continuent d'évoluer pour répondre aux exigences et offrir un éventail de fonctionnalités capables de satisfaire les besoins de tous les types d'utilisateurs.

Nous avons donc exploré l'architecture des systèmes de détection d'intrusions, leur principe de fonctionnement et les différentes approches pour la détection d'intrusions, en divisant les IDS en deux grandes catégories : les IDS basés sur le comportement et les IDS basés sur les signatures. Les IDS basés sur les signatures détectent les attaques en se fondant sur des signatures préalablement enregistrées, ce qui nécessite une mise à jour périodique de la base de signatures et rend la détection des nouvelles attaques impossible.

C'est pourquoi, dans ce travail, nous avons privilégié l'approche comportementale, qui offre la possibilité de détecter les attaques inconnues en s'appuyant sur qui seront détaillés dans les prochaines chapitres.

Chapitre 2

Etat de l'art« Apprentissage automatique et systèmes de détection d'intrusion »

2.1 Introduction

L'évolution rapide des technologies de l'information a apporté d'innombrables avantages à notre société, mais elle a également ouvert la porte à de nouvelles formes de menaces. Parmi celles-ci, les attaques DDoS (Distributed Denial of Service) figurent parmi les plus préoccupantes pour les entreprises et les organisations en ligne. Ces attaques, conçues pour submerger les serveurs ciblés avec un trafic malveillant, peuvent entraîner des temps d'arrêt coûteux, des pertes de données et une diminution de la confiance des utilisateurs. Face à cette menace croissante, les professionnels de la sécurité informatique recherchent constamment des moyens plus efficaces de détecter et de contrer les attaques DDoS. L'intelligence artificielle (IA) émerge comme une solution prometteuse dans cette lutte, offrant des capacités avancées pour analyser le trafic réseau, identifier les anomalies et réagir rapidement aux attaques en temps réel. Dans ce chapitre, nous avons étudié et analysé divers travaux de détection des attaques DDoS basés sur l'apprentissage profond .

2.2 L'intelligence artificielle

L'intelligence artificielle (IA) représente un domaine de l'informatique dédié à créer des systèmes capables de reproduire le fonctionnement du cerveau humain dans une certaine mesure, notamment en ce qui concerne la prise de décisions. Elle englobe un large éventail de techniques et de méthodes visant à doter les machines d'une forme d'intelligence semblable à celle des êtres humains. Parmi les différents sous-domaines de l'IA, le Machine Learning (ML) et le Deep Learning (DL)figure 2.1.

Les systèmes basés sur les règles métier utilisent des ensembles de règles logiques pour prendre des décisions, souvent définies par des experts humains dans des domaines spécifiques. Le Machine Learning, quant à lui, implique le développement d'algorithmes capables d'ap-

prendre à partir de données, permettant aux machines de reconnaître des schémas et de prendre des décisions autonomes sans être explicitement programmées. Enfin, le Deep Learning est une branche du Machine Learning qui utilise des réseaux de neurones artificiels à plusieurs couches pour réaliser des tâches d'apprentissage automatique complexes en capturant des représentations hiérarchiques des données.

L'IA offre des perspectives prometteuses dans de nombreux domaines, tels que la santé, la finance, l'industrie, et bien d'autres, en permettant aux machines de réaliser des tâches qui étaient auparavant réservées aux capacités humaines, voire de les surpasser dans certains cas.

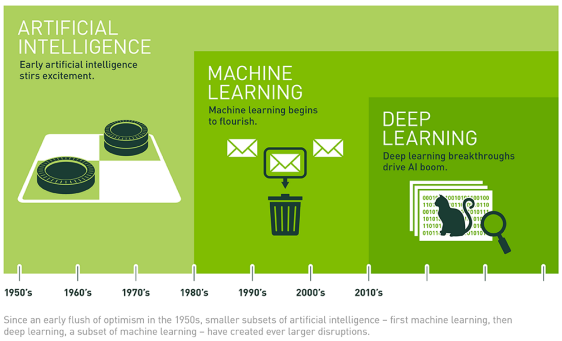


Figure 2.1 – Relation entre IA, ML et DL.

2.3 Les enjeux de l'IA dans la cybersécurité

Les progrès dans le domaine de la cybersécurité ont été considérables grâce à l'utilisation croissante de l'intelligence artificielle (IA), notamment les techniques d'apprentissage automatique et d'apprentissage profond, ainsi que l'exploration de données et les statistiques. Ces méthodes interdisciplinaires permettent de relever les défis de la cybersécurité en exploitant les vastes quantités de données provenant des réseaux, des systèmes d'exploitation et des systèmes d'information[27]. L'apprentissage automatique et l'apprentissage profond peuvent être appliqués efficacement dans les réseaux SDN pour la détection d'anomalies. Ces méthodes permettent de classifier, détecter et prévenir les cyberattaques en identifiant des modèles et des comportements typiques associés à différentes formes d'attaques. Grâce à leur capacité à détecter les attaques en temps réel et à prédire les attaques futures, ces techniques offrent une réponse proactive aux menaces. Les méthodes basées sur l'apprentissage en profondeur sont particulièrement prometteuses pour améliorer les systèmes de détection d'intrusion (IDS), car elles permettent de relever les défis associés à leur développement. Ces approches sont capables de gérer efficacement les mégadonnées générées par la collecte de données et le trafic réseau, en offrant des performances accrues en termes de détection d'anomalies tout en minimisant les taux de fausses alarmes. En résumé, l'utilisation de l'apprentissage automatique et de l'apprentissage en profondeur dans le domaine de la cybersécurité représente une avancée significative,

permettant une détection proactive des menaces et une meilleure protection des systèmes informatiques contre les cyberattaques.

2.4 Apprentissage automatique

2.4.1 Définition de l'apprentissage automatique

L'apprentissage automatique (ou apprentissage machine, machine learning) est un sous-domaine de l'IA qui s'intéresse en particulier aux capacités d'apprentissage. Le principe est de reproduire un comportement non pas en le programmant "à la main" dans un ordinateur, mais en concevant un système plus général capable d'apprendre à partir d'exemples à résoudre votre problème.

2.4.2 Les types de l'apprentissage automatique

- **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 [28]. Les algorithmes de classification supervisée de ML sont : Forest Random, Decision Trees, Logistic Regression, et le plus connu est SVM Support Vector Machines.
- **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[29].

2.4.3 Quelques algorithmes d'apprentissage automatique

- **k-nearest neighbor :** L'algorithme des k-nearest neighbor(KNN) est un des algorithmes de classification les plus simples. Le seul outil dont on a besoin est une distance entre les éléments que l'on veut classifier[30]. Chaque observation de l'ensemble d'apprentissage est représentée par un point dans un espace à n dimensions ou n est le nombre de variables prédictives. Pour prédire la classe d'une observation, on cherche les k points les plus proches de cet exemple. La classe de la variable cible, est celle qui est la plus représentée parmi les k plus proches voisins. Il existe des variantes de l'algorithme ou on pon-

dère les k observations en fonction de leur distance à l'exemple dont on veut classer[31], les observations les plus éloignées de notre exemple seront considérées comme moins importantes.

- **Support vector machines :**La recherche humaine sur les techniques d'apprentissage se concentre sur les machines à vecteur support. Les SVMs, développées par Vladimir Vapnik au début des années 90, sont un ensemble de méthodes d'apprentissage supervisé qui sont utilisées pour résoudre des problèmes de classification.Les classifiées linéaires sont généralisées en utilisant des SVM, qui sont basées sur une théorie mathématique. SVM fonctionne par mappage des données à un espace d'attributs haute dimension pour que les points de données puissent être classés, même lorsque les données ne sont pas séparables sur un plan linéaire. Un séparateur entre les catégories est identifié figure 2.3. Ensuite, les données sont transformées de sorte que le séparateur puisse être défini comme un hyperplan. Ensuite, les caractéristiques des nouvelles données peuvent être utilisées pour prédire le groupe auquel un nouvel enregistrement doit appartenir [32].
- **Bayesian classification :** La technique de classification bayésienne (Naive Bayes) repose sur le théorème de Bayes et suppose une indépendance entre les prédicteurs. Cette théorie repose sur le principe suivant : Assurez-vous que X est un échantillon de données dont la classe est inconnue et que vous souhaitez la déterminer, et que H est une hypothèse (par exemple, X appartient à la classe C). L'objectif est de calculer P(H/X) la probabilité de vérification de H après observation de X. La probabilité P(H/X) correspond à la probabilité postérieure, c'est-à-dire après avoir appris X, tandis que la probabilité à priori correspond à la probabilité de vérifier H pour n'importe quel exemple de données.Le théorème de Bayes propose une méthode de calcul de P(H/X) en utilisant les probabilités P(H), P(X) et P(X/H) :

$$P(H\,|\,X) = \frac{P(X\,|\,H)\cdot P(H)}{P(X)} \tag{2.1}$$

  P(H/X) est donc la probabilité d'appartenance de X à la classe C, P(H) la probabilité d'apparition de la classe C dans la population et qui peut être calculée comme le rapport entre le nombre d'échantillons appartenant à la classe C et le nombre total d'échantillons[30].
- **k-means :**L'algorithme k-means est l'algorithme de regroupement le plus connu et le plus utilisé, du fait de sa simplicité de mise en œuvre. Il partitionne les données d'une image en K clusters. Contrairement à d'autres méthodes dites hiérarchiques, qui créent une structure en « arbre de clusters » pour décrire les groupements, k-means ne crée qu'un seul niveau de clusters. L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde. Le k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster.

  La position initiale des centroïdes conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser

l'algorithme. K-means change les objets de cluster jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre de clusters. Les principales étapes de l'algorithme k-means sont :

- Choix aléatoire de la position initiale des K clusters.
- Réaffecter les objets à un cluster selon un critère de minimisation des distances,(généralement basé sur la distance euclidienne).
- Une fois que tous les objets sont placés, recalculer les K centroïdes.
- Répéter les étapes 2 et 3 jusqu'à ce qu'aucun réarrangement supplémentaire ne soit nécessaire.

2.5 Apprentissage profond

2.5.1 Définition de l'apprentissage profond

L'apprentissage profonde (Deep learning ou DL) appartient à une classe de techniques d'apprentissage automatique (machine learning ou ML), il obtient un grand succès dans de nombreuses tâches de l'intelligence artificielle (IA) par rapport aux algorithmes de ML classiques. Les architectures des modèles profondes sont relativement récentes où de nombreuses étapes de traitement non linéaire de l'information sont exploitées, dans lesquelles les informations sont traitées en couches hiérarchiques, chacune recevant et interprétant les informations de la couche précédente pour l'apprentissage des représentations de données[33]

2.5.2 Fonctionnement

Généralement, l'architecture des réseaux profondes est organisée en couches de neurones pour n'importe quel type de ces réseaux ; une Couche d'entrée (Input Layer), une ou plusieurs Couches cachées (Hidden Layers) et une Couche de sortie (Output Layer).

Chaque paire de couches voisines est connectée. Les connexions entre eux appelées poids (Weights). Les "neurones" d'une même couche généralement appelés "nœuds" n'ont aucune association, la figure 2.2 illustré une architecture standard d'un modèle de réseau de neurones profond.

L'apprentissage profond se présente comme un système de calcul avancé, il est constitué d'une variété de techniques issues du domaine de l'apprentissage automatique qui utilisent un déluge de neurones (nœuds) non linéaires disposés en plusieurs couches de traitement qui extraient et convertissent des valeurs de variables d'entité à partir du vecteur d'entrée pour créer plusieurs niveaux d'abstraction afin de représenter les données[34].

L'apprentissage du DNN c'est l'optimisation des paramètres de poids et le paramètre de biais entre deux couches voisines, Il évalue la justesse du modèle et permet de mieux l'adapter aux données d'apprentissage ses besoins. Lorsque le modèle arrive à une précision maximale

avec des paramètres optimaux, il sera généralisé pour les données réelles. La quantité et la qualité des données d'entraînement déterminent le degré d'apprentissage et donc la précision des modèles obtenus.

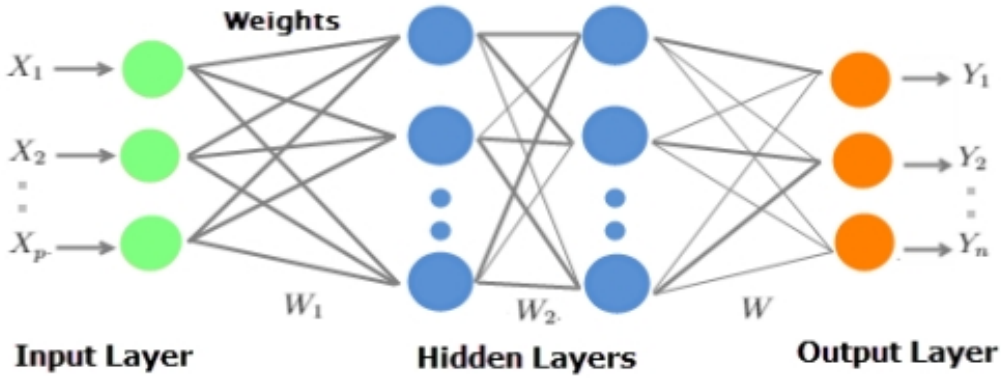


Figure 2.2 – L'architecture d'un modèle Deep Learning
[35].

2.5.3 Classification des méthodes DL

En pratique, toutes les approches d'apprentissage profond sont des réseaux de neurones (Neural networks), qui partagent certains propriétés de base communes. Ils sont tous constitués de neurones inter-connectés, ils sont organisés en couches. Ce qui les différencient, c'est l'architecture du réseau (où la manière dont les neurones sont organisés dans le réseau) et parfois la manière dont ils sont formés ont présenté une étude et analysé 10 différents approches du Deep learing les plus utilisées pour la détection des intrusions dans la cybersécurité. Ces approches peuvent être classées en trois modèles, en fonction de la manière dont elles sont formées et destinées à être utiliser.

• Deep learning pour l'apprentissage supervisé : Il est utilisé lorsque les données d'étiquette cible sont disponibles, il s'agit des modèles profonds discriminatoires à sa savoir le Deep neural networks (DNNs), Recurrent neural networks (RNNs), Convolutional neural networks (CNNs).
.• Deep learning pour l'apprentissage non-supervisé : Il est Utilisé lorsque les données d'entrée ne sont pas étiquetées ,il s'agit des modèles génératifs visent à regrouper les données selon certains critères de similarité à des fins de reconnaissance ou de synthèse de modèles, à savoir le deep belief networks (DBN), deep autoencoders (DA), Restricted Boltzmann machine (RBM) et deep Boltzmann machines (DBM).

.• Deep learning hybrides : une combinaison hybride de ces modèles mentionnés ci-dessus. Les réseaux profonds non supervisés pourraient fournir une excellente initialisation sur la base pour laquelle la discrimination (l'apprentissage supervisé) pourrait être examinée.

2.6 L'importance de l'apprentissage profond

L'apprentissage automatique n'est pas utile lorsque vos travaillez avec des données de grandes dimensions, c'est-a-dire que nous avons un grand nombre d'entrée et de sorties. Ne pas résoudre des problèmes cruciaux d'intelligence artificielle comme NLP, la reconnaissance d'image etc. Extraction de caractéristiques est un des grands défis des modèles d'apprentissage machine traditionnels Cette extraction automatisée de caractéristiques permet aux modèles de Deep learning d'atteindre un taux de précision particulièrement élevé pour la tache de vision par ordinateur (la figure 2.3). Les modèles d'apprentissage profond son capables de se concentrer sur les fonctionnalités appropriées par eux-mêmes nécessitant peu de conseils de part du programmeur.

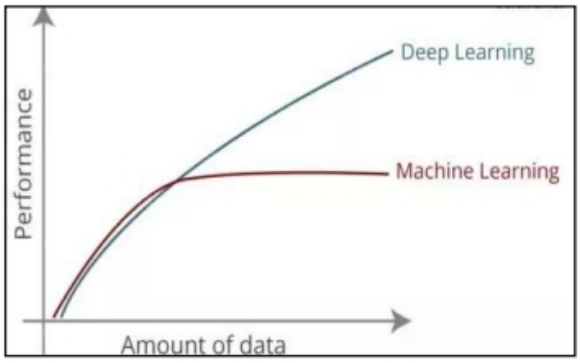


Figure 2.3 – Illustration de performance de l'apprentissage profond et l'apprentissage automatique .

2.7 Travaux connexes pour la détection d'intrusion

Plusieurs articles récents se penchent sur l'efficacité des systèmes de détection d'intrusions (IDS) en utilisant des approches d'apprentissage automatique avancées. **Farage et ses collègues (2021)** ont présenté trois modèles IDS basés sur l'apprentissage profond, chacun exploitant des architectures neuronales spécifiques. Leur étude, qui s'appuie sur les ensembles de données CIC-DDoS2019 et TON-IoT, démontre une nette amélioration des performances par rapport aux méthodes classiques, avec le modèle basé sur CNN en tête.

Dans une autre contribution majeure, **Sharafaldin et al(2019)** ont introduit une nouvelle taxonomie des attaques DDoS pour la couche applicative et ont lancé l'ensemble de données CICDDoS2019. Leur analyse approfondie des attaques DDoS et des algorithmes d'apprentissage automatique a abouti à une précision significative de f1=69%.

Un autre groupe de chercheurs, **Sandee et al (2019)**, a proposé un système NIDS combinant un auto-encodeur sparse pour l'apprentissage des caractéristiques non supervisées et une régression logistique pour la classification binaire. Leur approche, évaluée sur l'ensemble de données NSL-KDD, a obtenu une précision globale de 87,2%.

Enfin, **GAO et ses collègues (2014)** ont utilisé un réseau de croyance profonde (DBN) pour la détection des intrusions, surpassant les méthodes traditionnelles telles que le SVM et l'ANN sur l'ensemble de données KDD CUP 99. Leur modèle, basé sur des réseaux d'apprentissage mutli-couches non supervisés et supervisés, a démontré une performance supérieure dans la détection des intrusions.

Travail	Techniques	Dataset	Mesures de performances
Farage et al. [28]	CNN, DNN, RNN	TON_IoT, CICD-DoS2019	CNN 95%, DNN 94%, RNN 94%
Sharafaldin et al. [29]	Id3, RF, Naive Bayes, Logistic regression	CICDDoS2019	F1 score : 0.69
Sandee et al. [30]	Deep auto-encoder	NSL-KDD	ACC=99.99% PR = 84.6%, 92.8% spe = 80.7%
GAO et al. [31]	DBN	KDD'99	ACC=74.22% - 93.49% DR = 75.6% - 92.33% FAR = 3.15% - 0.76%

Table 2.1 – Travaux antérieurs connexes pour la détection d'intrusion

2.8 Ensemble de données d'évaluation de détection d'intrusion

L'évaluation de tous les algorithmes d'apprentissage profond recommandés repose sur les ensembles de données utilisés dans les recherches publiées sur l'application de l'apprentissage profond à la cybersécurité. Cependant, certains de ces ensembles de données ne sont pas accessibles gratuitement en raison de préoccupations liées à la confidentialité. Voici des exemples d'ensembles de données dédiés à la détection de différentes attaques :

2.8.1 KDD 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).

2.8.2 NSL-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.

2.8.3 MAWI

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.

2.8.4 ISCX

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.

2.8.5 CICIDS2017

Il s'agit d'un ensemble de données public accessible gratuitement à [36] . 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 (FridayWorkingHours-Morning.pcap.ISCX) et l'avons testé dans notre proposition maquette. Ce fichier comprend 191033 trafic bénin et 1966 trafic Botnet.

2.8.6 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.

Dataset	**Type**	**Étiqueté**	**Nombre de classes**	**Année**
KDD Cup'99[37]	Trafic du réseau	Oui	5 classes (normal, DoS, R2L, U2R, Probe)	1999
NSL-KDD[38]	Trafic du réseau	Oui	5 classes (normal, DoS, R2L, U2R, Probe)	2009
MAWI[39]	Trafic internet	Oui	Multi classe	2011
ISCXIDS[40]	Trafic du réseau	Oui	Multi classe	2012
CICIDS2017[41]	Trafic du réseau	Oui	7 classes (dos, portscan, Bot, bruteforce, webattack, infiltration)	2017
CICDDOS2019[42]	Trafic du réseau	Oui	14 classes ()	2019

Table 2.2 – Collecte de données publiées sur la cybersécurité.

2.9 Les mesures d'évaluation des modèles

• **Accuracy (Acc) :** est la proportion du nombre total de prédictions correctes. Elle est donnée par la relation suivante.

$$Ac = \frac{TP + TN}{TP + TN + FP + FN}$$

• **Précision (P r) :** le pourcentage des attaques DDoS identifiées comme des attaques T P parmi tous les exemples pré dictés comme attaque, il est donné par :

$$P_r = \frac{T_{PAttack}}{T_{PAttack} + F_{PBENIGN}}$$

• **Recall (Rc) :** le pourcentage des attaques DDoS identifiées comme des attaques T P parmet tous les attaques dans l'ensemble de données :

$$R_c = \frac{T_{PAttack}}{T_{PAttack} + F_{NAttack}}$$

• **F1-score (F1) :** la moyenne harmonique pondérée de précision et de rappel (Recall), il est donné par :

$$F1 = \frac{2 \cdot (P_r \cdot R_c)}{P_r + R_c}$$

• **Matrice de confusion :** Est une disposition de tableau spécifique permettant de visualiser les

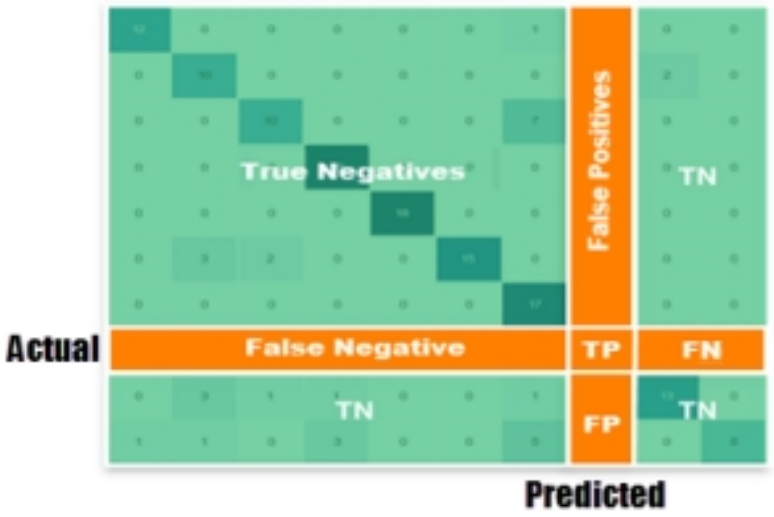


Figure 2.4 – Illustration d'une matrice de confusion.

performances d'un algorithme ML pour un problème de classification, elle est connue sous le nom de matrice d'erreur.

