MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

**Faculté des Sciences Exactes et d'Informatique**

**Département de Mathématiques et informatique**

**Filière : Informatique**

RAPPORT DE MINI-PROJET

Option : **Ingénierie des Systèmes d'Information**

THEME :

# Système de Détection d'Intrusion Réseau Basé sur l'Apprentissage Automatique

Etudiant: « **Zahaf Mohamed El-Bachir** »

Encadrante : « Bentaouza Chahinez Mériem »

Année Universitaire 2022-2023

# Remerciements

*Tout d'abord, Je dois remercier Dieu de m'avoir donné une seconde chance de vivre après avoir vu la mort de mes propres yeux suite à un accident et le courage de terminer ce travail.*

### *A mon Encadrante*

*Je tiens à remercier mon encadrante Madame. Bentaouza Chahinez Mérièm pour m'avoir accepté pour mener à bien ce projet de fin d'études et pour son temps précieux, ses conseils et sa disponibilité tout au long de ce travail.*

### *A mes membres du Jury*

*Je remercie également les membres du jury d'avoir accepté d'honorer mon travail par leur jugement.*

### *A mes Professeurs*

*J'adresse mes remerciements à tous les professeurs pour leurs conseils et leurs critiques qui ont guidé ma réflexion au cours de mes recherches.*

### *A mon Famille*

*Je tiens à exprimer ma profonde gratitude à mes parents qui m'ont soutenu tout au long de mon projet. Ainsi que toutes les familles et amis pour leur soutien indéfectible.*

# *Dédicaces*

## *A ma Famille*

*Je dédie ce travail à ma famille, mon père, ma mère et mes frères*

*J'espère être à la hauteur de leurs attentes et ne jamais les décevoir.*

## *A mon Encadrante*

*A mon encadrante Bentaouza Chahinez Mériem vous m'avez beaucoup aidé dans cette recherche cordialement je vous souhaite longue vie.*

## *A mes Amis*

*A tous mes chers amis qui m'aident et m'encouragent.*

*A mes amis Féderico Klez, et Djamel Mekki, pour leur soutien.*

*A tous les camarades de département, compagnons de ces années d'études je leur souhaite bonne chance*

## *Zahaf mohamed el-bachir*

## Résumé

Les attaques deviennent de plus en plus courantes, et les données stockées sur Internet sont maintenant plus ciblées que jamais. Assurer la protection des informations critiques est d'une importance capitale. Dans le cadre de cet effort, mon projet vise à développer une approche basée sur l'apprentissage automatique pour la détection des intrusions sur le réseau. Plus précisément, je propose de combiner les algorithmes NIDS (Network Intrusion Detection System) avec les CNN (Convolutional Neural Network) afin d'identifier et de prévenir les attaques quotidiennes auxquelles nous sommes confrontés sur les réseaux. Cette approche innovante représente une solution prometteuse pour détecter et atténuer efficacement les menaces de cybersécurité. En utilisant la base de données kdd99 comme référence pour mon étude, je peux mettre en œuvre l'apprentissage en profondeur en utilisant l'algorithme CNN, car il s'est avéré être le plus précis parmi les trois méthodes (DNN, RNN, CNN).

## Mots-clés

Sécurité des Réseaux, IDS, Apprentissage profond, apprentissage automatique, Réseau de neurones et DNN, CNN, RNN, KDD99.

## Summary

Attacks are becoming increasingly common, and data stored on the Internet is now more targeted than ever. Ensuring the protection of critical information is of utmost importance. In support of this effort, my project aims to develop a machine learning-based approach for network intrusion detection. Specifically, I propose to combine NIDS (Network Intrusion Detection System) algorithms with CNN (Convolutional Neural Network) to identify and prevent the daily network attacks we face. This innovative approach represents a promising solution to detect and mitigate cybersecurity threats effectively. By utilizing kdd99 as the foundational database for my study, I can implement deep learning using the CNN algorithm, as it has proven to be the most accurate among the three methods (DNN, RNN, CNN).

## Key-words

Network Security, IDS, Deep Learning, Machine Learning, Neural Network and DNN, CNN, RNN, KDD99.

