République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

MEMOIRE

Présenté par

DJEDID RAYANE

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Détection d'intrusion dans un système e-Learning

Soutenu le : 22/06/2024

Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Président	Mme. MAATALLAH MAJDA	MCB	Chadli Bendjedid El-Tarf
Rapporteur	**Mrs. MAKHLOUF AMINA**	**MCB**	**Chadli Bendjedid El-Tarf**
Examinateur	Mme. AHMED MALEK NADA	MAA	Chadli Bendjedid El-Tarf
Co-encadreur	Mr. SEDRAOUI BRAHIM KHALIL		

Année Universitaire : 2023/2024

Remerciement

Je tiens à remercier tout d'abord ALLAH le tout puissant de m avoir donne le courage, la patience et la santé pour réaliser ce modeste travail.

Mes remerciements les plus chaleureux et les plus profonds s'adressent à mon encadrant Dr. MAKHLOUF .A qui m'a guide par ses précieux conseils, ses encouragements et ses orientations. Ainsi que pour l'aide et le temps qu'il n'a jamais manque de m'apporter tout au long de l'élaboration ce travail.

Je remercie également Monsieur BRAHIM KHALIL SEDRAOUI qui m'a aidé beaucoup dans ce travail.

Mes remerciements les plus vifs s'adressent aussi aux nombre de jury pour l'intérêt qu'ils ont porté à ma recherche tout en acceptant d'évaluer ce travail.

Enfin, je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

DJEDID RAYANE

Dédicace

Je Dédié Ce Modeste Travail a:

Toute Ma Grande Famille

«Pour Leurs Encouragements Continu Et Leurs Soutien»

Mes Amis

«Dont La Liste Est Longue »

Résumé

La détection des intrusions dans les systèmes e-Learning est un aspect crucial de la sécurité de ces plateformes. Actuellement, les technologies de détection populaires se reposent sur des algorithmes d'apprentissage automatique traditionnels pour entraîner des modèles de détection d'intrusions à partir d'échantillons. Cependant, ces méthodes présentent souvent des taux de détection relativement faibles. L'apprentissage profond représente une avancée significative, permettant l'extraction automatique de caractéristiques pertinentes des échantillons. Face à la précision limitée des méthodes traditionnelles, cette étude propose un modèle de détection d'intrusions dans les systèmes e-Learning basé sur un réseau neuronal convolutif (CNN). Ce modèle est conçu pour extraire efficacement les caractéristiques des échantillons d'intrusion, améliorant ainsi la précision du processus de classification. Les résultats expérimentaux sur les ensembles de données KDD_NSL montrent que le modèle proposé peut grandement améliorer la précision de la détection des intrusions dans les systèmes e Learning.

Mots clés: KDD_NSL, cyber sécurité, intrusion Détection, CNN, l'apprentissage en profondeur

Abstract

Intrusion detection in e-learning systems is a crucial aspect of the security of these platforms. Currently, popular detection technologies rely on traditional machine learning algorithms to train intrusion detection models from samples. However, these methods often have relatively low detection rates. Deep learning represents a significant advance, enabling the automatic extraction of relevant features from samples. Faced with the limited accuracy of traditional methods, this study proposes an intrusion detection model in e-learning systems based on a convolutional neural network (CNN). This model is designed to efficiently extract features from intrusion samples, thereby improving the accuracy of the classification process. Experimental results on KDD_NSL datasets show that the proposed model can greatly improve the accuracy of intrusion detection in eLearning systems.

Keywords: KDD_NSL, Intrusion Detection, CNN, Deep learning

الملخص :

يعد اكتشاف التسلل في أنظمة التعلم الإلكتروني جانبًا مهمًا لأمن هذه المنصات. تعتمد تقنيات الكشف الشائعة حاليًا على خوارزميات التعلم الآلي التقليدية لتدريب نماذج كشف التسلل من العينات. ومع ذلك، فإن هذه الأساليب غالبًا ما تكون معدلات اكتشافها منخفضة نسبيًا. يمثل التعلم العميق تقدمًا كبيرًا، حيث يتيح الاستخراج التلقائي للميزات ذات الصلة من العينات. في مواجهة الدقة المحدودة للطرق التقليدية، تقترح هذه الدراسة نموذجًا لكشف التسلل في أنظمة التعلم الإلكتروني استنادًا إلى الشبكة العصبية التلافيفية **.(CNN)** تم تصميم هذا النموذج لاستخراج الميزات من عينات التسلل بكفاءة، وبالتالي تحسين دقة عملية التصنيف. تظهر النتائج التجريبية على مجموعات بيانات **KDD_NSL** أن النموذج المقترح يمكن أن يحسن بشكل كبير دقة كشف التسلل في أنظمة التعلم الإلكتروني**.**

الكلمات المفتاحية :الأمن السيبراني **KDD_NSL** ، كشف التسلل، **CNN**، التعلم العميق

Sommaire

Remerciement ii

Dédicace iii

Résumé i

Abstract ii

ملخص: iii

Sommaire iv

Liste des tableaux : vii

Liste des Figure: viii

Introduction générale ..... ي

Chapitre 01 Etat de l'art

1.1 Introduction 11

1.2 E-Learning 11

1.2.1 Historique 11

1.2.2Définition 11

1.3 Les attaques informatiques 12

1.3.1 Les différentes catégories d'attaques 12

1.3.1.1 Les principaux attaquent DOS 13

1.4 Sécurité informatique 19

1.5 Intrusions informatiques 19

1.6 La détection d'intrusion 20

1.7 Les systèmes de détection d'intrusion (IDS) 21

1.7.1 Evaluation des IDS 22

1.7.2 Les différents types d'IDS 23

1.7.3 Efficacité des systèmes de détection d'intrusions 26

1.8.1 Détection d'intrusions par signature 27

1.8.1.1 Principe 27

1.8.1.2 La base de signature :.....28
1.8.2 Détection d'intrusion comportementale.....29
1.8.2.1 Principe 29
1.8.3 Une approche hybride.....31
1.8.4 Les Systèmes experts.....32
1.9 La détection la d'intrusion basée sur le deep learning.....32
1.9.1 Définition de l'apprentissage profond 32
1.9.2Fonctionnement 33
1.9.3 Des méthodes d'apprentissage profond 34
1.9.4 Les ensembles de data set pour évaluer les IDSs basé à deep learning 41
1.9.5 Des Travaux associés pour détecter les intrusions basées sur le DL 42
1.10 Conclusion.....47

Chapitre 02 Conception

2.1 Introduction 49
2.2 Architecture du système.....49
2.3Collecte de données 49
2.4 Les données NSL-KDD 50
2.5 Caractéristiques du Dataset.....50
2.7 Optimiser.....53
2.8 Model architecture.....53
2.8.1 Model CNN 53
2.9 Configuration du modèle et nombre de paramètres 55
2.10 Evaluation metrics 56
2.10 Conclusion.....58

Chapitre 03 Implémentation Et résultat

3 Implémentation Et résultat.....60
3.1 Introduction 60

3.2.2 Bibliothèques utilisés dans l'implémentation 60
3.3. Préparation data 62
3.3.1Teste and train data 62
3.3.1.1Train_test_split 62
3.3.1.2 Sauvegarder data 63
3.3.1.3 Train et test shap 64
3.3 Expérimentations, comparaison et discussion des résultats obtenus 64
3.4 Expériences et résultats 64
3.4.1En termes de modèles 65
3.4.2 En termes de modèle LSTM 71
3.5. Comparaison enter le model CNN ET LTSM 72
3.6 Représentant notre interface système 73
3.7 Conclusion 77
Conclusion général 78
BIBLIOGRAPHIE 82

Liste des tableaux :

TABLEAU1. 1 : ATTAQUES DE PROBE 17
TABLEAU1. 2:ATTAQUES DE USER TO ROOT.....18
TABLEAU1. 3:ATTAQUES DE REMOTE TO USER 18
TABLEAU1. 4 : ENSEMBLES DE DONNEES PUBLIC RELATIVES AU CYBER SECURITE 41
TABLEAU1. 5 : DES TRAVAUX ASSOCIES POUR DETECTER LES INTRUSIONS BASEES SUR LE DL. 46

TABLEAU 2. 1: CARACTERISTIQUES DE BASE DES CONNEXIONS TCP INDIVIDUELLES.....50
TABLEAU 2. 2 : FONCTIONNALITES DE CONTENU DANS UNE CONNEXION SUGGEREE PAR LA CONNAISSANCEDU DOMAINE.....51
TABLEAU 2. 3 : CARACTERISTIQUES DU TRAFIC CALCULE A L'AIDE D'UNE FENETRE DE TEMPS DE DEUX SECONDES. 52
TABLEAU 2. 4 : SHOWS THE CONFUSION MATRIX.....56

TABLEAU 3. 1: DESKTOP HARDWARE.....60
TABLEAU 3. 2 : MODEL PERFORMANCE.....70

Liste des Figure:

FIGURE 1:FONCTIONNEMENT D'UNE ATTAQUE UDP FLOOD....16
FIGURE 2: MODELE SIMPLIFIE D'UN SYSTEME DE DETECTION D'INTRUSIONS 22
FIGURE 3: RESEAU HIDS 23
FIGURE 4: MODELE D'ARCHITECTURE DE L'IDS BASE SUR LE RESEAU....25
FIGURE 5:CARACTERES COMPLET ET CORRECT DU MODELE DE COMPORTEMENT NORMAL 30
FIGURE 6: APPROCHE HYPRIDE 31
FIGURE 7: L'ARCHITECTURE D'UN MODELE DEEP LEARNING 33
FIGURE 8: L'ARCHITECTURE D'UN MODELE DE RESEAU NEURONAL CONVOLUTIF 36
FIGURE 9 : CONVOLUTION....36
FIGURE 10 : POOLING....37
FIGURE 11 : L'ARCHITECTURE D'UN MODELE RNN 39
FIGURE 12 : L'ARCHITECTURE D'UN MODELE LSTM_GRU 40
FIGURE 13 : L'ARCHITECTURE DE SYSTEME 49
FIGURE 14 : MODEL ARCHITECTURE 54
FIGURE 15: LA CONFIGURATION DE MODEL....56
FIGURE 16: TRAIN TEST SPLIT....62
FIGURE 17 : SAUVEGARDER DATA 63
FIGURE 18 : TRAIN SHAPE....64
FIGURE 19 TRAIN SHAPE....64
FIGURE 20: CHART OF ACCURACY AND LOSS OF MODEL CNN1....66
FIGURE 21: CHART OF ACCURACY AND LOSS OF MODEL CNN1(A)....66
FIGURE 22: CHART OF ACCURACY AND LOSS OF MODEL CNN1(B) 67
FIGURE 23: CHART OF ACCURACY AND LOSS OF MODEL CNN2....68
FIGURE 24: CHART OF ACCURACY AND LOSS OF MODEL CNN2(A)....69
FIGURE 25: CHART OF ACCURACY AND LOSS OF MODEL CNN3.....70
FIGURE 26:INSTALLATION DE FLASK 73
FIGURE 27: FIGURE : INSTALLATION DE TENSORFLOW....74
FIGURE 28 : RUNNING D'APP.PY 74
FIGURE 29 : HOME PAGE....75
FIGURE 30 : LISTE DES INPUTS 75
FIGURE31 : AFFICHAGE RESULTAT....76

Introduction Générale

Introduction générale

Après la pandémie de COVID-19, l'e-Learning est devenu un élément crucial de l'éducation et de la formation à travers le monde. Avec les restrictions de déplacement et les mesures de distanciation sociale, de plus en plus de gens se tournent vers l'e-Learning pour poursuivre leurs études et leur développement professionnel. Cette transition a transformé la façon dont les individus acquièrent des connaissances, offrant une flexibilité sans précédent et éliminant les barrières géographiques.

Parallèlement, avec l'évolution des réseaux et en particulier les réseaux internet, les techniques de l'information et de communication offrent actuellement des facilités incontournables en matière d'apprentissage à distance. Les technologies émergentes telles que les vidéos conférences et les distributeurs automatiques, ainsi que l'omniprésence croissante des outils informatiques, ont ouvert de nouvelles perspectives dans le domaine de l'éducation.

Cependant, cette avancée vers un apprentissage plus flexible et connecté n'est pas sans risques. Les systèmes d'e-learning et les réseaux informatiques sont confrontés à de nouvelles vulnérabilités de sécurité, alimentées par les cybermenaces omniprésentes. Les réseaux et systèmes connectés font face à des menaces intentionnelles ou accidentelles, allant du piratage de la vie privée aux attaques par déni de service, posant ainsi des défis majeurs en matière de cybersécurité.

Afin de protéger les aspects cruciaux de la technologie de l'information, y compris l'accès, le stockage, le traitement et la transmission des données, la cybersécurité joue un rôle essentiel. Des techniques telles que les systèmes de détection d'intrusion (IDS) sont déployées pour prévenir les menaces imminentes et protéger les réseaux et systèmes informatiques contre les attaques.

Face à l'évolution constante des cyberattaques, les entreprises de cybersécurité se tournent vers des approches plus avancées, telles que l'apprentissage machine (ML) et l'apprentissage profond (Deep Learning), pour renforcer l'efficacité de leurs produits. Ces technologies permettent une détection plus précise et proactive des menaces, notamment dans des domaines critiques tels que la sécurité des infrastructures électriques et industrielles.

Ainsi, bien que l'e-learning offre des opportunités d'apprentissage sans précédent, il est essentiel de reconnaître et de traiter les défis de sécurité associés à cette transition numérique. En combinant l'innovation en matière d'apprentissage en ligne avec des mesures de cybersécurité robustes, nous pouvons créer un environnement d'apprentissage connecté et sécurisé pour les apprenants du monde entier.

Dans cette étude de master, notre objectif principal était d'améliorer les systèmes de détection d'intrusion dans l'apprentissage en ligne en utilisant l'algorithme d'apprentissage CNN et une base de données KDD_NSL.

Notre mémoire est organisé comme suite :

1- Etat de l'art

Le premier chapitre est dédié à l'exposition des divers aspects de la sécurité informatique et des systèmes de détection d'intrusion. Le chapitre explore les domaines problématiques de l'IDS et les différents types de systèmes IDS. De plus, nous passons en revue les travaux et études de recherche connexes qui ont contribué à l'avancement de l'IDS.

2- Conception

Dans le deuxième chapitre se concentre sur l'étude conceptuelle de notre système de détection d'intrusion.

Nous discutons de la conception du système, y compris de la sélection des outils et des cadres de développement appropriés. Le chapitre couvre également la présentation de l'ensemble de données utilisé pour la formation et l'évaluation, ainsi que les techniques d'optimisation utilisées pour améliorer les performances du modèle. De plus, nous présentons l'architecture de notre modèle d'apprentissage profond. Les méthodes et les métriques d'évaluation du modèle sont également abordées dans ce chapitre.

3- Implémentation et résultat

Dans le dernier chapitre, nous présentons les résultats et les travaux de mise en œuvre de notre système de détection d'intrusion. Nous discutons de la représentation des outils et des cadres de développement utilisés dans la mise en œuvre du système. Les résultats expérimentaux sont présentés, mettant en évidence la précision et les performances atteintes par notre modèle. Nous comparons et discutons les résultats obtenus avec les approches précédentes dans le domaine. De plus, nous présentons les résultats de prédiction et les mesures d'évaluation du modèle pour évaluer l'efficacité du système. De plus, nous présentons l'interface utilisateur de notre système et discutons du processus de test pour valider sa convivialité et ses fonctionnalités.

Chapitre 01
Etat de l’art

1.1 Introduction

La sécurité des systèmes informatiques est essentielle pour assurer le bon fonctionnement des plateformes d'e-Learning, tout en protégeant les données sensibles des utilisateurs.

La présentation des attaques informatiques, la politique de sécurité et les systèmes de détection d'intrusion sont les sujets abordés dans ce chapitre. Soulignant ainsi l'importance cruciale de la protection contre les menaces en ligne dans le domaine de l'éducation à distance.

1.2 E-Learning

1.2.1 Historique

Historique de l'e-Learning commence officiellement en 1999 avec l'introduction du terme "e-Learning" lors d'un séminaire sur les systèmes de formation en ligne [1]. Avant cela, des cours en ligne existaient déjà mais sous des noms tels que Computer Based Training (CBT) [1], et des efforts pour fournir des informations à distance datent des années 1960[1].

Le développement de l'Internet a joué un rôle crucial dans l'expansion de l'e-learning, car il a permis de communiquer et de partager des connaissances de manière globale [1]. Au fil des années, l'e-Learning a connu une popularité grandissante grâce à l'innovation numérique et à la diminution des coûts des équipements informatiques [1].

Aujourd'hui, l'e-Learning inclut des formats variés tels que les Massive Open Online Courses (MOOCs), les cours en ligne, les tutoriels et les livres électroniques (ebooks)[1]. Cette forme de formation est maintenant largement adoptée dans les universités et les entreprises, et elle offre de nombreuses opportunités pour apprendre de manière flexible et accessible [1].

1.2.2Définition

L'e-Learning, également appelé apprentissage en ligne, est un système d'apprentissage basé sur un enseignement formalisé mais avec l'aide de ressources électroniques. Il permet aux apprenants de se former à distance en utilisant des outils numériques tels que des ordinateurs et Internet.

Ce type d'enseignement peut se dérouler à l'intérieur ou à l'extérieur des salles de classe, mais l'utilisation d'ordinateurs et d'Internet constitue la composante principale de l'e-Learning. Il s'agit essentiellement d'un transfert de compétences et de connaissances via un réseau, permettant la diffusion de l'éducation à un grand nombre de destinataires [2].

L'e-learning offre une flexibilité et une accessibilité accrues pour l'apprentissage, ce qui le rend populaire dans les universités, les entreprises et divers autres domaines. Il permet aux

apprenants de participer à des expériences d'apprentissage organisées, peu importe leur emplacement physique, et offre une variété de formats tels que les MOOCs, les cours en ligne et les tutoriels [1, 2].

1.3 Les attaques informatiques

La notion d'« attaque » désigne l'utilisation d'une faille dans un système informatique (système d'exploitation, logiciel ou même l'utilisateur) à des fins inconnues par l'utilisateur et généralement préjudiciables.

1.3.1 Les différentes catégories d'attaques

En informatique, il y a différents types d'attaques que l'on peut classer en quatre catégories [3].

- **L'attaque DOS (denial of service)**

Le déni de service, également appelé « Denial Of Service » ou encore DOS, est une attaque visant à rendre les services ou les ressources d'une organisation indisponibles pendant une période déterminée. En règle générale, ce genre d'attaque vise à compromettre les machines, les serveurs et les accès d'une entreprise, les rendant ainsi inaccessibles pour leurs clients. Une telle attaque ne vise pas à modifier ou supprimer des données, ni même à divulguer des informations. [4].

Il s'agit de compromettre la réputation de sociétés présentes sur Internet en entravant le bon déroulement de leurs opérations. Il n'est pas très difficile de créer un DOS déni de service, mais il est tout aussi efficace. On peut attaquer différents équipements réseau tels que les serveurs, les routeurs et les switchs. 99% de la planète est touchée par cela car la majorité des problèmes de service exploitent des vulnérabilités liées au protocole TCP-IP. Les conséquences des attaques DOS peuvent être classées en deux catégories :

- La saturation du service DOS consiste à submerger une machine de demandes, ce qui la rend incapable de répondre aux demandes réelles.
- Les problèmes de service DOS causés par l'utilisation de vulnérabilités, qui impliquent l'exploitation d'une faille du système cible pour le rendre inaccessible.

Ces attaques DOS se basent sur l'envoi de paquets ou de données de taille ou de constitution inhabituelle, dans le but de causer une saturation ou un état instable des équipements victimes, les empêchant ainsi de fournir les services réseau qu'elles sont censées fournir. Dans certaines situations extrêmes, ce genre d'attaque peut entraîner la défaillance de l'équipement visé. Ainsi, le déni de service DOS est une forme d'attaque très coûteuse car elle

perturbe le fonctionnement normal des transactions d'une organisation. Étant donné qu'à l'heure actuelle, les montants et les défis d'une entreprise sont souvent considérables, cela peut engendrer de sérieux problèmes si une telle situation survient, même quelques heures. Imaginez les conséquences :

- Comment financer un site de vente en ligne majeur dont la plateforme d'hébergement serait indisponible pendant les fêtes de Noël?
- Sur le modèle d'une banque incapable de recevoir ses courriels?
- Pour vous, en tant qu'entreprise qui établit des liens avec vos clients?

Il est difficile de mettre en œuvre des contre-mesures et elles doivent être adaptées à un type d'attaque DOS. Comme les attaques par déni de service exploitent les services et protocoles habituels d'Internet, il serait nécessaire de s'en prémunir en coupant les voies de communication normales, étant donné que c'est la raison d'être principale des machines concernées (sites web, messagerie, extranet,...). [4].

Il faut donc essayer de se protéger au mieux contre certains comportements anormaux, ce qui implique notamment la vérification de l'intégrité des paquets, la surveillance du trafic, l'établissement de profils types et de seuils... La protection n'est donc pas totale, mais il est tout de même possible de se protéger de manière astucieuse et souple.

1.3.1.1 Les principaux attaquent DOS

- **Attaque ARP poisoning:**

L'attaque repose sur l'envoi de données ARP falsifiées. Donc, les divers équipements du réseau local sont confrontés à des erreurs de correspondance entre les adresses IP et les MAC.

Cette situation entraîne la rupture de toutes les communications entre deux équipements IP. Souvent, les serveurs et les routeurs sont cibles, ce qui rend les services associés indisponibles [5].

- **Attaque par fragmentation (fragment attack)**

Cette méthode d'attaque repose sur la division de l'IP. Le but est de perturber la connexion IP de la cible en ajustant les numéros de séquences. Effectivement, le protocole IP est conçu pour séparer les données de grande taille provenant de la couche 4 du modèle OSI. Ainsi, le

datagramme est divisé en plusieurs paquets IP qui ont chacun un numéro de séquence et un numéro d'identification commun. Lorsque les fragments sont reçus, la cible regroupe les paquets en utilisant les valeurs de décalage (en anglais offset) qu'ils renferment.

Les numéros de séquence peuvent être modifiés pour créer des blancs ou des recouvrements lors du réassemblage de la pile IP cible. Certains appareils ne pouvaient pas le supporter, ce qui entraînait différentes conséquences, comme l'arrêt du service TCPIP.

Toutefois, revenons à la réalité, aujourd'hui, tout comme pour le Ping de la mort, cette méthode n'est plus réalisable en raison de l'évolution des piles IP. Cette attaque est racontée en détail, ce qui permet de se remémorer des souvenirs [5].

- **Attaque ping de morte (ping of death)**

Cette méthode d'attaque DOS est aujourd'hui obsolète, mais elle a démontré son efficacité à l'époque. La plupart des piles IP présentaient une faiblesse dans leur mise en place en envoyant un paquet ICMP d'une taille non conforme (plus de 64 octets). Cela a entraîné une défaillance directe de la pile IP attaquée.