2.10 Conclusion

La détection des attaques DDoS représente un défi majeur pour les professionnels de la cybersécurité, et l'utilisation de l'intelligence artificielle offre des solutions prometteuses pour faire face à cette menace croissante. Au fil de ce chapitre, nous avons exploré les avancées les plus récentes dans ce domaine, mettant en évidence les techniques d'apprentissage automatique et d'apprentissage profond qui révolutionnent la façon dont les attaques DDoS sont détectées et contrées.

L'intégration de l'intelligence artificielle dans les systèmes de détection des attaques DDoS permet une analyse plus rapide et plus précise du trafic réseau, permettant ainsi aux entreprises et aux organisations de réagir plus efficacement aux attaques en temps réel. Les modèles d'apprentissage automatique et d'apprentissage profond, tels que les réseaux de neurones convolutionnels (CNN) et les réseaux de neurones récurrents (RNN), sont capables de détecter les anomalies dans le trafic, même lorsque celles-ci sont subtiles ou complexes.

Cependant, malgré les progrès réalisés, des défis persistent dans le domaine de la détection des attaques DDoS. L'obtention de données étiquetées pour l'apprentissage supervisé reste un défi, tout comme la nécessité de gérer les fausses alertes et de minimiser les temps d'arrêt des services légitimes pendant les attaques.

Pour surmonter ces défis, il est crucial de poursuivre la recherche et le développement dans le domaine de l'intelligence artificielle pour la cybersécurité. Les collaborations interdisciplinaires entre les chercheurs en sécurité informatique, les experts en apprentissage automatique et les praticiens de l'industrie sont essentielles pour créer des solutions robustes et innovantes.

En conclusion, l'intelligence artificielle offre un potentiel considérable pour améliorer la détection et la mitigation des attaques DDoS. En tirant parti de ces avancées technologiques, nous pouvons renforcer la résilience de nos systèmes informatiques et assurer la disponibilité et la sécurité des services en ligne dans un monde numérique en constante évolution.

Chapitre 3

Système de détection d'intrusion par d'apprentissage profond

3.1 Introduction

Dans cette chapiter explore un éventail crucial de sujets liés à l'apprentissage profond, en mettant en lumière des méthodes telles que les réseaux de neurones convolutionnels (CNN), les réseaux de neurones récurrents (RNN), les deep neural networks (DNN), et les unités de mémoire à court terme (LSTM). En outre, il plonge dans des aspects pratiques indispensables, notamment l'analyse du dataset utilisé, la classification des attaques DDoS, la préparation des données qui englobe la réduction des données, la résolution des problèmes d'étiquetage, les pré-traitements des données, la normalisation, ainsi que l'utilisation stratégique de SMOTE pour équilibrer les données.

Le point central de ce chapitre réside dans la conception et l'implémentation d'un système de détection d'intrusion dédié aux attaques DDoS, s'appuyant sur une diversité de modèles de deep learning tels que les CNN, les DNN, et les RNN_LSTM. Un schéma conceptuel détaillé accompagne cette présentation, illustrant l'approche méthodique adoptée pour mettre en œuvre ces modèles dans le domaine de la sécurité des réseaux.

3.2 Apprentissage profond

Dans le domaine de l'apprentissage automatique, les méthodes d'apprentissage profond ont transformé la manière dont les machines analysent et traitent les données. Parmi ces méthodes, les Réseaux de Neurones Convolutifs (CNN), les Réseaux de Neurones Profonds (DNN) et les Réseaux de Neurones Récursifs (RNN) sont particulièrement remarquables. Chacune de ces approches offre des avantages spécifiques en fonction du type de données et de la nature de la tâche à accomplir.

3.3 Quelques méthodes d'apprentissage profond

3.3.1 Réseaux de neurones Convolutionnels (CNN)

Un réseau neuronal convolutionnel ou CNN est une extension des réseaux de feed forward traditionnels (FFN) dans le cadre de l'inspiration des facteurs biologiques [43]. Ceux-ci ont été initialement étudiés pour le traitement d'images dans lesquelles des motifs répétitifs peuvent être trouvés - par exemple, une image avec des bords répétitifs et d'autres motifs. Les CNNs surpassent tous les autres algorithmes ML classiques et fait un grand succès dans les tâches de traitement de vision par ordinateur (Computer Vision Tasks), ils ont des larges applications dans le traitement d'image et vidéo,le traitement du langage naturel (NLP), les systèmes de recommandation . . .etc. Les réseaux convolutifs sont particulièrement efficaces grâce à plusieurs types de couches spéciales : des couches de convolution, des couches groupement (Pooling) et de couches entièrement connectées [19], la figure 3.1 illustre un modèle d'un réseau conventionnel unidimensionnel (1D CNN).

Convolution Layers : L'objectif de la convolution est d'extraire les caractéristiques de haut niveau. Il est constitué d'un ensemble de filtres (ou noyaux) apprenants, chacun représente une certaine fonctionnalité indépendante avec le volume d'entrée. Ces filtres sont constitué d'une couche de poids de connexion, ils ont un petit champ de réception (la taille du noyau), mais lors de la passe en avant (feed forward), chaque filtre est convolé sur la largeur et la hauteur du volume d'entrée, calculant le produit des points entre les entrées et les valeurs du filtre produisant une nouvelle carte de caractéristiques qui représente mieux l'information. En conséquence, le réseau apprend les filtres qui s'activent lorsqu'il détecte un type de caractéristique importante et spécifique à une certaine position spatiale dans l'entrée. La figure 3.2 présente une opération de convolution 1D avec une entré de dimension 1.

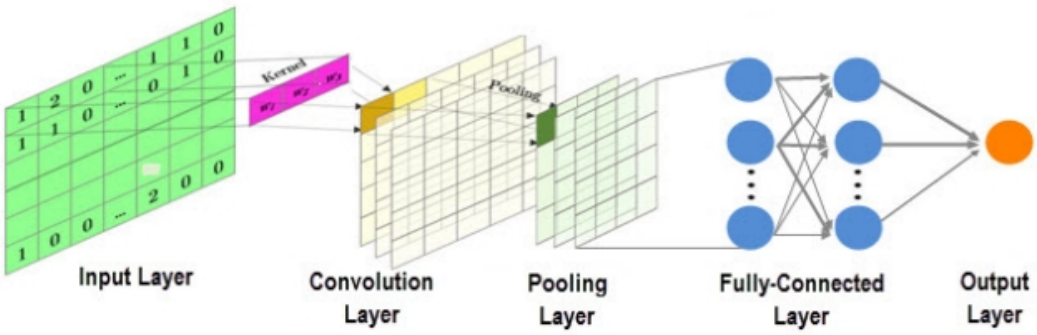


Figure 3.1 – L'architecture d'un modèle de réseau neuronal convolutif.

Une couche convolutionnelle partage le même noyau de convolution, ce qui réduit considérablement le nombre de paramètres nécessaires pour l'opération de convolution. Une fonction d'activation non linéaire sera appliqué immédiatement après chaque couche convolutionnelle. Les CNN profonds avec la fonction d'activation "Rectified Linear Units **ReLU"** $f(x) =$

$\max(0, x)$, renvoie x pour toutes les valeurs de x > 0 et renvoie 0 pour toutes les valeurs de x <= 0. s'entraînent plusieurs fois plus vite que leurs équivalents avec unités "Tanh Units"[44].

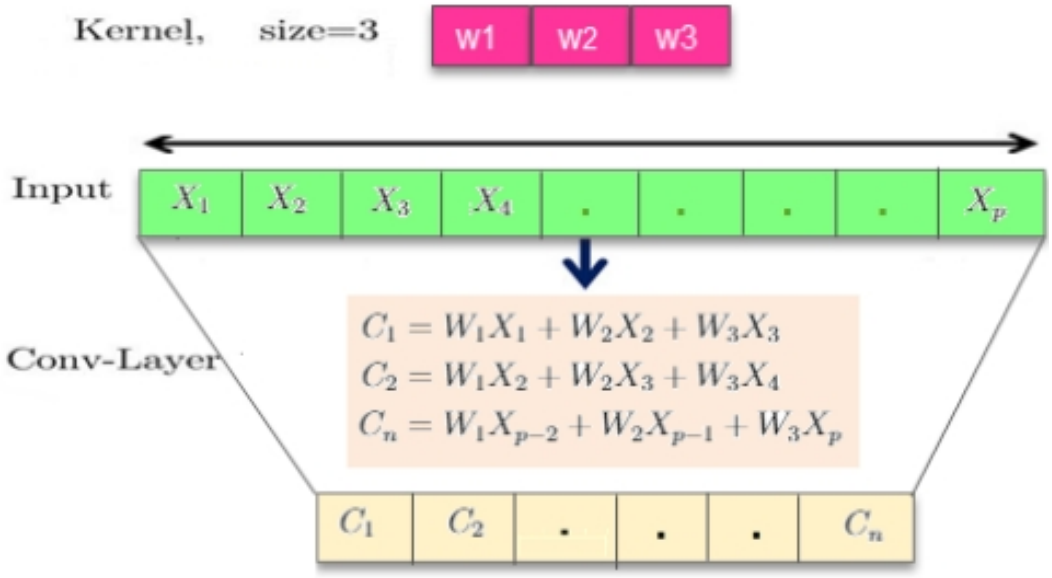


Figure 3.2 – Convolution.

Pooling Layers : Après la transformation ReLU, l'opération de mise en commun (Pooling) regroupe l'activation des neurones d'une couche en un seul neurone de la couche suivante. La couche de pooling fonctionne indépendamment sur chaque entité d'entrée, elle permet de réduire progressivement la taille des représentations afin de réduire le nombre de paramètres ou de poids, ce qui diminue le coût de calcul dans le réseau, tout en préservant les informations les plus critiques. Elle permet aussi de contrôler le sur-apprentissage.

Il peut utiliser deux méthodes de mise en commun différentes :

• **La mise en commun maximale (Max-Pooling)** : utilise la valeur maximale de chaque groupe de neurones de la couche précédente.

• **La mise en commun moyenne (Average-Pooling) :** utilise la valeur moyenne de chaque groupe de neurones de la couche précédente.

Le Pooling est une forme de sous-échantillonnage non linéaire fonctionne de manière similaire à la convolution. le noyau de pooling se convolé sur le volume d'entrée et le diviser en un ensemble de région qui ne se chevauchent pas, et chaque sous-région produit une seule valeur en sortie qui est la valeur maximale pour Max-Pooling ou la valeur moyenne pour Average-Pooling la figure 3.3 décrit lopération de Max-Pooling avec un entré 1D et un noyau de taille 2.

la couche de Pooling n'a aucun paramètre pouvant être appris. De ce fait, ces couches ne sont généralement pas incluses dans le nombre total de couches de réseaux de convolution.

Fully Connected Layers : À la fin d'un réseau CNN, il y a une ou plusieurs couches entièrement connectées (chaque nœud de la première couche est connecté à chaque nœud de la

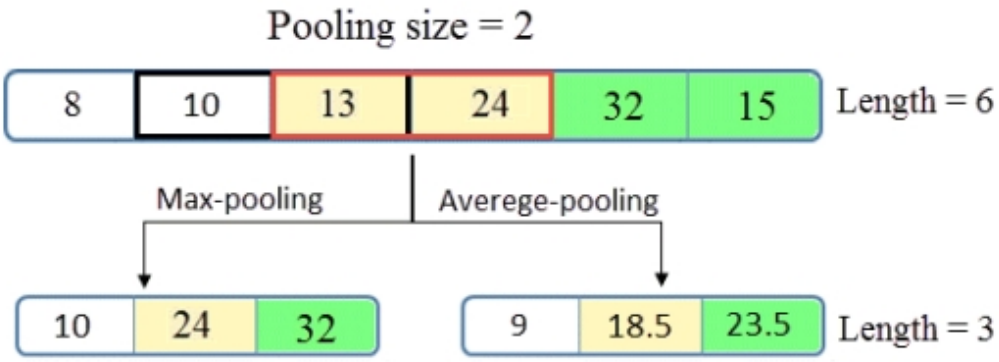


Figure 3.3 – Pooling.

couche suivante). Elles consistent à effectuer une classification basée sur les caractéristiques extraites des convolutions. La couche finale contient une fonction d'activation Softmax, qui génère une valeur de probabilité de 0 à 1 pour chacune des étiquettes de classes que le modèle tente de prédire. Dans certaines architectures de réseaux CNNs récentes, les couches entièrement connectées peuvent se remplacer par plusieurs couches de mise en commun moyennes (average-pooling). Cela permet à ces réseaux de réduire considérablement le nombre total des paramètres et qui permet une meilleure prévention de sur-apprentissage[45].

3.3.2 Réseaux de neurones récurrents (RNN)

les réseaux neuronal s'inspirent du fonctionnement des neurones biologiques du cerveau humain, ces neurones sont considère comme le centre de réflexion, et parfois ils doivent mémoriser certains évènements pour les utilisé ultérieurement avant de prendre la décision. Les réseaux neuronal traditionnel n'ont pas ce propriété, alors le fonctionnement d'un réseau de neurones récurrents (RNN) est motivé par le fait qu'un être humain raisonne en s'appuyant sur les connaissances qu'il a acquises et qui qu'il a mémorisé précédemment [].

Les réseaux RNNs sont des réseaux de type Feed-Forward ayant un état interne (ou mémoire) qui prennent en compte tout ou partie des données vues précédemment (déjà fournies au réseau), en plus de la donnée vue actuellement pour adapter leur décision. L'idée clé de base de ces réseaux est le déploiement d'un calcul récurrent grâce aux boucles dans l'architecture du réseau. La sortie de réseau est une combinaison de son état interne (mémoire d'entrées) et le dernier l'entrée, au même temps, l'état interne change pour intégrer cette nouvelle donnée saisie. cela permet aux informations de persister en mémoire, (la figure 3.4).

En raison de ces propriétés, les réseaux récurrents sont adaptés aux cas où la présence d'une forme n'est pas la seule information discriminante mais également un ordre d'apparition par exemple. ils sont de bons candidats pour les tâches qui traitent des données séquentielles, telles que les données textuelles ou des données avec des caractéristiques temporelles. La description mathématique du processus de transfert de mémoire est comme suit :

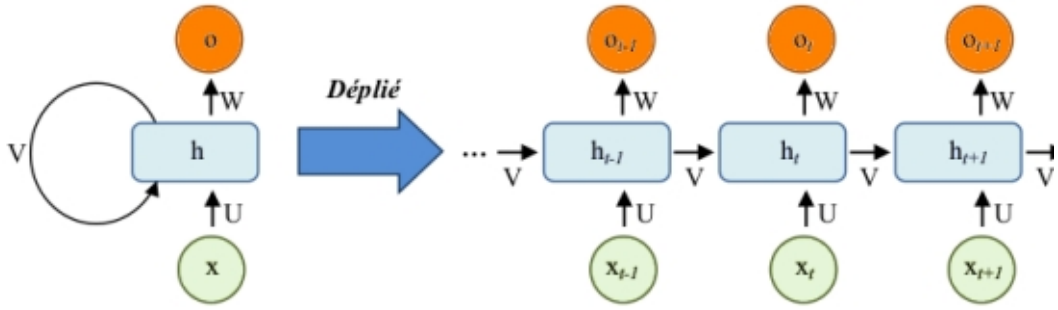


Figure 3.4 – L'architecture d'un modèle RNN.

$$h_t = \delta(Ux_t + Vh_{t-1} + b_h) \tag{3.1}$$

$$o_t = \delta(Wh_t + b_y) \tag{3.2}$$

ou :

• **ht :** est l'état caché au temps t.

• **xt :** est l'entrée au même temps t.

• **U, V, W :** sont les matrice de pondération, Input-to- Hidden, Hidden-to-Hidden et Hidden-to-Output respectivement .

• **bh :** est la valeur du biais de l'état caché.

• **by :** est la valeur du biais de sortie.

• **Ot :** est la valeur de sortie au temps t.

• **δ :** est une fonction de non-linéarité appelées fonctions d'activation. (soit une fonction sigmoïde logistique ou tanh) qui est un outil standard de changement d'échelle pour condenser des valeurs très grandes ou très petites dans un espace logistique, ainsi que pour rendre les gradients exploitables pour la rétro-propagation.

Un bloc de réseau neuronal, examine une entrée xt et génère une valeur ot . Une boucle de rétroaction se produit à chaque pas de temps, chaque état caché ht contient des traces non seulement de l'état masqué précédent, mais également de tous ceux qui ont précédé ht−1 aussi longtemps que la mémoire peut persister.

3.3.3 Deep Neural Network (DNN)

Les Deep Neural Networks (DNN) sont un ensemble de neurones organisés en une séquence de couches multiples appelée Multilayer Perceptrons (MLP). Ils se distinguent des réseaux neuronaux traditionnels (Artifical Neural Network) par leur profondeur et le nombre de couches, de nœuds (neurones) qui composent le réseau. Lorsqu'un ANN possède deux couches cachées ou plus, il est connu sous le nom de réseau neuronal profond. Ils tentent à modéliser

des données contenant des architectures complexes en combinant différentes transformations non linéaires[46].

Le concept de base du perception a été introduit par Rosenblatt en 1958 [47]. La perception calcule une sortie unique à partir de multiples entrées à valeurs réelles (xi) en formant une combinaison linéaire en fonction de ses poids (w) d'entrée, puis en plaçant la sortie via une fonction d'activation non linéaire. Mathématiquement, cela peut être écrit comme suit :

$$y = \delta(\sum_{n=1}^{n} W_i x_i + b) = \delta(W^T X + b)$$

Avec :

• W : est le vecteur des poids.

• X : est le vecteur des entrées.

• b : désigne le biais.

• δ : représente la fonction d'activation.

Un réseau typique de perception multi-couches (MLP) comprend un ensemble de nœuds sources formant la couche d'entrée, une ou plusieurs couches cachées de nœuds de calcul et une couche de sortie de nœuds. Le signal d'entrée se propage couche par couche sur le réseau.Le flux de signaux d'un tel réseau avec une couche cachée est illustré par la (figure 2.2).

Les réseaux DNNs sont généralement utilisés dans les problèmes d'apprentissage supervisé. La formation de modèle (l'apprentissage) signifie l'adaptation de tous les poids et les biais à leurs valeurs optimales.

3.3.4 Unités de mémoire à court terme (LSTM)

Le réseau RNN a un long pas de temps car il prend en compte l'état sauvegardé précédent lors de la mise à jour du poids, les gradients lorsque l'entraînement devient de plus en plus petit et après quelques étapes, les erreurs n'ont pas pu être propagées à la fin du réseau. il n'y aura pas de différence significative dans le résultat, donc il ne peut pas faire de mise à jour des poids. Ce problème du RNN est appelé gradients de disparition (Vanishing Gradients). Pour surmonter ce problème, une architecture à mémoire longue et courte durée (LSTM) a été proposée au milieu des années 90 par les chercheurs allemands Sepp Hochreiter et Juergen Schmidhuber pour les réseaux neuronaux récurrents et aussi des étapes supplémentaires appelées Gated Recurrent Units (GRU). Ces étapes sont été utilisées pour améliorer les performances et la précision des RNNs.

L'idée clé de la méthode LSTM est l'état de la cellule. Elle a la capacité de supprimer ou d'ajouter des informations à l'état de la cellule. Cette technique est réglée par des structures appelées portes (Gates). Ces derniers pourraient être une fonction sigmoïde où une valeur de 1 signifie que toutes les informations passent et une valeur de 0 signifie le contraire.

Les architectures LSTM et GRU se fonctionnent de la même manière. Cependant le GRU utilise moins de paramètres d'entraînement et donc moins de mémoires et s'entraînent plus rapidement que les LSTM. Alors que le LSTM est plus précis sur les ensembles de données utilisant une séquence plus longue.

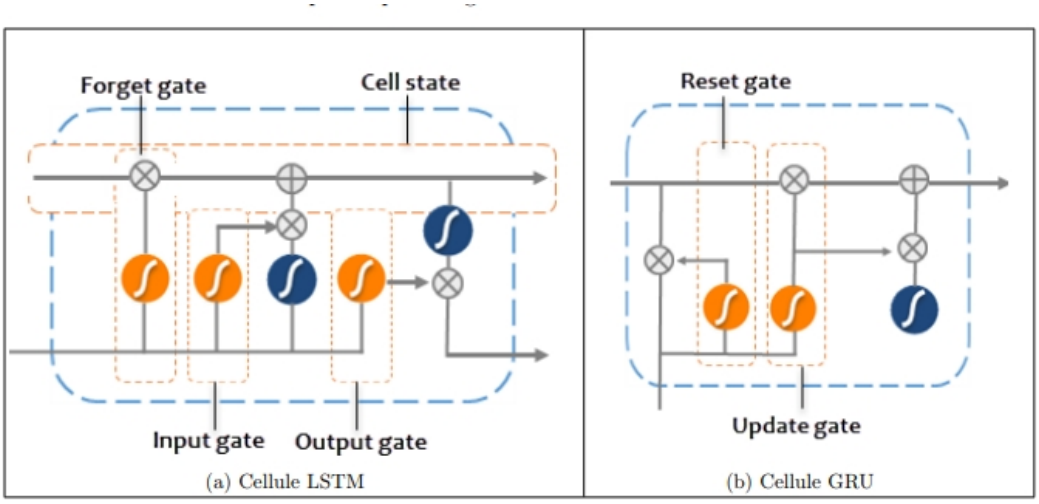


Figure 3.5 – L'architecture d'un modèle LSTM GRU.