# ملخص

أصبحت الهجمات شائعة بشكل متزايد ، وأصبحت البيانات المخزنة على الإنترنت أكثر استهدافًا من أي وقت مضى. ضمان حماية المعلومات الهامة له أهمية قصوى. دعماً لهذا الجهد ، يهدف مشروعي إلى تطوير نهج نظام) NIDS قائم على التعلم الآلي لاكتشاف اختراق الشبكة. على وجه التحديد ، أقترح دمج خوارزميات (الشبكة العصبية التلافيفية) لتحديد ومنع هجمات الشبكة اليومية التي CNN اكتشاف اختراق الشبكة) مع نواجهها. يمثل هذا النهج المبتكر حلاً واعدًا لاكتشاف تهديدات الأمن السيبراني والتخفيف من حدتها بشكل كقاعدة بيانات أساسية لدراستي ، يمكنني تنفيذ التعلم العميق باستخدام kdd99 فعال. من خلال استخدام .(DNN ، RNN ، CNN) حيث ثبت أنها الأكثر دقة من بين الطرق الثلاث ، CNN خوارزمية



## الكلمات الدالة

أمن الشبكات، نظام كشف التسلل، التعلم العميق، التعلم الآلي، الشبكة العصبية، الشبكة العصبية التلافيفية، الشبكة العصبية المتكررة، اكتشاف المعرفة والبيانات**99**. الشبكة العصبية العميقة، الشبكة العصبية المتكررة أو الشبكة العصبية التلافيفية


**Remarque :** Cette étude a été réalisée à l'aide d'un système très médiocre, étant donné le manque de capacités, faites donc une véritable étude, cette étude et ses résultats ne sont basés que sur un système médiocre, le résultat peut varier ou changer en fonction des capacités du système utilisé pour l'étude.

# Liste des figures

# Liste des tableaux

# Liste des abréviations

| Abréviation | Expression Complète |
|---|---|
| NIDS | Network Intrusion Détection System |
| IDS | Intrusion Détection System |
| CNN | Convolutional neural network |
| RNN | Recurrent neural network |
| KDD99 | Knowledge Discovery and Data |

# Table des matières

# Introduction Générale

Internet est devenu l'un des outils les plus importants et la meilleure source d'information dans notre monde actuel. Il joue un rôle clé dans les activités éducatives, créatives et commerciales. Cependant, avec cette dépendance croissante à Internet, la protection des données contre les intrusions est devenue primordiale.

La sécurité sur Internet est l'une des principales préoccupations de notre époque, car il fait face à diverses menaces. C'est pourquoi des systèmes de détection d'intrusion (IDS) ont été développés pour protéger ces données et les utilisateurs. Les administrateurs réseau personnalisent ces systèmes pour empêcher les attaques malveillantes, les rendant ainsi essentiels dans la gestion de la sécurité.

Il existe différentes méthodes pour détecter les anomalies et les comportements malveillants, notamment l'utilisation d'apprentissage automatique avec des réseaux de neurones artificiels, également connus sous le nom d'apprentissage profond. L'apprentissage automatique est une approche qui utilise des étapes séquentielles de traitement de l'information pour identifier des motifs et apprendre des caractéristiques ou des représentations.

Dans l'ensemble, l'apprentissage automatique joue un rôle essentiel dans la détection des intrusions. Il est largement utilisé dans divers domaines tels que la reconnaissance de la parole, la modélisation de graphes, la surveillance des motifs, la vision par ordinateur, le traitement du langage naturel et le traitement du signal. Les avancées dans les algorithmes d'apprentissage offrent un potentiel pour améliorer les capacités des systèmes de détection d'intrusion, augmenter les taux de détection et réduire les fausses alarmes. Cependant, il est important de noter que la mise en œuvre de l'apprentissage automatique dans les opérations de détection d'intrusion présente certaines limites qu'il convient de prendre en compte.

Dans ces 3 chapitres, je parlerai de Système de Détection d'Intrusion puis d'Apprentissage Automatiques, et de la détection d'intrusion basse sur l'Apprentissage Automatique et après cela je pourrai appuyer sur la partie implémentation.

*'' Je sais que l'adoption correcte de l'apprentissage automatique dans les opérations IDS est difficile en raison des divers héritages des approches précédentes. Et c'est peut-être en partie parce que j'utilise des modes d'apprentissage automatiques supervisés par des réseaux de neurones traditionnels (CNN) ou que j'entends parler de la complexité des modes d'apprentissage automatiques.''*

# Chapitre 1 : Un Système de Détection d'Intrusion

# Chapitre 1

# Un Système de Détection d'Intrusion

## 1.1 Introduction

Surtout avec le développement rapide de la technologie des réseaux et des réseaux sans fil, la sécurité de ces réseaux et des terminaux qui y sont connectés contre diverses menaces est devenue un problème important.

Toutes les informations affectées par les technologies Internet et les informations stockées dans des bases de données et transmises sur le réseau doivent être protégées. Les intrusions sont de véritables menaces qui peuvent être des activités non autorisées ou des utilisations malveillantes de ressources d'information qui enfreignent les politiques de sécurité.

Les systèmes et les techniques traditionnels de prévention des intrusions comme les pare-feu, le cryptage et le contrôle d'accès sont dans la plupart du temps inefficaces face à l'évolution des nouvelles menaces sophistiquées. Comment surmonter les défis du cybersécurité, identifier les intrusions et protéger nos données est un problème clé qui ne doit jamais être contourné. Un nouveau concept de détection d'intrusion a été proposé par James Anderson en 1980, dans le but d'identifier toute activité non autorisée dans un réseau [1].