Toutefois, revenons à la réalité, aujourd'hui, tout comme pour l'attaque par fragmentation, cette méthode n'est plus efficace en raison de l'évolution des piles IP. Il est donc possible d'échanger en utilisant un datagramme ICMP de grande taille sans aucun problème (heureusement).

Pour raconter l'histoire et se remémorer des souvenirs, il est important de détailler cette attaque [6].

On confond souvent l'attaque DOS Ping de la mort en pensant qu'elle repose sur la saturation d'une bande passante en ICMP. Il n'en est pas ainsi, car ce principe est mis en œuvre par l'attaque DOS Ping Flood et non par Ping mort.

- **Attaque unreachable host**

Les messages ICMP de type « Host Unreachable » sont envoyés à une cible par cette attaque DOS, ce qui entraîne la déconnexion des sessions et paralyse la victime. La particularité de cette attaque DOS réside dans le fait qu'elle requiert un débit limité, car les envois de datagramme ICMP peuvent être effectués à une cadence faible [5].

- **Attaque ICMP redirect**

La cible de cette attaque DOS est un serveur ou un routeur, et elle envoie des messages ICMP de type « Redirect ». Le datagramme fera savoir à la victime qu'il est nécessaire de suivre un autre chemin. Cela entraînera donc une indisponibilité du réseau WAN.

- **Attaque ping flood (ICMP Flood)**

De nombreux hôtes Internet, qu'ils soient publics ou privés, répondent aux paquets ICMP, il est donc aisé de les saturer de ce flux pour les rendre indisponibles. En outre, quelle que soit la réponse des cibles à l'ICMP, il est primordial de saturer leurs capacités d'accès réseau, processeurs et mémoire.

Ping Flood est l'attaque la plus courante par déni de service, car de nombreux individus et amateurs s'amusent simplement à faire du piège à un host distant. Et évidemment, ils apprécient d'ajouter les options qui permettent d'augmenter la vitesse au maximum.

Toutefois, Microsoft a restreint les possibilités de son Ping, ce qui entraîne une attente d'une seconde entre chaque Ping. Cela évite aux individus de se divertir avec cette attaque, mais il est important de rester réaliste, Windows n'est pas le seul système d'exploitation et Ping. L'application exe n'est pas la seule.…

- **Attaque UDP Flood**

L'idée de cette attaque DOS est la même que celle du Ping Flood. Il consiste à épuiser les ressources de la cible en ce qui concerne le débit, le processeur et la mémoire en utilisant le datagramme UDP.

De même, que la cible soit ou non répondante au flux abondant émis, cela n'affecte pas le résultat.

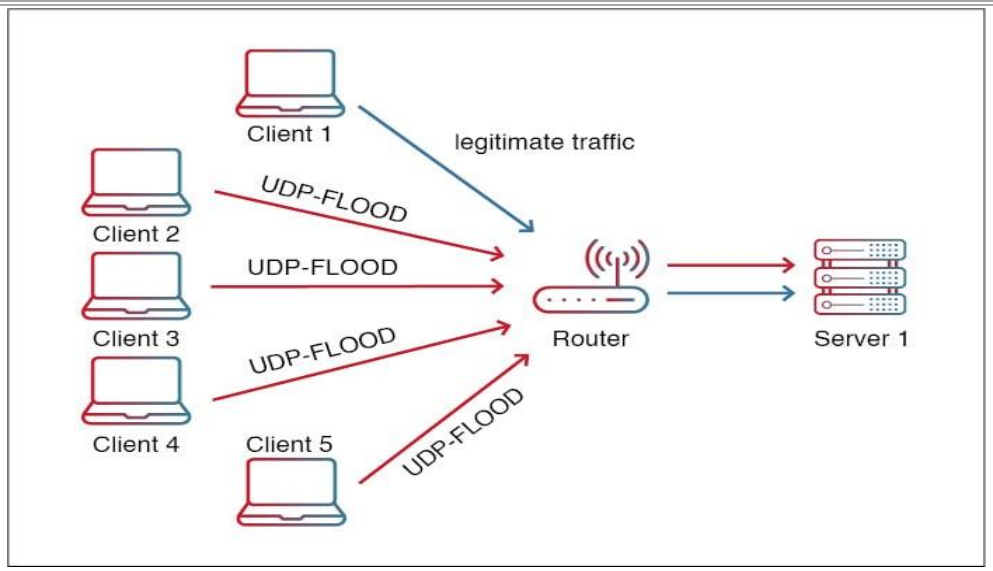


Figure 1:**fonctionnement d'une attaque UDP flood**

- ## Attaque * Flood

J'ai récemment inventé le nom de cette attaque DOS, qui consiste à saturer une cible de la même manière que Ping Flood et UDP Flood. Toutes ces attaques reposent sur la transmission massive de demandes à destination de la cible. Ces demandes ne reposent pas nécessairement sur ICMP ou UDP, mais elles utilisent TCP, IGMP, IP_raw,... C'est pourquoi l'attaque est appelée * Flood, ce qui signifie que l'on peut saturer une cible avec n'importe quel flux IP.

- ## Attaque Land (Land Attack)

L'objectif de cette attaque DOS est de lancer une session TCP en utilisant un SYN à destination d'un port ouvert de la machine cible. L'attaque consiste à spécifier l'adresse IP source identique à l'adresse IP destination et le port source identique au port destination. La personne touchée reçoit cette trame a alors l'impression de discuter avec elle-même, ce qui entraîne habituellement un accident. Toutefois, revenons à la réalité aujourd'hui, tout comme pour le Ping de la mort, cette méthode n'est plus efficace en raison de l'évolution des piles TCP-IP. Grâce à la description détaillée de cette attaque DOS, on peut raconter l'histoire et se souvenir de souvenirs. De plus, tous les systèmes d'IDS et de pare-feu sont fonctionnels pour empêcher tout type de trame.

- **Probing**

La classe d'attaque débute par une enquête sur la future victime, ce que l'on nomme un scan. Ce sondage examinera chaque port IP pour obtenir des informations sur les services proposés par le système (OS, configuration du réseau, sécurité mise en place,...). Après avoir effectué ce balayage, la machine de l'intrus (celui qui effectue l'intrusion) essaie alors de déterminer le système d'exploitation utilisé par cette victime et d'exploiter les données qu'elle a collectées. La plus large catégorie d'attaque nécessite une expertise technique minimale. Les attaques telles que Ipsweep, Mscan, Nmap, Saint, Satan sont des exemples de ce genre [4].

Type d'attaques	Service	Mécanisme	L'effet de l'attaque
Ipsweep	Icmp	Abus des propriétés	Identification des machines actives
Mscan	Plusieurs	Abus des propriétés	Recherche des vulnérabilités
nmap	Plusieurs	Abus des propriétés	Identification des ports active sur une machine
Saint	Plusieurs	Abus des propriétés	Recherche des vulnérabilités
Stan	Plusieurs	Abus des propriétés	Recherche des vulnérabilités

Tableau1. 1: Attaques de Probe [4].

- **U2R (user to root)**

L'objectif de cette catégorie d'attaques est d'exploiter les vulnérabilités pour obtenir la main de l'administrateur système (Root) à partir d'un simple compte utilisateur. Les plus célèbres exploits sont les débordements fréquents des buffers causés par des erreurs de programmation. Les attaques les plus courantes de ce genre sont Ejecta, Ffbconfig, Fdformat, Loadmodule, Perl, Ps, Xterm.

Type d'attaques	Service	Mécanisme	L'effet de l'attaque
Eject	Session utilisateur	Débordement du buffer	Gagne le Shell root
Ffbconfig	Session utilisateur	Débordement du buffer	Gagne le Shell root
Fdformat	Session utilisateur	Débordement du buffer	Gagne le Shell root
Load module	Session utilisateur	Un mauvais système d'installation	Gagne le Shell root
Perl	Session utilisateur	Un mauvais système	Gagne le Shell root

		d'installation	
Ps	Session utilisateur	Une mauvaise gestion des fichiers temporels	Gagne le shell root
Xterm	Session utilisateur	Débordement du buffer	Gagne le shell root

Tableau1. 2:**Attaques de User to Root. [4]**

- **R2L (Remote To Local)**

Dans cette classe, l'adversaire (une machine distante) envoie des paquets vers une machine du réseau cible, puis il exploite les faiblesses de cette machine pour obtenir un accès illégal en tant qu'utilisateur. Différentes attaques de R2L existent, les plus répandues utilisent ou exploitent les bugs ou les configurations incorrectes des applications ou des systèmes. Les attaques de cette classe comprennent le dictionnaire, le Ftp_write, le Guest, l'Imap, le Named, le Phf, le Sendmail, le Xlock et le Xsnoop, comme le montre le tableau 1.3.

Type d'attaques	Service	Mécanisme	L'effet de l'attaque
Dictionnaire	Telent, rlogin, pop, ftp, imap	Abus des propriétés	Gagne un accès utilisateur
Ftp_write	ftp	Mauvaise configuration	Gagne un accès utilisateur
Guest	telnet, rlogin	Mauvaise configuration	Gagne un accès utilisateur
Imap	Imap	Bug	Gagne un accès root
Named	Dns	Bug	Gagne un accès root
Phf	http	Bug	Exécute des commandes autant que utilisateur http
Sendmail	Smtp	Bug	Exécute des commandes autant que root
Xlock	Smtp	Mauvaise configuration	Mystifie un utilisateur pour obtenir le mot de passe
Xsnoop	Stmp	Mauvaise configuration	Contrôle le stockage des clés à distance

Tableau1. 3:**Attaques de Remote to User [4].**

1.4 Sécurité informatique

Le domaine de la sécurité informatique consiste à utiliser la technologie, les politiques et l'éducation des individus afin de garantir la confidentialité, l'intégrité et l'accessibilité des données pendant leur stockage, leur traitement et leur transmission. La préservation des informations doit être basée sur le système à protéger. Par conséquent, d'après ce dernier, on mettra plus ou moins l'accent sur l'intégrité, la confidentialité ou la disponible.

Confidentialité : La confidentialité stipule que seules les personnes autorisées à accéder à une certaine information peuvent l'obtenir. Dans cette optique, les contrôles d'accès et le chiffrement sont employés. Il est possible de constater une violation de la confidentialité lorsque des informations confidentielles sont devenues publiques, que ce soit grâce aux journaux du système ou aux modifications de comportement d'une personne envers l'organisation [13].

Intégrité : L'intégrité stipule que seuls les individus autorisés peuvent modifier les informations présentes dans le système. Il est souvent protégé de la même manière que celui qui est assuré par la confidentialité. En effet, en surveillant l'accès à une information, on garantit également son intégrité. On peut détecter une violation de cette dernière en comparant l'information avec ses copies ou ses données de hashing, par exemple. Une solution à cela consiste à corriger cette information [13].

Disponibilité : La disponibilité se réfère à la possibilité d'obtenir une information lorsque l'on en a besoin. Le déni de service est l'une de ces attaques bien connues. Celui-ci peut être réduit en restreignant les ressources consommables du système par un individu. Une solution à cela consiste à diminuer la charge ou à accroître les capacités du système [13].

Assurer la confidentialité, intégrité et la disponibilité : Afin de garantir tous ces éléments, un ensemble de mesures de sécurité est mis en place : supprimer les programmes non utilisés, utiliser des pare-feu, utiliser des contrôles d'accès, configurer les programmes de manière adéquate, utiliser des antivirus, utiliser des systèmes de détection d'intrusion,... Notre attention sera portée sur les IDS.

Les IDS assurent la surveillance du système, ce qui nécessite leur restriction afin de rester conformes à la législation du pays où le détecteur est mis en place.

1.5 Intrusions informatiques

Grâce à l'émergence d'Internet, de nouvelles opportunités ont émergé et de nouvelles perspectives ont émergé. La majorité des possibilités se trouvent dans le secteur commercial,

où les entreprises ont la possibilité de présenter leurs produits et services à travers le monde entier grâce à des sites internet. Chaque jour, on effectue des transactions avec des sommes considérables, ce qui met les entreprises en danger de diverses menaces. Auparavant, l'attention était beaucoup plus portée aux bénéfices de cette nouvelle technologie et il était rare de voir ou de mettre en place des mesures et des ressources pour garantir une sécurité minimale [3].

Le plus préoccupant est que de nombreuses entreprises sont exposées à des risques sans en être conscientes, et que les administrateurs réseau ou les personnes responsables de la sécurité informatique ne sont pas conscientes des mesures de protection à prendre. De nos jours, on estime que le temps moyen nécessaire pour qu'un ordinateur non protégé connecté à Internet soit victime d'une tentative d'intrusion ou soit infecté par un programme malveillant est de moins de 4 minutes. Des communautés de pirates informatiques ont mis en œuvre différentes méthodes. Chacune d'elles concerne un aspect spécifique de l'outil informatique. Il est possible de classer les diverses méthodes de piratage informatique en deux catégories :

- **Attaques réseaux :** Cette catégorie d'attaques regroupe toutes les méthodes développées pour exploiter les vulnérabilités des réseaux ou les attaques visant des éléments du réseau.
- **Attaques applicatives :**Dans cette deuxième catégorie, on retrouve les méthodes qui se basent sur les vulnérabilités et les erreurs des applications, ce qui permet d'utiliser ces dernières à des fins malveillantes [3].

1.6 La détection d'intrusion

La détection d'intrusion en sécurité informatique consiste à repérer les actions visant à compromettre la confidentialité, l'intégrité ou la disponibilité d'une ressource. Il est possible de détecter l'intrusion de manière manuelle ou automatique. Lorsqu'il s'agit de détecter une intrusion manuelle, un analyste humain examine les fichiers de journaux pour repérer tout signe suspect pouvant suggérer une intrusion.

Un système qui effectue une détection d'intrusion automatisée est appelé système de détection d'intrusion (IDS). Quand un IDS détecte une intrusion, il peut prendre différentes mesures sur le terrain, comme l'enregistrement des informations pertinentes dans un fichier ou une base de données, la génération d'une alerte par e-mail ou d'un message sur un pager ou un téléphone mobile. Le domaine de la détection d'intrusion ne comprend généralement pas la détermination de la nature réelle de l'intrusion détectée et l'action nécessaire pour y mettre fin ou l'empêcher de se reproduire. Néanmoins, il est possible de mettre en place différentes

formes de réaction automatique en interaction avec l'IDS et des systèmes de contrôle d'accès tels que les pare-feu [4].

- **Que doit assurer la détection d'intrusion ?**

Les organisations peuvent protéger leurs systèmes contre les menaces qui ne cessent de croître en raison de l'augmentation de la connectivité du réseau public (Internet) et de la confiance accordée aux systèmes informatiques qui présentent des bugs en détectant les intrusions. Pour les experts en sécurité, la question ne devrait pas être de savoir s'il est nécessaire d'utiliser la détection d'intrusion, mais plutôt de déterminer quels dispositifs utiliser et quelles sont leurs capacités de détection d'intrusion.

Les systèmes de détection d'intrusion ont été reconnus comme un élément essentiel de l'infrastructure de sécurité informatique de chaque entreprise. En effet, les systèmes de détection d'intrusion peuvent être achetés et utilisés pour diverses raisons [5].

• Afin de repérer les attaques et autres violations de sécurité qui ne sont pas mises en échec par d'autres outils de protection.

• Pour mettre en évidence les dangers présents au sein d'une organisation, c'est-à-dire repérer les lacunes avant qu'elles ne soient exploitées par un pirate.

• Pour jouer le rôle de vérificateur de qualité dans la conception de sécurité, notamment dans les grandes et complexes sociétés.

• Afin de fournir des renseignements précieux concernant les intrusions qui ont eu lieu, et de procéder à des diagnostics, des recouvrements et des corrections des causes.
• Pour empêcher les intrusions et réduire les dommages. Malheureusement, cela ne peut pas toujours être réalisé en raison de la complexité et de la variété des intrusions, ainsi que de la création de nouveaux types d'intrusions liés au développement des nouvelles technologies d'information. Les mesures actives de contre-intrusion sont souvent facultatives dans presque tous les systèmes de détection d'intrusion.

1.7 Les systèmes de détection d'intrusion (IDS)

Un système de détection d'intrusion est un dispositif qui vise à détecter des activités inhabituelles ou suspectes sur la cible étudiée (réseau, hôte). Debar rend le processus de détection d'intrusion plus simple en utilisant un détecteur qui analyse les données provenant du système surveillé. [5]

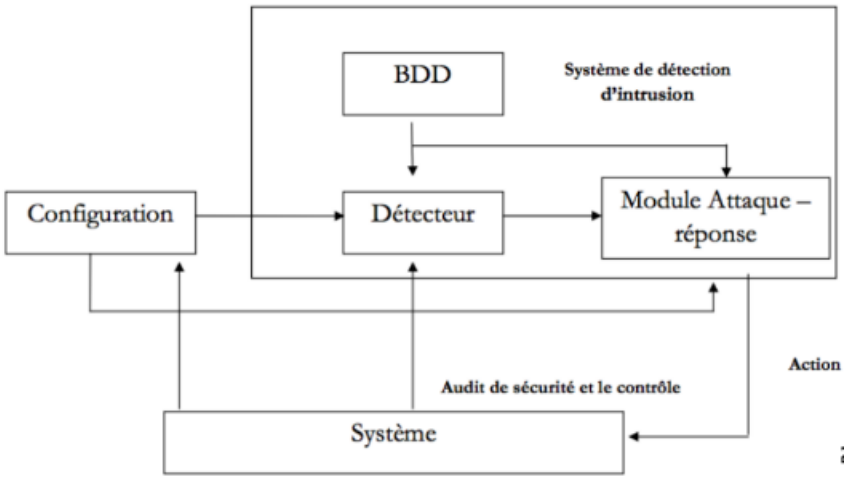


Figure 2: Modèle simplifie d'un système de détection d'intrusions [5].

1.7.1 Evaluation des IDS

Philipe et définit trois critères pour évaluer l'efficacité des systèmes de détection d'intrusion [16] :

- **L'exactitude (accuracy) L'exactitude** est un critère utilisé lorsque le système de détection d'intrusion déclare une activité légale comme malicieux, ce qui correspond au faux positif.
- **La performance (performance)** Le taux de traitement des évènements est influencée par la performance du système de détection d'intrusions. En cas de faible taux, il est donc impossible de détecter en temps réel.
- **La complétude (completeness)** on parle de la complétude quand le système de détection d'intrusion rate la détection d'une attaque. Le critère le plus complexe réside dans le fait qu'il est impossible d'avoir une compréhension globale des attaques. Ce critère correspond au vrai négatif.

Debar et al dans [17] a rajouté également les deux critères suivante :

- **La tolérance aux fautes (faulttolerance)** le système de détection d'intrusion doit luimême résisté aux attaques, particulièrement au déni de service. Ceci est important, parce que plusieurs systèmes de détection d'intrusion s'exécutent sur des matériels ou logiciels connuscomme vulnérables aux attaques.
- **La réaction à temps (timeliness)** Il est essentiel que le système de détection d'intrusion fonctionne et diffuse les résultats de l'analyse dès que possible, afin de permettre à l'officier de sécurité de réagir avant que des dommages graves ne se produisent. Cela ne se limite pas à un simple calcul de performance, car il s'agit non

seulement du temps de traitement des événements, mais également du temps nécessaire pour leur propagation et leur réaction.

1.7.2 Les différents types d'IDS

➢ IDS basé sur l'hôte

Un IDS (Système de Détection d'Intrusion) basé sur l'hôte surveille plusieurs domaines pour détecter les activités malveillantes ou abusives à l'intérieur d'un réseau, ainsi que les intrusions provenant de l'extérieur. Ces systèmes analysent les journaux de diverses sources, tels que le noyau, le système, les serveurs, le réseau, les pare-feu, etc., en les comparant à une base de données interne de signatures d'attaques connues.

Les IDS basés sur UNIX et Linux tirent parti de la commande syslog, qui permet de séparer les événements enregistrés selon leur gravité (par exemple, des messages mineurs d'imprimante par rapport à des avertissements critiques du noyau). Cette commande est souvent disponible via le paquetage sysklogd, inclus notamment dans RedHat Enterprise Linux, offrant ainsi une fonctionnalité de journalisation du système et la capture des messages du noyau.

Pour filtrer les journaux souvent volumineux et complexes, les IDS basés sur l'hôte les analysent, les marquent avec un système d'évaluation de la gravité propre à l'IDS, puis les rassemblent dans un journal spécialisé. Ces journaux spécialisés sont ensuite examinés par les administrateurs pour détecter les activités suspectes et prendre les mesures appropriées en réponse.

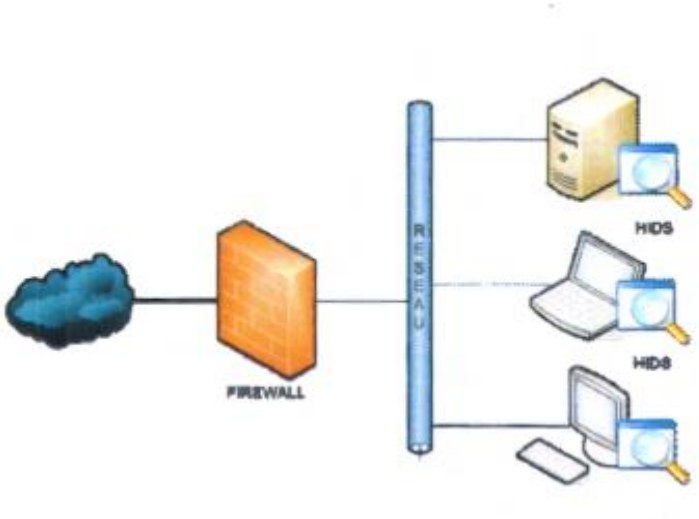


Figure 3: **réseau HIDS**

Les systèmes d'alarme basés sur l'hôte ont également la capacité de vérifier l'intégrité des données de fichiers et d'exécutables essentiels. Ils effectuent une vérification d'une base de données de fichiers confidentiels (ainsi que de tout fichier ajouté par l'administrateur) et calculent une somme de contrôle pour chaque fichier en utilisant un logiciel d'analyse de fichiers messages tels que la commande md5sum (algorithme 128-bit) ou la commande sha1sum (140-bit). Les systèmes de détection d'intrusion basés sur l'hôte stockent donc les sommes dans un fichier texte clair et, parfois, comparent les sommes de contrôle de fichiers avec les valeurs du fichier texte. En cas de non-conformité d'une des sommes, les IDS informent l'administrateur par courrier électronique ou par courrier pager.

- **Les avantages d'un IDS basehôte**
  - ❖ La possibilité de superviser de manière précise les activités locales des utilisateurs.
  - ❖ Possédant la capacité de décider si une attaque est réussie.
  - ❖ La capacité à travailler dans des contextes crêpes.
  - ❖ L'IDS base hôte utilise les données d'audit des systèmes d'exploitation afin de repérer différents types d'attaques (par exemple : Cheval de Troie).
- **Les inconvénients d'un IDS base hôte**
  - ❖ La possibilité d'être vulnérable aux attaques de déni de service, car l'IDS peut se trouver dans l'hôte cible lors des attaques.
  - ❖ La complexité de mise en place et de gestion, en particulier lorsque le nombre d'hôtes nécessitant une protection est élevé.
  - ❖ Il est impossible pour ces systèmes de repérer des attaques contre de nombreuses cibles dans le réseau.