Les cellules LSTM sont les plus efficaces pour retenir les informations utiles lors de la rétro-propagation du gradient. Ce qui a leur permet de corriger les différences entre les prédictions sortantes et les catégories de référence en calculant le gradient de l'erreur pour chaque neurone, en allant de la dernière couche vers la première. La figure 3.5 illustre les quatre couches interactives (sigmoïde et tanh), les trois portes et les opérations Pointwise qui traitent le vecteur x à l'intérieur d'une cellule LSTM à un temps t.

3.4 Base de données

Dans cette étude l'ensemble de donnée utilisé est **CIC_DDoS_2019** ,publiquement disponible sur le site Web de l'Institut canadien de cybersécurité[48],Elle regroupe des informations provenant des flux réseau réels, avec diverses formes d'attaques DDOS les plus récentes et les plus courantes. Ces informations sont des formats plus simplifiés qui renferment principalement des informations métas sur des connexions réseau. Chaque donnée (chaque flux de réseau) regroupe tous les paquets qui partagent certaines caractéristiques dans une période temporelle et qui ne contiennent pas de charge utile (Payload). Les données de l'ensemble comprennent deux versions différentes, à savoir des données PCAP brutes et des données CSV. L'analyseur du trafic CICFlowMeter-V3 a été employé par les auteurs afin d'extraire plus de 80 caractéristiques des fichiers PCAPs. Les résultats ont été sauvegardés dans des fichiers CSVs structurés et étiquetés par l'Université de New Brunswick[49].

L'ensemble de données se divisé en deux jours distincts :

- Le premier jour (12/01/2019) est nommé jour d'entraînement, ce dernier se présente sous forme de 12 attaques DDoS différentes et comprend 11 fichiers CSV. Les détails de ce jour sont résumés dans le tableau 3.1.

Attaque	Nb d'instances	Pourcentage
Syn	39995	9.32%
DrDoS_SNMP	39990	9.32%
DrDoS_LDAP	39985	9.32%
DrDoS_SSDP	39980	9.32%
DrDoS_NetBIOS	39900	9.30%
DrDoS_MSSQL	39854	9.29%
TFTP	39826	9.28%
DrDoS_UDP	39789	9.27%
DrDoS_DNS	39637	9.24%
UDP-lag	39225	9.14%
DrDoS_NTP	37446	8.72%
BENIGN	4298	1.00%
WebDDoS	75	0.02%
Nb parcentage	429346	100%

Table 3.1 – Nom et nombre d'attaques trouvées dans le premier jour.

L'ensemble complet des données contient 429346 instances, dont 425048 sont des attaques DDoS ansi 4298 sont des instances d'un trafic réseau bénin (légitime /normale), le nombre d'instances pour chaque type d'attaque DDOS est indiqué dans le tableau 3.1.

- Le deuxième jour (11/03/2019), également connu sous le nom de jour du test, comporte 7 attaques DDoS différentes. Cet ensemble contient 7 fichiers CSV. Tous les détails sont répertoriés dans le tableau 3.2 .

  L'ensemble complet des données contient 218 000 instances, dont 215761 sont des attaques DDoS et 2 239 sont des instances d'un trafic réseau bénin.le nombre d'instances pour chaque type d'attaque DDOS est indiqué dans le tableau 3.2.

Attaque	Nb d'instances	Pourcentage
NetBIOS	79 950	36.66%
UDP	57 282	26.30%
Syn	56 296	25.85%
MSSQL	42 836	19.66%
Portmap	39 033	17.92%
BENIGN	2 239	1.03%
LDAP	1 977	0.91%
UDPLag	387	0.18%
nb parcentage	218 000	100%

Table 3.2 – Nom et nombre d'attaques trouvées dans le deuxième jour.

3.5 Taxonomie des attaques DDoSs

Une variété d'études ont créé des taxonomies pour les attaques DDoS, mais leur portée reste limitée. Pour combler cette lacune, nous avons analysé de nouvelles attaques exploitant les protocoles TCP/UDP au niveau applicatif, et avons proposé une nouvelle taxonomie. Cette taxonomie, détaillée dans la figure 3.6, distingue les attaques réflexives des attaques d'exploitation, offrant ainsi une vue complète des différentes catégories d'attaques DDoS.[50]

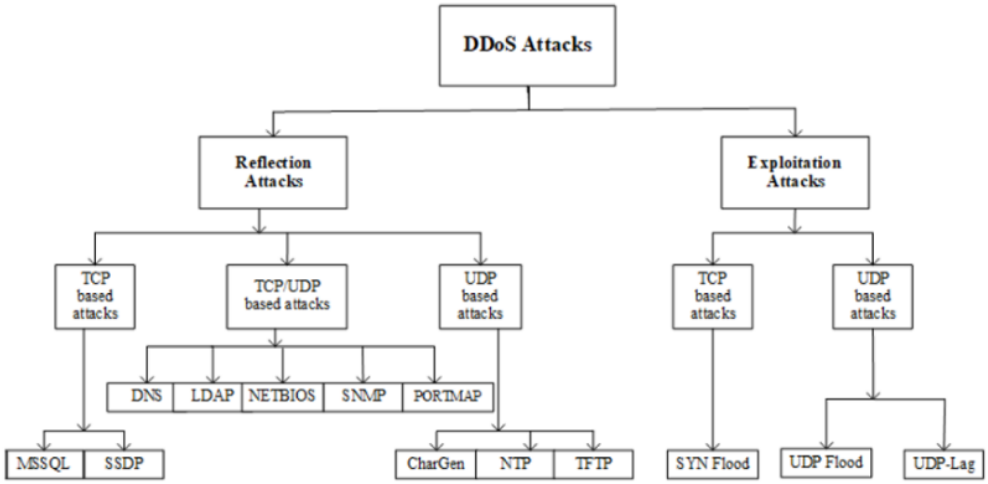


Figure 3.6 – Taxonomie des attaques DDoSs [50].

Les attaques DDoS basées sur la réflexion : Les attaques DDoS basées sur la réflexion consistent à masquer l'identité de l'attaquant en utilisant des composants tiers légitimes. Les attaquants envoient des paquets à des serveurs réflecteurs avec une adresse IP source falsifiée pour correspondre à celle de la victime ciblée, noyant cette dernière sous des paquets de réponse. Ces attaques peuvent exploiter des protocoles de couche applicative via TCP ou UDP, ou les deux. Les attaques TCP incluent MSSQL et SSDP, tandis que celles basées sur UDP incluent CharGen, NTP et TFTP. Certaines attaques telles que DNS, LDAP, NETBIOS et SNMP peuvent utiliser soit TCP soit UDP[50].

Les attaques DDoS basées sur l'exploitation : dissimulent l'identité de l'attaquant en utilisant des composants tiers légitimes. Elles consistent à envoyer des paquets à des serveurs réflecteurs avec une adresse IP source falsifiée pour submerger la victime de paquets de réponse. Ces attaques exploitent les protocoles TCP et UDP au niveau applicatif pour perturber le fonctionnement des serveurs cibles. Les attaques basées sur TCP incluent le SYN flood, tandis que celles basées sur UDP comprennent le UDP flood et le UDP-Lag[50].

Les attaques	Description
UDP-lag	L'attaque UDP-Lag vise à perturber la connexion entre le client et le serveur[50].
TFTP	L'attaque exploite la vulnérabilité de débordement de tampon dans un serveur Trivial File Transfer Protocol (TFTP).
WebDDoS	Une attaque DDoS Tsunami sur le Web est une forme évoluée d'attaque DDoS Flood HTTP, sophistiquée et agressive, difficile à détecter et à contrer sans bloquer le trafic légitime.
DNS	Une attaque par réflexion, exploitée les vulnérabilités du DNS (Domain Name Service Protocol).
MSSQL	Attaque visant à exploiter les serveurs Microsoft SQL Server en les surchargeant de requêtes malveillantes.
LDAP	Les attaques par injection LDAP utilisent des techniques similaires aux attaques par injection SQL c'est une attaque utilisée pour exploiter les applications web . Elles exploitent les paramètres introduits par l'utilisateur pour générer une requête LDAP[51]
NetBIOS	Le système d'entrée/sortie de base du réseau (NetBIOS) est le mécanisme utilisé par les systèmes Microsoft Windows pour partager des ressources, notamment des partages de fichiers et d'imprimantes. NetBIOS utilise les ports 137, 138 et 139[52].
NTP	Le NTP est une attaque par amplification dans laquelle l'attaquant exploite les serveurs NTP accessibles au public pour surcharger la cible de trafic UDP[53].
SSDP	Une attaque SSDP est un type d'attaque DDoS par réflexion utilisant les protocoles UPnP pour envoyer un trafic amplifié au serveur de la victime[54].
SNMP	Dans cette variante d'attaque de surcharge, une attaque utilisant le protocole SNMP génère un flux massif de trafic dirigé vers des victimes issues de multiples réseaux.
Syn	L'attaque SYN flood envoie des paquets SYN répétés pour saturer le serveur en exploitant le protocole TCP, provoquant des plantages potentiels [50].
UDP	L'attaque par inondation UDP consiste à envoyer un grand nombre de paquets UDP à des ports aléatoires sur une machine distante à un rythme très élevé. Cela entraîne l'épuisement de la bande passante disponible du réseau, des crashs du système et une dégradation des performances[50].
PortMap	L'attaque cible le port 111 en TCP ou UDP, un service utilisé pour rediriger les clients vers le bon numéro de port pour communiquer avec le service de Remote Procedure Call (RPC).

Table 3.3 – Attaques DDoS basées sur la réflexion et sur l'exploitation.

3.6 La préparation des données

Les performances des méthodes de deep learning sont étroitement liées à la quantité et à la qualité des données d'apprentissage. Une plus grande quantité de données de qualité se traduit par une précision accrue et de meilleurs résultats. Dans notre cas, nous disposons d'une masse de données suffisamment importante. Cependant, en raison du déséquilibre entre les classes de données, il est nécessaire de réduire ces données.

La preprocessing des données comprend deux opérations essentielles avant leur traitement : la réduction des données et la résolution de l'étiquetage.

3.6.1 La réduction des données :

En raison du volume important de l'ensemble de données, nous devons réduire considérablement le nombre d'échantillons dans les données d'apprentissage et de test ensemble. Cette réduction de données est une pratique courante dans le domaine de l'apprentissage automatique, connue sous le nom de sous-échantillonnage, qui vise à améliorer l'efficacité des algorithmes tout en conservant la représentativité de l'ensemble de données.

Notre ensemble de données se compose de 11 fichiers CSV d'une taille totale de plus de 2,2 Go pour l'apprentissage, ainsi que de 7 autres fichiers CSV d'une taille totale de plus de 876 Mo pour le test. Ces fichiers contiennent des informations sur les caractéristiques et les comportements des utilisateurs, collectées à partir de diverses sources dans le cadre de notre recherche sur la sécurité des systèmes d'information.

La lecture et la préparation de ces données volumineuses représentaient un défi technique significatif, même avec l'utilisation de Google Colab et ses 12 Go de RAM. Nous avons donc dû recourir à des techniques de gestion de la mémoire telles que la fonction **reduce-mem-usage** pour optimiser l'utilisation des ressources et garantir des analyses efficaces.

Cette démarche de réduction des données n'affecte pas la qualité de notre recherche, mais elle nous permet de travailler de manière plus efficiente en concentrant notre attention sur les échantillons les plus représentatifs et pertinents pour nos objectifs d'analyse et de modélisation.

3.6.2 La résolution de l'étiquetage :

Comme indiqué dans le tableau 3.4, les étiquettes des données étaient incorrectes. Par conséquent, nous avons entrepris de réviser manuellement les étiquettes des données de test dans l'ensemble de données CICDDoS-2019. Les données de test se composent de 7 fichiers CSV représentant 8 classes. Parmi ces classes, 7 correspondent à des attaques DDoS, tandis que la 8e classe est dédiée au trafic normal (classe BENIGN). Voici les modifications apportées aux étiquettes des classes dans les données de testse fait comme la suivante :

- La classe NetBIOS a été modifiée en DrDoS_NetBIOS.
- La classe MSSQL a été modifiée en DrDoS_MSSQL.
- La classe LDAP a été modifiée en DrDoS_LDAP.

- La classe UDPLag a été modifiée en UDP-lag.
- La classe UDP a été modifiée en DrDoS_UDP.
- Les classes BENIGN, Portmap et Syn sont restées inchangées.

Les étiquettes des données de Test(jour 2)	Les étiquettes des données d'apprentissage(jour 1)
BENIGN	BENIGN
PortMap	DrDoS_DNS
NetBIOS	DrDoS_NetBIOS
LDAP	DrDoS_LDAP
MSSQL	DrDoS_MSSQL
UDP	DrDoS_UDP
UDP-Lag	UDP-lag
SYN	SYN
	DrDoS_SNMP
	TFTP
	DrDoS_SSDP
	DrDoS_NTP

Table 3.4 – Les étiquettes des différentes attaques dans la journée d'entraînement ainsi dans la journée de Test

La classe Portmap, qui n'existe pas dans les données d'apprentissage, a été éliminée pour la première fois pour la classification multi-classes de 13 classes, mais elle est considérée pour la classification binaire (2_classes) et la classification multi-classes (8_classes) pour la détection des attaques DDoS.

Dans une autre expérience, ces classes ont été regroupées sous l'étiquette "Other", qui désigne une autre attaque DDoS. De même, dans les données d'apprentissage, les attaques qui ne se produisent que pendant l'entraînement, telles que DrDoS_NTP, DrDoS_SNMP, DrDoS_SSDP, DrDoS_UDP, TFTP, UDP-lag et WebDDoS, absentes dans les données de test, ont été regroupées sous cette même étiquette.

Par un processus de criblage, nous avons extrait trois groupes de données. Le premier sous-ensemble, présenté dans le tableau 3.5, a été traité en remplaçant tous les types d'attaques existants par "Attaque" et "Benign (trafic légitime)". Pour le deuxième sous-ensemble, présenté dans le tableau 3.7, nous avons sélectionné les données du jour 1 (12/01/2019). Quant au troisième sous-ensemble, présenté dans le tableau 3.6, nous avons pris les données du jour 2 (11/03/2019).

	Les Classes	Nb d'instances pour L'apprentissage	Nb d'instances pour le Test
Base de données_1 (2_Classes)	BENIGN	1372144	343037
	ATTAQUE	1372145	343036

Table 3.5 – Sous ensembles_1 constitue de 2 classes pour la détection des attaques DDoS (Base de données_1).

	Les Classes	Nb d'instances pour L'apprentissage	Nb d'instances pour le Test
Base de données_2 (8_Classes)	BENIGN	12905	3256
	DrDoS_NetBIOS	229153	56925
	DrDoS_MSSQL	159367	39948
	DrDoS_LDAP	81925	20569
	Portmap	73981	18464
	Syn	178821	44691
	DrDoS_UDP	192146	48525
	UDP-lag	74181	18270

Table 3.6 – Sous ensembles_2 constitue de 7 différentes DDoS attaques (Base de données_2).

	Les Classes	Nb d'instances pour L'apprentissage	Nb d'instances pour le Test
Base de données_3 (13_Classes)	BENIGN	12905	Utilisé l'échantillonnage stratifié (Stratified sampling) avec 25% des données d'apprentissage afin d'éviter l'erreur d'échantillonnage des données déséquilibrées pourcentage et aussi assurer que les données d'entraînement et de test ont le même pourcentage de division pour chaque classe.
	DrDoS_NetBIOS	229153	
	DrDoS_MSSQL	159367	
	DrDoS_LDAP	81925	
	Syn	178821	
	DrDoS_UDP	192146	
	UDP-lag	74181	
	TFTP	75461	
	WebDDoS	158	
	DrDoS_SSDP	78596	
	DrDoS_SNMP	78516	
	DrDoS_NTP	74374	
	DrDoS_DNS	75489	

Table 3.7 – Sous ensembles_3 constitue de 12 différents DDoS attaques (Base de données_3).

3.6.3 Les pré-traitements des données :

Afin de construire un modèle précis et d'obtenir les meilleurs résultats, il est très important d'effectuer des analyses exploratoires sur l'ensemble de données et ses caractéristiques Le pré-traitement de l'ensemble de données est effectué avant d'être appliqué au réseau neuronal profond. les étapes de pré-traitement sont comme les suivantes :

Tout d'abord, filtré l'ensemble de données pour supprimer les lignes redondantes représentant les instances de classe, une analyse a été réalisée pour détecter toute valeur **'NAN'** (Not A Number) ou **'INF'** (Infinite Value). Ces valeurs peuvent être considérées comme des valeurs manquantes, ce qui peut avoir un impact négatif sur les performances des modèles de deep learning ou de machine learning en général. Il s'avère que les données sélectionnées pour cette étude présentent plusieurs valeurs 'NAN' dans la colonne **Flow Bytes**. Pour conserver cette caractéristique et étant donné que nous disposons de suffisamment de données, les lignes contenant des valeurs NAN ou INF ont été supprimées.

Les statistiques descriptives qui résument la dispersion et la distribution de l'ensemble de données ont montré qu'il y a des colonnes vides (ses valeurs sont toujours 0), ces caractéristiques ne contiennent aucune information discriminatoire permettant de différencier les classes d'attaque, par contre ils peuvent donner des mauvaises résultats, ces colonnes sont : **' Bwd PSH Flags',' Fwd URG Flags',' Bwd URG Flags','FIN Flag Count',' PSH Flag Count',' ECE Flag Count', 'Fwd Avg Bytes/Bulk',' Fwd Avg Packets/Bulk',' Fwd Avg Bulk Rate',' Bwd Avg Bytes/Bulk', ' Bwd Avg Packets/Bulk','Bwd Avg Bulk Rate'** qui ont été supprimés car ils se sont installés à 0 ou 1 (Figure 3.7)dans toutes les attaques .

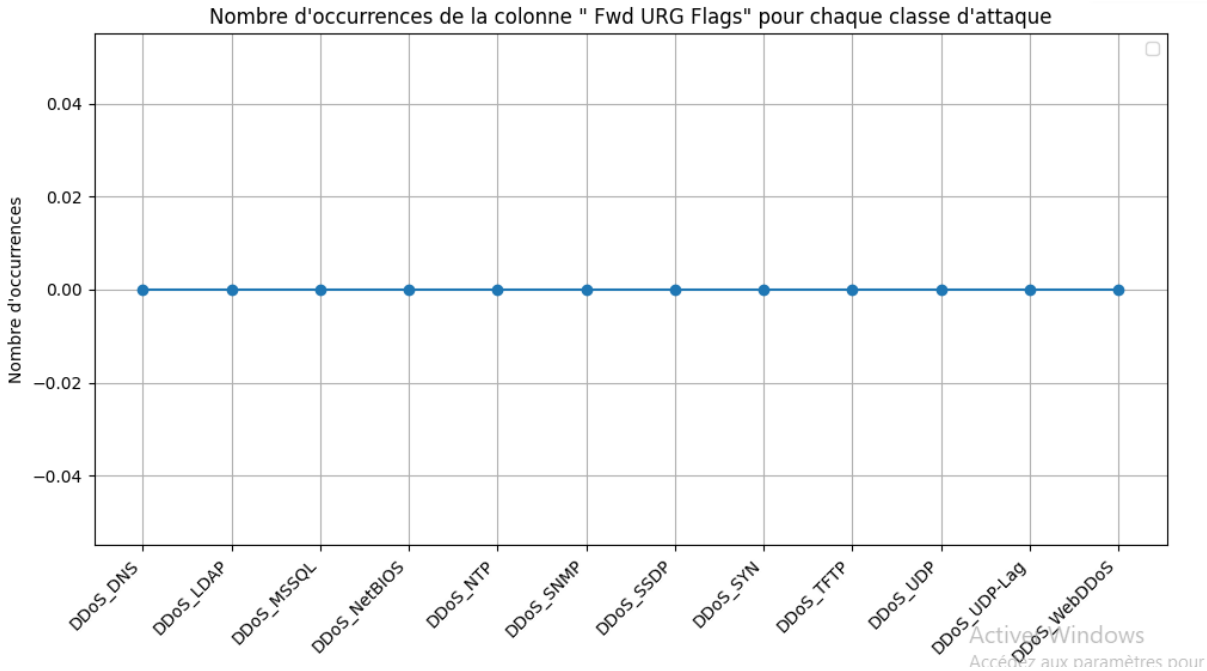


Figure 3.7 – Exemple pour la colonne supprimer (stabilise a 0).

Il y a un certain nombre de caractéristiques de type catégoriel dans l'ensemble de données qui doivent être encodées. la colonne **Flow Packets/s** a été convertie en une colonne numérique. Cependant, les autres colonnes catégorielles comme les colonnes Adresse IP et Timestamp ont été supprimées. On a considéré que ces caractéristiques sont liées aux informations de connexion et ne représentent pas les propriétés des attaques DDoS, car ce dernier peut être produit à tout moment par n'importe quelle machine contre n'importe quelle machine victime, pour cette raison les caractéristiques **'Flow ID', 'Source IP', ' Destination IP', ' Timestamp'** et **'Inbound'**. sont aussi supprimées pour qu'il ne reste que les caractéristiques du trafic réseau (Trafic features).

L'ensemble des caractéristiques qu'ils nous reste pour cette étude sont indiquées dans le tableau 3.8.

Features	Type	Features	Type
Unnamed : 0	int32	Average Packet Size	float16
Source Port	int32	Avg Fwd Segment Size	float16
Destination Port	int32	Avg Bwd Segment Size	float16
Protocol	int8	Fwd Header Length.1	int64
Flow Duration	int32	Subflow Fwd Packets	int16
Total Fwd Packets	int16	Subflow Fwd Bytes	int32
Total Backward Packets	int16	Subflow Bwd Packets	int16
Total Length of Fwd Packets	float32	Subflow Bwd Bytes	int32
Total Length of Bwd Packets	float32	Init_Win_bytes_forward	int32
Fwd Packet Length Max	float16	Init_Win_bytes_backward	int32
Fwd Packet Length Min	float16	act_data_pkt_fwd	int16
Fwd Packet Length Mean	float16	min_seg_size_forward	int32
Fwd Packet Length Std	float16	Active Mean	float32
Bwd Packet Length Max	float16	Active Std	float32
Bwd Packet Length Min	float16	Active Std	float32
Bwd Packet Length Mean	float16	Active Max	float32
Bwd Packet Length Std	float16	Active Min	float32
Flow Bytes/s	float64	Idle Mean	float32
Flow Packets/s	float64	Idle Std	float32
Flow IAT Mean	float32	Idle Max	float32
Flow IAT Std	float32	SimillarHTTP	float64
Flow IAT Max	float32	Max Packet Length	float16
Flow IAT Min	float32	Packet Length Mean	float16
Fwd IAT Total	float32	Packet Length Std	float16
Fwd IAT Mean	float32	Packet Length Variance	float32
Fwd IAT Std	float32	SYN Flag Count	int8
Fwd IAT Max	float32	SYN Flag Count	int8
Fwd IAT Min	float32	RST Flag Count	int8
Bwd IAT Total	float32	ACK Flag Count	int8
Bwd IAT Mean	float32	URG Flag Count	int8
Bwd IAT Std	float32	CWE Flag Count	int8
Bwd IAT Max	float32	Down/Up Ratio	float16
Bwd IAT Min	float16	Bwd Packets/s	float32
Fwd PSH Flags	int8	Fwd Packets/s	float32
Fwd Header Length	int64	Bwd Header Length	int16
Min Packet Length	float16	Label	object

Table 3.8 – L'ensemble des caractéristiques utilisées pour la détection des intrusions basé réseau (NIDS).

Pour la colonne **Label** qui représente la classe de chaque instance, il a été encodé avec une technique populaire appelée **"One-Hot-Encoding"**. Le codage va convertit les lignes contenant des catégories en leur propre colonne avec une valeur 1 signifie vrai (cette instance est celle de cette classe) ou 0 signifie faux (cette instance n'est pas de cette classe), et nous passerons à la normalisation des données.

3.6.4 Normalisation des données :

La normalisation des données est généralement requise lorsque les chercheurs appliquent des techniques d'apprentissage en profondeur à des données qui ont des échelles différentes sur les attributs, Nous avons recherché une comparaison des performances du modèle avec ou sans la normalisation des traits effectuée par Wang et al [55], Les modèles discriminatoires gagnent en efficacite,Dans cette travaille, les imitateurs sont StandardScaler. La fonction de mappage **StandardScaler** est illustrée dans Equation3.3.

$$f'_{:,i} = \frac{f_{:,i} - \text{mean}(f_{:,i})}{\text{std}(f_{:,i})} \tag{3.3}$$

Ensuit ,Les données sont divisée en 2, des données pour l'apprentissage et pour la test de modèle.

3.6.5 SMOTE(Synthetic Minority Oversampling Technique)

Comme on a mentionné auparavant,ces ensembles des données sont très déséquilibres, on aura de mauvaises performances sur les classes minoritaires. Dans notre cas, la classe minoritaire BENIGN est la plus importante pour différencier le trafic normale d'une trafic malicieux contenant des attaques DDoS. Un taux élevé de vrai négatifs va réduit le taux de fausse alarme du système. Pour traiter ce problème, on a essayé l'un des algorithmes de sur-échantillonnage (oversampling) :**Synthetic Minority Oversampling Technique (SMOTE)** (figure 3.8), Comme le montre l'image, les valeurs étaient faibles, et après les avoir ajoutées, nous l'avons appliquée aux données d'apprentissage afin d'augmenter la taille de. les classes minoritaires dans 3 expériences.

Synthetic Minority Oversampling Technique (SMOTE) SMOTE est une technique de sur-échantillonnage où des échantillons synthétiques sont générés pour la classe minoritaire. Cet algorithme aide à surmonter le problème de surajustement posé par le suréchantillonnage aléatoire. Il se concentre sur l'espace des caractéristiques pour générer de nouvelles instances à l'aide de l'interpolation entre les instances positives qui se trouvent ensemble[56].

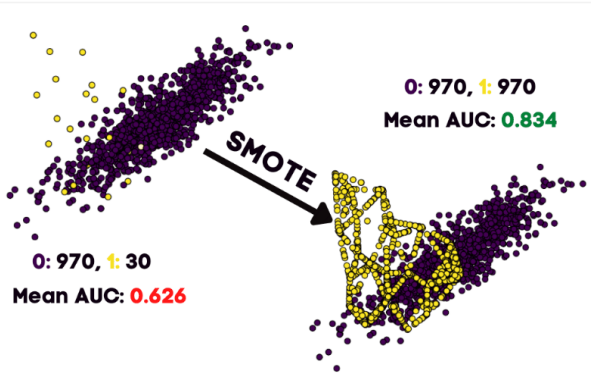


Figure 3.8 – Le sur-échantillonage via SMOTE (Oversampling Technique). [57].

3.7 Un système de détection d'intrusion pour la détection des attaques DDoS dans les Réseaux

En utilisant les techniques de classification du Deep Learning, nous avons développé et évalué trois modèles discriminatoires basés sur l'apprentissage profond : le réseau de neurones profond (DNN), le réseau de neurones convolutif (CNN) et le réseau de neurones récurrent (RNN). Ces modèles ont été testés sur trois ensembles de données différents provenant du jeu de données CIC_DDoS_2019.

1. Le premier test consistait en une classification multi-classe (7_ classes) avec la classe "BENIGN" représentant le trafic réseau légitime et six autres catégories d'attaques différentes. Les données d'apprentissage ont été utilisées pour entraîner et valider les modèles, qui ont ensuite été évalués sur l'ensemble de données de test de CIC_DDoS_2019.
2. Le deuxième test était une classification binaire (2_classes), où les classes ont été regroupées en "BENIGN" et "ATTAQUE", cette dernière incluant toutes les attaques DDoS. Les données d'apprentissage ont été utilisées pour l'entraînement et la validation, suivies de l'évaluation sur l'ensemble de données de test de CIC_DDoS_2019. L'objectif était d'évaluer l'efficacité et le taux de détection des anomalies des attaques DDoS.
3. Le troisième test impliquait une classification multi-classe (13_ classes) avec 12 classes différentes d'attaques DDoS. Les données d'apprentissage ont été utilisées pour entraîner et évaluer les modèles. Cette expérimentation visait à évaluer l'efficacité de détection du système face à plusieurs types d'attaques DDoS avec des comportements différents, ainsi que sa capacité à identifier le type d'attaque. Les performances du système ont été évaluées en fonction du taux de détection et de la précision globale de classification

3.7.1 L'architecture d'un modèle et le modèle proposé

Nous avons implémenté trois types d'approches de réseaux profonds (DNN, CNN etRNN_LSTM), l'architecture de chacune de ces approches a été modifiée et améliorée en essayant plusieurs combinaisons de plusieurs paramètres pour chaque expérimentation. Néanmoins, ces architectures des modèles proposées partagent certains propriétés et paramètres communs :

.La fonction d'activation utilisée était ReLU, différentes autres fonctions comme tanh et sigmoid ont été expérimentées, mais le ReLU a toujours les meilleurs résultats.

.La fonction de perte (Loss function) sélectionnée était " categorical-cross-entropy " pour la classification multi-classes et " binary-cross-entropy " pour la classification binaire (normale/Attack).

.L'optimiseur " Adam " a été utilisé avec un taux d'apprentissage (Learning Rate) de 0.001, au lieu de l'algorithme d'optimisation stochastique du gradient descendant (SGD) qui nous a données des mauvais résultats. La fonction de perte va mesurer l'écart entre les prédictions de modèle et les résultats attendus. Ensuite, l'algorithme d'optimisation "Adam" va dicter com-

ment mettre à jour les poids d'un réseaux de neurones pour diminuer le perte, qui au modèle de converge rapidement et obtient des meilleurs prédiction avec le minimum d'erreur.

3.7.2 Un modèle de détection d'intrusion basé sur les réseaux de neurones convolutionnels (CNN)

l'implémentation d'un modèle de réseau neuronal convolutif (CNN) en utilisant la bibliothèque Keras pour effectuer une classification multi-classes, spécifiquement dans le domaine de la détection d'intrusion. Le modèle CNN est construit en empilant plusieurs couches, chaque couche ayant un rôle spécifique dans l'extraction et la transformation des caractéristiques des données en entrée.

Le modèle commence par des couches convolutives qui appliquent des opérations de convolution sur les données d'entrée pour extraire des motifs et des caractéristiques importantes. Ces couches convolutives sont suivies de fonctions d'activation ReLU pour introduire de la non-linéarité dans le modèle. Ensuite, des couches de pooling sont utilisées pour réduire la dimensionnalité des caractéristiques extraites tout en préservant les informations importantes.

Après la phase de convolution et de pooling, les caractéristiques extraites sont aplaties dans une seule dimension et alimentées dans des couches entièrement connectées. Ces couches permettent au modèle d'apprendre des représentations plus abstraites et complexes des données.

Pour éviter le sur-apprentissage, une couche de dropout est ajoutée, ce qui aide à régulariser le modèle en désactivant aléatoirement un pourcentage des neurones pendant l'entraînement. Enfin, une couche de sortie est ajoutée avec une activation softmax pour la classification multi-classes, où chaque classe est associée à une probabilité.

Le modèle est ensuite compilé en spécifiant l'optimiseur Adam et la fonction de perte d'entropie croisée catégorique. Ensuite, il est entraîné sur les données d'entraînement pendant un certain nombre d'époques avec une taille de lot spécifiée. Les performances du modèle sont surveillées sur un ensemble de validation pour évaluer sa généralisation aux données non vues.

Une fois l'entraînement terminé, le modèle est évalué sur les données de test pour estimer ses performances en termes de perte et de précision. De plus, des prédictions sont générées sur les données de test pour produire un rapport de classification et une matrice de confusion, permettant d'évaluer la capacité du modèle à classifier correctement les différentes classes.

Enfin, des courbes de perte et de précision sont tracées pour visualiser les performances du modèle au fil de l'entraînement, ce qui permet de détecter d'éventuels problèmes tels que le sur-apprentissage ou le sous-apprentissage. En résumé, ce code offre une implémentation complète d'un modèle CNN pour la détection d'intrusion, avec des métriques d'évaluation et des visualisations pour évaluer ses performances(voir figure 3.9).

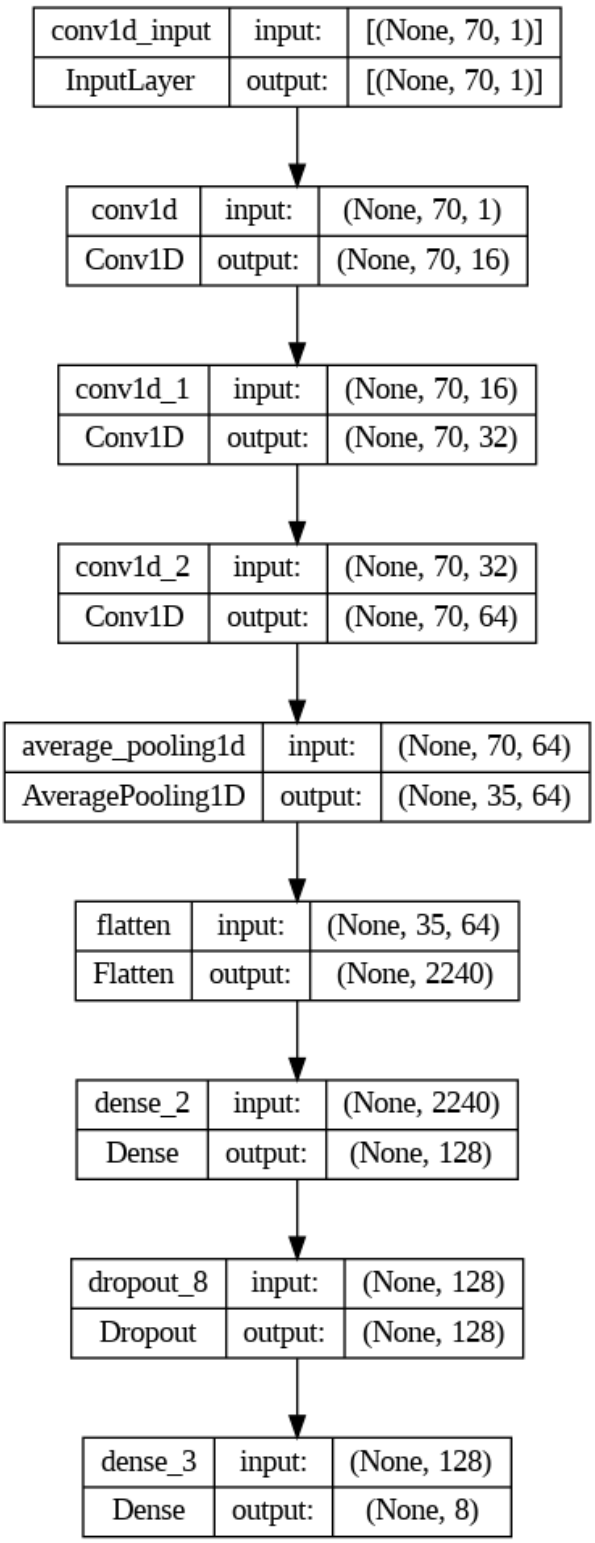


Figure 3.9 – Architecteur de Modle CNN.

3.7.3 Un modèle de détection d'intrusion basée sur les réseaux de neurones profonds (DNN)

L'architecture proposée du modèle DNN de détection d'intrusion basé sur un réseau neuronal profond (DNN) en utilisant la bibliothèque Keras. Tout d'abord, les données sont préparées en divisant l'ensemble de données en ensembles d'entraînement et de test. Ensuite, un modèle de réseau neuronal séquentiel est défini à l'aide de Keras, composé de plusieurs couches denses. La première couche cache compte 256 neurones avec une fonction d'activation ReLU, suivie d'une autre couche cachée de 64 neurones également activés par ReLU. La couche de sortie a une taille correspondant au nombre de classes et utilise une activation softmax pour la classification multi-classes. Le modèle est ensuite compilé avec l'optimiseur Adam, une fonction de perte de 'categorical crossentropy', et l'exactitude comme métrique de performance. Ensuite, le modèle est entraîné sur les données d'entraînement pour un total de 10 époques avec une taille de lot de 64. Une fois l'entraînement terminé, le modèle est évalué sur les données de test pour calculer la perte et l'exactitude. De plus, des prédictions sont générées sur les données de test et utilisées pour produire un rapport de classification et une matrice de confusion afin d'évaluer les performances du modèle de détection d'intrusion. Enfin, les courbes de perte et d'exactitude sont tracées pour visualiser la performance du modèle au cours de l'entraînement(voir figure 3.10).

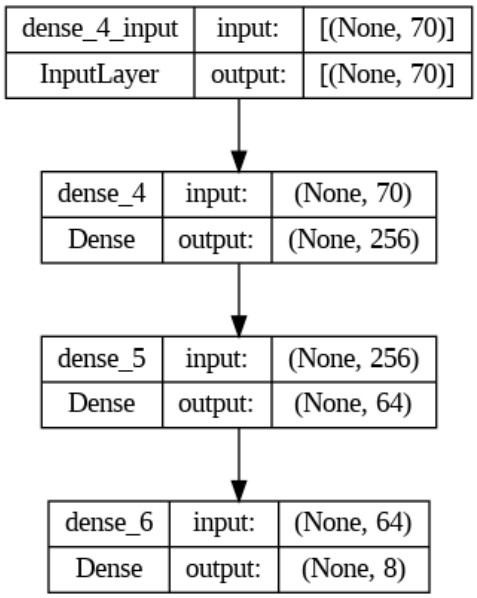


Figure 3.10 – Architecteur de Modle DNN.

3.7.4 Un modèle de détection d'intrusion basé sur les réseaux de neurones récurrents (RNN_LSTM)

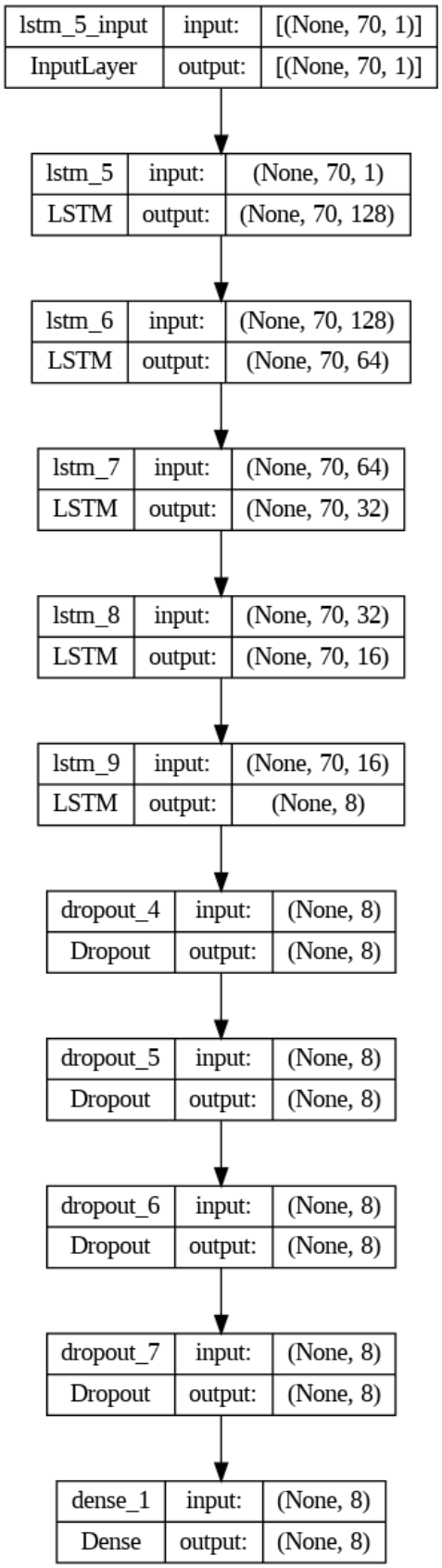


Figure 3.11 – Architecteur de Modle RNN_LSTM.

Nous avons implémenté un réseau RNN_LSTM qui classe les événements de trafic du réseau comme des données de séries chronologiques.

Tout d'abord, les données sont divisées en ensembles d'entraînement et de test. Ensuite, le nombre de classes est défini comme 8, ce qui semble être le nombre de classes dans le problème de classification.

Le modèle RNN-LSTM est ensuite construit séquentiellement en utilisant la classe 'Sequential' de Keras. Plusieurs couches LSTM sont ajoutées au modèle, chacune avec un nombre spécifié d'unités LSTM (128, 64, 32, 16). L'option 'return-sequences=True' est utilisée pour que chaque couche LSTM renvoie la séquence complète plutôt que seulement la sortie à l'état final. La dernière couche LSTM ne renvoie pas la séquence complète.

Des couches de dropout sont ajoutées après chaque couche LSTM pour la régularisation, ce qui aide à prévenir le sur-apprentissage en désactivant aléatoirement un pourcentage des neurones pendant l'entraînement.

Enfin, une couche dense de sortie avec une activation softmax est ajoutée pour la classification multi-classes. Le modèle est ensuite compilé avec l'optimiseur Adam et la fonction de perte d'entropie croisée catégorique.

Le modèle est ensuite entraîné sur les données d'entraînement pour un total de 10 époques avec une taille de lot de 64. Les performances du modèle sont surveillées sur un ensemble de validation représentant 20 des données d'entraînement.

Une fois l'entraînement terminé, le modèle est évalué sur les données de test pour estimer sa performance en termes de perte et de précision. Des prédictions sont générées sur les données de test, suivies d'un rapport de classification et d'une matrice de confusion pour évaluer la capacité du modèle à classer correctement les différentes classes.

Enfin, des courbes de perte et de précision sont tracées pour visualiser les performances du modèle au fil de l'entraînement. Cela permet de surveiller les performances du modèle et d'identifier tout signe de sur-apprentissage ou de sous-apprentissage. En résumé, ce code fournit une implémentation complète d'un modèle RNN_LSTM pour la classification multi-classes dans le domaine de la détection d'intrusion, avec des métriques d'évaluation et des visualisations pour évaluer ses performances (voir figure 3.11).

3.7.5 Schéma conceptuel de notre méthode d'implémentation DL

La figure3.12 représente un diagramme de flux illustrant le processus de traitement et d'utilisation du jeu de données CIC_DDOS_2019 pour développer des modèles d'apprentissage profond destinés à détecter les attaques DDoS. Le processus commence par la séparation des données en ensembles de test et d'entraînement. Ensuite, les données sont préparées et classées en différentes catégories (8 classes, 13 classes, 2 classes).

Après cela, les données subissent un prétraitement (encodage/normalisation) pour les ensembles de test et d'entraînement. Le processus de formation des modèles implique l'élaboration et l'initialisation des paramètres des modèles profonds, suivi de l'entraînement des mo-

dèles. Enfin, les modèles sont testés et évalués pour vérifier leurs performances en matière de détection des attaques DDoS.