- **IDS basé sur le réseau IDS**

Les systèmes de détection d'intrusions basés sur le réseau fonctionnent en analysant le trafic réseau au niveau de l'hôte ou du routeur. Ils examinent les paquets réseau, les analysent pour détecter des comportements suspects, et enregistrent ces paquets dans un journal spécial avec des détails. Ensuite, ils comparent ces paquets avec une base de données de signatures d'attaques connues pour déterminer le niveau de sévérité. Si le niveau est assez élevé, une alerte est déclenchée, généralement sous forme de message électronique ou d'appel de pager,

pour alerter l'équipe de sécurité. Cela leur permet d'investiguer plus en profondeur et de prendre des mesures pour contrer l'intrusion.

Les IDS basés sur le réseau sont devenus très appréciés avec l'augmentation de la taille et du trafic de l'internet.

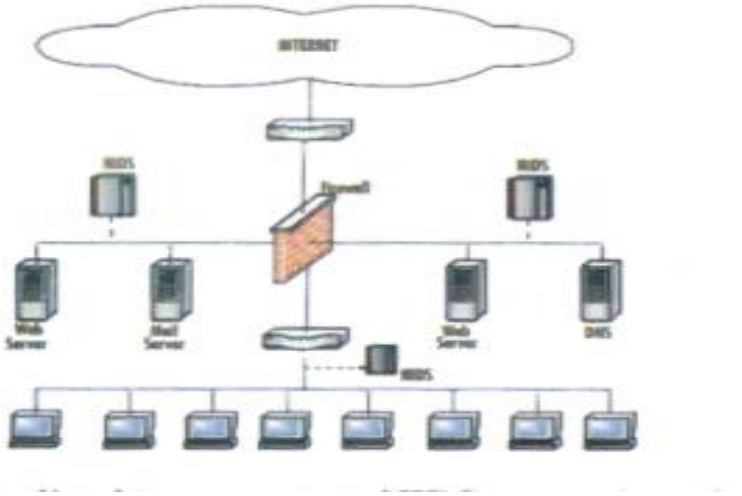


Figure 4: **modèle d'architecture de l'IDS base sur le réseau**

Les systèmes de détection d'intrusion (IDS) capables de scanner les grandes quantités d'activités réseau et de détecter avec succès les transmissions suspectes sont appréciés dans le domaine de la sécurité. Étant donné la nature peu sûre des protocoles TCP/IP, il est devenu essentiel de créer des scanneurs, des renifleurs et d'autres outils d'analyse de réseau et de détection afin d'éviter les vulnérabilités liées aux activités réseau malveillantes telles que :

• Le vol d'identité

•Les attaques par déni de service

• La corruption de cache-arp

• La défaillance des noms DNS

• Les attaques à l'intérieur du groupe

- **Les avantages d'un IDS basé réseau**
  - ❖ L'IDS basé sur le réseau peut superviser un grand nombre d'hôtes avec un coût de déploiement réduit.
  - ❖ Il n'a pas d'impact sur les résultats des entités surveillées.

- ❖ L'IDS basé sur le réseau a la capacité de repérer les attaques provenant de plusieurs hôtes.
- ❖ L'IDS basé sur le réseau garantit une sécurité élevée contre les attaques car il est invisible pour les attaquants.

- **Les inconvénients d'un IDS basé réseau**

- ❖ Il est impossible que l'IDS base réseau fonctionne dans des environnements crêpes.
- ❖ Ce genre d'IDS ne peut pas garantir la réussite d'une tentative d'attaque.
- ❖ En raison de la variété des sources de données disponibles et de la représentativité des données utilisées lors des tests, l'évaluation et la comparaison des systèmes de détection d'intrusions représentent un défi en soi. Une. Traditionnellement, les systèmes de détection d'intrusions sont évalués en fonction de deux critères :

La fiabilité de l'IDS : En effet, toute intrusion doit entraîner une alerte. Une intrusion non signalée est considérée comme une panne de l'IDS, connue sous le nom de faux négatif. Le taux de faux négatifs (c'est-à-dire le pourcentage d'intrusions non détectées) d'un système de détection d'intrusions est essentiel pour garantir sa fiabilité.

La pertinence des alertes : Toute alerte doit correspondre à une véritable intrusion. La présence de toute « fausse alerte » (également appelée faux positif) diminue la pertinence de l'IDS.

Il est essentiel qu'un IDS efficace ait un nombre minimal de faux positifs.

Il n'est pas suffisant de repérer de manière adéquate les intrusions ; Il est essentiel de ne pas produire trop d'alertes erronées. L'administrateur de sécurité risque de considérer toutes les alertes comme des faux positifs, comme si plus de 1% des alertes étaient des faux positifs (1% étant un chiffre prudent), et de ne pas les analyser et de ne pas prendre des mesures de protection au cas où l'alerte ne serait pas un faux positif. Dans ce contexte, le système de détection d'intrusions est inutile.

1.7.3 Efficacité des systèmes de détection d'intrusions

Pour évaluer l'efficacité des systèmes de détection d'intrusion, Philip établit trois critères :

• **l'exactitude (accu race)** L'exactitude est évoquée lorsque le système de détection d'intrusion déclare une activité légale comme malveillante. Il s'agit du critère du faux positif.

• **la performance (performance)** Le taux de traitement des évènements est responsable de la performance du système de détection d'intrusion. En cas de faible taux, il est donc impossible de détecter en temps réel.

•**La complétude (complétées) :** La complétude est évoquée lorsque le système de détection d'intrusion échoue à détecter une attaque. Il s'agit du critère le plus complexe, car il est impossible d'avoir une vision d'ensemble des attaques. Cette mesure correspond à la vérité négative [6].

Les deux critères suivants ont également été ajoutés par Debar :

• **la tolérance aux fautes (fautl tolérance) :** Il est essentiel que le système de détection d'intrusion soit capable de faire face aux attaques, notamment au déni de service. Il est crucial de souligner que de nombreux systèmes de détection d'intrusion fonctionnent sur des appareils ou des logiciels identifiés comme vulnérables aux attaques.

• **la réaction à temps :** Il est essentiel que le système de détection d'intrusion fonctionne et diffuse les résultats de l'analyse dès que possible, afin de permettre à l'officier de sécurité de réagir avant que des dommages graves ne se produisent. Cela ne se limite pas à un simple calcul de performance, car il s'agit non seulement du temps de traitement des événements, mais également du temps nécessaire pour leur propagation et leur réaction [4].

1.8 Détection d'intrusion par signature et comportementale

On peut classer les systèmes de détection d'intrusions en fonction de leur méthode d'analyse des données. La littérature a suggéré deux grandes méthodes : la détection d'intrusions par signature et la détection d'intrusions comportementales. Leur principe de détection diffère entre ces deux approches : l'approche par signature repose sur la recherche de traces d'attaques ou d'intrusions, tandis que l'approche comportementale cherche les variations du comportement de l'entité observée par rapport à un modèle du comportement normal de cette entité. Même si la première méthode suggérée par Anderson en 1980[14] est de nature comportementale, nous allons tout d'abord examiner les méthodes par signature qui sont appréciées par les industriels de la sécurité, mais qui présentent des inconvénients intrinsèques difficilement contournables.

1.8.1 Détection d'intrusions par signature

1.8.1.1 Principe

Les systèmes de détection d'intrusions par signature reposent sur la détection de signatures d'attaques dans le flux d'événements générés par le biais d'une ou plusieurs sondes. Une

signature représente, dans le déroulement des événements, des scénarios d'attaques préétablis. Un système d'identification par signature est composé de :

• Une ou plusieurs sondes qui produisent un flux d'événements, qu'ils soient de type réseau ou hôte.

•À partir d'une base de signature.

• D'un dispositif permettant de repérer les motifs dans le flux d'événements.

1.8.1.2 La base de signature :

L'efficacité de l'IDS repose principalement sur la qualité de la base de données, car seules les attaques dont la signature est présente dans la base peuvent être détectées. On décrit les signatures en utilisant des langages de description d'attaques [18]. La majorité du temps, elles sont établies par un opérateur, même si des recherches récentes ont permis la génération automatique des signatures [19]. Il est également important de préserver la base de signatures :

• Il est nécessaire d'intégrer les nouvelles attaques détectées par la communauté à la base.

• En fonction des décisions de l'administrateur de sécurité, il est nécessaire de fournir les signatures correspondantes.

Une éventuelle intrusion (comme la mise à jour, le remplacement ou la suppression d'un logiciel ou d'un système d'exploitation, par exemple) peut être retirée de la base de données. La préservation de la base de signatures revêt une importance capitale. Si l'IDS n'est pas entretenu, il ne peut pas repérer les nouvelles attaques. La propagation rapide de certains vers tels que Spammer et al dans [20] met également en évidence une limite de cette méthode, car le temps nécessaire pour l'administrateur pour mettre à jour la base de signatures est plus long que le temps nécessaire pour la propagation du ver.

Le système de reconnaissance de motifs a pour objectif de détecter les motifs présents dans la base de signature, dans le flux d'événements. Différents systèmes de reconnaissance de motifs ont été définis dans la littérature. Différents systèmes existent, allant des systèmes simples basés sur les règles de Verne Paxon dans [21] ou les correspondances de chaînes de caractères de Mukherjee et al dans [22] (string matching) à des systèmes beaucoup plus complexes basés sur des systèmes experts tels que Phillip et al dans [23] ou la modélisation

d'états de Steven et al [24], qui peuvent donner une certaine abstraction pour repérer des attaques inconnues mais qui correspondent à une même classe d'attaques. Pour plus de détails sur ces systèmes, consultez la classification d'Axels son. [25].

- **Avantage de la détection par signature**

❖ Il est envisageable de prendre en considération les comportements précis des attaquants potentiels.

- **Inconvénients de la détection par signature**

❖ La construction de la base des règles est essentielle, ce qui peut parfois être difficile.

❖ Les performances du système expert sont restreintes par celles de l'expert humain qui a élaboré les règles. Cependant, les officiers de sécurité ont peu de connaissances en matière de détection d'intrusion car, la plupart du temps, le grand nombre de fichiers d'audit les a découragées de toute analyse.

1.8.2 Détection d'intrusion comportementale

1.8.2.1 Principe

Une méthode, suggérée par J.P. ANDERSON [14] puis reprise et développée par D.E. DENNING [26], repose sur l'hypothèse que l'exploitation d'une vulnérabilité du système traduit un usage anormal de celui-ci. On peut donc identifier une intrusion comme une rupture par rapport au comportement habituel d'un utilisateur. Voici quelques illustrations qui étendent cette hypothèse :

- Si un utilisateur inconnu du système essaie d'entrer, cela entraînera un taux anormal de mots de passe erronés.
- L'attaquant qui se déguise se connecte à une heure inhabituelle, il utilise de nombreuses commandes lui permettant de changer de répertoire, peut-être n'utilise-t-il jamais l'utilitaire favori de l'utilisateur habituel.
- Un utilisateur légitimement connecté au système qui tente de contourner la politique de sécurité se connectera la nuit, exécutera des programmes qu'il n'a pas l'habitude d'utiliser, générera un volume accru d'enregistrements d'audit, utilisera une imprimante sur laquelle il ne sort habituellement pas de documents, etc.
- En termes d'utilisation des ressources d'entrée/sortie, un cheval de Troie diffèrera du programme légal dont il a pris la place.

- Une attaque par déni de service entraînera un taux d'utilisation anormalement élevé de certaines ressources du système.

Évidemment, les phénomènes mentionnés dans ces exemples peuvent être dus à une autre cause que l'attaque du système, comme un changement de fonction de l'utilisateur au sein de l'entreprise. Nous nous concentrerons donc sur la recherche de méthodes ayant le taux de discrimination le plus élevé possible (c'est-à-dire avec le taux de détection d'intrusion le plus élevé et le taux de fausses alarmes le plus faible). En outre, nous utiliserons un seuil au-delà duquel le comportement sera considéré comme intrusif.

On désigne cette approche comme étant basée sur la question de savoir si le comportement actuel de l'utilisateur est en accord avec son comportement passé. L'approche la plus directe pour définir le comportement normal d'un utilisateur (on parle de modèle de comportement) est d'utiliser des méthodes statistiques. On peut aussi envisager l'emploi de systèmes spécialisés ou de réseaux de neurones. Nous allons vous exposer.

Selon la politique de sécurité, le modèle de comportement normal est considéré comme complet lorsqu'il reflète entièrement le comportement légitime de l'entité surveillée. Dans cette situation, toutes les alertes sont des intrusions: il n'y a pas de faux positifs.

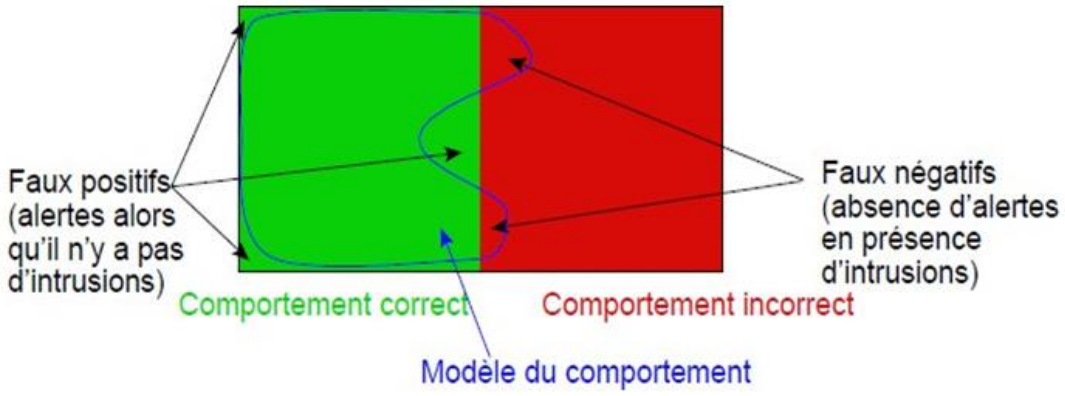


Figure 5:**Caractères complet et correct du modèle de comportement normal [26].**

o Avantages de la détection comportementale

- Il est envisageable de détecter une intrusion inconnue.

o Inconvénients de la détection comportementale

- La décision concernant les divers paramètres du modèle statistique est assez complexe
- et dépend de l'expérience de l'officier de sécurité.
- Il n'y a pas de preuve de l'hypothèse d'une distribution normale des différentes mesures.

- Il est difficile de sélectionner les mesures à retenir pour un système cible spécifique.
- Si l'environnement du système cible subit une modification significative, le modèle statistique émet un flux constant d'alertes, au moins pendant une période transitoire.
- Un utilisateur a la possibilité de modifier progressivement son comportement afin de familiariser le système avec un comportement intrusif.
- Il est complexe de déterminer si les observations effectuées pour un utilisateur spécifique
- Correspondent à des activités que l'on souhaite interdire.
- Pour une personne avec un comportement erratique, toute activité est considérée comme normale. Il ne sera pas possible de repérer une attaque par déguisement sur son compte.
- Les tentatives de collusion entre utilisateurs ne sont pas prises en considération, bien que cet aspect soit très crucial, en particulier dans le cas des réseaux.

1.8.3 Une approche hybride

Tombini et al ont proposé une approche hybride [27]. Cette méthode implique de sérialiser un IDS comportemental puis d'utiliser un IDS par signature. Les requêtes normales sont filtrées par l'IDS comportemental, ce qui signifie que seules les requêtes anormales sont passées à l'IDS par signature. Malgré la simplicité de l'IDS comportemental utilisé, cela permet de diminuer le nombre de faux positifs générés dans l'ensemble. Le fichier d'audit du serveur web est utilisé comme source d'entrées. Ainsi, cet IDS rencontre les mêmes difficultés que les autres utilisateurs de cette source de données. Il apparaît donc nécessaire d'utiliser en même temps une approche comportementale et une approche par signature afin de bénéficier des bénéfices de l'une et de l'autre.

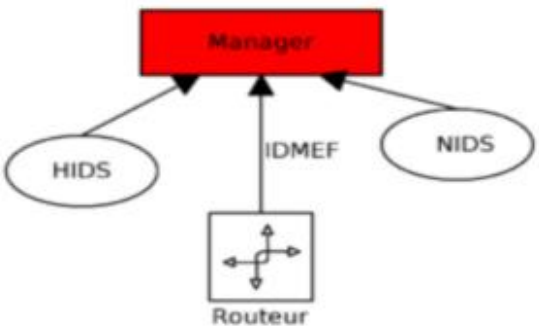


Figure 6: **approche hypride**

1.8.4 Les Systèmes experts

L'usage « normal » d'un utilisateur du système peut être représenté par un ensemble de règles plutôt qu'un modèle statistique. Cela offre la possibilité d'employer un système-expert comme moyen de détecter les intrusions. Il est possible d'entrer manuellement les règles d'un tel système expert ou de les générer automatiquement à partir des enregistrements d'audit. Par exemple, l'utilisation de l'entrée manuelle permettra d'exprimer une politique de sécurité. De leur côté, les règles émises décrivent des comportements. L'un des inconvénients souvent mentionnés des systèmes-experts est que la base de règles n'est ni simple à établir, ni simple à maintenir.

1.9 La détection la d'intrusion basée sur le deep learning

En raison de la disponibilité de grandes quantités de données provenant de la cybeinfrastructure, des réseaux, des systèmes d'exploitation ou des systèmes d'information, des méthodes et des techniques telles que l'apprentissage automatique (machine learning), le minage de données, les statistiques et d'autres compétences interdisciplinaires ont été utilisées pour faire face aux défis de la cybersécurité [46].

Les Systèmes IDS basés sur la signature ou basés sur la détection des anomalies pourraient être exploités grâce à l'apprentissage profond qui fait partie de l'apprentissage automatique. Il est possible d'utiliser ces techniques de classification et de prédiction afin de repérer les motifs et les comportements inhabituels des différentes cyberattaques, ce qui permet un cyber réponse en temps réel. Ils sont capables de repérer les attaques dès qu'elles ont eu lieu et ont également la capacité de prédire les éventuelles futures attaques [35].

Les techniques d'apprentissage approfondi peuvent contribuer à résoudre les difficultés associées à la création d'un IDS efficace [51, 41].

1.9.1 Définition de l'apprentissage profond

Les techniques d'apprentissage profond (Deep Learning ou DL) font partie d'une catégorie de techniques d'apprentissage automatique (machine Learning ou ML), et elles connaissent un succès considérable dans de nombreuses tâches de l'intelligence artificielle (IA) par rapport aux algorithmes standards de ML. Les modèles profonds sont relativement récents et utilisent de nombreuses étapes de traitement non linéaire de l'information. Ces architectures impliquent l'utilisation de couches hiérarchiques pour traiter les informations, chacune recevant et

interprétant les informations de la couche précédente afin d'apprendre les représentations de données [52].

1.9.2Fonctionnement

Dans la plupart des cas, l'organisation des réseaux profonds est structurée en différentes couches de neurones, telles qu'une couche d'entrée (Input Layer), une ou plusieurs couches cachées (HiddenLayers) et une couche de sortie (Output Layer). Chacune des couches adjacentes est reliée. Les liens entre eux sont connus sous le nom de poids (Weights). La figure 2.1 présente une architecture standard d'un modèle de réseau de neurones profond, où les "neurones" d'une même couche sont appelés "noeuds".

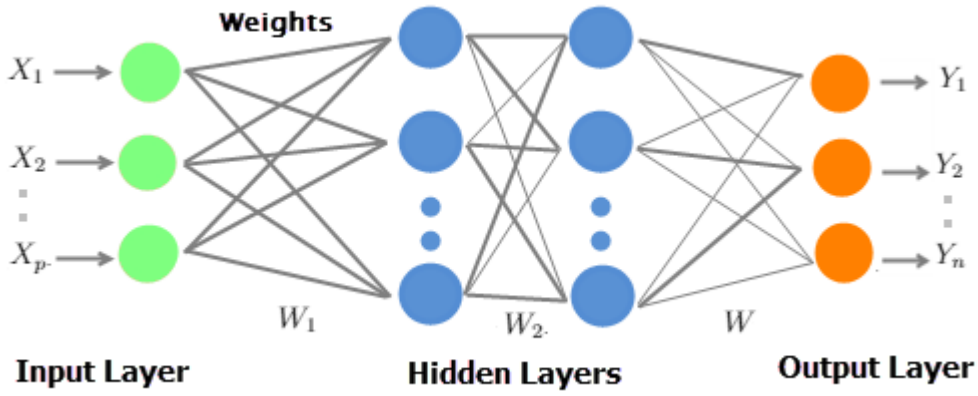


Figure 7: l'architecture d'un modèle deep Learning [34]

Le système de calcul avancé appelé apprentissage profond est composé d'une multitude de techniques provenant du domaine de l'apprentissage automatique. Il utilise un ensemble de neurones non linéaires disposés en plusieurs couches de traitement pour extraire et convertir des valeurs de variables d'entité à partir du vecteur d'entrée, créant ainsi plusieurs niveaux d'abstraction pour représenter les données [54].

Le processus d'apprentissage du DNN consiste à optimiser les paramètres de poids et de biais entre deux couches voisines. Cela permet d'évaluer la précision du modèle et de mieux l'adapter aux données d'apprentissage. Quand le modèle atteint une précision maximale avec des paramètres optimaux, il sera appliqué aux données réelles. Le degré d'apprentissage et donc la précision des modèles obtenus sont influencés par la quantité et la qualité des données d'entraînement.

1.9.3 Des méthodes d'apprentissage profond

Les réseaux neuronaux profonds sont un ensemble de neurones organisés en une séquence de couches interconnectés. Ce qui les différencie, c'est l'architecture du réseau (la manière dont les neurones sont organisés dans le réseau et la manière dont ils se fonctionnent. Parmi de nombreuses implémentions de modèles d'apprentissage profond :

Deep Neural Network (DNN)

Les réseaux neuronaux profonds (DNN) sont un groupe de neurones structurés en une série de couches multiples connue sous le nom de Perceptrons multi-couches (MLP). Les réseaux neuronaux artificiels se démarquent des réseaux neuronaux classiques (Artifical Neural Network) par leur profondeur et le nombre de couches et de neurones qui les composent. Quand un réseau neuronal profond comporte deux couches cachées ou plus, on parle de réseau neuronal profond.

En combinant différentes transformations non linéaires, ils cherchent à représenter des données qui présentent des architectures complexes. [9]

Rosenblatt a introduit le concept fondamental de la perception en 1958 [40]. En utilisant une combinaison linéaire en fonction de ses poids (w) d'entrée, la perception calcule une sortie unique en utilisant une fonction d'activation non linéaire. En mathématiques, on peut l'écrire de la manière suivante :

$$y = \delta(\sum_{n=1}^{n} w_{ixi} + b) = \delta(w^T X + b\ ) \qquad (1)$$

Avec :

• W : est le vecteur des poids.

• X : est le vecteur des entrées.

• b : désigne le biais.

• δ : représente la fonction d'activation.

La couche d'entrée d'un réseau de perception multi-couches (MLP) est constituée d'un ensemble de nœuds sources, d'une ou plusieurs couches cachées de nœuds de calcul et d'une couche de sortie de nœuds. Le signal d'entrée se répand à travers le réseau couche par couche. La (figure 2.1) représente le flux de signaux d'un tel réseau avec une couche dissimulée. En règle générale, les réseaux DNN sont employés dans les domaines de l'apprentissage supervisé. L'apprentissage implique l'ajustement de tous les poids et les biais à leurs valeurs optimales.

Convolutional neural networks (CNNs)

Un réseau neuronal convolutionnel, également connu sous le nom de CNN, est une variante des réseaux de feed forward classiques (FFN) dans le domaine de l'incitation des facteurs biologiques [30].

On les a d'abord étudiés pour le traitement d'images où on peut trouver des motifs répétitifs : par exemple, une image avec des bords répétitifs et d'autres motifs. Les réseaux neuronaux convolutionnels (CNN) dévalent tous les autres algorithmes de machine learning classiques et connaissent un succès considérable dans les tâches de traitement de la vision par ordinateur (Computer Vision Tasks). Ils sont largement utilisés dans le traitement d'images et de vidéos, le traitement du langage naturel (NLP), les systèmes de recommandation, etc.

Plusieurs types de couches spéciales sont utilisés dans les réseaux convolutifs, telles que les couches de convolution, les couches de groupement (Pooling) et les couches entièrement connectées [38]. La figure 8 présente un modèle d'un réseau conventionnel unidimensionnel (1D CNN).

Convolution layers :

La convolution vise à extraire les caractéristiques de niveau supérieur. Il est composé d'un groupe de filtres (ou noyaux) apprenants, chacun représentant une fonctionnalité distincte en fonction du volume d'entrée. Ces filtres sont composés d'une couche de poids de connexion et ont un champ de réception restreint (la taille du noyau). Cependant, lors de la passe en avant (feedforward), chaque filtre est convolé sur la largeur et la hauteur du volume d'entrée, ce qui permet de calculer le produit des points entre les entrées et les valeurs du filtre, ce qui crée une nouvelle carte de caractéristiques qui représente mieux l'information.

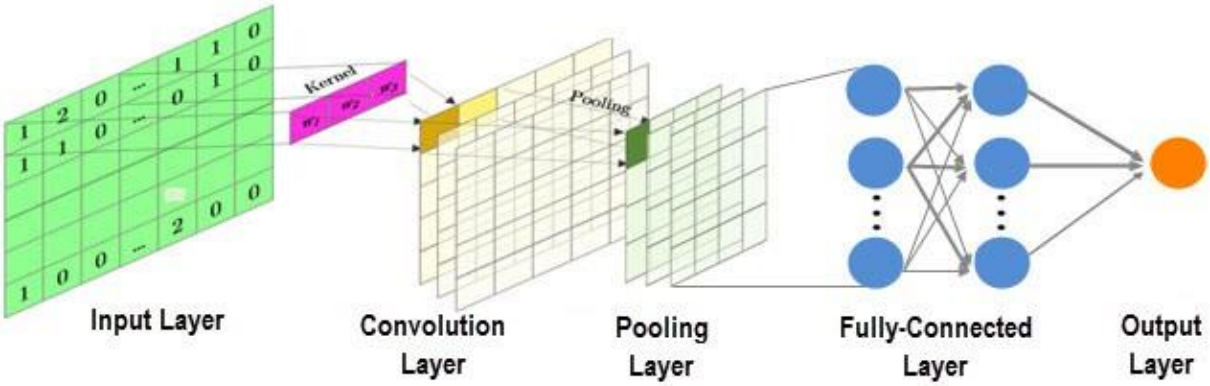


Figure 8: L'architecture d'un modèle de réseau neuronal convolutif

Par conséquent, le réseau acquiert des connaissances sur les filtres qui se mettent en place lorsqu'il détecte un type de caractéristique importante et spécifique à une certaine position spatiale dans l'entrée. Une opération de convolution 1D avec une entrée de dimension 1 est illustrée dans la figure 9.

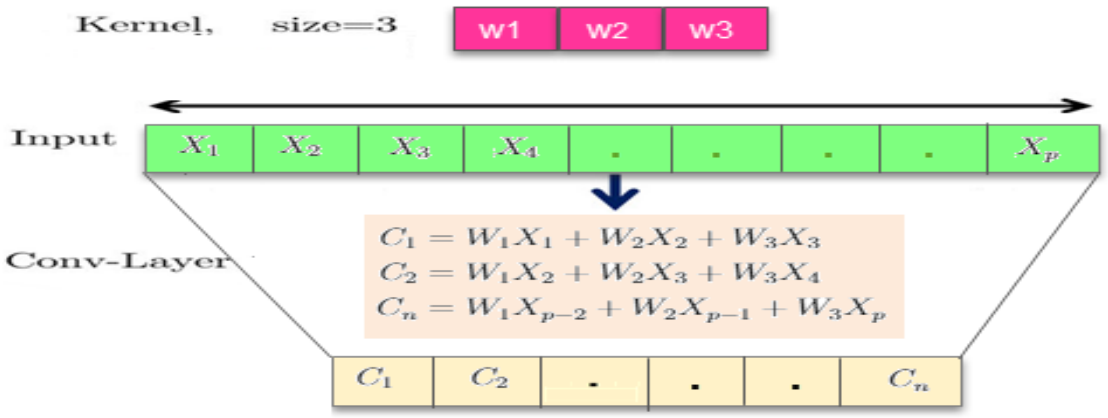


Figure 9 : convolution

Le noyau de convolution d'une couche convolutionnelle est commun, ce qui permet de diminuer considérablement le nombre de paramètres requis pour l'opération de convolution. Il sera immédiatement appliqué une fonction d'activation non linéaire après chaque couche convolutionnelle. La fonction d'activation "RectifiedLinearUnitsReLU" est utilisée pour les CNN profonds.

$$[(\boldsymbol{f(x) = max(0, x)}] \qquad (2)$$

, x renvoie à toutes les valeurs de x > 0 et à toutes les valeurs de x <= 0. Ils sont plus rapides que leurs concurrents avec des unités "TanhUnits" [29].

Pooling layers :

L'opération de mise en commun (Pooling) consiste à regrouper l'activation des neurones d'une couche en un seul neurone de la couche suivante après la transformation ReLU. Dans chaque entité d'entrée, la couche de pooling fonctionne de manière autonome, ce qui permet de diminuer progressivement la taille des représentations pour réduire le nombre de paramètres ou de poids, ce qui réduit le coût de calcul dans le réseau tout en préservant les informations les plus importantes. Elle permet également de réguler l'excès d'apprentissage.

Il a la possibilité d'utiliser deux différentes méthodes de mise en commun :

• Max-Pooling : utilise la valeur maximale de chaque groupe de neurones de la couche précédente.

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

Le pooling est une méthode non linéaire de sous-échantillonnage qui fonctionne de la même manière que la convolution. Le noyau de pooling se concentre sur le volume d'entrée et le divise en un ensemble de régions qui ne se superposent pas, et chaque sous-région génère une seule valeur en sortie, soit la valeur maximale pour Max-Pooling, soit la valeur moyenne pour Average-Pooling. La représentation la figure 10 illustre l'opération de Max-Pooling avec un entrée en 1D et un noyau en 2.

La couche de Pooling ne permet d'apprendre aucun paramètre. Ainsi, il est courant que ces couches ne soient pas prises en compte dans le nombre total de couches des réseaux de convolution.

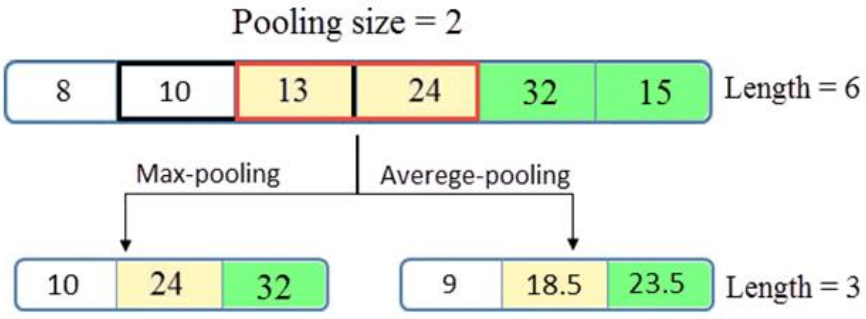


Figure 10 : pooling

Fully connected layers

Une ou plusieurs couches sont entièrement connectées à la fin d'un réseau CNN (chaque nœud de la première couche est connecté à chaque nœud de la couche suivante). Il s'agit de réaliser une classification en se basant sur les caractéristiques extraites des convolutions. Une fonction d'activation Softmax est présente dans la couche finale, générant une probabilité de 0 à 1 pour chacune des étiquettes de classes que le modèle essaie de prédire. Dans certaines structures de réseaux CNN récentes, il est possible de substituer les couches entièrement connectées par plusieurs couches de mise en commun moyennes (poolingaverage). Cela aide ces réseaux à diminuer de manière significative le nombre total de paramètres, ce qui favorise une meilleure prévention du sur-apprentissage [11].

Recurrent neural networks (RNNs)

Les réseaux neuronaux sont basés sur le fonctionnement des neurones biologiques du cerveau humain, qui sont perçus comme le centre de la réflexion et qui doivent parfois mémoriser certains événements afin de les utiliser ultérieurement sur la prise de décision. Les réseaux neuronaux classiques ne possèdent pas cette caractéristique, ce qui explique pourquoi le fonctionnement d'un réseau de neurones récurrents (RNN) repose sur le fait qu'un individu raisonne en se basant sur les connaissances qu'il a acquises et mémorisées auparavant [12].

Les réseaux RNNs sont des réseaux de type Feed-Forward qui utilisent un état interne (ou mémoire) pour prendre en compte tout ou partie des données vues précédemment (déjà fournies au réseau), en plus de la donnée vue actuellement, afin d'ajuster leur décision.

Le concept fondamental de ces réseaux repose sur la mise en place d'un calcul récurrent à travers les boucles dans l'architecture du réseau. La sortie du réseau est un mélange de son état interne (mémoire d'entrées) et de l'entrée finale, tandis que l'état interne évolue pour prendre en compte cette nouvelle donnée saisie. Cela favorise la persistance des informations en mémoire, comme illustré dans la figure 11.

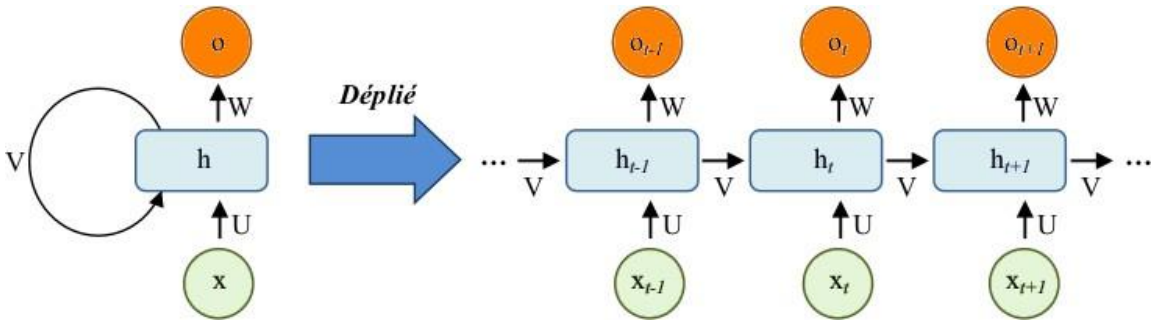


Figure 11 : L'architecture d'un modèle RNN

Grâce à ces caractéristiques, les réseaux récurrents conviennent aux situations où la présence d'une forme n'est pas seulement une information discriminante, mais aussi un ordre d'apparition, par exemple. Ils conviennent parfaitement aux tâches impliquant des données séquentielles, comme les données textuelles ou les données avec des caractéristiques temporelles. L'explication mathématique du transfert de mémoire est la suivante :

$$h_t = \delta(U_{xt} + V_{ht-1} + b_h) \quad (3)$$

$$O_x = Wh_t + b_y \quad (3.1)$$

Ou:

•h_t: Est l'état caché au temps t.

• x_t: Est l'entrée au même temps t.

• U, V, W : sont les matrices de pondération, Input-to-Hidden, Hidden-to-Hidden et Hidden-to-Output respectivement (connue comme des matrices de transition).

• b_h: Est la valeur du biais de l'état caché.

• b_y: Est la valeur du biais de sortie.

• O_t: Est la valeur de sortie au temps t.

• δ : La fonction d'activation est une fonction de non-linéarité. Une fonction sigmoïde logistique, également appelée tanh, est un outil courant de changement d'échelle qui permet de réduire les valeurs très grandes ou très petites dans un espace logistique, tout en rendant les gradients exploitables pour la régression.

Un réseau neuronal, en analysant une entrée xt, produit une valeur □□, À chaque fois que le temps passe, une boucle de rétroaction se produit. Chaque état masqué ht renferme des traces non seulement de l'état masqué précédent, mais aussi de tous ceux qui ont précédé ht−1, tant que la mémoire peut rester.

Unités de mémoire à court terme (LSTM)

Le réseau RNN est très lent car il tient compte de l'état sauvegardé précédent lors de la mise à jour du poids, des gradients lorsque l'entraînement devient de plus en plus petit et des erreurs qui n'ont pas pu se propager à la fin du réseau. Il n'y aura pas de changement significatif dans les résultats, ce qui signifie qu'il ne peut pas mettre à jour les poids. Les gradients de disparition (VanishingGradients) sont un problème du RNN. Afin de résoudre ce problème, les chercheurs allemands Sepp Hochreiter et Juergen Schmidhuber ont suggéré une architecture à mémoire longue et courte durée (LSTM) pour les réseaux neuronaux récurrents, ainsi que des étapes supplémentaires appelées GatedRecurrentUnits (GRU). On a employé ces étapes afin d'optimiser les performances et la précision des RNNs.

La méthode LSTM repose principalement sur l'état de la cellule. Son pouvoir consiste à supprimer ou à ajouter des données à l'état de la cellule. Des structures connues sous le nom de portes (Gates) régulent cette technique. Il est possible que ces derniers soient une fonction sigmoïde où une valeur de 1 indique que toutes les informations sont transmises et une valeur de 0 indique le contraire.

Le fonctionnement des architectures LSTM et GRU est identique. Toutefois, le GRU nécessite moins de paramètres d'entraînement, ce qui signifie qu'il utilise moins de mémoire et s'entraîne plus rapidement que les LSTM. Les ensembles de données utilisant une séquence plus longue sont plus précis avec le LSTM.

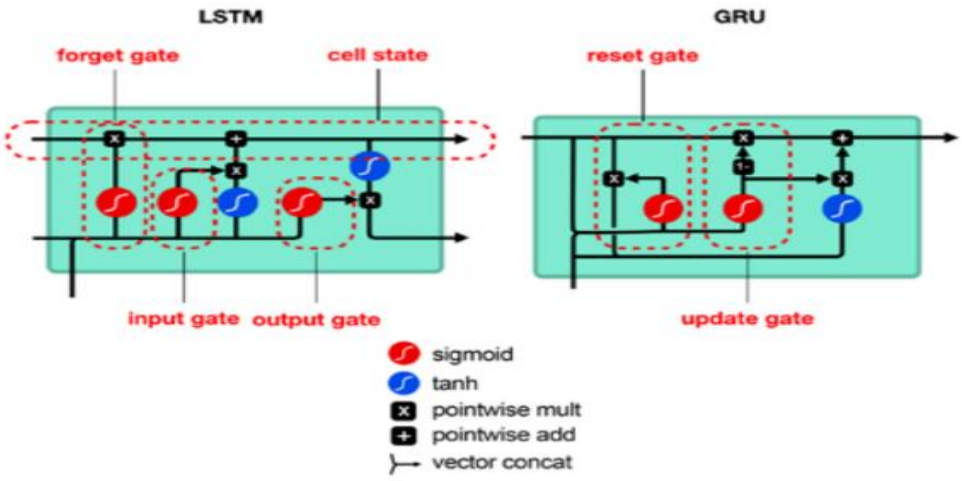


Figure 12 : L'architecture d'un modèle LSTM_GRU

Les cellules LSTM se révèlent les plus performantes pour stocker les données pertinentes lors de la rétro-propagation du gradient. Leur capacité à rectifier les disparités entre les prédictions sortantes et les catégories de référence réside dans le calcul du gradient d'erreur pour chaque neurone, en partant de la dernière couche et en passant à la première. La figure 12 présente les quatre couches interactives (sigmoïde et tanh), les trois portes et les opérations Pointwise qui gèrent le vecteur x à l'intérieur d'une cellule LSTM pendant un temps t.

1.9.4 Les ensembles de data set pour évaluer les IDSs basé à deep learning

Les ensembles de données employés dans les études publiées pour l'application de l'apprentissage approfondi dans le domaine de la cybersécurité jouent un rôle crucial dans la validation de toutes les approches DL proposées. Il est difficile d'accéder à certains de ces ensembles de données en raison de problèmes de confidentialité. Les ensembles de données spécialement conçus pour détecter les intrusions en matière de cybersécurité peuvent être de [30] :

Data-set public	**Type**	**Étiqueté**	**Année**	**[Réf]**
KDD99	Trafic du réseau	Oui	1999	[49]
NSL-KDD	Trafic du réseau	oui	2009	[37]
MAWI	Trafic internet	oui	2011	[36]
ISCX data set	Trafic du réseau	oui	2012	[37]
CIC DoS data set	Trafic du réseau	oui	2017	[37]
Bot-IoT data set	Trafic IoT	oui	2018	[38]
CICD DoS	Trafic du réseau	oui	2019	[37]

Tableau1. 4 : Ensembles de données public relatives au cyber sécurité

• benchmark data sets : Le public peut accéder à ces ensembles de données pour des recherches et évaluer les performances des algorithmes proposés.

• data sets privé : Les ensembles de données privées sont des données collectées à partir de sources publiques et en temps réel. Il est interdit de consulter ces ensembles de données à des fins de recherche.

• en temps réel : Les données sont recueillies en temps réel et sont qualifiées des ensembles de données en temps réel.

• collectés à partir de source accessibles au public : Ces informations sont recueillies à partir de différentes sources ouvertes au public. La majorité du temps, ces ensembles de données ne sont pas accessibles au public pour des raisons de recherche.

Parmi les ensembles de données accessibles au public qui sont couramment utilisés comme des références, on retrouve : DARPA, le KDD99, le NSL-KDD et l'ADFA-LD. [1] [58, 55, 30]

1.9.5 Des Travaux associés pour détecter les intrusions basées sur le DL

Selon des recherches antérieures, il a été démontré que les méthodes DL de détection des anomalies sont plus efficaces que d'autres algorithmes d'apprentissage machinent tels que la machine à vecteur de support (SVM) et les réseaux artificiels neuronaux traditionnels (ANN) [57].

Approche	Auteur, année	Data set	Avantages	Inconvénients	Acc	Type d'approche
ANN	Sumaiya et al 2020 [63]	NSL-KDD UNSW-NB15	• L'intégration de CFS et ANN augmente la précision.	• Le temps consommé est élevé.	99.31% - 98.4%	**Simple**
CNN	Nguyen et al 2018 [64]	KDDcup99	• Une enquête sur la technique de normalisation pour la saisie des données a été réalisée avec une comparaison des performances de chaque cas.	• Le modèle proposé étudiait un type d'attaques.	99.87%	**Simple**
CNN	Hu et al 2021 [65]	CSI	• L'IDS peut détecter les mouvements humains sur des voies sans visibilité directe (NLOS) avec plus de sensibilité • augmenter la fiabilité du système.	/	98,69 %	**Simple**
DNN Colony algorithm	gulia et al 2023 [66]	KDD NSL	• Taux de détection élevé • Sélection optimale des	/	96%	**Simple**

e			caractéristiques • Classification avancée. • Combinaison innovante • Modèles hybrides et techniques d'optimisation			
DNN	Syariful et al 2022 [67]	UNSW BOT – LOT	• Précision Exceptionnelle • Efficacité en Cyber sécurité pour l'IoT • Identification et Atténuation des Attaques • Indicateurs de Performance Solides	/	99.99%	**Simple**
RNN	Ashwaq et al 2022 [68]	NSL KDD	• réponse aux vulnérabilités croissantes de l'IoT. • l'utilisation efficace des RNN pour la détection des intrusions. • Une précision notable. • l'utilisation d'un data set de référence.	• Précision Moyenne. • Complexité Computationnelle • peut entraîner des problèmes de performance et de précision dans la détection d'intrusions sur de longues périodes.	87%	**Simple**
MLP	Louati et ktata [69]	KDD cup 99	• Précision Exceptionnelle. • améliorant la performance globale du système. • L'approche multi agent	/	99.95%	**Simple**

			pour la détection des intrusions est novatrice et ouvre des perspectives pour de nouvelles recherches et améliorations dans le domaine des systèmes de détection d'intrusion.			
MLP	Abdulrahmane et al 2020 [70]	NSL - KDD	• Haute Précision de Détection des attaques • Efficacité Améliorée avec des Couches Cachées Supplémentaires • Capacité de Gestion des Attaques DDoS. • Robustesse des Modèles contre les attaques. • améliorer significativement les systèmes de détection d'intrusion en réseau.	/	95 .6%	**Simple**
RNN LSTM	Syndney manbweksong 2023 [71]	NSL-KDD	• L'intégration de l'algorithme • Performance Élevée en Classification Binaire. • Efficacité, performances robustes et cohérentes. • Optimisation des Temps d'Entraînement. • Réduction des Faux Positifs.	/	99.49%	**Hyprid**

CNN ,LSTM,CNN-LSTM	Alkahtani et aldhyani 2021 [72]	LOTid 20	• Haute Précision d'Identification des Intrusions. • Le modèle hybride CNN-LSTM combine les avantages des CNN pour l'extraction de caractéristiques spatiales et des LSTM pour l'analyse temporelle. • Cette combinaison offre une solution robuste pour la détection des intrusions.	• Complexité Computationnelle • Dépendance à la Qualité des Données. • Maintenance et Mise à Jour des Modèles.	98.80%	**Hyprid**
CNN-MLP	Ashikuetdagl2021 [73]	UNSW-ND15	• haute précision • une adaptabilité et une résilience accrues. • une approche de réglage des hyper paramètres semi-dynamique • des performances comparatives supérieures. • l'utilisation d'un jeu de données de référence solide.	/	95.4% ,95.6%	**Hyprid**
RNN, LTSM	Samutha et al [74]	UNSW-NB18	• Amélioration, performance. • Efficacité dans la Sécurité Réseau. • l'intégration.	• La complexité computationnelle. • la dépendance aux grandes quantités de	99.4%	**Hyprid**

				données. • la nécessité de maintenance et de mises à jour fréquentes.		
LTSM, CNN	Su et al 2020 [75]	NSL-KDD	• Le modèle proposé peut capturer les caractéristiques du trafic réseau de manière plus complète.	• N'évalue pas les performances en termes de complexité temporelle.	84%	**Hyprid**
CNN ,RNN	Khan 2021 [76]	CSE-CIC-IDS2018	• Le problème du déséquilibre des classes a été résolu.	• Le modèle proposé a été testé sur un seul ensemble de données.	97.75%	**Hyprid**

Tableau1. 5 : Des Travaux associés pour détecter les intrusions basées sur le DL.