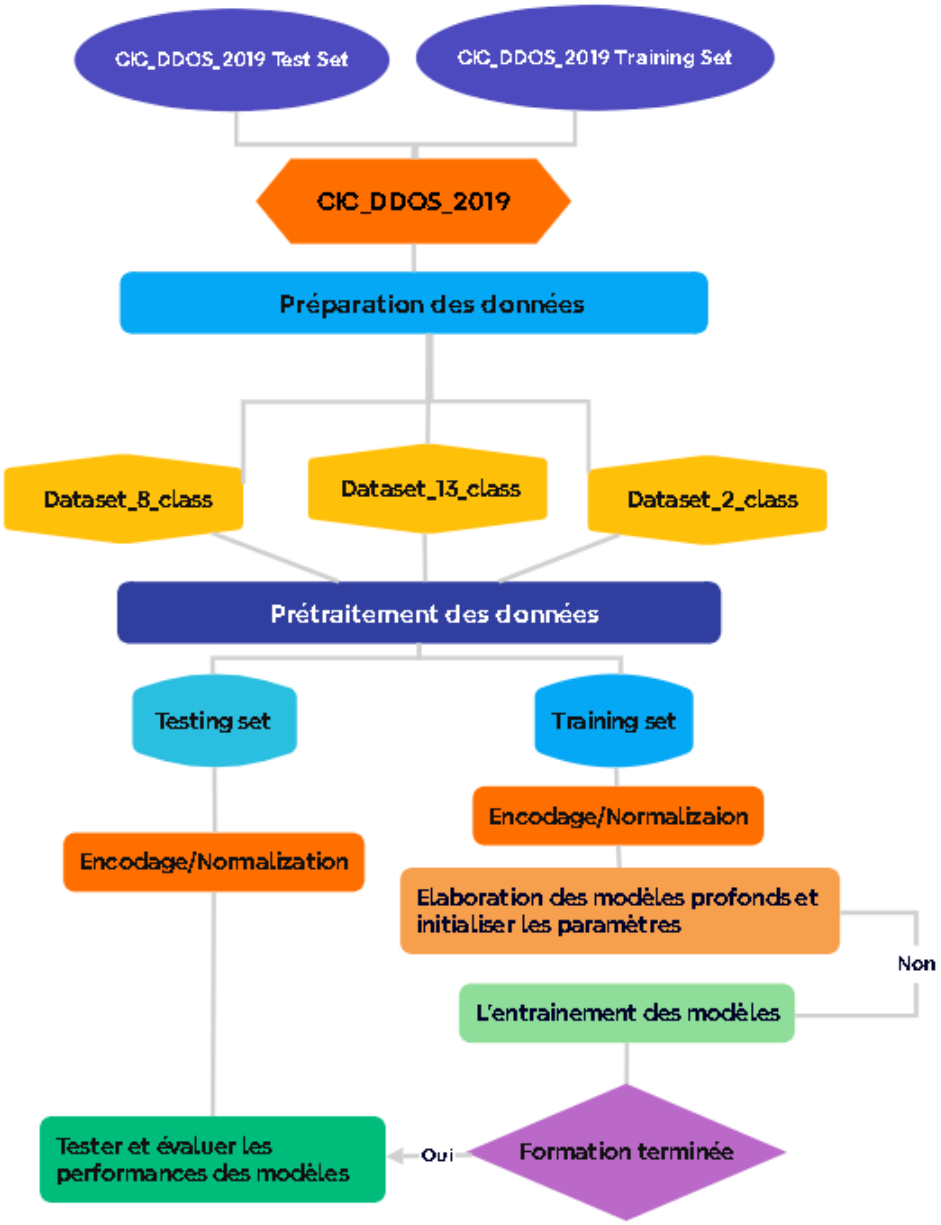


Figure 3.12 – Schéma conceptuel de notre méthode d'implémentation des méthodes DL proposés.

3.8 Conclusion

En conclusion, le chapitre 3 a offert un aperçu complet des méthodes d'apprentissage profond appliquées à la détection des attaques DDoS. Nous avons exploré les fondements des réseaux de neurones convolutionnels (CNN), des réseaux de neurones récurrents (RNN), des deep neural networks (DNN), et des unités de mémoire à court terme (LSTM), soulignant leur rôle

essentiel dans la modélisation et la prévision des attaques sur les réseaux informatiques.

L'étude approfondie du dataset utilisé, la taxonomie des attaques DDoS, et les étapes de préparation des données ont souligné l'importance de méthodologies rigoureuses pour garantir des résultats fiables et précis. La gestion des déséquilibres de données, la normalisation et l'utilisation de techniques comme SMOTE ont été abordées avec un souci constant de qualité et de pertinence dans la mise en œuvre.

Enfin, la conception et l'implémentation d'un système de détection d'intrusion basé sur des modèles de deep learning tels que les CNN, les DNN, et les RNN_LSTM, ont été présentées avec clarté, mettant en évidence l'efficacité de ces approches dans la lutte contre les attaques DDoS.

Chapitre 4

Implémentation,Résultats Et Discussions

4.1 Introduction

Les systèmes de détection d'intrusion basés sur l'apprentissage automatique et profond occupent une place centrale dans la recherche en cybersécurité. Notre étude s'est concentrée sur l'utilisation des ensembles de données CIC_DDoS_2019 et NSL-KDD.

Dans ce chapitre, nous explorons les ensembles de données utilisés ainsi que l'environnement de développement pour créer des modèles d'apprentissage profond et d'apprentissage automatique, avec les détails de leur implémentation. Nous comparons ces approches aux méthodes d'apprentissage profond et d'apprentissage machine traditionnelles, ainsi qu'aux bases de données elles-mêmes. Nous discutons également de l'utilisation de la validation croisée K-Fold pour évaluer la performance de ces modèles.

Enfin, nous présentons les résultats obtenus et entamons une discussion sur ces résultats. L'objectif principal est d'améliorer la capacité des IDS à identifier avec précision les activités malveillantes tout en minimisant les fausses alarmes.

4.2 Environnement de développement

Le domaine du Deep Learning requiert des ressources matérielles importantes, notamment des GPU capables d'effectuer des calculs intensifs. Dans un premier temps, nous avons mis en place un environnement de développement Python sur notre machine en utilisant la plateforme Anaconda.

- **Anaconda :** Anaconda est en effet un outil de distribution Python open source qui simplifie la gestion des packages Python et est très populaire dans les domaines du machine learning et de la science des données. Lorsqu'il est utilisé localement avec Jupyter Notebook, vous pouvez créer et partager des documents incluant du code en direct, des équations, des visualisations et du texte narratif. Cela facilite grandement l'exploration et l'analyse des données, en particulier lorsque vous effectuez des tâches de prétraitement nécessitant des étapes non informatiques.

Ensuite, pour passer à l'apprentissage profond, nous avons orienté vers le cloud, qui offre des ressources de calcul et de mémoire importantes au-delà de nos ordinateurs locaux. Le cloud peut fournir un accès au processeur graphique même sans GPU sur notre propre machine. Un des outils cloud les plus utilisés pour l'apprentissage automatique est Google Colab[58].

- **Google Colab :** est un service Cloud basé sur Jupyter Notebooks, permet de développer des applications en Deep Learning et Machine learning en Python, il offre un processeur GPU gratuit, 12 Go de RAM et plus de 100 Go de stockage. Pour l'accès dans ce service il nous suffit simplement d'avoir qu'un compte Google. Pour le langage de développement on a choisi Python, un langage de programmation interprété, multi- paradigme et multi-plateformes, il est aussi un langage plus commun et plus populaire pour l'apprentissage automatique et l'intelligence artificielle grâce à sa flexibilité et aussi parce qu'il a un nombre important de bibliothèques logicielles open source disponible. Permet ses bibliothèques utilisées dans notre projet : Pandas, Numpy, Scikit-Learn . . .etc. Pandas et Numpy sont utilisés pour la manipulation des données (le chargement, la réorganisation et le traitement des données), Scikit-Learn nous permet d'expérimenter différentes techniques et algorithmes d'apprentissage automatique et d'analyse de données prédéfinis rapidement et facile a utilisé. les frameworks TensorFlow et Keras ont été choisir pour l'implémentions des méthodes deep learning proposé. TensorFlow est une bibliothèque de deep Learning open source développée par Google utilisée pour effectuer des opérations numériques complexes et plusieurs autres tâches pour modéliser les architectures de Deep Learning. il peut déployer facilement des calculs sur plusieurs plates-formes comme les CPU, les GPU. Keras est une API de haut niveau qui vise à créer et à entraîner des modèles de Deep Learning basé sur python. Elle a été développée dans le but de permettre des expérimentations rapides. Parmi ses points forts :

  • Était capable d'aller de l'idée au résultat avec le plus faible délai possible. Et ça c'est la clé d'une recherche efficace.

  • Supporte à la fois les réseaux convolutifs (CNN) et les réseaux récurrents (RNN) ainsi que la combinaison des deux items Pas de fichiers séparés de configuration des modèles, tout est déclaré dans le code.

  Fonctionne sur CPU et GPU : Bien que Keras offre toutes les fonctionnalités générales nécessaires à la création de modèles d'apprentissage en profondeur, il ne propose pas autant de fonctionnalités avancées que TensorFlow. TensorFlow offre un contrôle plus poussé pour développer des types spécifiques de modèles et permet une meilleure compréhension de ce qui se passe à l'intérieur d'un réseau de deep learning.

  Pour tirer parti des avantages des deux frameworks, nous avons utilisé TensorFlow comme backend avec Keras. Nos expérimentations ont été menées dans l'environnement de Google Colab, ce qui nous a permis de bénéficier d'un processeur GPU et de 12 Go de RAM.

4.3 Base de données CIC_DDoS_2019

Nous évaluons notre classifieur proposé en utilisant le nouvel ensemble de données CIC-DDoS 2019 ,qui a été partagé par l'Institut canadien de cybersécurité [59]. L'ensemble de données contient une grande quantité de différentes attaques DDoS pouvant être réalisées via des protocoles de couche applicative utilisant TCP/UDP.

Cet ensemble de données comprend également 13 différents attaques DDoS le plus à jour, L'ensemble complet des données contient 50063112 instances, dont 50006249 sont des attaques DDoS et que 56863 sont des instances d'un trafic réseau bénin (légitime / normale), le pourcentage pour chaque type d'attaque DDoS est indiqué dans le figure 4.1. Ce base de données contient également 86 caractéristiques (Features), ou 6 d'eux sont étiquetés et caractérisés le flux lui-même, en fonction de **Source IP, Source Port, Destination IP, Destination Port, Protocole et Timestamp** (temps d'attaques), et plus de 80 caractéristiques sur le flux trafic du réseau . La Figure 4.1 résume les différentes attaques dans l'ensemble de données CIC_DDoS_2019, Après avoir regroupé les deux jours et renommé les attaques.

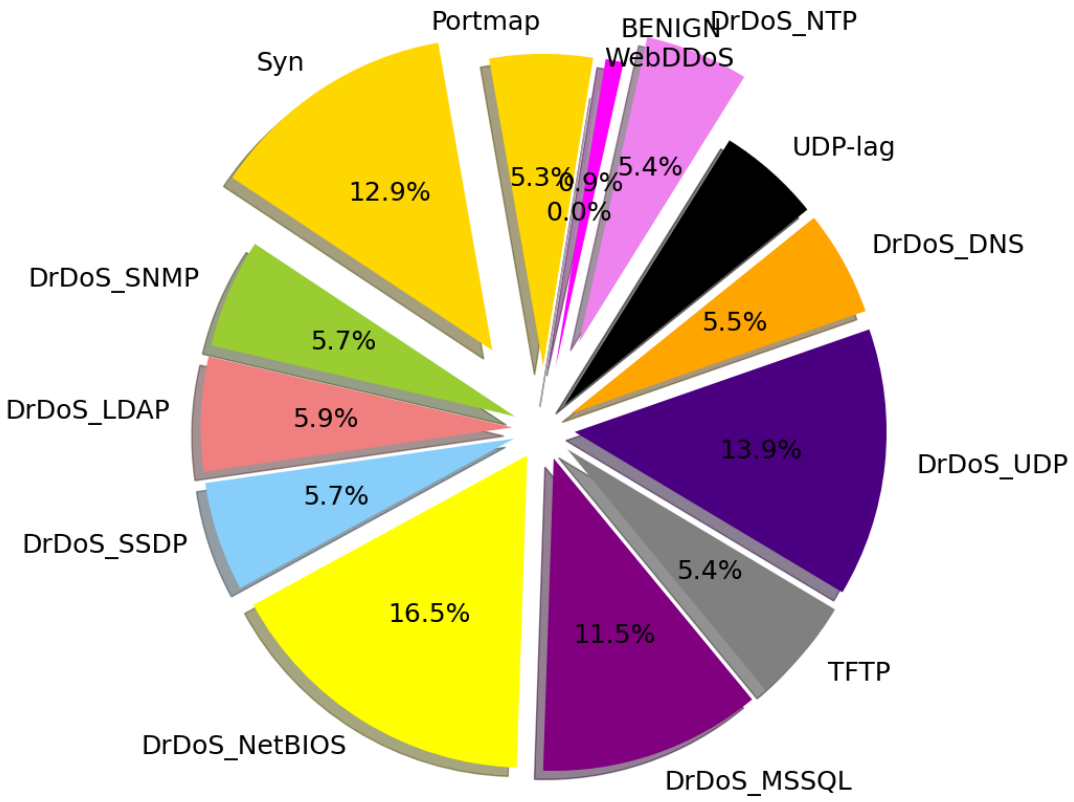


Figure 4.1 – Pourcentages des différentes d'attaques.

4.4 Selection de model d'apprentissage profond

4.4.1 Système de détection d'intrusion par Réseau de neurones convolutionnel (CNN) avec classification 8_classes :

Le figure 4.4 qui illustre les résultats de tests de notre classifieurs DL (DNN, CNN, RNN) et les classiffieurs ML.Il est clair que notre méthodes DL surpassent largement toutes les les autres méthodes ML, avec un taux de précision et de rappelle (Recall) élevé. Parmi notre méthodes le CNN a les meilleurs résultats avec une précision 91%, grâce à sa capacité de reconnaissance des motifs discriminatoires pour chaque classe, le tableau 4.1 montre le rapport de classification des attaques utilisant la méthode CNN.

Classe	Précision	Rappel	F1-score	Support
BENIGN	1.00	1.00	1.00	40184
LDAP	1.00	1.00	1.00	39994
MSSQL	0.98	0.99	0.98	39944
NetBIOS	0.77	0.43	0.55	39529
Portmap	0.61	0.87	0.71	39997
Syn	1.00	1.00	1.00	39800
UDP	0.99	0.97	0.98	40142
UDPLag	0.98	0.99	0.99	40236
accuracy			0.91	319826
Macro avg	0.92	0.91	0.90	319826
Weighted avg	0.92	0.91	0.90	319826

Table 4.1 – Le rapport de classification 8_classes avec CNN.

Comme les performances des systèmes IDSs reposent sur sa capacité de différencier les comportements normaux des comportements malicieux, le CNN a été bien identifié. La classe minoritaire BENIGN qui représente le trafic normal avec une précision converge **1** et un rappel de 100%.Cette précision signifie le nombre des fausses alarmes est minimale, le rappel signifié que le modèle est efficace dans l'identification des attaques réseaux avec un taux de fausse négative (FN) réduits. Le modèle CNN a été bien identifié aussi certains types d'attaques comme **LDAP, NetBIOS, UDP et Syn**.Cepandant, la classe **NetBIOS** a été mal classé avec 43% de rappel.

La figure 4.2 montre l'utilisation du sur-échantillonnage via SMOTE a permis d'améliorer l'exactitude des modèles de Deep Learning dans un contexte multiclasses, ce qui démontre l'effet bénéfique de cette méthode pour réduire le déséquilibre des classes. Avec SMOTE, l'exactitude du modèle DNN a augmenté de 0,85 à 0,90, celle du CNN de 0.86 à 0,91 et celle du RNN de 0,85 à 0.86, ce qui suggère que le déséquilibre des classes peut avoir un impact significatif sur les performances du modèle. En produisant des échantillons synthétiques destinés aux classes minoritaires, SMOTE a contribué à rendre l'ensemble de données plus équilibré, ce qui a renforcé la capacité de généralisation des modèles.

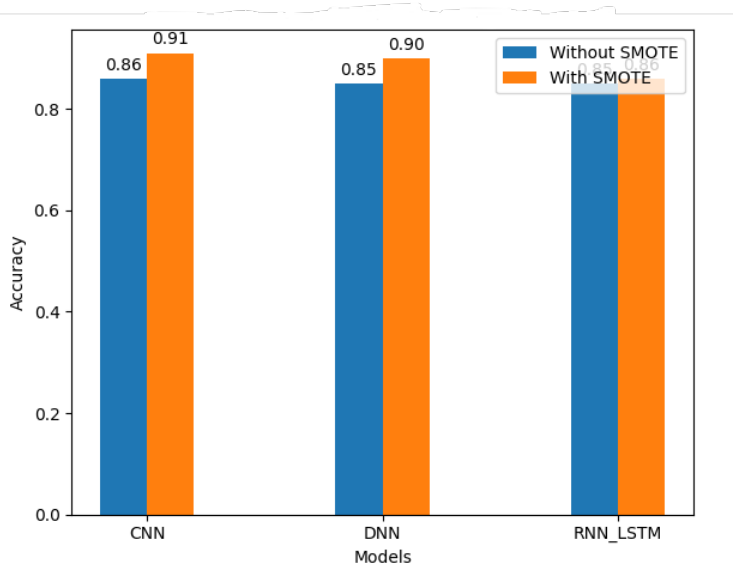


Figure 4.2 – Accuracy des 3 modèles avec et sans utiliser le sur-échantillonnage via SMOTE.

4.4.2 Système de détection d'intrusion par d'apprentissage profond avec classification binaire :

Dans cette expérimentation, nous avons testé l'efficacité de notre méthodes dans la détection des attaques DDoS quelles que soient ses types, les résultats de détection ont montré dans le tableau 4.2 On peut constater clairement que notre méthode DL proposée a été très performante. La classification binaire avec la classe BENIGN qui représente les trafiques normaux et la classe Attack qui représente les trafics malicieux a été effectué avec un taux de vrai positive (TPR) nettement élevé qui converge vers 1 et un taux de fausse négative très réduit. Cela signifie que les modèles proposés ont bien identifié le comportement normal du trafic ce qui permet de réduire le nombre des fausses alarmes aussi qu'ils ont bien identifié le comportement malicieux du trafic ce qui permet une sécurité très élevée contre ce type des attaques réseau.

Modèle	TP	FP	TN	FN	TPR%	FPR%
DNN	342495	9	343422	147	99.96%	0.0026%
CNN	342482	16	343415	160	99.96%	0.0047%
RNN	209735	52	210268	133	99.94%	0.0247%

Table 4.2 – Le rapport de classification binaire.

4.4.3 Système de détection d'intrusion par d'apprentissage profond avec classification 13_classes :

Dans cette implementation, nous avons évalué le sous ensemble de données de premier jour 3.1 par le biais de trois modèles d'apprentissage profond .Les résultats sont présentés dans le

tableau 4.3. Nous avons utlisé le Base de données_3 (3.7) pour évaluer l'efficacité de notre modèles face aux différentes d' attaques DDoS et sa capacité de différencier ces attaques entre eux et aussi avec le trafic bénin . La meilleure précision que nous ayons obtenue est 79%, cela se réfère au nombre de classes concernées dans cette étude, La matrice de confusion normalisée 4.3 ci-dessous montre que la classe BENIGN a été bien prédite avec d'autres classes d'attaques qui ont été prédites convenablement.Cependant, certains types d'attaques comme : DrDoS_LDAP, DrDoS_SSDP, DDoS_NetBIOS ont été mal classé, cela signifie que ces attaques ayant des comportements similaires et partagent certaines propriétés entre eux ce qui rendent la tache de reconnaissance de ces derniers plus difficiles.

Méthode	Précision	Rappel	F1-score
DNN	0.79	0.76	0.74
CNN	0.79	0.76	0.75
RNN	0.76	0.74	0.71

Table 4.3 – Les résultats des méthodes DL pour la Base de données_3.

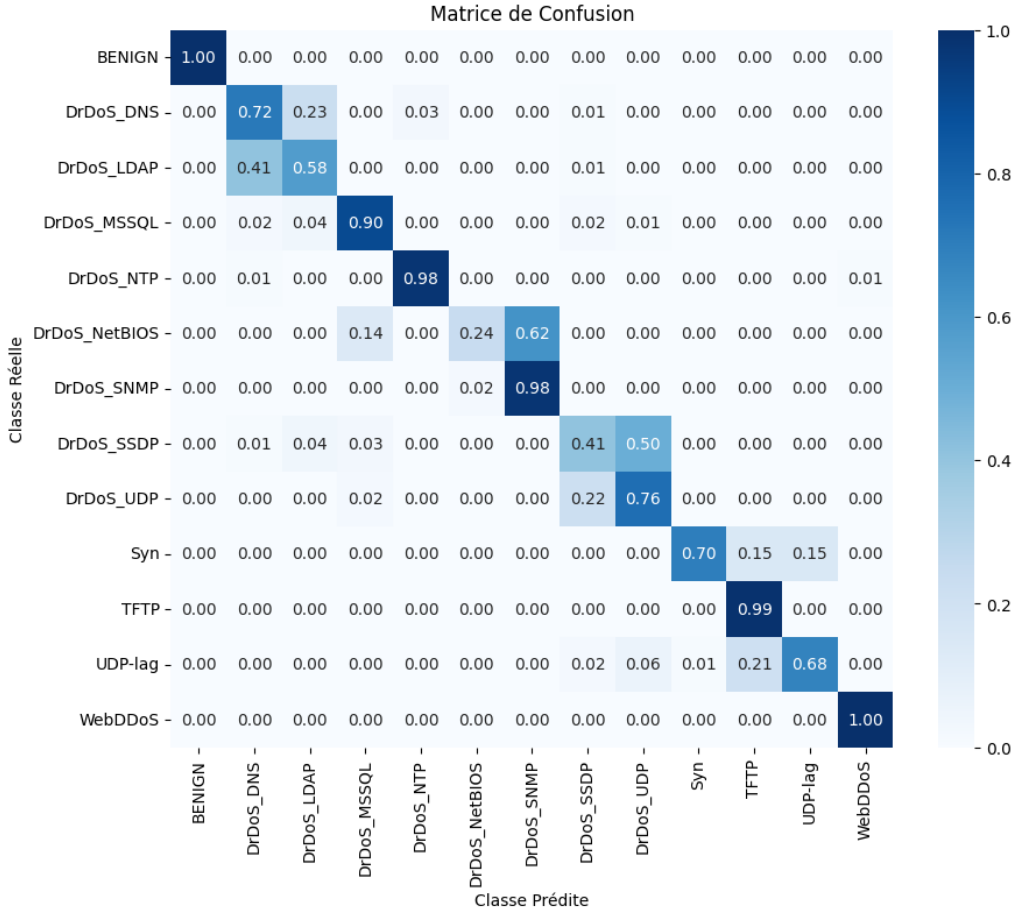


Figure 4.3 – Matrice de confusion du CNN (13_classes).

Les bonnes caractéristiques discriminantes de la classe BENIGN ont été examinées pour obtenir un taux de vrais négatifs (TNR) très élevé pour tous les modèles proposés. Le taux de

vrais négatifs (TNR) est défini comme suit :

$$\text{TNR} = \frac{TN_{\text{BEGNIN}}}{TN_{\text{BEGNIN}} + FP_{\text{BEGNIN}}}$$

Les modèles qui ont un TNR convergent vers 1, ce qui nous a donnés des résultats très satisfaits en matière de taux de détection des attaques (TPR ; sans considérer le type d'attaque) qui a été extrêmement élevé 100%, ainsi que le taux de fausses alarmes (FPR) a été trop faible, 1.99% pour le DNN , 1.96% pour CNN et 2.20% pour le RNN.