1.10 Conclusion

En conclusion, l'examen de la littérature existante souligne l'importance cruciale de la détection des intrusions dans les systèmes d'apprentissage en ligne. Des mécanismes efficaces de détection des intrusions protègent non seulement les données sensibles, mais maintiennent également l'intégrité et la fiabilité des environnements d'apprentissage en ligne. Néanmoins, plusieurs obstacles persistent, notamment la nature évolutive du cybermenaces, la nécessité de disposer d'ensembles de données complets pour une formation solide et les progrès continus des techniques d'intrusion.

L'apprentissage profond apparaît comme une voie prometteuse pour relever ces défis, grâce à sa capacité à discerner des modèles complexes et des anomalies dans le comportement des systèmes. Des études ont montré la supériorité des systèmes de détection d'intrusion basés sur l'apprentissage profond par rapport aux méthodes conventionnelles, indiquant une trajectoire prometteuse pour de nouveaux progrès dans ce domaine.

Essentiellement, ce chapitre souligne l'impératif de la détection des intrusions dans les systèmes d'apprentissage en ligne et met en évidence le potentiel de l'apprentissage profond pour renforcer les mesures de cybersécurité, favorisant ainsi une expérience d'apprentissage en ligne plus sûre et plus sécurisée pour toutes les parties prenantes impliquées. Des efforts de recherche continus sont indispensables pour affiner et faire progresser les technologies de détection des intrusions afin d'atténuer efficacement les menaces émergentes et de préserver l'intégrité des plateformes d'apprentissage en ligne.

Chapitre 02

Conception

2.1 Introduction

Le chapitre précédent a fourni un aperçu complet de la détection d'intrusion, y compris ses systèmes les plus populaires et les processus sous-jacents impliqués. Plus précisément, nous avons approfondi les subtilités des systèmes de détection d'intrusions, détaillant comment l'apprentissage profond et les réseaux neuronaux constituent l'épine dorsale de cette technologie.

Ce chapitre vise à présenter notre conception proposée pour le système de détection d'intrusion. Nous commencerons par aborder les différentes étapes impliquées dans le processus de détection d'intrusions. Ensuite, nous fournirons une description détaillée de la de la phase de détection, qui utilise un modèle CNN comme élément clé.

2.2 Architecture du système

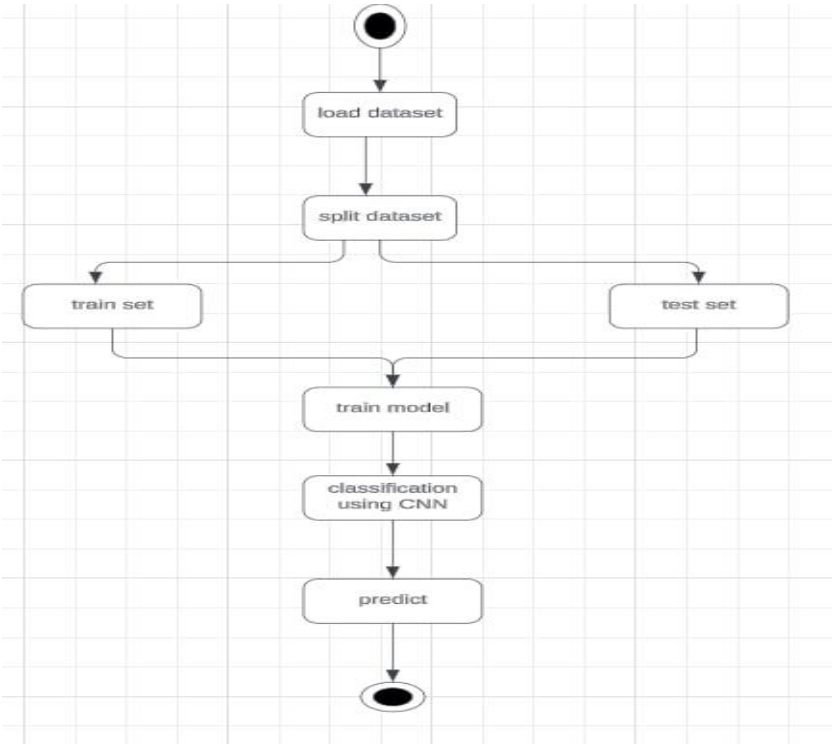


Figure 13 : **l'architecture de système**

2.3Collecte de données

Le processus de collecte de données implique la sélection de données de qualité à analyser. Nous avons utilisé l'ensemble de données d'intrusion KDD extrait de kaggle.com pour la mise en œuvre de l'apprentissage automatique. Le travail d'un analyste de données consiste à trouver des moyens et des sources pour collecter des données pertinentes et complètes, les interpréter et analyser les résultats à l'aide de techniques statistiques.

2.4 Les données NSL-KDD

L'ensemble de données NSL-KDD est proposé afin de résoudre certains des problèmes liés à l'ensemble de données KDD'99. Malgré les problèmes mentionnés par McHugh et le manque d'ensembles de données publiques pour les IDS basés sur les réseaux, cette nouvelle version de l'ensemble de données KDD peut ne pas être une représentation parfaite des réseaux réels existants. Cependant, nous croyons qu'elle peut être utilisée comme un ensemble de données de référence efficace pour aider les chercheurs à comparer différentes méthodes de détection d'intrusion. [62]

En outre, il est raisonnable de considérer le nombre d'enregistrements dans le train NSL-KDD et les ensembles de tests. Cette caractéristique permet de réaliser les expériences sur l'ensemble sans avoir à choisir au hasard une petite partie. Ainsi, les conclusions de l'évaluation des diverses études seront cohérentes et comparables.[62]

L'ensemble de données consiste en un vaste ensemble de trafics réseau capturés dans un environnement de réseau militaire simulé. Le trafic est composé à la fois de trafic normal et anormal, et il contient un total de 41 caractéristiques différentes, notamment le type de protocole, le type de service, le nombre de tentatives de connexion infructueuse, etc. Le jeu de données KDD99 est divisé en deux parties principales : le jeu d'apprentissage est le jeu de test. L'ensemble d'apprentissage contient 4 898 431 connexions réseau, tandis que l'ensemble de tests contient 311029 connexions réseau. Les deux ensembles sont étiquetés avec cinq types d'attaques différents :

Denial of Service (DoS), Probe, User to Root (U2R), Remote to Local (R2L) ET Normal.[59]

La répartition des données est très déséquilibrée, avec une majorité de connexions normales (97,06%) et une minorité de connexions malveillantes. L'attaque DoS est le type d'attaque le plus fréquent, représentant 80 % des connexions malveillantes. Bien que l'ensemble de données KDD99 ait été largement utilisé dans la recherche, il a également été critiqué pour ses limites, notamment son environnement simulé irréaliste, ses scénarios d'attaques obsolètes et l'absence de définition claire de ce qui constitue une attaque réussie. [60]

2.5 Caractéristiques du Dataset

Tableau 2. 1: Caractéristiques de base des connexions TCP individuelles.[61]

Nom de la fonction	Description	Type
Duration	length (number of seconds) of the connection	Continu
protocol_type	Type of protocol,e.g.tcp, udp.etc.	discrète
servive	network service on the destination e.g,http,telnet,etc.	discrète
src_bytes	number of data bytes from source to destination	Continu
Dst_bytes	number of data bytes from destination to source	Continu
Flag	normal or error status of the connection	Discrète
Land	Land 1 if connection is from/to the same host/post;0 otherwise	Discrète
Wrong_fragment	number of wrong fragments	Continu
urgent	number of urgent packets	Continu

Tableau2. 2 : Fonctionnalités de contenu dans une connexion suggérée par la connaissance du domaine.[61]

Nom de la function	Description	Type
Hot	number of hot indicators	continu
Num_failed_logins	number of failed login attempts	continu
Logged_in	1 if successfully logged in 0;otherwise	discrète
Mised	number of compromised conditions	continu
root_ shell	1if root shell is obtained ; 0 otherwise	discret
Su_attempted	If su root command attempted ;0 otherwise	discret
num_root	number of root accesses	continu
Num_file_creations	Number of file creation operations	continu
Num_shells	number of shell prompts	continu
Num_access_fils	number of operations on access control files	continu
Num_outbound_cmds	number of outbound commands in an ftp session	continu
Is_hot_login	1 if the login belongs to the hot list ;0 otherwise	discet
Is_guest_login	1 if the login is a guestlogin;0otherwise	discret

L'ensemble des données KDD99 est un ensemble de données volumineux et complexes avec de nombreuses fonctionnalités et types d'attaques. Pour le rendre plus accessible, une version retraitée appelée "kddcup.data_10_percent" a été créée. Cette version contient un échantillon aléatoire de 10 % de l'ensemble de données d'origine, ce qui réduit le coût et le temps de calcul nécessaires pour créer des modèles d'apprentissage automatique tout en fournissant un échantillon représentatif. L'utilisation de la version retraitée permet également de comparer et d'évaluer plus facilement les méthodes de détection d'intrusion sur le terrain.

Tableau 2. 3 : Caractéristiques du trafic calculé à l'aide d'une fenêtre de temps de deux secondes. [61]

Nom de la fonction	Description	Type
Count	number of connections to the same host as the current connection in the past two seconds	continu
	Note : the fellowing features refer to these sam_host connections.	
Serror_rate	% of connections that have “SYN” errors	continu
Rerror _rate	% of connections that have “REJ” errors	continu
Same_srv_rate	%of connections to the same service	continu
diff_srv_rate	%of connections to the different service	continu
Srv_count	number of connection to the same service as the current connection in the past two seconds	continu
	Note : the fellowing features refor to these sam_service connections .	
Srv_serror_rate	% of connections that have “SYN” errors	Continu
Srv_rerror_rate	% of connections that have “REJ” errors	Continu
Srv_diff_host_rate	%of connectionsto different hosts	Continu

Continu : Ils représentent des entités avec des valeurs numériques qui peuvent prendre n'importe quelle valeur dans une plage. Les exemples incluent la durée d'une connexion réseau, les octets transférés ou le temps entre les paquets.

Discret : Ils représentent des fonctionnalités avec un ensemble limité de valeurs ou de catégories. Les exemples incluent le type de protocole (TCP, UDP, ICMP), le type de service (HTTP, FTP, SSH) ou le type de connexion réseau (normal, suspect, attaque).

2.6 Fractionnement de l'ensemble de donnée

Le Fractionnement de l'ensemble de données en ensembles d'apprentissage et de test :

X_train : X irrévocables utilisés pour s'adapter au modèle d'apprentissage automatique.

X_test : X irrévocables utilisés pour évaluer l'ajustement du modèle d'apprentissage automatique.

Y_train : Y irrévocable utilisé pour s'adapter au modèle d'apprentissage automatique.

Y_test : Y irrévocable utilisé pour évaluer l'ajustement du modèle d'apprentissage automatique.

2.7 Optimiser

En apprentissage automatique, un optimiseur est un algorithme utilisé pour ajuster les paramètres d'un modèle afin de minimiser la fonction d'erreur ou de perte. La fonction de perte représente la différence entre la sortie prévue du modèle et la sortie réelle, et l'optimiseur est chargé de trouver l'ensemble de paramètres du modèle qui entraînera la perte la plus faible possible.

Le choix de l'optimiseur peut avoir un impact significatif sur les performances d'un modèle d'apprentissage automatique. Différents optimiseurs utilisent différents algorithmes pour rechercher l'ensemble optimal de paramètres de modèle, et certains peuvent être mieux adaptés que d'autres à certains types de données ou d'architectures de modèle.

Certains optimiseurs populaires utilisés dans l'apprentissage automatique incluent la descente de gradient stochastique (SGD), Adam, RMSprop et Adagrad. Ces optimiseurs utilisent diverses techniques telles que les taux d'apprentissage adaptatifs, l'élan et l'écrêtage de gradient pour améliorer l'efficience et l'efficacité du processus d'optimisation.

Dans le domaine de l'apprentissage automatique (machine learning), les optimiseurs sont des algorithmes utilisés pour minimiser une fonction de perte (ou coût) lors de l'entraînement d'un modèle. Ces fonctions de perte mesurent l'écart entre les prédictions du modèle et les valeurs réelles des données d'entraînement.

2.8 Model architecture

2.8.1 Model CNN

Le model consiste un réseau neuronal convolutif (CNN) en utilisant Keras pour une tâche de classification binaire. Le modèle commence par une couche d'entrée avec une forme de (40,1). Il inclut ensuite trois couches convolutives `Conv1D` avec respectivement 32, 64 et 32 filtres, toutes ayant une taille de noyau de 5 et utilisant la fonction d'activation ReLU. Chaque couche convolutive est suivie d'une couche de max-pooling pour réduire la taille des

représentations intermédiaires. Après les couches convolutives, une couche `Flatten` est utilisée pour transformer les sorties en un vecteur 1D. Ensuite, deux couches denses suivent : la première avec 128 unités et la seconde avec 64 unités, toutes deux utilisant l'activation ReLU. Enfin, la couche de sortie est une couche dense avec une seule unité sans fonction d'activation spécifiée, adaptée pour une classification binaire. Le modèle est compilé avec l'optimiseur Adam, la fonction de perte `BinaryCrossentropy` avec `from_logits=True`, et la métrique de précision.

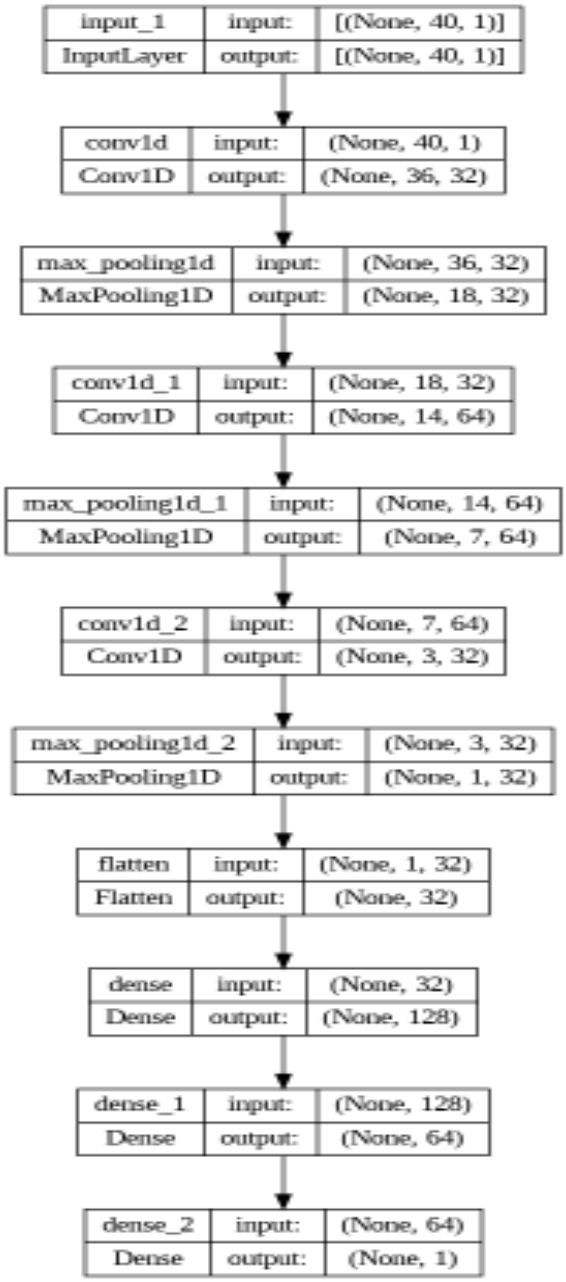


Figure 14 : notre architecture modèle

2.9 Configuration du modèle et nombre de paramètres

Le modèle "sequential_2" est une architecture de réseau de neurones convolutifs (CNN, Convolutional Neural Network) spécifiquement conçue pour l'analyse de données séquentielles. Sa structure est composée de trois couches de convolution 1D, chacune suivie d'une opération de max pooling qui réduit progressivement la dimensionnalité des séquences tout en capturant les caractéristiques les plus significatives. Les couches convolutives utilisent des filtres pour détecter des motifs locaux dans les données, ce qui est essentiel pour l'identification de tendances ou de motifs complexes dans les séries temporelles. Ensuite, une couche de flattening transforme les données multidimensionnelles en un vecteur unidimensionnel, permettant leur traitement par des couches denses, qui apprennent des représentations abstraites pour la prise de décision finale. Avec 33 313 paramètres entièrement trainables, ce modèle est relativement léger (environ 130,13 KB), ce qui le rend adapté pour des applications en temps réel ou sur des dispositifs avec des ressources limitées. Ce modèle est particulièrement efficace pour des tâches de classification ou de régression impliquant des données séquentielles, grâce à sa capacité à extraire et capturer des caractéristiques complexes présentes dans les séquences de données.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_6 (Conv1D)           (None, 36, 32)            192

 max_pooling1d_6 (MaxPoolin  (None, 18, 32)            0
 g1D)

 conv1d_7 (Conv1D)           (None, 14, 64)            10304

 max_pooling1d_7 (MaxPoolin  (None, 7, 64)             0
 g1D)

 conv1d_8 (Conv1D)           (None, 3, 32)             10272

 max_pooling1d_8 (MaxPoolin  (None, 1, 32)             0
 g1D)

 flatten_2 (Flatten)         (None, 32)                0

 dense_6 (Dense)             (None, 128)               4224

 dense_7 (Dense)             (None, 64)                8256

 dense_8 (Dense)             (None, 1)                 65

=================================================================
Total params: 33313 (130.13 KB)
Trainable params: 33313 (130.13 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 15: la configuration de model

2.10 Evaluation metrics

Parce que les performances des systèmes de détection d'intrusion sont basées sur la matrice de confusion pour évaluer la classification réelle et prévue, comme indiqué dans le tableau 2.4.

- True Positive (TP): -le modèle a correctement prédit la normale comme étant normale.
- TrueNegative (TN): - le modèle a correctement prédit l'attaquant attaquant.
- False Positive (FP): - le modèle identifie à tort une activité normale comme étant malveillante.
- False Negative(FN): - le modèle identifie à tort le trafic malveillant comme étant normal.

	Actual Class		
		Normal	**Attacker**
Predicted Class	**Normal**	TP	FP
	Attacker	FN	TN

Tableau2. 4 : shows the confusion matrix.

Les métriques suivantes sont les métriques d'évaluation les plus couramment utilisées : -

1) Accuracy :

Défini comme la moyenne des échantillons véritablement classés comme normaux ou attaqués sur le nombre total d'échantillons.

$$accuracy = \frac{TP+TN}{total\ number\ of\ samples} * 100 \qquad (4)$$

2) Précision (P) :

Est la proportion de prédictions positives faites par le classificateur qui sont vraies, comme dans l'équation suivante.

$$precision = \frac{TP}{TP+FP} \qquad (5)$$

3) Recall (R) :

C'est le pourcentage de bons positifs qui est véritablement détecté par le classificateur et appelé DR, TPR.

$$Recall = \frac{TP}{TP+FN} \qquad (6)$$

4) F1_Score (F1) :

C'est le résultat entre la précision et le rappel.

$$F1_{score} = 2 * \frac{precision*recall}{precision+recall} \qquad (7)$$

5) Courbe des caractéristiques de fonctionnement du récepteur (ROC)

ROC : - est un graphique basé sur le taux de vrais positifs (TPR) et le taux de faux positifs (FPR). Le modèle d'apprentissage automatique était meilleur si l'AUC était plus élevée.

$$AUC = \int_0^1 \frac{TP}{TP+FN} d\frac{FP}{TN+FP} \quad (8)$$

2.10 Conclusion

Sur la base des informations fournies, le chapitre **« Etude conceptuelle »** semble couvrir trois aspects principaux du projet :

• La conception du système :

Cela comprend probablement une discussion sur l'objectif général du système, la base d'utilisateurs prévue, ainsi que les principales caractéristiques et fonctionnalités du système.

• L'ensemble de données utilisé dans le projet :

Cela peut inclure des détails sur la taille et la composition de l'ensemble de données, la manière dont il a été collecté, tout prétraitement ou nettoyage effectué, ainsi que les défis ou limitations rencontrés lors de l'utilisation des données.

• Optimiseur

Cela comprend une description détaillée de l'Optimamazer utilisé dans le projet

• L'architecture et la configuration du modèle

Cela comprend probablement une description détaillée de l'architecture de réseau neuronal spécifique utilisée dans le projet, ainsi que des informations sur le nombre de paramètres et tous les hyper paramètres qui ont été réglés lors de la formation du modèle.

• Évaluation du modèle

La section Évaluation du modèle présente une analyse des performances du modèle en termes d'exactitude, de rappel, de précision, de score F1 et de matrice de confusion. Cette section fournit des informations essentielles sur l'efficacité du modèle et peut éclairer tout ajustement ou amélioration nécessaire du système.

Dans l'ensemble, le chapitre **« Étude conceptuelle »** fournit un aperçu de haut niveau des éléments clés du projet, y compris le problème abordé, les données utilisées pour entraîner le modèle, l'architecture et la configuration du modèle lui-même, l'évaluation du modèle.

Chapitre 03

Implémentation Et résultat

3 Implémentation Et résultat

3.1 Introduction

Ce chapitre est divisé en trois sections principales

• La première section décrit les outils et technologies choisis pour le développement du système. Il fournit un aperçu des langages de programmation, des bibliothèques et des Framework utilisés dans la mise en œuvre du système. Les outils choisis pour le projet sont discutés en détail, ainsi que leurs avantages et leurs limites.

• La deuxième section discute et compare les résultats obtenus à partir des tests du système.

Les mesures d'évaluation utilisées pour évaluer les performances du système sont présentés, et les résultats sont analysés et comparés aux attentes et aux exigences du système.

• La troisième et dernière section du chapitre présente l'interface du système et affiche quelques captures d'écran de la phase de test.

Dans l'ensemble, ce chapitre fournit un aperçu complet du développement et des tests du système de vision par ordinateur, depuis les outils et technologies utilisés jusqu'à l'interface utilisateur finale et les résultats des tests.

3.2 Représentation des outils de développement

3.2.1Environnement physique

Nous utilisons ce matériel et ces logiciels dans les processus de formation et de test.

Operating system	64-bit Windows 10.
CPU	Intel(R) Pentium(R) CPU B960 @ 2.20GHz 2.20 GHz
RAM	4.00 Go

Tableau 3. 1desktop hardware

3.2.2 Bibliothèques utilisés dans l'implémentation

Google Colab : Un service cloud basé sur Jupyter Notebook vous permet de développer des applications d'apprentissage en profondeur en Python. Il offre un processeur GPU gratuit, 12 Go de RAM et plus de 100 Go d'espace de stockage. Tout ce dont vous avez besoin pour accéder à ce service est un compte Google.

Python : Python est un langage de programmation interprété de haut niveau qui a été publié pour la première fois en 1991 par Guido van Rossum. Il est conçu pour être facile à lire, à écrire et à maintenir, avec une syntaxe qui met l'accent sur la lisibilité et la simplicité du

code. Python est un choix populaire pour un large éventail d'applications, notamment le développement Web, le calcul scientifique, l'analyse de données, l'intelligence artificielle et l'automatisation. Il dispose d'une vaste bibliothèque standard, ainsi que de nombreux packages et modules tiers, ce qui en fait un outil polyvalent. langage qui peut être utilisé pour une variété de tâches. Python est également connu pour son solide support communautaire, avec de nombreuses ressources, documentations et didacticiels disponibles en ligne, ce qui en fait un langage accessible aux développeurs de tous niveaux. Dans l'ensemble, Python est un langage puissant et flexible qui est devenu l'un des langages de programmation les plus populaires au monde.[Pyt]

TensorFlow : est un framework d'apprentissage automatique open source populaire développé par Google. Il permet aux développeurs de créer et de former divers modèles d'apprentissage automatique, notamment des réseaux de neurones, et de les déployer pour diverses applications. TensorFlow fournit une large gamme d'outils et de bibliothèques pour la manipulation des données, la formation de modèles et le déploiement, ce qui en fait un outil puissant et flexible pour l'apprentissage automatique. Il prend en charge divers langages de programmation, notamment Python, C++ et Java, et peut être utilisé sur diverses plates-formes, notamment les processeurs, les GPU et les appareils mobiles. TensorFlow a été largement adopté dans l'industrie et le monde universitaire pour un large éventail d'applications [80]

keras : est un framework d'apprentissage profond open source populaire développé en Python. Il fournit une interface de haut niveau pour créer et former des réseaux de neurones, permettant aux développeurs de créer et de prototyper rapidementmodèles d'apprentissage profond avec un minimum de code. Keras prend en charge plusieurs backends, notamment TensorFlow, CNTK et Theano, et peut être utilisé sur diverses plates-formes, notamment les processeurs, les GPU et les systèmes distribués. Grâce à sa simplicité, sa modularité et sa flexibilité, Keras a été largement adopté par les chercheurs et les développeurs pour créer et former des modèles d'apprentissage profond.[78]

NumPy : (abréviation de Numerical Python) est une bibliothèque open source populaire pour le calcul scientifique en Python. Il fournit un objet tableau multidimensionnel, ainsi qu'un large éventail de fonctions mathématiques et d'outils pour travailler avec des tableaux. NumPy est largement utilisé dans divers domaines du calcul scientifique, tels que la physique, la finance, l'ingénierie et l'apprentissage automatique, entre autres.[79]

Pandas : est une bibliothèque open-source Python populaire utilisée pour l'analyse de données et la manipulation de structures de données tabulaires. Elle offre des structures de

données puissantes et flexibles, notamment les DataFrames, qui permettent de stocker et de manipuler des données de manière efficace. Pandas est largement utilisé dans le domaine de la science des données et de l'analyse de données pour effectuer des opérations telles que le filtrage, le tri, l'agrégation et la visualisation de données tabulaires.

VS Code : est un éditeur de code open source gratuit développé par Microsoft. Il fournit un environnement puissant et flexible pour le codage, le débogage et la création d'applications dans un large éventail de langages et de plates-formes. VS Code offre un large éventail de fonctionnalités, notamment la prise en charge de la coloration syntaxique, de la complétion du code, du débogage, du contrôle de version et des extensions. [81]

3.3. Préparation data

Dans cette partie, nous discutons sur le test et le entrainement data .nous utilisons la fonction train-test split pour divise les données en ensembles de formation et de test et entrainement nous sauvegardons cette ensemble de donnée dans les tableau NumPy pour faciliter charger ces tableau en mémoire et nous utilisons train et test shap pour divise les donnée en ensemble entrainement et de test, o 20% taille de test et 80%taille de entraînement.