4.5 L'évaluation de modèle d'apprentissage machine

Les résultats du tableau 4.4 ont évalué cinq algorithmes d'apprentissage Machine (**Random Forest**, **Decision Tree**, **KNeighbors**, **Logistic Regression**, et **Naive Bayes**) sur le sous ensemble de données de classification 8_classes.

Algorithmes	**Accuracy(%)**	**Loss(%)**
Random Forest	87.08%	0.24%
Decision Tree	84.65%	5.53%
KNeighbors	83.56%	0.87%
Logistic Regression	82.67%	0.35%
Naive Bayes	49.14%	1.12%

Table 4.4 – Les résultats des algorithmes ML.

4.6 Comparaison des approches d'apprentissage auto- matique utilisées

Nous avons utilise algorithmes d'apprentissage automatique, dont Random Forest,Naive Bayes,KNN et Logistic Regression et l'autre en utilisant l'apprentissage en profondeur (CNN, RNN, DNN) sur l'ensemble de données CIC_DDoS_2019. Les mesures d'évaluation ainsi que les résultats obtenus sont illustrées dans le tableau ci-dessous 4.5 :

Méthode	**Précision**	**Rappel**	**F1-Score**
DNN	0.91	0.90	0.90
CNN	0.92	0.91	0.90
RNN	0.85	0.86	0.82
Random Forest	0.86	0.87	0.86
Naive Bayes	0.49	0.41	0.37
Logistic Regression	0.80	0.83	0.80
K-Nearest Neighbors	0.86	0.83	0.85

Table 4.5 – Les résultats des méthodes DL proposées et les algorithmes ML.

Les résultats du tableau 4.5 démontrent de manière significative que les modèles d'apprentissage en profondeur (DNN, CNN, RNN) surpassent généralement les algorithmes d'apprentissage automatique classiques (Random Forest, Naive Bayes, Logistic Regression, K-nearest neighbors)pour les détection des attaques DDoS. Cela se manifeste à travers des mesures telles que la précision, le rappel et le score F1, où le CNN se démarque avec une précision estimée à 92%, la plus élevée parmi toutes les méthodes. En revanche, Naive Bayes affiche des performances nettement inférieures, avec une précision de seulement 49% en comparaison.

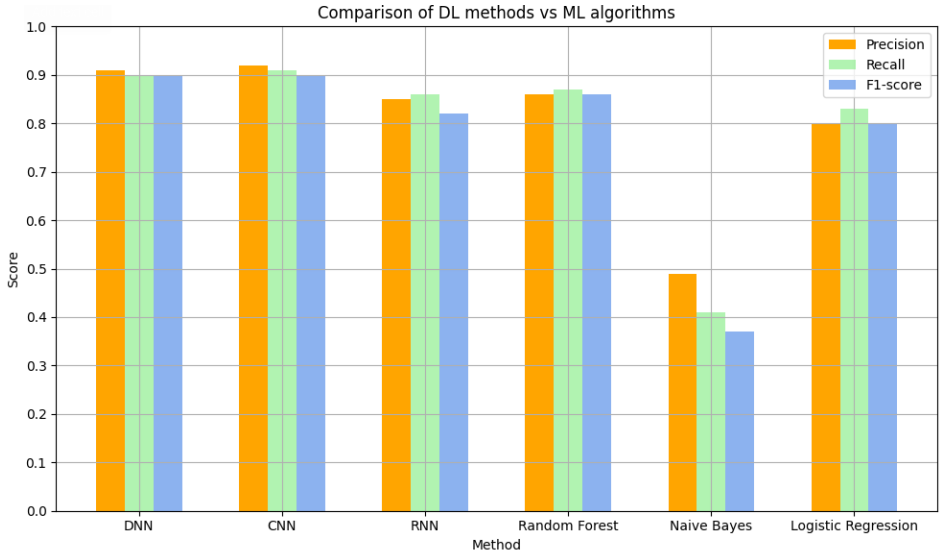


Figure 4.4 – Comparaison des résultats entre les méthodes DL proposés et les algorithmes ML.

4.7 Résultat et discussions

Nous avons implémenté trois modèles de deep learning : DNN, CNN et RNN_LSTM, ainsi que cinq algorithmes de machine learning. Ces modèles de deep learning ont été formés et testés sur trois sous-ensembles différents de la base de données CIC_DDoS_2019. Plusieurs tests ont été effectués afin de déterminer les meilleurs modèles. Une fois que nous avons obtenu un modèle avec un taux d'erreur minimal et une précision maximale, nous avons ensuite testé ce modèle sur le sous-ensemble de test. Les résultats de ces tests sont présentés dans les trois figures 4.5 ,4.6 et 4.7 ci-dessus. Les résultats des algorithmes de machine learning testés sur un sous-ensemble de la même base de données sont présentés dans les figures 4.8.

Système de détection d'intrusion par la apprentissage profond(classification binaire) réalisée sur le sous-ensemble de données du CIC_DDoS_2019, comme indiqué dans le tableau 3.5, les modèles ont été évalués directement sur l'ensemble de test . Le CNN et le DNN ont été formés sur 15 époques, tandis que le RNN a été formé sur 10 époques. Pour tous les modèles, nous remarquons une augmentation constante de l'exactitude d'apprentissage et de validation du début à la fin, atteignant une valeur maximale proche de 1. De plus, la valeur de perte diminue considérablement pendant l'entraînement et l'évaluation, atteignant une valeur minimale

proche de 0. Cela indique que ces modèles apprennent de manière efficace et font de meilleures prédictions à chaque itération d'optimisation.

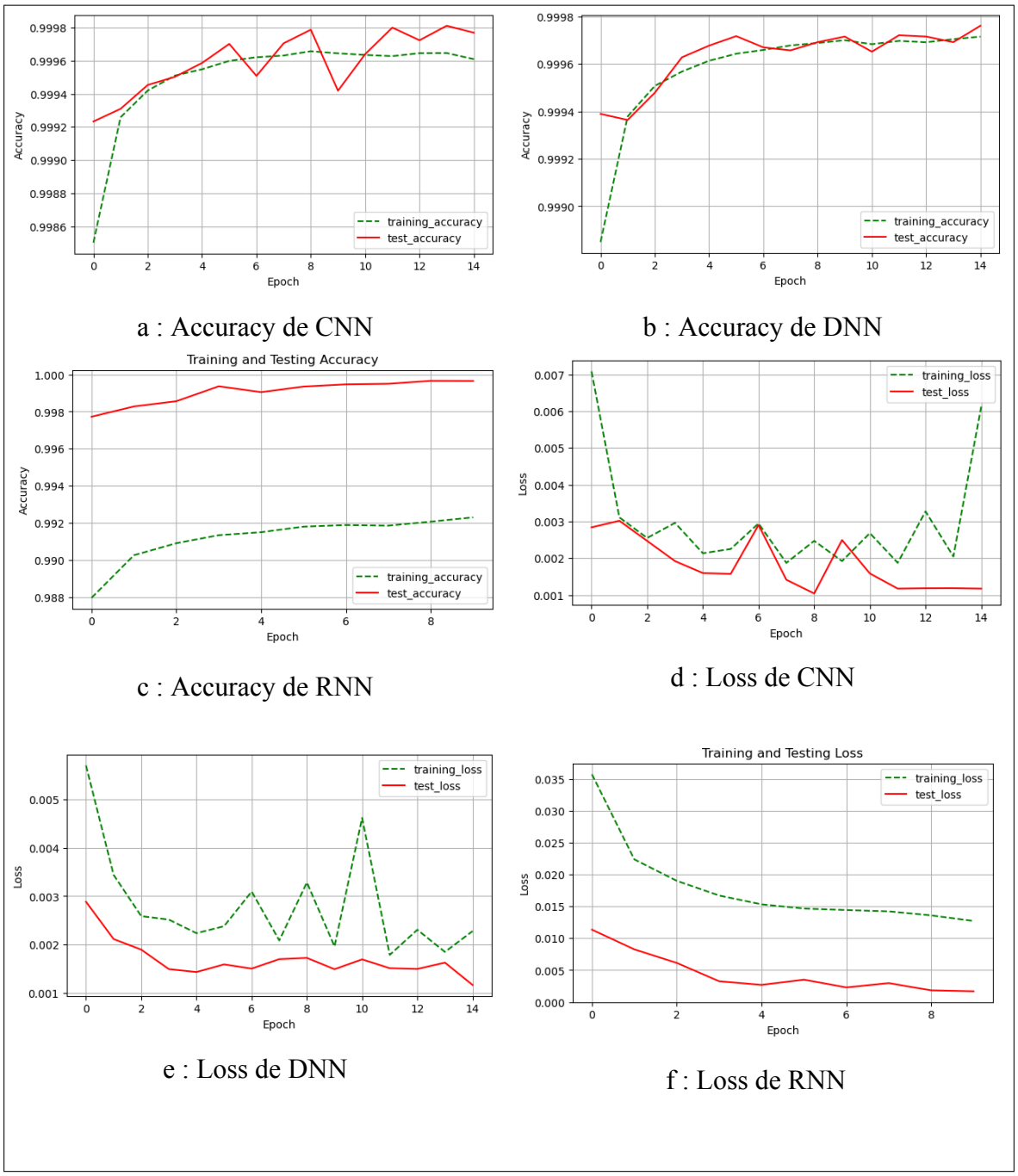


Figure 4.5 – L'exactitude et la perte des modèles proposés par rapport aux époques d'apprentissage et de validation pour Base de données_1 (2_classe).

Système de détection d'intrusion par l'apprentissage profond (classification en 8 classes) réalisée sur le sous-ensemble de données du CIC_DDoS_2019, indiqué dans le tableau 3.6, les données d'entraînement ont été divisées en deux : 80 % pour l'apprentissage et 20 % pour l'évaluation. L'apprentissage a pris beaucoup de temps, même si les modèles étaient relative-

ment simples, étant formés sur 15 à 20 époques. Les modèles ont obtenu une précision très élevée : 91 % pour le CNN, 90,2 % pour le DNN et 86,4 % pour le RNN. Nous notons ici que les trois modèles convergent vers une valeur de perte minimale, avec des valeurs de perte d'apprentissage et d'évaluation presque identiques. Ce qui indique que ces modèles seront généralisés bien au-delà de l'ensemble d'apprentissage. Ensuite, nous avons testé ces modèles sur l'ensemble de test.

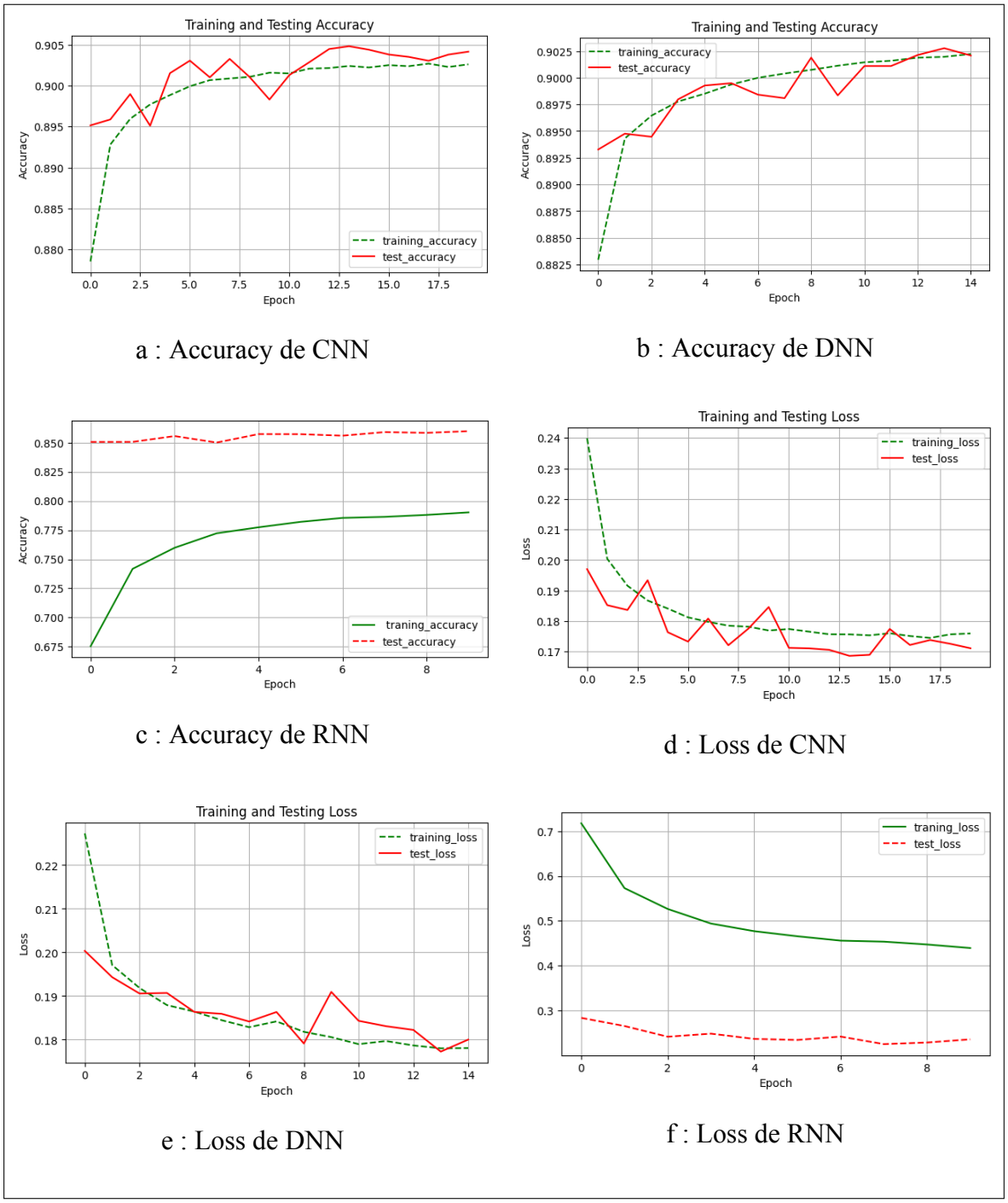


Figure 4.6 – L'exactitude et la perte des modèles proposés par rapport aux époques d'apprentissage et de validation pour Base de données_2 (8_classe)

Système de détection d'intrusion par la apprentissage profond (classification en 13 classes) réalisée sur le sous-ensemble de données du CIC_DDoS_2019, comme indiqué dans le tableau 3.7, les données ont été divisées en deux parties : 80 % pour l'apprentissage et 20 % pour le test. Les modèles ont été évalués et testés sur le sous-ensemble des données d'apprentissage. Le CNN et le DNN ont été formés sur 30 époques, tandis que le RNN a été formé sur 10 époques. Il est à noter que la valeur de perte d'apprentissage, la valeur de perte de test, ainsi que l'exactitude d'apprentissage et l'exactitude de test convergent toujours vers la même valeur pour tous les modèles. Cela signifie que les problèmes de sur-apprentissage et de sous-apprentissage ont été correctement traités. La meilleure précision obtenue était de 76,4 % pour le CNN. Les valeurs de perte des trois modèles étaient proches, diminuant de 1,3 à 0,4.

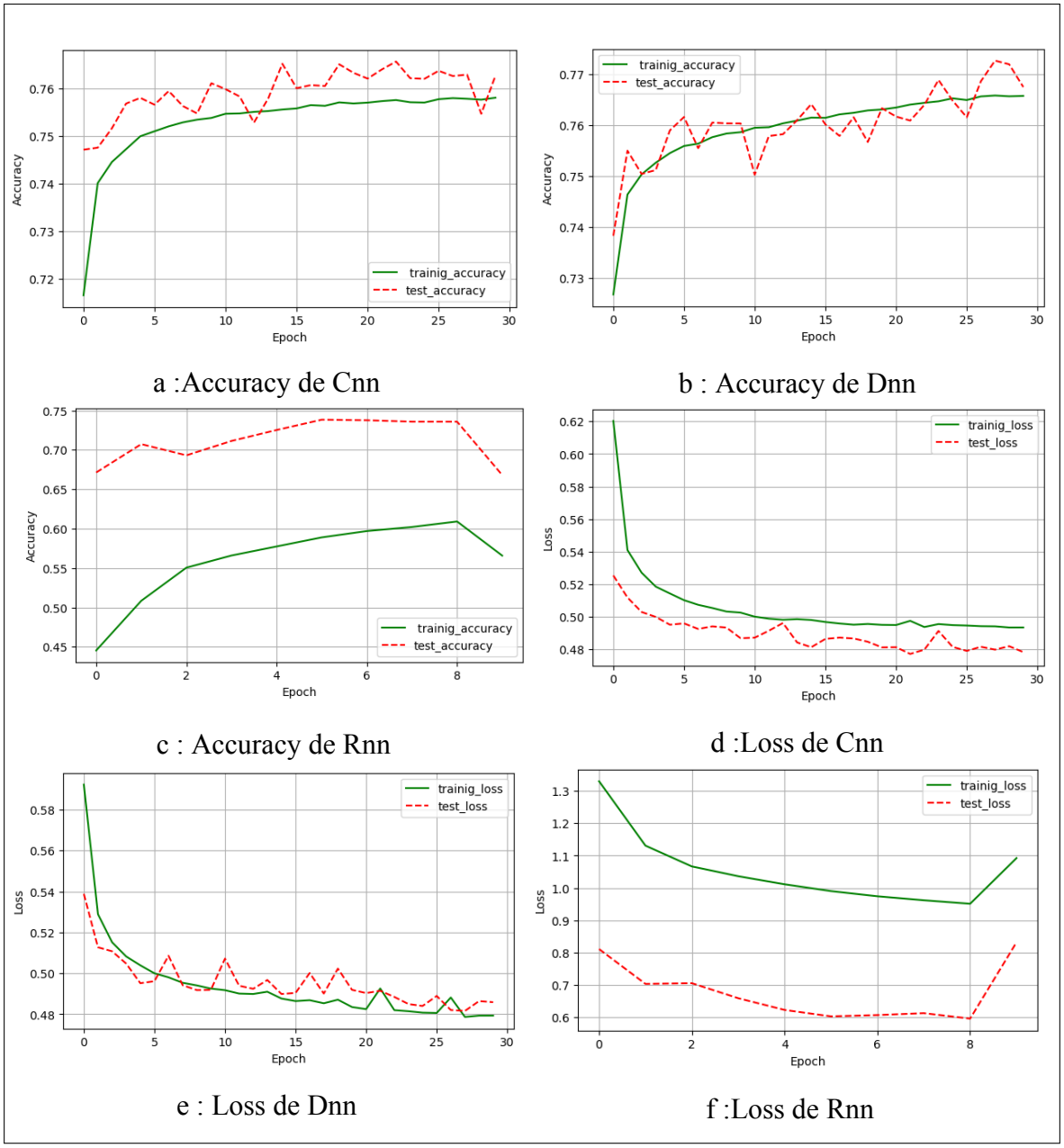


Figure 4.7 – L'exactitude et la perte des modèles proposés par rapport aux époques d'apprentissage et de validation Base de données_3 (13_classe).

Système de détection d'intrusion par la apprentissage machine(utilisant la classification en 8 classes) réalisée sur l'ensemble de données d'entraînement CIC_DDoS_2019, comme indiqué dans le tableau 3.6, la régression, la Random Forest, les KNeighbors les plus proches et les algorithmes logistiques ont pris plus de temps que l'arbre de décision tree et Naive Bayes, qui ont pris un temps négligeable. Grâce aux résultats obtenus, l'algorithme Random Forest a obtenu la meilleure performance avec une précision de 87,08% et une perte de 0,24%. Par contre, Naive Bayes a donné la plus faible performance avec une précision de 49,14% et une perte de 1,12%. Ces résultats montrent que les algorithmes arborescents, en particulier Random Forest, sont plus efficaces pour classer les différentes attaques DDoS.

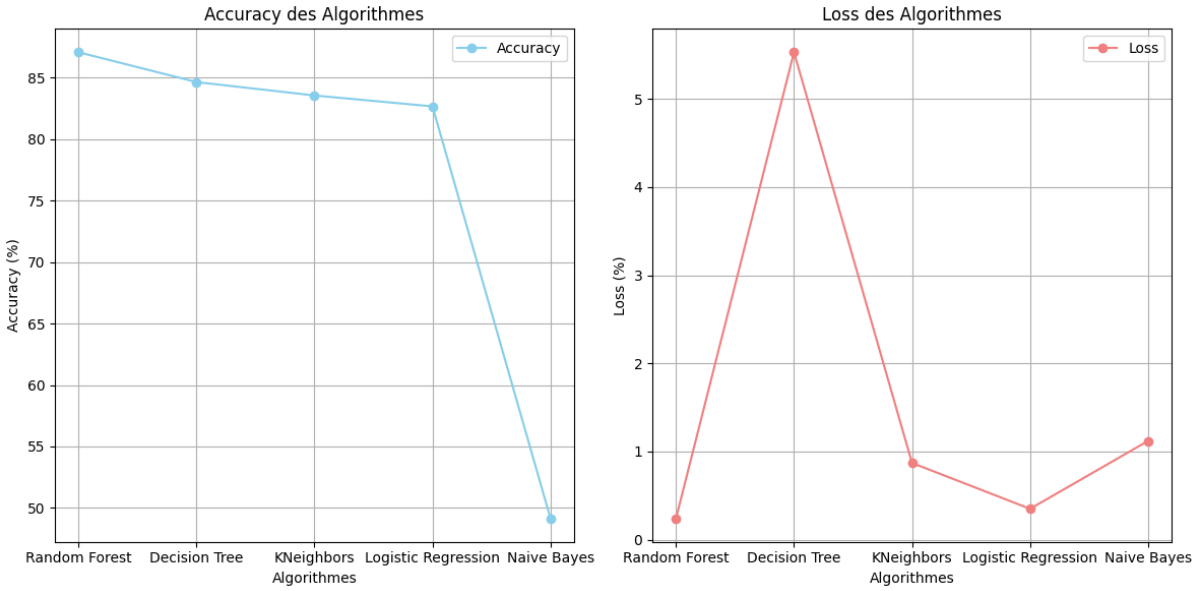


Figure 4.8 – Les résultats des algorithmes ML.