3.3.1Teste and train data

3.3.1.1Train_test_split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=2500)
```

Figure 16: train test split

Nous utilisons la fonction `train_test_split` de scikit-learn pour diviser les données en ensembles de formation et de test. Voici ce que représente chaque variable :

- **`X`**:features matrix.
- **`y`**:target variable or labels.
- **`test_size`** : la proportion de l'ensemble de données à inclure dans la répartition de test. Ici, il est fixé à 0,2, ce qui signifie que 20 % des données seront utilisées pour les tests et les 80 % restants pour la formation.

- **`random_state`** : Ceci est utilisé pour initialiser le générateur de nombres aléatoires, ce qui est important pour la reproductibilité. Le définir sur une valeur spécifique garantit que chaque fois que vous exécutez le code, vous obtenez la même répartition.

Après avoir exécuté cette ligne de code, nous avons « X_train » et « y_train » contenant 80 % de vos données pour la formation, et « X_test » et « y_test » contenant 20 % de vos données pour les tests.

3.3.1.2 Sauvegarder data

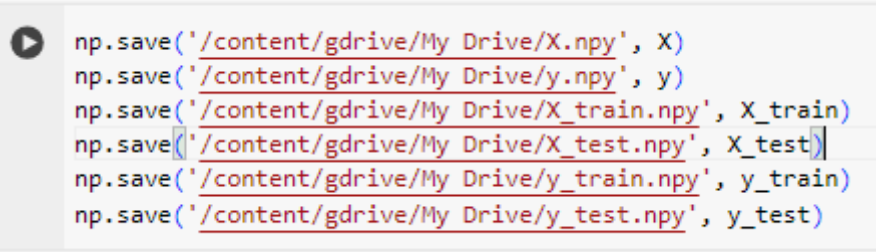


Figure 17 : sauvegarder data

Nous sauvegardons vos tableaux de données à l'aide de la fonction `np.save` de NumPy. Cette fonction vous permet de sauvegarder les tableaux NumPy sur le disque au format binaire.

Voici un aperçu de ce que vous faites :

- **`np.save ('/content/gdrive/My Drive/X.npy', X)`** : enregistrement de l'intégralité de la matrice de fonctionnalités `X` dans un fichier nommé "X.npy" dans le répertoire spécifié.

- **`np.save ('/content/gdrive/My Drive/y.npy', y)`** : enregistrement de la variable cible ou des étiquettes `y` dans un fichier nommé "y.npy".

- **`np.save ('/content/gdrive/My Drive/X_train.npy', X_train)`** : enregistrement des fonctionnalités de l'ensemble d'entraînement `X_train` dans un fichier nommé "X_train.npy".

- **`np.save ('/content/gdrive/My Drive/X_test.npy', X_test)`** : enregistrement des fonctionnalités de l'ensemble de tests `X_test` dans un fichier nommé "X_test.npy".

- **`np.save ('/content/gdrive/My Drive/y_train.npy', y_train)`** : enregistrement des étiquettes de l'ensemble d'entraînement `y_train` dans un fichier nommé "y_train.npy".

- **`np.save ('/content/gdrive/My Drive/y_test.npy', y_test)`** : enregistrement des étiquettes de l'ensemble de tests `y_test` dans un fichier nommé "y_test.npy".

De cette façon, vous pouvez facilement charger ces tableaux en mémoire ultérieurement en utilisant « np.load » pour une analyse plus approfondie ou un entraînement de modèle sans avoir besoin de les régénérer.

3.3.1.3 Train et test shap

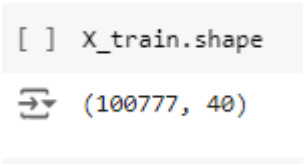


Figure 18 : Train shape

-Après avoir divisé les données en ensembles d'entraînement et de test en utilisant une taille d'entraînement de 0,8 (80 %), l'ensemble d'entraînement « X_train » contient 100 777 échantillons, chacun avec 40 fonctionnalités. Cela signifie que nous avons 100 777 instances dans l'ensemble de tests, chaque instance ayant 40 fonctionnalités.

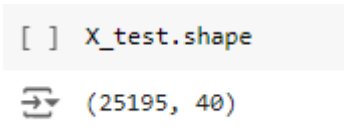


Figure 19 train shape

-Après avoir divisé les données en ensembles d'entraînement et de test en utilisant une taille de test de 0,2 (20 %), votre ensemble de test « X_test » contient 25 195 échantillons, chacun avec 40 fonctionnalités. Cela signifie que nous avons 25 195 instances dans l'ensemble de test, chaque instance ayant 40 fonctionnalités.

3.3 Expérimentations, comparaison et discussion des résultats obtenus

Dans cette section, nous discuterons et comparerons les résultats obtenus à partir d'un grand nombre de modèles développés pour notre projet. Chaque modèle sera analysé individuellement, avec un nom récapitulatif attribué à chacun en fonction de son architecture et de son nombre d'unités. Nous présenterons ensuite les résultats, y compris l'exactitude, la perte, l'exactitude de la validation et la perte de validation, et fournirons une discussion détaillée des résultats. Enfin, nous résumerons les résultats dans un tableau pour fournir un aperçu complet des performances de chaque modèle.

3.4 Expériences et résultats

Les modèles sont numérotés, et certains d'entre eux possèdent des sous-modèles.

Pour chaque modèle/sous-modèle, les mesures de performances suivantes sont fournies :

- **Accuracy:** a measure of how many predictions the model got right (number of correct predictions/ total number of predictions)
- **Loss:** a measure of the error of the model's predictions (usually measured as a difference between predicted and actual values)

- **Validation Accuracy:** the accuracy of the model's predictions on a validation dataset (a subset of the dataset that was not used during training, used to evaluate the model's performance on unseen data)
- **Validation Loss:** the loss of the model's predictions on the validation dataset
- **Epoch:**the number of epochs (complete passes through the dataset) that the model was trained for.

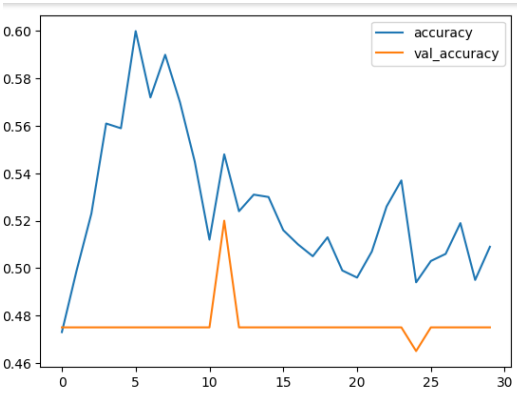
3.4.1En termes de modèles

Dans cette section, nous montrerons une partie de notre expérience avec le modèle CNN avec Adam optimazer:

Model CNN 1 : con2D-32-3-3-Relu-BatchNormalization-Maxpool2D-2-2-Dropout-0.25-con2D-64-3-3-Relu-BatchNormalization-MaxPool2D-2-2-Dropout-0.25-conv2D-128-3-3-Relu-BatchNormalization-MaxPool2D-2-2-Dropout-0.25-flatten-Dense-256-Relu-Dense-2-Sigmoid.

Le model utilise une couche convolutionnelle (Conv2D) avec 32 filtres de taille 3x3, suivie d'une fonction d'activation ReLU. Ensuite, une normalisation par lot (BatchNormalization) est appliquée pour normaliser les activations, suivie d'une couche de max pooling (MaxPool2D) avec un filtre de 2x2 pour réduire les dimensions spatiales. Une couche de dropout avec un taux de 0,25 est ajoutée pour éviter le surapprentissage. Ce bloc est répété avec des couches convolutionnelles successives utilisant 64 et 128 filtres respectivement, chacune suivie de la normalisation par lot, de max pooling et de dropout. Après ces couches de convolution, la sortie est aplatie (flatten) et passée à une couche entièrement connectée (Dense) de 256 unités avec une activation ReLU. Enfin, une couche de sortie dense avec 2 unités et une activation sigmoïde (Sigmoid) est utilisée pour la classification binaire.

Model accuracy



model loss

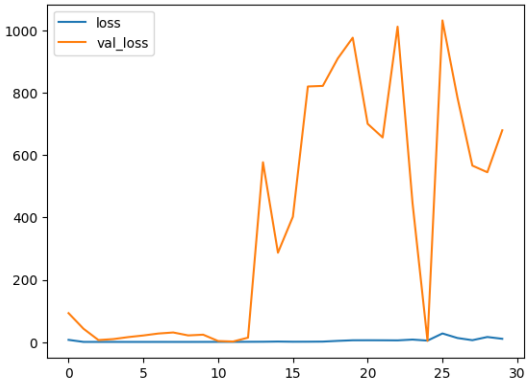
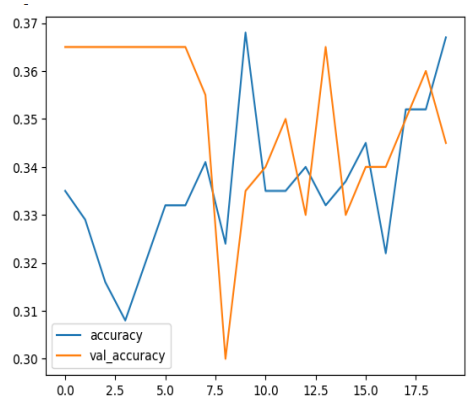


Figure 20: Chart of accuracy and loss of Model CNN1

Le graph présente un modèle de réseau neuronal entraîné sur 30 époques avec des résultats variables. À l'époque 1, la perte d'entraînement est de 7,2733 et la précision d'entraînement est de 47,30%, tandis que la perte de validation est de 92,9350 et la précision de validation est de 47,50%. Cependant, malgré une certaine amélioration initiale, les métriques de validation stagnent autour de 47,50% de précision et montrent des pertes très élevées dans les époques ultérieures, atteignant jusqu'à 1031,6179 à l'époque 26. À l'époque 30, la perte d'entraînement 10.6649 avec une précision de 50,90%, et la perte de validation à 679.4560 avec une précision de47.50%.

Model CNN1(a) :conv2D-32-3-3-Maxpool2D-3-3-Dropout-0.5-conv2D-256-3-3-Dropout-0.1-conv2D-128-3-3-Maxpool2D-Dropout-0.3-Dense-128-Dense-3.

Model accuracy



Model loss

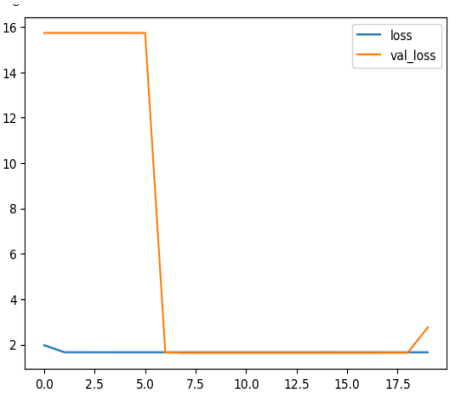


Figure 21: Chart of accuracy and loss of Model CNN1(a)

Le graph présent un modèle de réseau neuronal entraîné sur 20 époques démontre des défis importants dans l'amélioration des performances du modèle. Initialement, le modèle montre

une précision d'entraînement de 33,50 % et une perte de 1,9673. Au fil des époques, la précision de l'entraînement fluctue légèrement, se terminant à 36,70 % avec une perte de 1,6552 aux 20 époques. Sur l'ensemble de validation, la précision commence à 36,50 % et connaît des fluctuations mineures, pour finalement s'établir à 34,50 %, avec une perte finale de 2,7494.

Model CNN1(b) : No dropout

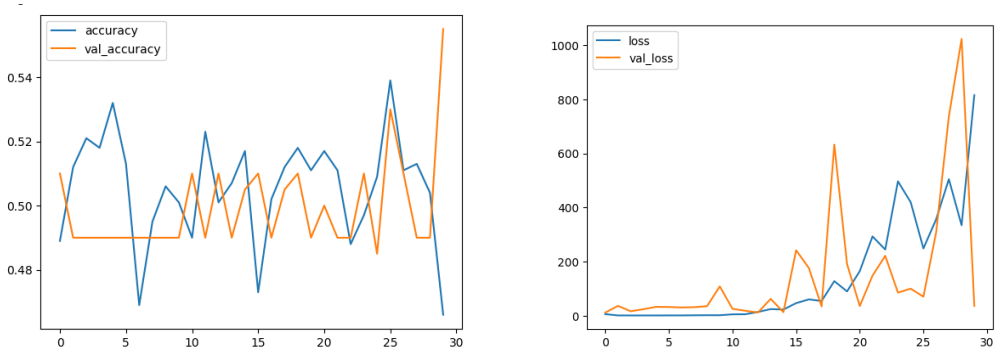


Figure 22: Chart of accuracy and loss of Model CNN1(b)

Le graph présente un modèle de réseau neuronal entraîné sur 30 époques indique une instabilité significative et une mauvaise convergence tant en performance d'entraînement que de validation. Initialement, le modèle commence avec une perte d'entraînement relativement élevée (5.4452) mais une précision modérément bonne (48.90%). La perte de validation est également élevée (11.0632), mais la précision de validation est comparable à celle de l'entraînement (51.00%). À partir de la deuxième époque, la perte de validation varie de manière spectaculaire, atteignant des valeurs aussi élevées que 632.7533 et même 1023.6921. Malgré ces pertes élevées, la précision de validation reste autour des valeurs initiales (49.00% à 55.50%), les précisions d'entraînement et de validation restent relativement stagnantes autour de 49-51%.

ModelCNN2 :con1D-62-3-Same-Relu-Maxpool1D-2-flatten-dropout-0.5-conv1D-62-3 Same-Relu-Maxpool1D-2-flatten-Dropout-0.5-conv1D-124-3-Same-Relu-Maxpool1D-2-flatten-Dropout-0.5-Dense-256-Relu-Dropout-0.5-Dense-5-softmax.

Compile le model avec le jeu de donnée **mnist [mnist=tf.keras.datasetsmnist] :**

Le model utilise une couche convolutionnelle (Conv1D) avec 62 filtres de taille 3, une fonction d'activation ReLU et un padding "same" pour conserver les dimensions d'entrée. Cette couche est suivie d'une couche de max pooling (MaxPool1D) avec un filtre de taille 2 pour réduire la dimension des caractéristiques. Ensuite, la sortie est aplatie (flatten) et une

couche de dropout avec un taux de 0,5 est appliquée pour éviter le surapprentissage. Ce bloc est répété une deuxième fois avec une couche convolutionnelle (Conv1D) ayant les mêmes spécifications : 62 filtres de taille 3, activation ReLU, padding "same", suivie d'une couche de max pooling, un aplatissement, et une dropout à 0,5. La troisième convolutionnelle (Conv1D) utilise 124 filtres de taille 3, avec une activation ReLU et un padding "same", suivie encore une fois par une couche de max pooling, un aplatissement et un dropout à 0,5. Après les couches convolutionnelles, une couche dense (Dense) de 256 unités avec une activation ReLU est ajoutée, suivie d'une couche de dropout à 0,5 pour prévenir le surapprentissage.

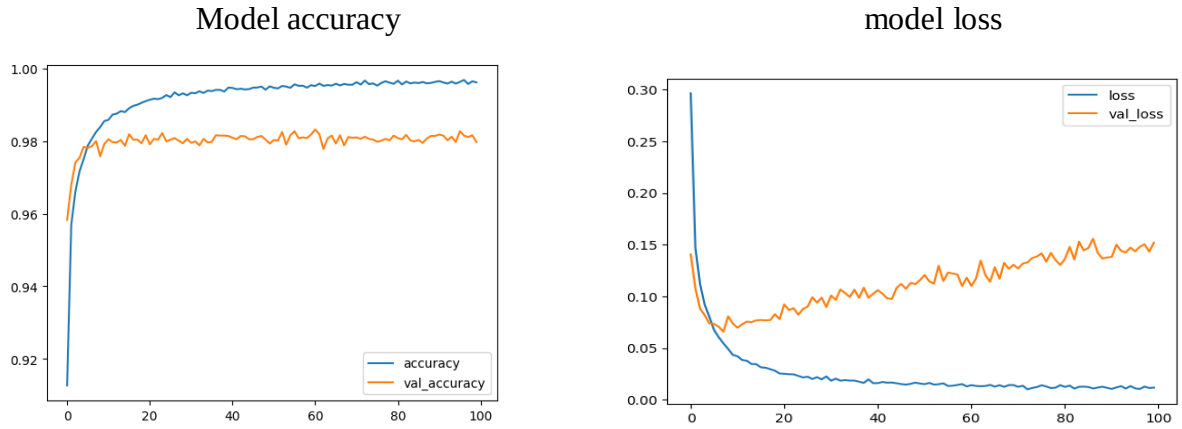


Figure 23: Chart of accuracy and loss of Model CNN 2

Le graph présente un modèle de réseau neuronal entraîné sur 100 époques démontrent des améliorations significatives des mesures de performance, à la fois sur les ensembles de formation et de validation. Initialement, la précision de l'entraînement commence à 91,27 % à l'époque 1 et atteint de manière impressionnante 99,58 % à l'époque 100, ce qui indique l'apprentissage efficace du modèle. Parallèlement, la perte de formation diminue régulièrement, passant de 0,2963 à 0,0133, démontrant ainsi la capacité du modèle à minimiser les erreurs. Du côté de la validation, la précision commence à 95,83 % et s'améliore à 98,12 % à l'époque finale, ce qui suggère que le modèle se généralise bien aux données invisibles malgré quelques fluctuations. La perte de validation commence à 0,1406 et se termine à 0,1472, montrant une tendance initiale à la baisse mais avec plus de variabilité au cours des époques ultérieures, ce qui pourrait indiquer un surajustement potentiel à mesure que le modèle devient de plus en plus spécialisé dans les données d'entraînement.

Model CNN2(a) : flatten-28-28-Dense-130-Relu-Dropout-0.5-Dense-10-softmax

Le model utilise une couche Flatten qui transforme l'image bidimensionnelle en un vecteur unidimensionnel. Ensuite, une couche Dense de 130 unités applique la fonction d'activation ReLU pour introduire de la non-linéarité dans le modèle. Pour régulariser le modèle et

prévenir le sur apprentissage, une couche de Dropout est ajoutée, désactivant aléatoirement la moitié des neurones à chaque itération d'entraînement. Enfin, la couche de sortie est une couche Dense avec 10 unités, activée par softmax.

Model accuracy model loss

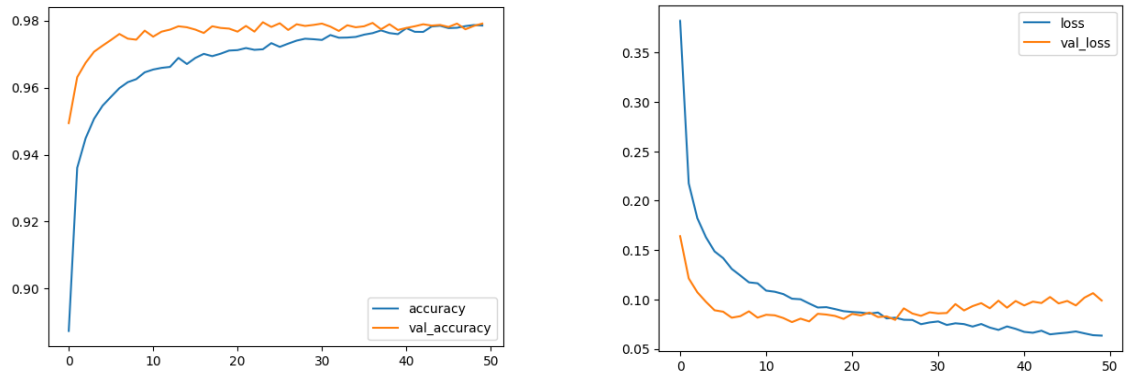


Figure 24: Chart of accuracy and loss of Model CNN 2(a)

Le graph présente un modèle de réseau neuronal entraîné sur 50 époques présente une image détaillée de son processus d'apprentissage. Initialement, le modèle présente une précision de 88,74 % et une perte de 0,3821, qui s'améliore progressivement jusqu'à une précision de 97,85 % et une perte de 0,0634 à l'époque finale. Cette diminution constante des pertes et cette augmentation de la précision sur l'ensemble d'entraînement démontrent la capacité du modèle à apprendre et à généraliser efficacement des modèles à partir des données d'entraînement. Sur l'ensemble de validation, la précision commence à 94,94 % et atteint 97,91 %, avec une perte commençant à 0,1641 et se terminant à 0,0989. Malgré quelques fluctuations, la précision de la validation montre également une amélioration globale, indiquant que le modèle se généralise bien aux données invisibles. Cependant, il existe de légers signes de surapprentissagevers la fin de la formation, car la perte de validation commence à s'écarter légèrement de la perte de formation.

Model CNN 3:conv1D32-5-Relu-MaxPool1D-64-5-Relu-MaxPool1D-32-5-Relu-MaxPool1D-flatten-128-Dense-Relu-64-Dense-Relu-1-Dense .

Le model utilise une couche d'entrée suivie de trois blocs de couches convolutives 1D et de couches de max-pooling, où chaque bloc extrait progressivement des caractéristiques de plus en plus complexes des données d'entrée. Le premier bloc utilise une couche convolutive avec 32 filtres et un noyau de taille 5 suivie d'une max-pooling, le deuxième bloc utilise 64 filtres et un noyau de taille 5 avec une max-pooling, et le troisième bloc revient à 32 filtres et un noyau de taille 5 suivi d'une max-pooling. Ensuite, une couche de flatten convertit les sorties 2D en un vecteur 1D, qui est ensuite passé à deux couches denses entièrement connectées

avec des activations ReLU, comptant respectivement 128 et 64 unités. Enfin, une couche dense avec une seule unité produit la sortie pour la classification binaire. Le modèle est compilé avec l'optimiseur Adam.

Model accuracy model loss

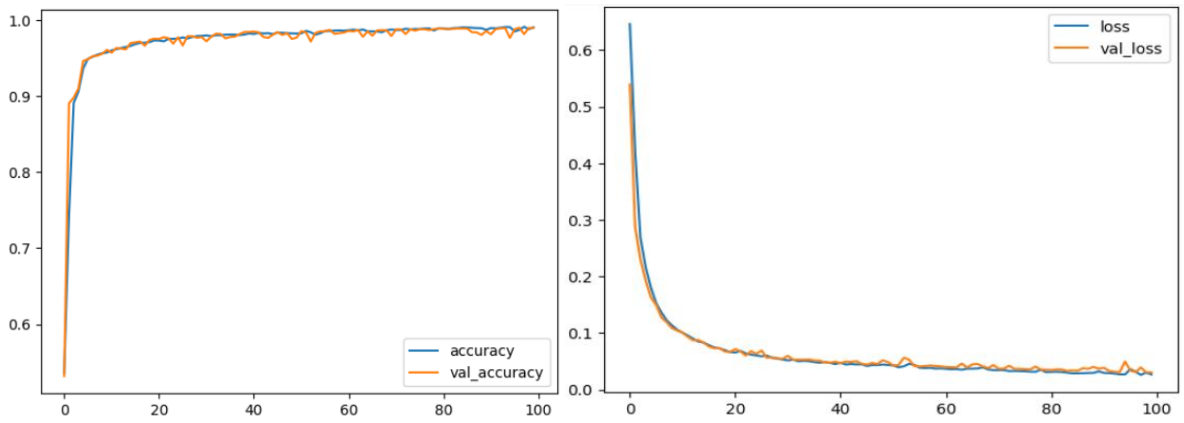


Figure 25: Chart of accuracy and loss of Model CNN 3.

Le graph présente un modèle de réseau neuronal entraîné sur 100 époques avec des performances impressionnantes. Initialement, à l'époque 1, la perte d'entraînement est de 0,6461 et la précision d'entraînement est de 53,53%, tandis que la perte de validation est de 0,5394 et la précision de validation est de 53 ,16%. À la fin de l'entraînement, à l'époque 100, la perte d'entraînement chute à 0,0268 avec une précision de 99,09%, et la perte de validation à 0,0303 avec une précision de 99,03%.

Son tableau montre les mesures de performances de plusieurs modèles, notamment la précision, la perte, la précision de validation, la perte de validation et le nombre d'époques. Le modèle le plus performant de chaque groupe est surligné en vert, tandis que le modèle le moins performant est surligné en rouge.

Model	**Accuracy**	**loss**	**Val accuracy**	**Val loss**	**Epoch**
1	50.90%	10.6649	47.50%	676.4560	30
1(a)	36.70%	1.6552	34.50	2.7494	20
1(b)	46.60%	815.7731	55.50%	35.4357	30
2	99.62%	0.0118	97.97%	0.1519	100
2(a)	97.85%	0.0634	97.91%	0.0989	50
3	99.09%	0.0268	99.03%	0.0303	100

Tableau3. 2 : Model Performance

Sur la base des résultats fournis, il est difficile de déterminer le « meilleur » modèle car il dépend de la tâche et des critères spécifiques. Voici cependant quelques observations :

- **Le modèle 2** présente la précision la plus élevée et la perte la plus faible sur l'ensemble d'apprentissage, mais sa précision de validation est inférieure à celle de certains autres modèles.
- **Le modèle 2(a)** a une grande précision sur les ensembles d'entraînement et de validation, avec une faible perte sur l'ensemble de validation.
- **Le modèle 3** présente une grande précision à la fois sur les ensembles de formation et de validation, avec la perte de validation la plus faible parmi tous les modèles.

Après avoir analysé les performances de plusieurs modèles, le mode 3 a été sélectionné comme le modèle le plus performant en raison de sa grande précision de validation et de sa faible perte de validation. Par conséquent, le modèle 3 peut être considéré comme le choix optimal pour la tâche donnée.

3.4.2 En termes de modèle LSTM

Model LSTM :Layers-40-1-layers-LSTM-20-Tanh-Layers-Reshape-20-1-layers-Reshape-20-1-Layers-LSTM-20-Tanh-layers-Reshape-20-1-layers-LSTM-16-Tanh-Layer-Dense-128-Relu-Layers-Dense-64-Relu-layers-Dense-1.

Le modèle LSTM est composé de plusieurs couches, en commençant par une couche LSTM avec 40 unités suivie de couches LSTM successives avec 20 et 16 unités, toutes utilisant la fonction d'activation Tanh pour capturer les dépendances temporelles complexes dans les données. Chaque couche LSTM est suivie d'une couche de reshaping pour ajuster la forme des données pour la prochaine étape. Ensuite, des couches denses avec 128 et 64 neurones utilisant la fonction d'activation ReLU sont incluses pour extraire et raffiner les caractéristiques pertinentes avant la couche de sortie finale à une seule unité, utilisée pour la classification. Cette architecture permet de capturer les séquences temporelles et d'appliquer des transformations non linéaires pour une meilleure performance en classification.

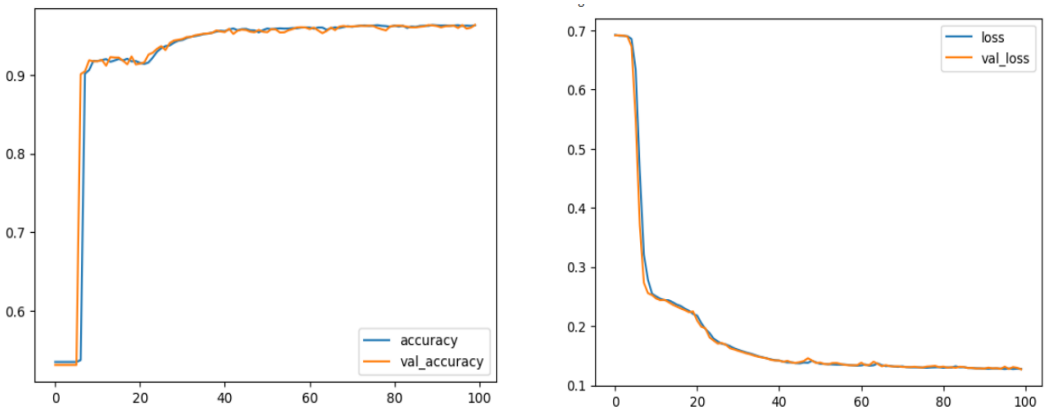


Figure 25: Chart of accuracy and loss of Model LSTM

Le graph présente un modèle entraîné 100 époques le modèle a montré une amélioration significative des performances, particulièrement notable après les époques initiales. Partant d'une précision d'entraînement et de validation juste supérieure aux estimations aléatoires (environ 53 %), le modèle a connu des difficultés au départ, ce qui suggère des difficultés dans l'apprentissage des caractéristiques de l'ensemble de données. Cependant, un tournant s'est produit vers l'époque 6 avec une baisse substantielle des pertes et une augmentation de la précision de la validation, marquant la capacité du modèle à commencer à apprendre des modèles significatifs. Une amélioration majeure a été observée à l'époque 7, avec une précision de validation grimpant à environ 90 %. Une amélioration continue a été observée jusqu'à l'époque 33, où la précision de la validation a culminé à environ 94,8 %. À partir de l'époque 40, les mesures de précision se sont stabilisées, oscillant entre 95 % et 96 %, avec des valeurs de perte stabilisées, indiquant que le modèle avait appris à bien généraliser. À l'époque finale, le modèle a atteint une précision de formation et de validation louable d'environ 96,3 %.

3.5. Comparaison enter le model CNN ET LTSM

Apres entraînez deux modèles différents, un modèle LSTM et un modèle CNN, pour une tâche de classification. Les journaux de formation que vous avez fournis montrent les mesures de perte et de précision pour chaque époque au cours du processus de formation.

Pour le modèle LSTM :

- L'entraînement a débuté avec une précision d'environ 53,53% et s'est progressivement amélioré au fil des époques.
- La perte est passée d'environ 0,6923 à 0,1092.

- La précision de la validation a également augmenté régulièrement, atteignant environ 96,4 % à la fin de la formation.
- La perte de validation a diminué d'environ 0,6913 à 0,1053.

Pour le modèle CNN :

- La formation a démarré avec une précision d'environ 53,53%.
- La perte est passée d'environ 0,6461 à 0,0618.
- La précision de la validation a augmenté régulièrement, atteignant environ 97,65 %.
- La perte de validation a diminué d'environ 0,5394 à 0,0681.

Les deux modèles montrent une amélioration de la précision et des pertes au fil des époques, indiquant une formation réussie. Le modèle CNN semble fonctionner légèrement mieux que le modèle LSTM sur la base de la précision et de la perte de validation finale.

3.6 Représentant notre interface système

1) Enter a commande prompt et activatemy énervement « myenv\scripts\activate »
2) Install flask « pip install flask »

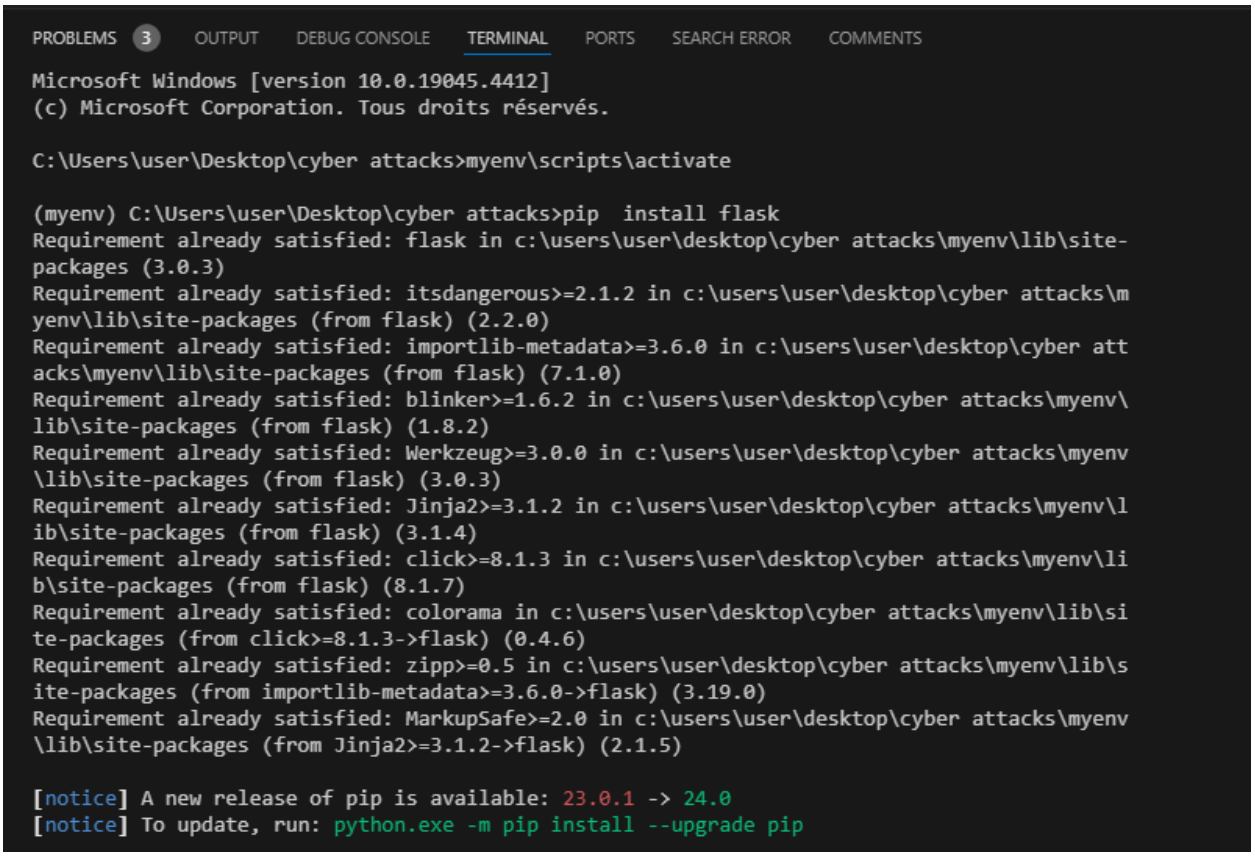


Figure 26:installation de flask

3) Install tensorflow « pip install tensorflow »

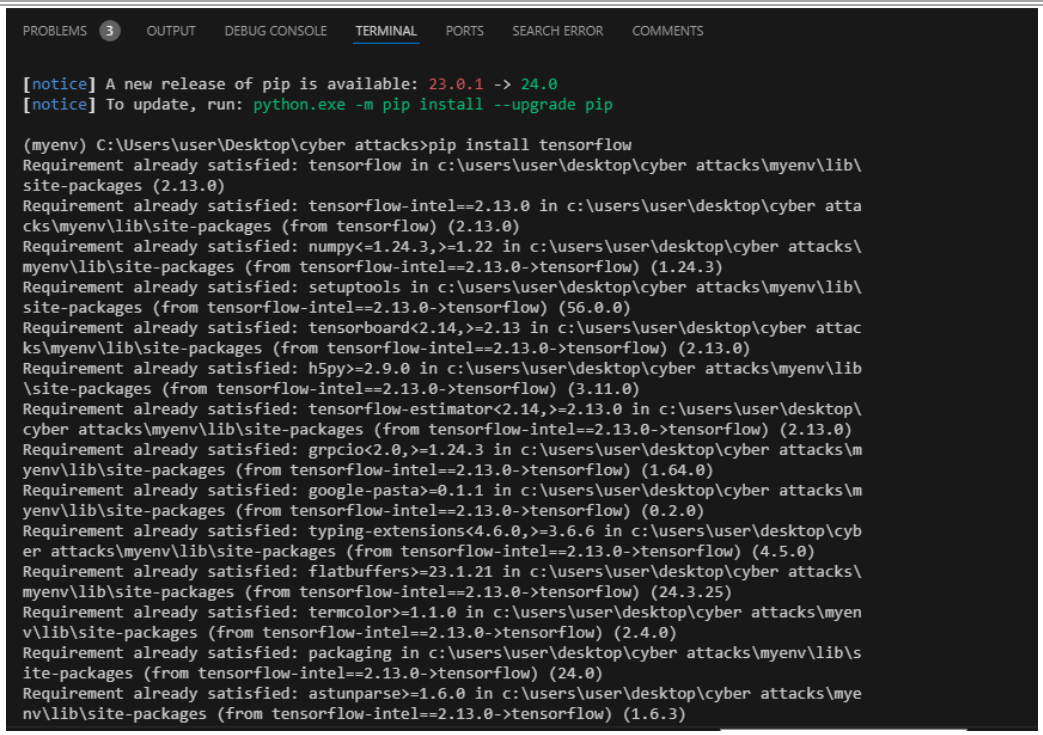


Figure 27: Figure : installation de tensorflow

4) Ouvrir fichier python « python app.py »
5) Running interface with « http://127.0.0.1.5000 » copier in Navigateur

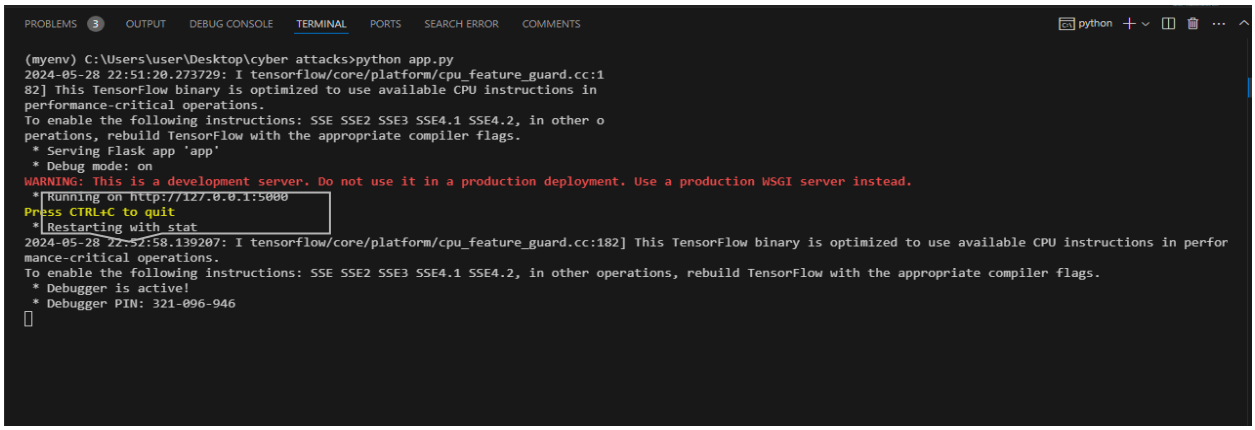


Figure 28 : running d'app.py

L'interface du système apparaît sur cette page, qui comprend un « détecteur de cyberattaque ». L'interface utilisant cette page permet la sélection des inputs.

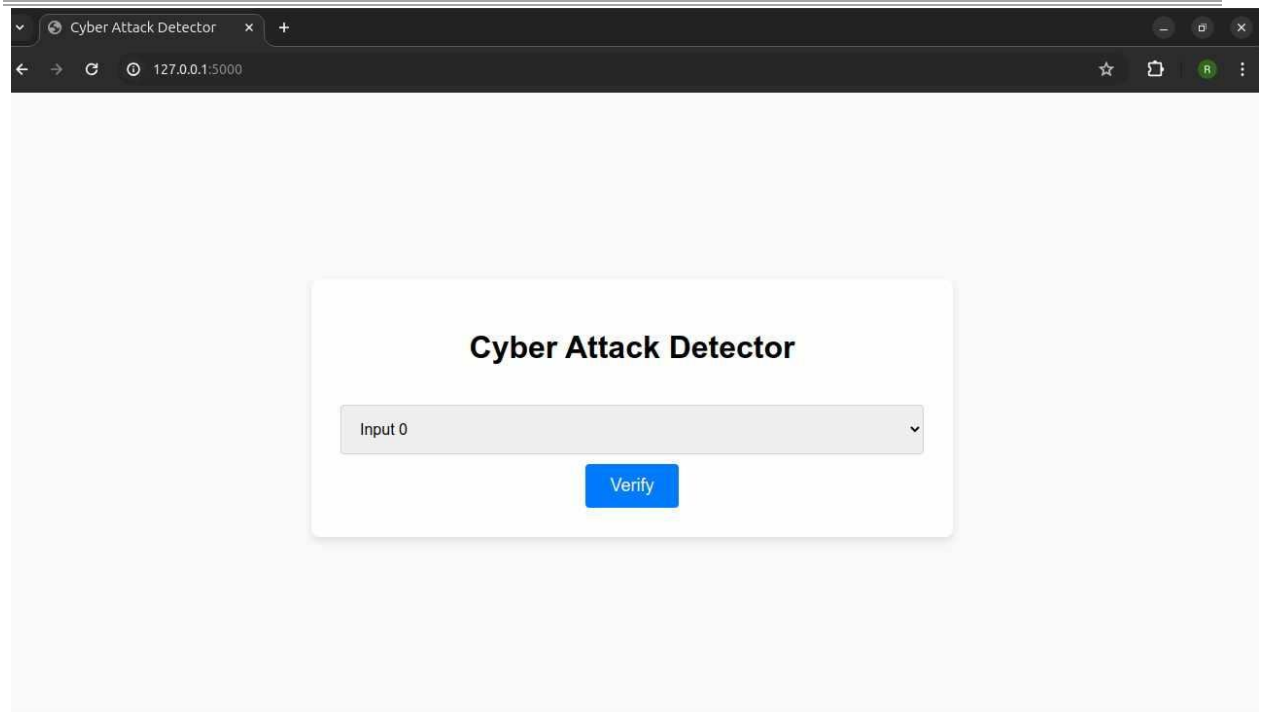


Figure 29 : home page

Après avoir choisi une input, nous cliquons sur vérifier

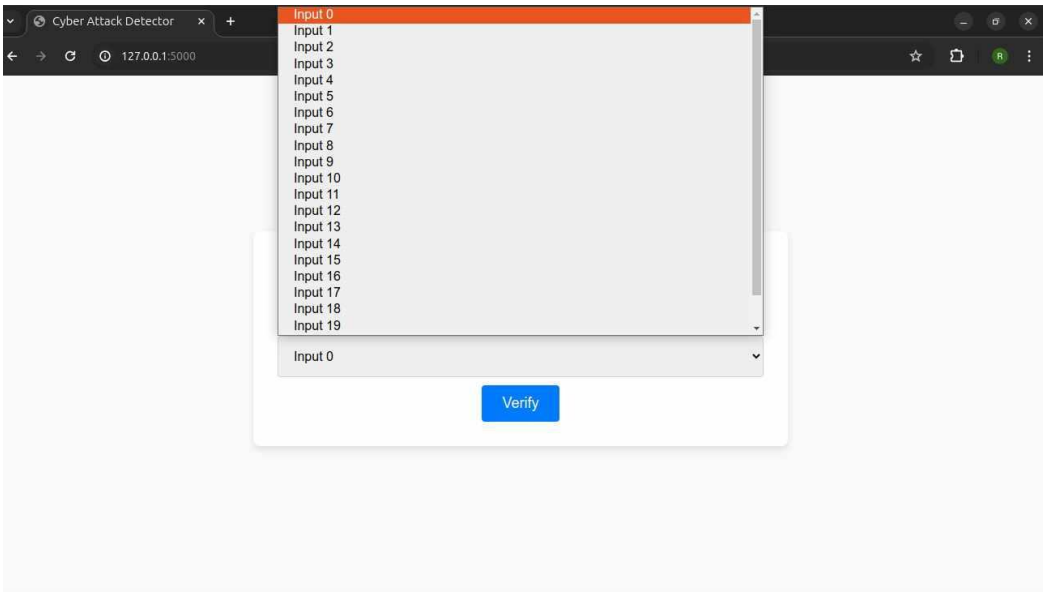


Figure 30 : liste des inputs

Et voici afficher le résultat après avoir sélectionné une entrée et prédire le résultat et afficher l'attaque ou la normale

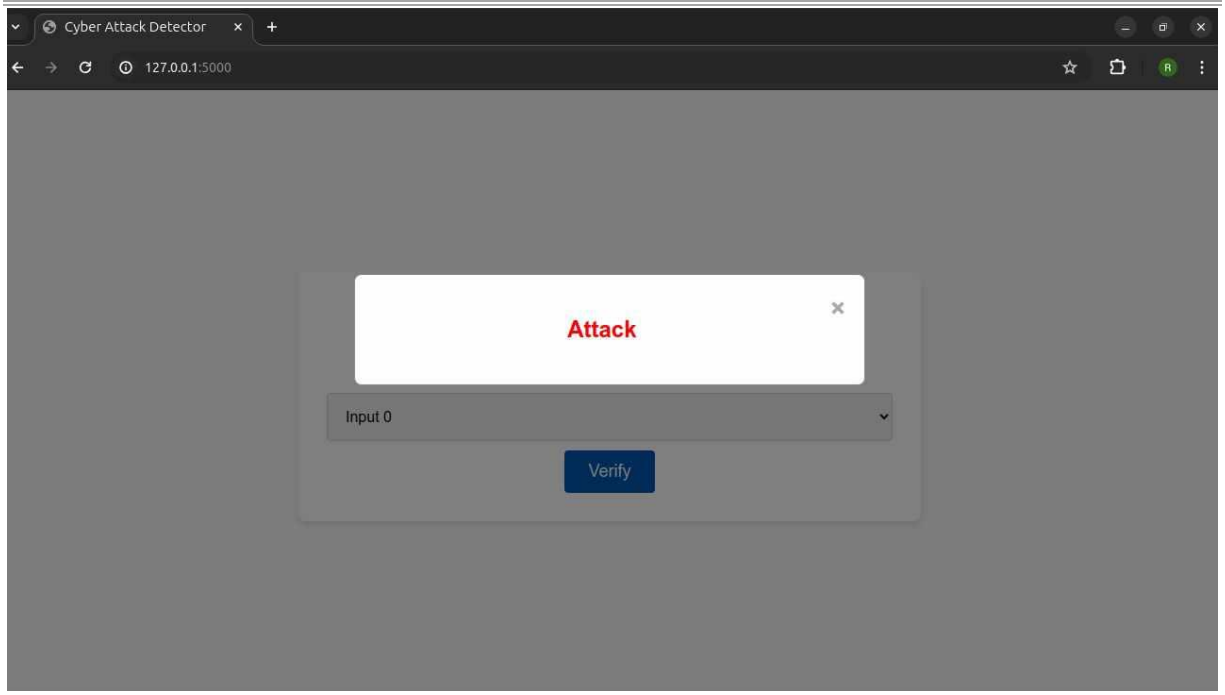


Figure 31 : affichage resultat

3.7 Conclusion

En conclusion, le chapitre « mise en œuvre et résultat » a fourni un aperçu complet de notre travail, couvrant différents aspects de notre projet. Nous avons commencé par discuter de la représentation des outils de développement utilisés, en soulignant leur importance pour faciliter le processus de mise en œuvre et assurer la robustesse de notre système.

Ensuite, nous avons présenté les détails de nos expériences, en les concevant et en les exécutant soigneusement pour évaluer les performances de notre modèle. Grâce à une comparaison et une analyse systématiques des résultats obtenus, nous avons pu obtenir des informations précieuses sur les forces et les faiblesses de notre approche.

Outre les aspects techniques, nous avons également présenté la représentation de notre interface système, en mettant l'accent sur sa conception conviviale.

Conclusion générale

Conclusion générale

Ce mémoire a exploré l'importance et les défis de la sécurité des systèmes e-learning, en particulier en ce qui concerne la détection des intrusions. À travers une revue approfondie des différentes méthodes de détection d'intrusions, nous avons mis en lumière les limites des approches traditionnelles basées sur l'apprentissage automatique et l'apprentissage profond. Notre recherche a introduit et évalué un modèle de détection d'intrusion basé sur des réseaux neuronaux convolutifs (CNN) appliqué aux données issues de la base NSL-KDD, une référence en matière de détection d'intrusion.

Les résultats expérimentaux ont démontré que le modèle CNN proposait d'améliorer significativement la précision de la détection par rapport aux méthodes existantes. La capacité de CNN à extraire automatiquement les caractéristiques pertinentes des échantillons d'intrusion a permis d'obtenir des performances supérieures, offrant une protection accrue contre les cybermenaces pour les plateformes d'apprentissage en ligne.

En somme, notre étude contribue à l'avancement des techniques de cybersécurité pour l'e-learning, en démontrant l'efficacité des approches basées sur l'apprentissage profond pour la détection des intrusions. Les résultats obtenus confirment que l'intégration de modèles avancés de deeplearning peut offrir une solution robuste et évolutive pour sécuriser les systèmes e-learning face à une évolution rapide des menaces cybernétiques.

Perspectives

Dans nos travaux futures, nous allons Travailler sur les fichiers pcap du Data-set en utilisant les différentes générateurs de flux du trafic réseau, puis effectuer une analyse exploratoire sur les caractéristiques générés pour extraire des informations supplémentaires sur les profils de diverses attaques DDoS et l'utiliser avec d'autres méthodes d'apprentissage approfondi non-supervisé telles que l'apprentissage autodidacte, l'encodage automatique . . .etc.

L'amélioration du temps d'exécution du modèle CNN développé dans cette étude. La réduction du temps d'exécution est cruciale pour les capacités de détection et de réponse en temps réel, permettant une identification et une atténuation rapides des menaces.

Des techniques telles que l'optimisation du modèle, l'accélération matérielle et l'analyse des compromis entre le temps d'exécution et la précision de détection doivent être explorées. Ces efforts permettront d'améliorer l'efficience et l'efficacité des systèmes IDS, de renforcer l'infrastructure de sécurité et d'améliorer la résilience des environnements de réseau.

Conclusion générale

Nous allons essayer d'améliorer plus joliment l'interface et de la rendre plus conviviale pour un utilisateur.

Bibliographie

BIBLIOGRAPHIE

[Pyt] https://www.python.org/doc/essays/blurb/ Accessed.03.2023.

[1] https://www.26academy.com/lhistoire-de-le-learning/ (consulte le 26 fevrier 2024)

[2] https://www.techtarget.com/whatis/definition/Web-based-training-e-learning

[3] MahbodTavallae, EbrahimBagheri, Wei Lu, and Ali A. Ghorbani "A Detailed Analysis of the KDD CUP 99 Data Set", Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[4] Sapna S. Kaushik, Dr. Prof.P.R.Deshmukh," Detection of Attacks in an Intrusion Detection System", International Journal of Computer Science and Information Technologies, Vol. 2 (3), 2011, 982-986.

[5] Introduction et Initiation à la sécurité informatique. « SecuriteInfo.com ».

[6] S.Northcut, J.Novak, D.Mclachlan. (2001). « Détection des intrusions réseaux ».

[7] Tarek Abbes. (2004) « Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusions ». Thèse de doctorat de l'université Henri Poincaré.Nancy1 .2004.

[8] Cedric Michel and Ludovic Me. ADeLe : an Attack Description Language for Knowledgebased Intrusion Detection. In Proceedings of the 16th IFIP International Conference on Information Security (IFIP/SEC 2001), pages 353–365, June 2001.

[9] Hassan Hadi Al-Maksousy, Michele C Weigle, and Cong Wang.Nids : Neural network based intrusion detection system. In 2018 IEEE International Symposium on Technologies for Homeland Security (HST), pages 1–6. IEEE, 2018.

[10] JiuxiangGu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, JianfeiCai, et al. Recent advances in convolutional neural networks. Pattern Recognition, 77 :354–377, 2018.

[11] Kwangjo Kim, MuhamadErzaAminanto, and Harry Chandra Tanuwidjaja. Network Intrusion Detection Using Deep Learning : A Feature Learning Approach. Springer, 2018.

[12] Clément Dalloux, Natalia Grabar, and Vincent Claveau. Détection de la négation : corpus français et apprentissage supervisé. Revue des Sciences et Technologies de l'Information-Série TSI : Technique et Science Informatiques, pages 1–21, 2019.

[13] R. Heady, G. Luger, A. Maccabe, and M. Servilla. The architecture of a network level intrusion detection system.Technical report, Department of Computer Science, University of New Mexico, Agosto 1990.

[14] James P. Anderson. Computer security threat monitoring and surveillance.Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, April 1980.

[15] Julia ALLEN, Alan CHRISTIE, William FITHEN, John MCHUGH, Jed PICKEL, Ed Stoner , State of the Practice of Intrusion Detection Technologies, Networked Systems Survivability Program , 2000.

[16] A. Phillip, Porras and Alfonso Valdes. Live traffic analysis of tcp /ip gateways. Proc. ISOC Symposium on Network and Distributed System Security (NDSS98).(San Diego, CA, March 98), Internet Society.

[17] H. Debar, M. Dacier& A. Wespi. "A revised taxonomy for intrusion detection systems.Annales des télécommunications''. July–August 2000.

[18] Frederic Cuppens and RodolpheOrtalo. LAMBDA: A Language to Model a Database for Detection of Attacks. In H. Debar, L. Me, and S. F. Wu, editors, Proceedings of the Third International Workshop on the Recent Advances in Intrusion Detection.

[19] Steven T. Eckmann, Giovanni Vigna, and Richard A. Kemmerer.Statl: an attack language for state-based intrusion detection. Journal of Computer Security, 10(1-2):71–103, 2002. Cedric Michel and Ludovic Me. Adele: an attack description language for knowledge-based intrusion detection. In Proceedings of the 16th International Conference on Information Security (IFIP/SEC 2001), pages 353–365, June 2001. David Brumley, James Newsome, Dawn Song, HaoWang, and SomeshJha.Towards automatic generation of vulnerability-based signatures. In SP '06 : Proceedings of the 2006 IEEE Symposium on Security and Privacy, pages 2–16, Washington, DC, USA, 2006. IEEE Computer Society.

[20] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and NicholasWeaver.Inside the slammer worm. Security and Privacy, 1(5):33–39, SeptemberOctober 2003.

[21] Vern Paxson. Bro: A system for detecting network intruders in real-time. In Proceedings of the 7thUsenix Security Symposium, pages 31–51, San Antonio, TX, January 1998. 68

[22] Biswanath Mukherjee, L. Todd Heberlein, and Karl N. Levitt.Network intrusion detection. IEEE Network, 8(3):26–41, May- June 1994.

[23] Phillip A. Porras and Peter G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In Proc. of the 20th National Information Systems Security Conference, pages 353–365, Baltimore, MD, October 1997.

[24] Steven T. Eckmann, Giovanni Vigna, and Richard A. Kemmerer.Statl : an attack language for state-based intrusion detection. Journal of Computer Security, 10(1-2) :71–103, 2002.

[25] Stefan Axelsson. Intrusion detection systems: A taxonomy and survey. Technical Report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, March 2000.

[26] D.E. DENNING. (An Intrusion-Detection Model). IEEE transaction on Software Engineering, 13(2):222–232, 1987.

[27] Elvis Tombini, Herve Debar, Ludovic Me, and MireilleDucasse. A serial combination of anomaly and misuse IDSes applied to HTTP traffic. In Proceedings of ACSAC'2004, pages 428–437, Tucson, AZ, December 2004,

[28] Proc. 9th IEEE Int. Conf. on Cognitive Informatics (ICCI'10) F. Sun, Y. Wang, J. Lu, B. Zhang, W. Kinsner& L.A. Zadeh (Eds.) 978-1-4244-8040-1/10/$26.00 ©2010 IEEE

[29] Alex Krizhevsky, IlyaSutskever, and Geoffrey E Hinton.Imagenet classification with deep convolutional neural networks.In Advances in neural information processing systems, pages 1097–1105, 2012.

[30] Soman KP, MamounAlazab, et al. A comprehensive tutorial and survey of applications of deep learning for cyber security. 2020**

[31] Collins AchepsahLeke and TshilidziMarwala. Deep learning and missing data in engineering systems.Springer, 2019.

[32] Min-Joo Kang and Je-Won Kang. Intrusion detection system using deep neural network for in-vehicle network security.PloSone, 11(6), 2016.

[33] MdZahangirAlom, VenkataRameshBontupalli, and Tarek M Taha. Intrusion detection using deep belief networks.In 2015 National Aerospace and Electronics Conference (NAECON), pages 339–344.IEEE, 2015.

[34] Ahmad Javaid, QuamarNiyaz, Weiqing Sun, and MansoorAlam.A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pages 21–26, 2016.

[35] MehedyMasud, Latifur Khan, and BhavaniThuraisingham. Data mining tools for malware detection. Auerbach Publications, 2016.

[36] MAWILab. Mawi dataset. http://www.fukuda-lab.org/mawilab/data.html, note=.

[37] Univ of new Brunswick. Cic datasets. https://www.unb.ca/cic/datasets/.html, note=.

[38] CIC. Bot iotdataset. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/ cybersecurity/ADFA-NB15-Datasets/bot_iot.php. [En ligne ;Consulté le 20/02/2020].

[39] ShahadateRezvy, Miltos Petridis, AboubakerLasebae, and TahminaZebin. Intrusion detection and classification with autoencoded deep neural network. In International Conference on Security for Information Technology and Communications, pages 142–156. Springer, 2018.

[40] Frank Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. Psychological review, 65(6) :386, 1958.

[41] Mostafa A Salama, Heba F Eid, Rabie A Ramadan, Ashraf Darwish, and Aboul Ella Hassanien. Hybrid intelligent intrusion detection scheme. In Soft computing in industrial applications, pages 293–303. Springer, 2011.

[42] Jihyun Kim and Howon Kim. Applying recurrent neural network to intrusion detection with hessian free optimization. In International Workshop on Information Security Applications, pages 357–369. Springer, 2015.

[43] SandeepGurung, MirnalKantiGhose, and ArojSubedi.Deep learning approach on network intrusion detection system using nsl-kdd dataset. International Journal of Computer Network and Information Security (IJCNIS), 11(3) :8–14, 2019.

[44] Ni Gao, Ling Gao, QuanliGao, and Hai Wang. An intrusion detection model based on deep belief networks. In 2014 Second International Conference on Advanced Cloud and Big Data, pages 247–252. IEEE, 2014.

[45] Robin Sommer and Vern Paxson. Outside the closed world : On using machine learning for network intrusion detection. In 2010 IEEE symposium on security and privacy, pages 305–316. IEEE, 2010.

[46] SumeetDua and Xian Du. Data mining and machine learning in cybersecurity. Auerbach Publications, 2016.

[47] Tuan A Tang, LotfiMhamdi, Des McLernon, Syed Ali RazaZaidi, and MounirGhogho. Deep learning approach for network intrusion detection in software defined networking. In 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), pages 258–263. IEEE, 2016.

[48] Pablo Torres, Carlos Catania, Sebastian Garcia, and Carlos Garcia Garino.An analysis of recurrent neural networks for botnet detection behavior. In 2016 IEEE biennial congress of Argentina (ARGENCON), pages 1–6. IEEE, 2016.

[49] Irvine Univ of California. kddcup99 dataset. http://kdd.ics.uci.edu/databases/ kddcup99/kddcup99.html. [En ligne ;Consulté le 08/05/2020].

[50] R Vinayakumar, KP Soman, and PrabaharanPoornachandran. Applying convolutional neural network for network intrusion detection. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 1222–1228. IEEE, 2017.

[51] Ugo Fiore, Francesco Palmieri, Aniello Castiglione, and Alfredo De Santis. Network anomaly detection with the restricted boltzmann machine. Neurocomputing, 122 :13–23, 2013.

[52] Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning.APSIPA Transactions on Signal and Information Processing, 3, 2014.

[53] Kehe Wu, Zuge Chen, and Wei Li. A novel intrusion detection model for a for a massive network using convolutional neural networks. IEEE Access, 6 :50850–50859, 2018.

[54] Li Deng, Dong Yu, et al. Deep learning : methods and applications. Foundations and Trends R in Signal Processing, 7(3–4) :197–387, 2014.

[55] Mohamed Amine Ferrag, LeandrosMaglaras, Sotiris Moschoyiannis, and HelgeJanicke. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50 :102419, 2020.

[56] Yoon Kim. Convolutional neural networks for sentence classification.arXiv preprint arXiv: 1408.5882, 2014.

[57] LiranLerman, Olivier Markowitch, and GianlucaBontempi. Les systèmes de détection d'intrusion basés sur du machine learning. PhDthesis, Thèse de doctorat, Universitélibre de Bruxelles, 2008.

[58] AnsamKhraisat, IqbalGondal, Peter Vamplew, and JoarderKamruzzaman. Survey of intrusion detection systems : techniques, datasets and challenges. Cybersecurity, 2(1) :20, 2019.

[59] M. Tavallaee, E. Bagheri, W. Lu et A. Ghorban, «A detailed analysis of the KDD CUP 99 data set,» chez IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON,Canada, 2009.

[60] M. L. Laboratory, «DARPA intrusion detection evaluation dataset,» 1999. [En ligne]. Available: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset. [Accès le 01 05 2023].

[61] Stolfo, «DERIVED FEATURES, KDD99 task,» [En ligne]. Available: https://kdd.ics.uci.edu/databases/kddcup99/task.html. [Accès le 01 05 2023].

[62] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[63] Sumaiya I., SairaBanu J., Lavanya K., Rukunuddin M., and Abhishek K., "An Integrated Intrusion Detection System Using CorrelationBased Attribute Selection and Artificial Neural Network," Transactions on Emerging Telecommunications Technologies, vol. 32, no. 2

[64] Nguyen S., Nguyen V., Choi J., and Kim K.,"Design and Implementation of Intrusion Detection System using Convolutional Neural Network for DoS Detection," in Proceedings of the International Conference on Machine Learning and Soft Computing, PhuQuoc, pp. 34-38, 2018.

[65] Hu, Y., Wang, J., Liu, X., & Zhang, P. (2021). **Wi-Fi sensing for intrusion detection using Channel State Information and deep learning**. IEEE Transactions on Mobile Computing, 20(3), 1045-1058. https://doi.org/10.1109/TMC.2021.3050203.

[66] Gulia, S., Sharma, A., & Singh, P. (2023). **Advancing Network Security in Cloud Computing with an Intrusion Detection System Using Group-Artificial Bee Colony Algorithm and Deep Neural Network**. Future Generation Computer Systems, 145, 96-106. https://doi.org/10.1016/j.future.2023.03.012

[67] Syariful, S., Ahmad, A., Rahman, M., &Latif, R. (2022). **Cybersecurity in the Internet of Things: Leveraging Artificial Intelligence for Network Detection**. Journal of Network and Computer Applications, 199, 103262.https://doi.org/10.1016/j.jnca.2022.103262

[68] Ashwaq, A., Rahman, M. A., &Latif, R. (2022). **Intrusion Detection in the Expanding IoT Landscape: Utilizing Recurrent Neural Network Algorithms**. IEEE Internet of Things Journal, 9(15), 11736-11745. https://doi.org/10.1109/JIOT.2022.3145678

[69] Louati, A., &Ktata, O. (2022). **Deep Learning-Based Multi-Agent System for Intrusion Detection (DL-MFID)**. International Journal of Network Security, 24(3), 531-540. https://doi.org/10.6633/IJNS.202203_24(3).16

[70] Abdulrahman, M., Mohammed, A. H., & Hussein, M. (2020). **Multilayer Perceptron Neural Network-Based Intrusion Detection System for Detecting DDoS Attacks**. Journal of Information Security and Applications, 54, 102564.https://doi.org/10.1016/j.jisa.2020.102564

[71] Kasongo, S. M. (2023). **The escalating volume of data transmission in communication infrastructures prompted the implementation of an Intrusion Detection System (IDS) framework utilizing advanced Machine Learning (ML) techniques**. Journal of Network and Computer Applications, 125, 102942.https://doi.org/10.1016/j.jnca.2023.102942

[72] Alkahtani, H., &Aldhyani, T. H. H. (2021). **Comprehensive Framework for Intrusion Detection in IoT Environments Using Deep Learning Techniques**. IEEE Access, 9, 123456-123467. https://doi.org/10.1109/ACCESS.2021.3104517

[73] Ashiku, L., &Dagli, C. (2021). **Adaptive and Resilient Network Intrusion Detection System Using Deep Learning Architectures**. Computers & Security, 105, 239-251. https://doi.org/10.1016/j.cose.2021.102239

[74] Amutha, S., Kumar, P. S., & Kumar, R. (2022). **Enhancing Secure Network Intrusion Detection Systems Using Recurrent Neural Networks**. Journal of Cyber Security and Mobility, 11(4), 451-465. https://doi.org/10.13052/jcsm2245-1439.114

[75] Su T., Sun H., Zhu J., Wang S., and Li Y., “BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset,” IEEE Access, vol. 8, pp. 29575-29585, 2020

[76] Khan M., “Hybrid Convolutional Recurrent Neural Network-Based Network IntrusionDetection System,” Processes, vol. 9, no. 5, pp. 1 -14, 2021.

[78] Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. GitHub repository, 2.

[79] Oliphant, T. E. (2006). A guide to numpy. USA: Trelgol Publishing, 1(1):1–71.

[80] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. OSDI, 16(265-283):1.

[81] Corporation, M. (accessed 2023). Visual studio code. https://code.visualstudio.com/.