4.8 La validation Croisée K_FOLD

La validation croisée k-fold est l'un des types de validation croisée qui vise à évaluer les performances d'un algorithme en divisant aléatoirement un échantillon de données en k groupes. Ensuite, l'un de ces groupes k-fold est utilisé comme données de test, tandis que les autres groupes sont utilisés comme données d'entraînement.

K-Fold Cross-Validation est une technique largement utilisée en apprentissage automatique et apprentissage approfondi pour évaluer la performance des modèles de manière robuste et fiable. Elle permet d'estimer la capacité de généralisation d'un modèle à partir d'un échantillon de données sans avoir besoin de données de test séparées.Parmi les points forts de cette technique sont les suivants :

. Utilisation Efficace des Données : Chaque point de données est utilisé à la fois pour l'entraînement et le test, maximisant ainsi l'utilisation du jeu de données.

. Évaluation Robuste : En utilisant différents ensembles de test, vous réduisez le risque de biais dû à un seul ensemble de test.

. Réduction du Surapprentissage : La validation croisée K-Fold aide à prévenir le surapprentissage en diversifiant les ensembles de test.

Comme mentionné précédemment, les étapes suivies dans cette étude consistent à diviser les données en ensembles d'entraînement et de test, avec l'utilisation de sous ensemble de données et une division croisée de kfold=5, ce qui donne 80% de données d'entraînement et 20% de données de test à chaque étape. L'image 4.9 illustre la validation croisée dans cette étude.

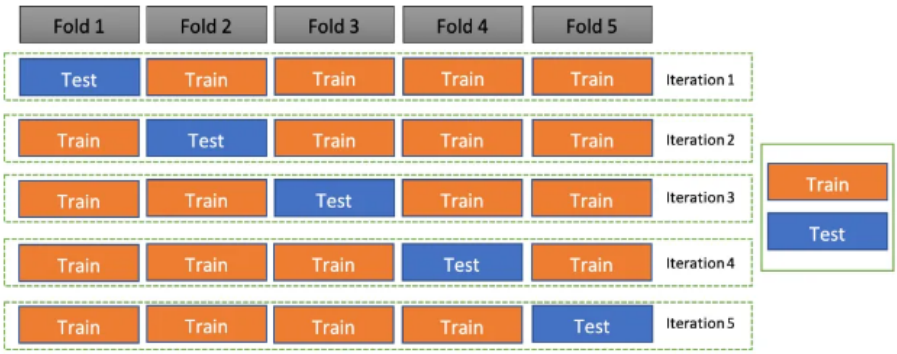


Figure 4.9 – Application de validation croisée sur la Base de données_2.

Les processus des pré-traitements des données commence ici par la création d'un modèle avec des données d'entraînemen, suivi de son implémentation sur des données de test en utilisant les trois algorithmes de DL (CNN et DNN). Le tableau 4.6 ce qui donne comme résultat final des valeurs d'exactitude et des résultats de classification comme suit.

Model	KFold	Test Accuracy	Précision	Rappel	F1-score	Test Loss
CNN	1	0.859	0.892	0.828	0.828	0.0251
	2	0.856	0.892	0.827	0.822	0.0254
	3	0.857	0.885	0.829	0.824	0.0254
	4	0.858	0.892	0.829	0.826	0.0250
	5	0.858	0.890	0.829	0.827	0.0256
	Moyenne	0.858	0.890	0.828	0.825	0.0253
DNN	1	0.857	0.883	0.830	0.824	0.0259
	2	0.858	0.883	0.831	0.825	0.0261
	3	0.859	0.884	0.832	0.826	0.0254
	4	0.859	0.891	0.830	0.826	0.0272
	5	0.858	0.892	0.818	0.821	0.0301
	Moyenne	0.858	0.887	0.828	0.824	0.0301

Table 4.6 – Résultats des tests de performance des modèles CNN et DNN.

Par conséquent, à travers les résultats présentés dans le tableau, nous pouvons conclure qu'en divisant les données en 20 % pour l'entraînement et 80 % pour le test, et en utilisant une validation croisée à 5 plis, le CNN a obtenu la meilleure précision de 89,20 % avec une perte de 2.51% dans le premier pli (Kfold 1). En comparaison, le DNN a atteint une précision de 89,10 % avec une perte de 2.54% dans le troisième pli (Kfold 3). Après avoir mené cette expérience,

nous avons constaté que, en termes d'évaluation, la meilleure classification est obtenue avec le modèle CNN.

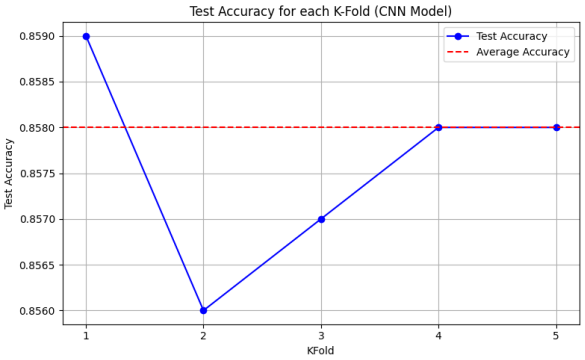


Figure 4.10 – Test Accuracy de modèle CNN avec validation croisée K_FOLD .

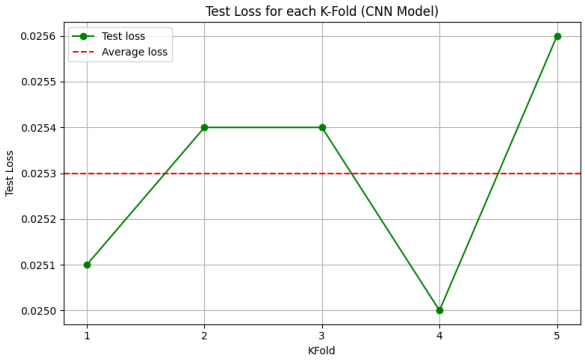


Figure 4.11 – Test Loss de modèle CNN avec validation croisée K_FOLD.

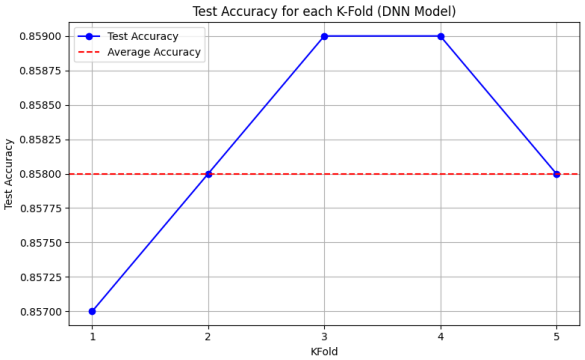


Figure 4.12 – Test Accuracy de modèle DNN avec validation croisée K_FOLD.

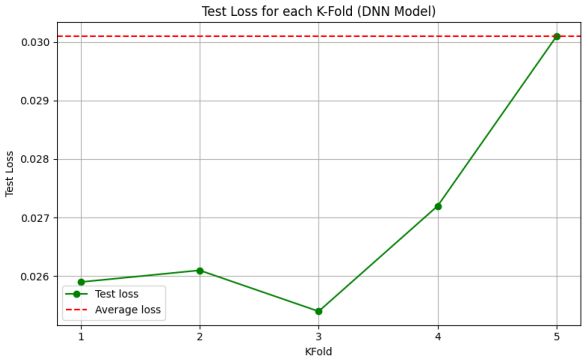


Figure 4.13 – Test Loss de modèle DNN avec validation croisée K-FOLD

4.9 Base de données NSL-KDD

L'ensemble de données NSL-KDD est une version améliorée de l'ensemble de données KDD99 précédent. Dans ce projet, le jeu de données NSL-KDD est analysé et utilisé pour évaluer l'efficacité de divers algorithmes de classification dans la détection des anomalies dans les modèles de trafic réseau.

L'ensemble de données NSL-KDD, comprenant 41 attributs et un attribut de classe, apporte des améliorations par rapport au jeu de données KDD99 en supprimant les instances en double pour éviter des résultats de classification biaisés.

L'ensemble de données NSL-KDD est analysé et classé en quatre groupes différents représentant les types d'attaques les plus courants. Une analyse approfondie est réalisée sur les jeux de données d'entraînement et de test. L'entraînement est effectué sur les données KDDTrain qui contiennent 22 types d'attaques, tandis que les tests sont effectués sur les données KDDTest qui contiennent 17 types d'attaques supplémentaires.

Ces attaques peuvent être catégorisées en quatre types différents avec certaines propriétés communes, comme illustré dans le tableau 4.7 pour l'entraînement et les tests [60].

Denial of Service (DoS) : Une tentative malveillante de bloquer les ressources et services du système ou du réseau.

Probe : Cette attaque collecte des informations sur les vulnérabilités potentielles du système cible qui peuvent ensuite être utilisées pour lancer des attaques sur ces systèmes.

Remote to Local (R2L) : Capacité non autorisée à envoyer des paquets de données à un système distant sur le réseau et à obtenir un accès soit en tant qu'utilisateur soit en tant qu'administrateur pour mener des activités non autorisées.

User to Root (U2R) : Dans ce cas, les attaquants accèdent au système en tant qu'utilisateur normal et exploitent les vulnérabilités pour obtenir des privilèges administratifs.

Catégories d'attaques	Attaque
DOS	Back,Land,Neptune,Pod,Smurf,Teardrop ,Mail-bomb,Processtable,Udpstorm,Apache2,Worm
Probe	Satan,IPsweep,Nmap,Portsweep,Mscan,Saint
R2L	Guess_password,Ftp_write,Imap,Phf, Multi hop,Warezmaster,Xlock,Xsnoop,Snmpgue ss ,Snmpgetattack,Httptunnel,Sendmail,Named
U2R	Buffer_overflow,Loadmodule,Rootkit,Perl,Sqlattack, Xterm,Ps

Table 4.7 – Les attaques dans les ensembles de données d'entraînement et de test

Le tableau 4.8 présente les attaques dans l'ensemble d'entraînement et de test de NSL-KDD, tandis que le tableau 4.8 décrit les classes d'attaques avec le nombre d'instances par classe. NSL-KDD contient 125 973 instances dans l'ensemble d'entraînement et 22 544 instances dans l'ensemble de test.

Ensemble de données	Classe	Nb d'instances
Entraînement	Normal	67343
	Dos	45927
	Probe	11656
	R2L	995
	U2R	52
Test	Normal	9711
	Dos	7458
	Probe	2421
	R2L	2754
	U2R	200

Table 4.8 – Les classes et nombre d'instances dans l'ensembles d'entraînement et de test.

4.9.1 Les pré-traitements des données

Le processus de prétraitement des données est essentiel pour rendre le jeu de données compatible avec les algorithmes d'apprentissage machine et d'apprentissage profond, afin de créer un modèle de classification fiable et cohérent. Pour le jeu de données NSL-KDD, trois étapes principales ont été suivies : la numérisation, la normalisation et la sélection des attributs.

La numérisation consiste à transformer les attributs catégoriels en valeurs numériques. Par exemple, les valeurs "tcp", "udp" et "icmp" de l'attribut "protocol_type" sont respectivement encodées en 0, 1 et 2. Les types d'attaques à la fin du jeu de données sont également convertis en leurs catégories numériques. Ensuite, la normalisation ajuste ces valeurs numériques pour qu'elles se situent toutes entre 0 et 1, en utilisant la méthode **Min-Max** pour garantir l'efficacité et les performances du modèle.

$$\text{Min-Max}(x) = \frac{x - \text{Min}(x)}{\text{Max}(x) - \text{Min}(x)}$$

où x est la valeur à normaliser, $\text{Min}(x)$ est la valeur minimale dans la plage, et $\text{Max}(x)$ est la valeur maximale dans la plage.

Pour améliorer les performances de notre modèle de détection d'intrusions basé sur le jeu de données NSL-KDD, nous avons choisi les attributs les plus pertinents parmi les 42 disponibles. Utiliser tous les attributs aurait pu avoir un impact négatif sur les performances du modèle en termes de ressources et de temps de réponse. Nous avons donc suivi plusieurs étapes pour sélectionner les attributs les plus significatifs :

1. **Calcul de l'entropie :**

$$H(x_i) = -\sum_{j=1}^{n} p(x_j|c_1)\log_2 p(x_j|c_1) + p(x_j|c_2)\log_2 p(x_j|c_2)$$

Où :
— c_1, c_2 représentent les deux classes de classification (normale, attaque).
— x_i représente un attribut.
— x_j représente une valeur particulière de l'attribut x_i.
— n dénote le nombre de valeurs de l'attribut x_i.
— p représente la probabilité.

2. **Calcul du gain :**

$$\text{Gain} = \text{Entropie\_E} - H(x_i)$$

Où :

$$\text{Entropie\_E} = \frac{nb_{normal}}{nb_{connexion}}\log_2\left(\frac{nb_{normal}}{nb_{connexion}}\right) + \frac{nb_{attack}}{nb_{connexion}}\log_2\left(\frac{nb_{attack}}{nb_{connexion}}\right)$$

Où :
— nb_{normal} représente le nombre de connexions classées comme normales.

— nb_{attack} représente le nombre de connexions classées comme une tentative d'attaque.
— $nb_{connexion}$ représente le nombre total de connexions dans la base d'apprentissage.

3. **Élimination des attributs :** Nous avons retiré les attributs dont le gain était inférieur à un seuil donné (0,5 dans notre cas).
4. **Génération du modèle :** Nous avons construit le modèle de classification en utilisant les attributs restants après la troisième étape. Les attributs sélectionnés dans notre cas sont les suivants :

Attributs	**Gain**	**Attributs**	**Gain**
service	3.886	diff_srv_rate	2.230
flag	1.528	dst_host_srv_count	5.578
src_bytes	6.143	dst_host_same_srv_rate	4.191
dst_bytes	6.056	dst_host_diff_srv_rate	3.509
count	6.485	dst_host_same_src_port_rate	3.195
serror_rate	1.246	dst_host_serror_rate	1.625
srv_serror_rate	1.060	dst_host_srv_serror_rate	1.265
same_srv_rate	2.853	difficulty	2.281

Table 4.9 – Liste des attributs sélectionnés.

Après avoir sélectionné les caractéristiques pour l'ensemble d'entraînement, nous avons effectué une classification en utilisant les algorithmes d'apprentissage automatique (Random Forest, K-Nearest Neighbors et Naive Bayes) ainsi que les algorithmes d'apprentissage profond (CNN et DNN). Enfin, le modèle a été évalué par prédiction avec l'ensemble de test.

4.10 Selection de model d'apprentissage profond(NSL-KDD)

- **Système de détection d'intrusion par Réseau de neurones convolutionnel (CNN) et Réseau de neurones profonds (DNN) avec classification multiple :**

Les résultats obtenus démontrent l'efficacité des réseaux de neurones pour la détection d'intrusion. Le modèle basé sur un réseau de neurones profonds (DNN) a atteint une précision de 0.90, un rappel de 0.82, et un F1-score de 0.85. De son côté, le réseau de neurones convolutionnel (CNN) a légèrement surpassé le DNN avec une précision de 0.91, un rappel de 0.83, et un F1-score de 0.86. Ces performances indiquent que les deux modèles sont capables de classifier avec précision les différentes formes d'intrusions, avec une légère supériorité du CNN dans ce contexte spécifique. Ces résultats soulignent l'importance de l'utilisation de techniques avancées d'intelligence artificielle pour renforcer la sécurité des systèmes informatiques.

Model	**Précision**	**Rappel**	**F1-score**
DNN	0.90	0.82	0.85
CNN	0.91	0.83	0.86

Table 4.10 – Les résultats d'apprentissage profonde (CNN et DNN)

- **Système de détection d'intrusion par Réseau de neurones convolu- tionnel (CNN) et Réseau de neurones profonds (DNN) avec classi- fication binaire :**

L'étude démontre des performances exceptionnelles de modèles de réseaux neuronaux pour la détection d'intrusions à l'aide d'un binaire de classification. Les réseaux neuronaux profonds (DNN) rapportent une précision, un rappel et un score F1 de 0,99, tandis que les réseaux neuraux convolutifs (CNN) les surpassent légèrement avec une precision, une rappel, et un résultat F1- de 0,99. Ces résultats soulignent l'importance des techniques d'intelligence artificielle pour la cybersécurité.

La matrice de confusion 4.14 montre que le modèle de détection d'intrusion fonctionne avec une grande efficacité. Elle classe correctement 99% des activités bénignes et malveillantes, avec un taux d'erreur de seulement 1% pour chaque catégorie.

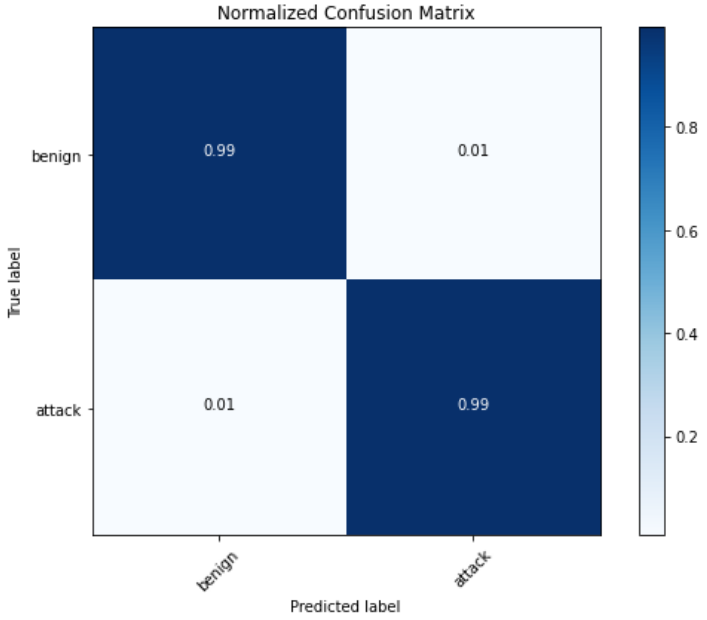


Figure 4.14 – Matrice de confusion du CNN classification binaire.

4.11 L'évaluation de modèle d'apprentissage machine(NSL-KDD)

- **Système de détection d'intrusion par classification binaire et mul-tiple :**

Cette section présente la classification binaire et multiclasse. Nous avon évalué les algorithmes d'apprentissage machine sur l'ensemble de donnée. Pour l'ensemble d'entraînement, la plupart des algorithmes de Random Forest et de KNN ont montré une précision d'entraînement atteignant 100%. Les résultats de la classification dans l'ensemble de test sont présentés comme suit : Durant la phase de test, les résultats du Random Forest pour la classification binaire et de l'algorithme de KNN pour la classification multiclasse sont examinés. Les résultats du Random Forest en termes de précision sont prometteurs pour la classification binaire, avec des scores de

précision, de rappel et de F1-score satisfaisants, comme indiqué dans le tableau. 4.11.

Classe	Précision	Rappel	F1-score	Support
abnormal	1.00	1.00	1.00	14720
normal	1.00	1.00	1.00	16774
accuracy			1.00	31494
macro avg	1.00	1.00	1.00	31494
weighted avg	1.00	1.00	1.00	31494

Table 4.11 – Rapport de Classification binaire

Les résultats des tests de KNN pour la classification multiclasse sont illustrés dans le tableau 4.12, la performance de KNN est évaluée sur l'ensemble de test pour toutes différentes d' attaques , et sa capacité de différencier ces attaques entre eux et aussi avec le trafic bénin . le système a obtenu une bonne précision pour le classe normal et Dos a été bien prédite avec d'autres classes d'attaques qui ont été prédites convenablement. Les résultats des KNN en termes de précision donnent de bons résultats en classification multiclasse, comme indiqué dans le tableau 4.12 .

Classe	Précision	Rappel	F1-score	Support
Dos	1.00	1.00	1.00	9181
Probe	0.99	0.99	0.99	2357
R2L	0.92	0.92	0.92	224
U2R	0.50	0.18	0.27	11
normal	1.00	1.00	1.00	13422
Accuracy			1.00	25195
Macro Avg	0.88	0.82	0.83	25195
Weighted Avg	1.00	1.00	1.00	25195

Table 4.12 – Rapport de Classification multiclass

4.12 Résultats et discussions

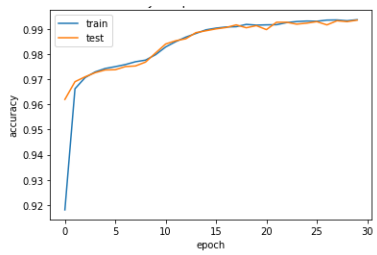
Cette section présente en détail les résultats du système proposé, incluant ceux de la classification binaire et multiclasse. Le système de détection d'intrusion réseau proposé a été implémenté en utilisant l'ensemble de données NSL-KDD et évalué avec des algorithmes d'apprentissage profond (CNN et DNN) ainsi que des algorithmes de machine learning (Random Forest, KNN, Naive Bayes).

Les résultats expérimentaux des approches d'apprentissage profond pour la classification binaire et multiclasse sont présentés dans la Figure 4.10.

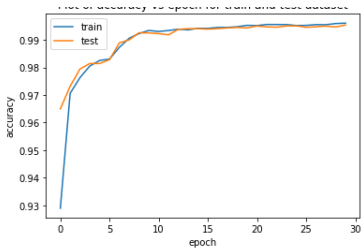
Pour la classification multiclasse sur l'ensemble de données NSL-KDD, les données ont été divisées en 80 % pour l'apprentissage et 20 % pour le test. Les modèles CNN et DNN,

entraînés sur 30 à 60 époques, ont montré une excellente de test accuracy de 99,20 % et 99,32 %, respectivement, avec une valeur de perte similaire de 2%.

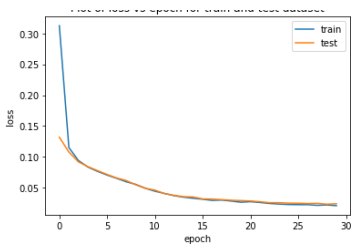
Pour la classification binaire, les mêmes données ont été divisées de la même manière, et les modèles ont été entraînés sur 30 itérations. Les résultats montrent une test accuracy de 99.19 % pour le CNN et 99.39 % pour le DNN, avec une valeur de perte similaire de 2%. Ces résultats indiquent que les modèles d'apprentissage profond apprennent efficacement et fournissent des prédictions précises après chaque phase d'optimisation.



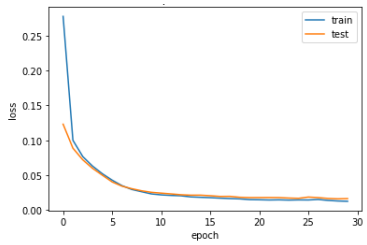
a : Accuracy de CNN classification Binaire



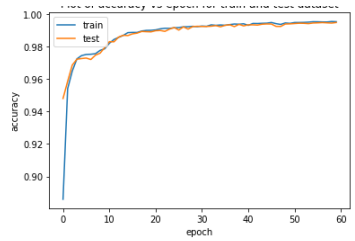
b : Accuracy de DNN classification Binaire



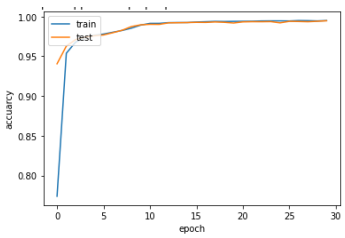
c : Loss de CNN classification Binaire



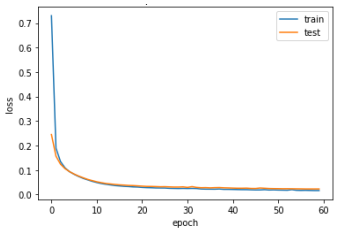
d :Loss de DNN classification Binaire



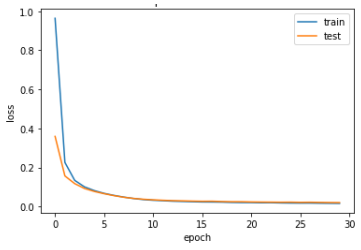
e : Accuracy de CNN Multiclass



f :Accuracy de DNN Multiclass



g : Loss de CNN Multiclass



h : Loss de DNN Multiclass

Figure 4.15 – Les résultats expérimentaux des approches d'apprentissage profond pour la classification binaire et multiclasse utilisant les modèles CNN et DNN.

Les résultats expérimentaux des approches d'apprentissage machine pour la classification binaire et multiclasse, présentés dans les figures 4.16 et 4.17, montrent une variation significative des performances entre les deux types de classification.

Pour la classification multiclasse (Figure 4.16), la Random Forest a obtenu d'excellents résultats avec une précision de 1.00, un rappel de 0.85 et un score F1 de 0.88. En revanche, la méthode Naïve Bayes a montré des résultats moins efficaces avec une précision de 0.56, un rappel de 0.66 et un score F1 de 0.42. La méthode des KNN a atteint une précision de 0.88, un rappel de 0.82 et un score F1 de 0.83.

Pour la classification binaire (Figure 4.17), la Random Forest et les KNN ont tous deux montré une performance parfaite avec une précision, un rappel et un score F1 de 1.00. La méthode Naïve Bayes a amélioré ses résultats par rapport à la classification multiclasse, avec une précision de 0.89, un rappel de 0.85 et un score F1 de 0.85.

Les résultats soulignent l'importance de sélectionner la méthode de classification adéquate en fonction du type d'attaque à détecter et des caractéristiques propres aux données, afin d'obtenir des performances optimales dans la détection des attaques.

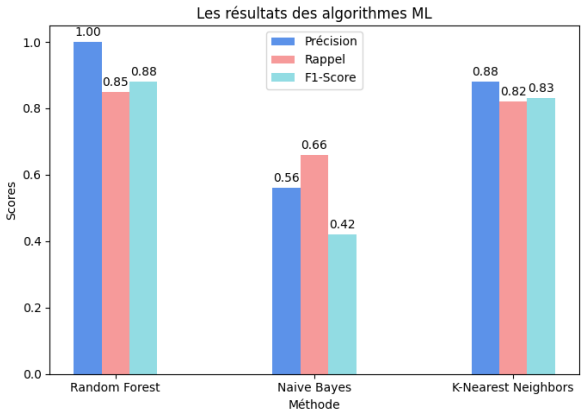


Figure 4.16 – Résultats de Classification Multiclasses.

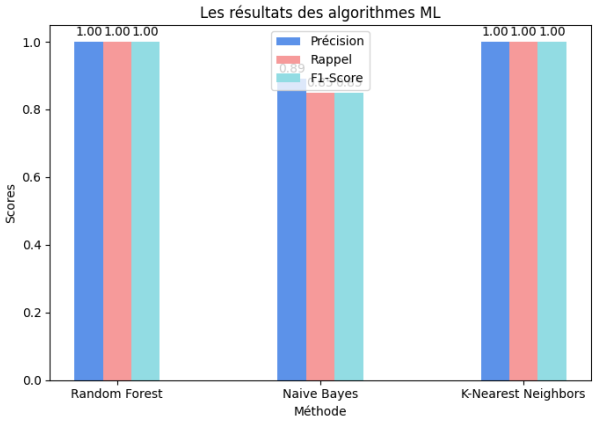


Figure 4.17 – Résultats de Classification Binaire.

4.13 Comparaison des résultats de la base de données cic-ddos-2019 et nsl-kdd

La comparaison des performances des algorithmes de machine learning et de deep learning sur les ensembles de données CIC_DDoS_2019 et NSL-KDD révèle des écarts significatifs. Pour l'algorithme de machine learning Random Forest, l'exactitude est de 0.87 sur CIC_DDoS_2019 et de 1.00 sur NSLKDD, tandis que pour KNN, elle est de 0.83 sur CIC_DDoS_2019 et de 0.97 sur NSL-KDD. Quant au modèle de deep learning CNN, l'exactitude est de 0.91 sur CIC_DDoS_2019 et de 0.99 sur NSL-KDD, tandis que pour DNN, elle est de 0.90 sur CIC_DDoS_2019 et de 0.99 sur NSL-KDD. Ces résultats indiquent généralement une meilleure performance sur le jeu de données NSL-KDD par rapport à CIC_DDoS_2019 pour la plupart des modèles, notamment Random Forest, KNN, CNN, et DNN.

Dataset	Méthode	Accuracy	Précision	Recall	F1-Score
NSL-KDD	KNN	1.00	0.88	0.82	0.83
	Random Forest	1.00	1.00	0.85	0.88
	Naive Bayes	0.71	0.56	0.66	0.42
	CNN	0.99	0.99	0.99	0.99
	DNN	0.99	0.90	0.82	0.85
CIC_DDoS_2019	KNN	0.83	0.86	0.83	0.85
	Random Forest	0.87	0.89	0.88	0.88
	Naive Bayes	0.49	0.49	0.41	0.37
	CNN	0.91	0.92	0.91	0.90
	DNN	0.90	0.91	0.90	0.90

Table 4.13 – Les résultats de l'apprentissage automatique pour les bases de données NSL-KDD et CIC_DDoS_2019

Ces écarts pourraient être dus à plusieurs facteurs, tels que la complexité des ensembles de données, la distribution des classes, le nombre de classes, la diversité des caractéristiques utilisées, et le prétraitement des données. Les ensembles de données comme CIC_DDoS_2019, centrés sur des attaques DDoS, peuvent présenter une complexité accrue ou un déséquilibre des classes, rendant la classification plus difficile, tandis que NSL-KDD, un ensemble de données plus ancien, pourrait avoir une structure plus homogène. Le choix des caractéristiques et des techniques de normalisation peut également jouer un rôle dans ces différences de performance.

4.14 Conclusion

Nous avons implémenté trois modèles de deep learning et quatre modèles de machine learning en utilisant le jeu de données CIC_DDoS_2019 pour détecter les attaques DDoS. Malgré des défis liés au volume des données, au déséquilibre et à l'étiquetage des données, ainsi qu'aux limitations des ressources matérielles, nous avons surmonté ces obstacles en créant trois sous-ensembles de données pour différentes classifications. Les modèles de deep learning ont montré une performance supérieure par rapport aux modèles de machine learning traditionnels, avec une haute précision de détection pour divers types d'attaques DDoS, même pour celles inconnues durant la phase d'apprentissage. Ces modèles peuvent servir de base pour un système de détection d'intrusion (IDS) basé sur la détection des anomalies du trafic réseau. Nous avons également mis en œuvre deux modèles d'apprentissage profond et trois modèles d'apprentissage machine en utilisant l'ensemble de données NSL-KDD pour détecter les attaques. Ce processus n'a pas pris beaucoup de temps et nous n'avons rencontré aucune difficulté, car la taille des données était petite car dispose de quatre types de classes d'attaque (classes) et de 43 feautre performances des modèles d'apprentissage automatique et d'apprentissage profond étaient excellentes, avec une grande précision dans la détection de différents types d'attaques.

Enfin, nous concluons dans notre comparaison entre les deux bases de données, CIC_DDoS_2019 et NSL-KDD, que la taille des données joue un rôle crucial dans les performances des modèles. Avec CIC_DDoS_2019, qui possède un volume de données très important, les modèles ont ren-

contré des défis importants liés au traitement des données, ce qui a entraîné des performances globalement plus faibles malgré des efforts de suréchantillonnage et de tuning des hyperparamètres. En revanche, avec NSL-KDD, le volume de données étant beaucoup plus faible, les modèles de machine learning et de deep learning ont montré des performances excellentes, avec une grande précision dans la détection des attaques. Cela souligne l'importance de la gestion du volume des données dans le développement et l'optimisation des modèles de détection d'intrusion.

Conclusion générale

Conclusion générale

La cybersécurité est un ensemble de pratiques qui consistent à sécuriser les éléments vulnérables grâce aux technologies de l'information et de communication (TIC). Les systèmes de détection d'intrusions ont fait partie de ces pratiques de surveillance dans le but de couvrir l'insuffisance des différents modules de sécurité comme les logiciels antivirus ou les pare-feu. Ces logiciels sont dans la plupart du temps inefficaces face à l'évolution des nouvelles menaces plus sophistiquées. Cette étude a été menée afin de démontrer l'efficacité du deep learning pour le domaine de la cybersécurité. Notre objectif est d'implémenter des méthodes de détection d'intrusions basées sur l'apprentissage profond et évaluer ses performances.

Nous avons commencé par le choix de l'ensemble de données où nous avons choisi de travailler avec NSL-KDD pour une comparaison avec une base de données très récente nommée CIC-DDoS-2019 pour la détection des différentes attaques (DoS et DDoS). Ces cyberattaques réseau sont les plus fréquentes et les plus répandues, et comme elles peuvent être lancées à distance et répercutées par des utilisateurs légitimes sur les réseaux, il est difficile de les détecter et de les prévenir. Notre objectif est d'explorer la détection de ces attaques, et en particulier celles qui sont apparues ces dernières années.

Par la suite, nous avons décidé de mettre en place trois modèles discriminatoires d'apprentissage profond (apprentissage supervisé) : un réseau neuronal profond (DNN), un réseau de neurones convolutif (CNN) et un réseau neuronal récurrent (RNN) pour la classification. Ces méthodes ont été sélectionnées lorsqu'elles sont en accord avec un ensemble de données étiquetées.Donc, lorsque elles ont obtenu des résultats satisfaisants dans des travaux précédents liés.

Les résultats obtenus sont extrêmement satisfaisants, où seules les caractéristiques du trafic réel d'un réseau public sont prises en considération sans aucune information concernant les terminaux connectés, ce qui nous laisse penser que le taux de détection serait encore plus élevé si nous appliquons ces méthodes sur un réseau particulier. Il vous faut simplement sauvegarder les poids du modèle et installer un capteur de réseau et un analyseur qui permettent de lire les flux en temps réel et de les envoyer dans le modèle pour prédire. Selon la complexité des modèles (nombre de paramètres du modèle), le temps de réponse pour une seule prédiction doit être suffisamment faible pour être utilisé comme un système de détection d'alarme en temps réel.

Dans nos travaux futurs, nous envisageons d'explorer l'application de Federated Learning sur nos données en utilisant des architectures distribuées pour entraîner des modèles d'apprentissage automatique sans centraliser les données. En outre, nous chercherons à développer des stratégies d'agrégation sécurisées pour fusionner les mises à jour de modèle provenant de différentes sources tout en préservant la confidentialité des données. Nous prévoyons également d'explorer l'intégration de techniques d'apprentissage fédéré avec d'autres approches d'apprentissage en profondeur telles que l'apprentissage auto-supervisé et les autoencodeurs pour améliorer la performance de la détection d'intrusion dans les réseaux informatiques.

Références

[1] RIAHLA, *Introduction à la sécurité informatique*. PhD thesis, Département de physique/Infotronique IT/S6 de l'université de Boumerdes, 2008-2009.

[2] D. Riquet, *Discus : Une architecture de détection d'intrusions réseau distribuée basée sur un langage dédié*. PhD thesis, Université Lille 1-Sciences et Technologies, 2015.

[3] C. Asma, "Sécurité d'une application web à l'aide d'un systéme de détection d'intrusions comportementale," 2011-2012.

[4] C. Llorens, L. Levier, D. Valois, and B. Morin, *Tableaux de bord de la sécurité réseau*. Editions Eyrolles, 2011.

[5] F. Vinet, J.-C. Gaillard, J.-C. Denain, E. Clavé, F. Leone, S. Giyarsih, and S. Bachri, "Enjeux et modalités spatiales de la reconstruction post-tsunami à banda aceh," *Tsunarisque : le tsunami du*, vol. 26, pp. 233–270, 2004.

[6] P.-L. Lussan, "Les 10 types de cyberattaques les plus courants." `https://blog.netwrix.fr/2018/07/04/les-10-types-de-cyberattaques-les-plus-courants/`, 17 octobre 2022 [En ligne ; Consulté le 10/03/2024].

[7] L. Gallon and J. Aussibal, "Analyse spectrale d'outils classiques de ddos," in *1er Colloque sur les Risques et la Sécurité d'Internet et des Systèmes (CRISIS'2005)*, 2005.

[8] M. K. Nicolas Baudoin, *NT Réseaux IDS et IPS*. 2003-2004.

[9] M. E.-S. GADELRAB, *Evaluation des systèmes de détection d'intrusion*. PhD thesis, université de Toulouse III, 2008.

[10] H. A. Jabou Chaouki, Schillings Michaël, "Ter détection d'anomalies sur le réseau," *Rapport de projet*, 2009.

[11] R. G. Yende, "Support de cours de sécurité informatique et crypto.," 2018.

[12] S. S. T. Denis de REYNAL, Jehan-Guillaume de RORTHAIS, "présentation sur les vpn," *Informatique et Réseaux 3éme année*, février 2004.

[13] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems : A survey of common practices," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, pp. 1–41, 2015.

[14] P. Duc, *Hybrid IDS*. PhD thesis, Ecol d'ingénieurs du Canton de Vaud, 18 décembre 2003.

[15] A. Maxime DUMAS, *VISUALISATION RADIALE DE GRAPHES BIPARTIS APPLIQUÉE AUX SYSTÈMES DE DÉTECTION D'INTRUSIONS SUR DES RÉSEAUX INFORMATIQUES MONTRÉAL*, ch. chapiter 1. 16 DÉCEMBRE 2011.

[16] M. T. V. T. M. DOMINGUEZ Hugo, *LE SYSTÈME DE DÉTECTION DES INTRUSIONS ET LE SYSTÈME D'EMPÊCHEMENT DES INTRUSIONS (ZERO DAY)*. Montréal, Février 2005.

[17] D. S. Abderrahim, *Cours sécurité des systèmes informatique, Support de Cours pour Sécurité des Systèmes d'Informations*, ch. chapier 1. 2022-2023.

[18] "Sécurité des systèmes informatiques/version imprimable — wikilivres." `https://fr.wikibooks.org/w/index.php?title=Sécurité_des_systèmes_info`, 2016. [En ligne ; Consulté le 28/04/2024].

[19] P. Illy, "Les systèmes de détection d'intrusion (ids)." `https://www.researchgate.net/figure/Modele-generique-de-la-detection-dintrusions-propose-par-lIDWG-Ladministrateur_fig1_335639245`. [En ligne ; Consulté le 10/03/2024].

[20] M. ROMAIN COUSSEMENT, *D'AIDE À LA DÉCISION POUR LES IDS DANS LES RÉSEAUX VANETS*. JANVIER 2014.

[21] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system : A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[22] M. H. Olivier Lourme, *Contribution à l'adoption des IDS dans l'IoT*. 2021.

[23] V. A. PORRAS (EA.), "Live traffic analysis of tcp/ip gateways," *Proceedings of the Network and Distributed System Security Symposium, NDSS 1998, San Diego, California, USA*, p. 293, 29 February 2016.

[24] A. W. Herv6 DEBAR, Marc DACIER, "A revised taxonomy for intrusion-detection systems," pp. pp.361–378, ANN. TELECOMMUN., 55, n 7-8, 2000.

[25] W. P. Nasrin Sultana, Naveen Chilamkurti and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches.," *Peer-to-Peer Networking and Applications*, p. 12(2) :493–501, 2019.

[26] O. M. Liran Lerman and G. Bontempi, *Les systèmes de détection d'intrusion basés sur du machine learning*. PhD thesis, Thèse de doctorat, Université libre de Bruxelles, 2008.

[27] S. Dua and X. Du, *Data mining and machine learning in cybersecurity*. CRC press, 2016.

[28] J. Singh and S. Behal, "Detection and mitigation of ddos attacks in sdn : A comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, p. 100279, 2020.

[29] S. AMARA, *Une Approche IntelligenteDeep LerningPour La Detection Des Attaques Ddos Pour Le ReseauSdn*. PhD thesis, Université de Larbi Tebessi–Tebessa, 2022.

[30] A. Djeffal, "Cours fouille de données avancée," *Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie : Université Mohamed Khider-Biskra*, pp. 6–8, 2014.

[31] N. Halabi, O. Rivoire, S. Leibler, and R. Ranganathan, "Protein sectors : evolutionary units of three-dimensional structure," *Cell*, vol. 138, no. 4, pp. 774–786, 2009.

[32] P. Mariot, C. Golbreich, J.-P. Cotton, F. Vexler, and A. Berger, "Méthode, modèle et outil ardans de capitalisation des connaissances," *Revues des Nouvelles Technologies de l'Information, RNTI E-12, 7èmes journées francophones Extraction et Gestion des Connaissances, Namur, Belgium*, pp. 187–207, 2007.

[33] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA transactions on Signal and Information Processing*, vol. 3, p. e2, 2014.

[34] L. Deng, D. Yu, *et al.*, "Deep learning : methods and applications," *Foundations and trends® in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[35] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[36] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "A detailed analysis of the cicids2017 data set," in *Information Systems Security and Privacy : 4th International Conference, ICISSP 2018, Funchal-Madeira, Portugal, January 22-24, 2018, Revised Selected Papers 4*, pp. 172–188, Springer, 2019.

[37] "Kdd cup 1999 data." `https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`, 1999. [En ligne ; Consulté le 10/04/2024].

[38] "Iscx nsl-kdd dataset 2009." `https://www.unb.ca/cic/datasets/nsl.html`, 2009. [En ligne ; Consulté le 10/04/2024].

[39] "Mawi datasets 2011." `https://faculty.nps.edu/cabollma/MAWI_Datasets/Datasets.html`, 2011. [En ligne ; Consulté le 11/04/2024].

[40] "Intrusion detection evaluation dataset (iscxids2012)." https://www.unb.ca/cic/datasets/ids.html, 2012. [En ligne ; Consulté le 10/04/2024].

[41] "Intrusion detection evaluation dataset (cic-ids2017)." https://www.unb.ca/cic/datasets/ids-2017.html, 2017. [En ligne ; Consulté le 10/04/2024].

[42] "Ddos evaluation dataset (cic-ddos2019)." https://www.unb.ca/cic/datasets/ddos-2019.html, 2019. [En ligne ; Consulté le 10/04/2024].

[43] Y. LeCun *et al.*, "Generalization and network design strategies," *Connectionism in perspective*, vol. 19, no. 143-155, p. 18, 1989.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[45] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, *Network intrusion detection using deep learning : a feature learning approach*. Springer, 2018.

[46] H. H. Al-Maksousy, M. C. Weigle, and C. Wang, "Nids : Neural network based intrusion detection system," in *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, pp. 1–6, IEEE, 2018.

[47] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Soft computing in industrial applications*, pp. 293–303, Springer, 2011.

[48] G. D. G. M. M. Arash Habibi Lashkari and A. Ghorbani, ""characterization of tor traffic using time based features"," *in In Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP)*, 2017.

[49] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 international carnahan conference on security technology (ICCST)*, pp. 1–8, IEEE, 2019.

[50] A. A. G. Iman Sharafaldin, Arash Habibi Lashkari, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," October 2019.

[51] M. B. R. B. Jose Maria ALONSO1, Antonio GUZMAN2, "Ldap injection techniques," *Wireless Sensor Network*, July 5, 2009.

[52] "What is a netbios vulnerability, what is the risk and how can you mitigate that risk?." https://www.skywaywest.com/2021/01/what-is-a-netbios-vulnerability/, [En ligne ; Consulté le 15/04/2024].

[53] "Ntp amplification ddos attack." https://www.cloudflare.com/en-in/learning/ddos/ntp-amplification-ddos-attack/, [En ligne ; Consulté le 15/04/2024].

[54] "Ssdp ddos attack." https://ddos-guard.net/en/terms/ddos-attack-types/ssdp-ddos-attack. [En ligne ; Consulté le 15/04/2024].

[55] W. Wang, X. Zhang, S. Gombault, and S. J. Knapskog, "Attribute normalization in network intrusion detection," in *2009 10th international symposium on pervasive systems, algorithms, and networks*, pp. 448–453, IEEE, 2009.

[56] S. SATPATHY, "Smote for imbalanced classification with python." https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/, [En ligne ; Consulté le 24/04/2024].

[57] A. A. Awan, "An introduction to smote." https://www.kdnuggets.com/2022/11/introduction-smote.html, [En ligne ; Consulté le 24/04/2024].

[58] M. Dr.Poornima G.Naik, Dr.Girish R.Naik, "Conceptualizing python in google colab," January 2022.

[59] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," pp. 1–8, 2019.

[60] B. Ingre and A. Yadav, "Performance analysis of nsl-kdd dataset using ann," in *2015 international conference on signal processing and communication engineering systems*, pp. 92–96, IEEE, 2015.