Ce chapitre présente les systèmes de détection d'intrusion (IDS) et définit un modèle de base pour ces systèmes. Il a également détaillé la taxonomie (IDS) et présenté une analyse des différentes techniques de détection possibles et des différentes mesures d'évaluation du système (IDS).

## 1.2 Définition d'un système de détection d'intrusion

Un système de détection d'intrusions (« Intrusion Detection Systems » ou IDS) est un appareil ou une application qui alerte l'administrateur en cas de faille de sécurité, de violation de règles ou d'autres problèmes susceptibles de compromettre son réseau informatique.

Les systèmes de détection d'intrusions surveillent et analysent les activités d'un réseau, analysent ses configurations et ses vulnérabilités, et vérifient l'intégrité des fichiers. Ils peuvent reconnaître des schémas d'attaque classiques. Pour ce faire, ils analysent les comportements anormaux et suivent les violations de règles par les utilisateurs. Certains systèmes industriels de détection d'intrusions peuvent également réagir à des menaces détectées.

Un système IDS est en général à double détente. La première étape, que l'on peut qualifier de passive, intervient sur la machine. Il s'agit de l'inspection des fichiers de configuration du réseau, notamment pour détecter les paramètres déconseillés et les violations de règles. La seconde étape, que l'on peut qualifier d'active, intervient sur le réseau. Ici, les mécanismes réutilisent des méthodes d'attaque identifiées et enregistrent les réactions [2].

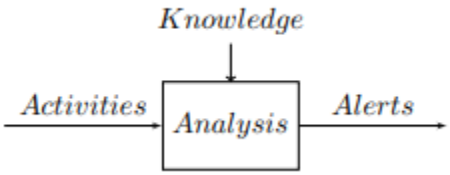


**Figure 1.1 – Intrusion Detection System function** [3]

## 1.3 Le modèle de base d'un système de détection d'intrusion

Un système de détection d'intrusion peut être composé de plusieurs outils, ou chaque outil peut avoir sa propre tâche, et son objectif global est de détecter une intrusion la première fois et d'alerter un opérateur ou le personnel informatique d'une attaque potentielle. Un modèle général pour la structure d'un système de détection d'intrusion a été proposé par le groupe de travail sur la détection d'intrusion (IDWG) de l'IETF, qui couvre et normalise la structure des systèmes de détection d'intrusion. La figure suivante montre en détail les différents composants de ce système.

Comme le montre la figure 4.1, un IDS n'inclut pas nécessairement tous ses composants complètement indépendants. Certains IDS combinent ces composants en un seul module, tandis que d'autres ont plusieurs instances de ces modules [3].

- **Administrateur :** il s'agit de la personne responsable de la création des politiques de sécurité organisationnelles qui déploient et configurent les différents composants d'IDS. Support des déclarations prédéfinies d'activités pouvant être réalisées sur un réseau ou sur un hôte particulier pour répondre aux besoins d'un système d'information.
- **Source de donnée :** Il existe différents types de données provenant de plusieurs sources (réseaux, systèmes, applications, alertes). Les systèmes IDS ne se limitent pas aux sources de données qu'ils utilisent, ils utilisent donc des capteurs appropriés pour analyser les informations provenant de ces sources afin de détecter les activités frauduleuses ou indésirables.
- **Capteurs et analyseurs :** Composants critiques du système. Tout d'abord, le capteur accède aux données brutes, collecte toutes les informations sur les activités en cours et les envoie sous forme d'événements (séries d'activités) à l'analyseur. Ce dernier analyse et rend compte de ces événements Activités ou événements non autorisés ou indésirables susceptibles d'intéresser les administrateurs de sécurité. Dans la plupart des IDS existants, le capteur et l'analyseur font partie d'un seul composant.
- **Gestionnaire :** Également un élément clé, à partir duquel les opérateurs peuvent gérer divers composants du système. Les fonctions du gestionnaire incluent généralement (mais sans s'y limiter) la configuration du capteur, la configuration de l'analyseur, la gestion des notifications d'événements, l'intégration des données et la gestion des rapports.
- **Réponses :** il s'agit des mesures prises en réponse à l'événement. Cela peut être fait automatiquement par une entité dans l'architecture IDS pour une initiation humaine. L'envoi d'une notification à un opérateur est une réponse très courante. D'autres réponses incluent (mais sans s'y limiter) les journaux d'activités, les journaux de données brutes (de la source). Données caractérisant les événements, les arrêts du réseau ou de l'utilisateur, ou les modifications des cessions d'applications, des contrôles d'accès au réseau ou au système).

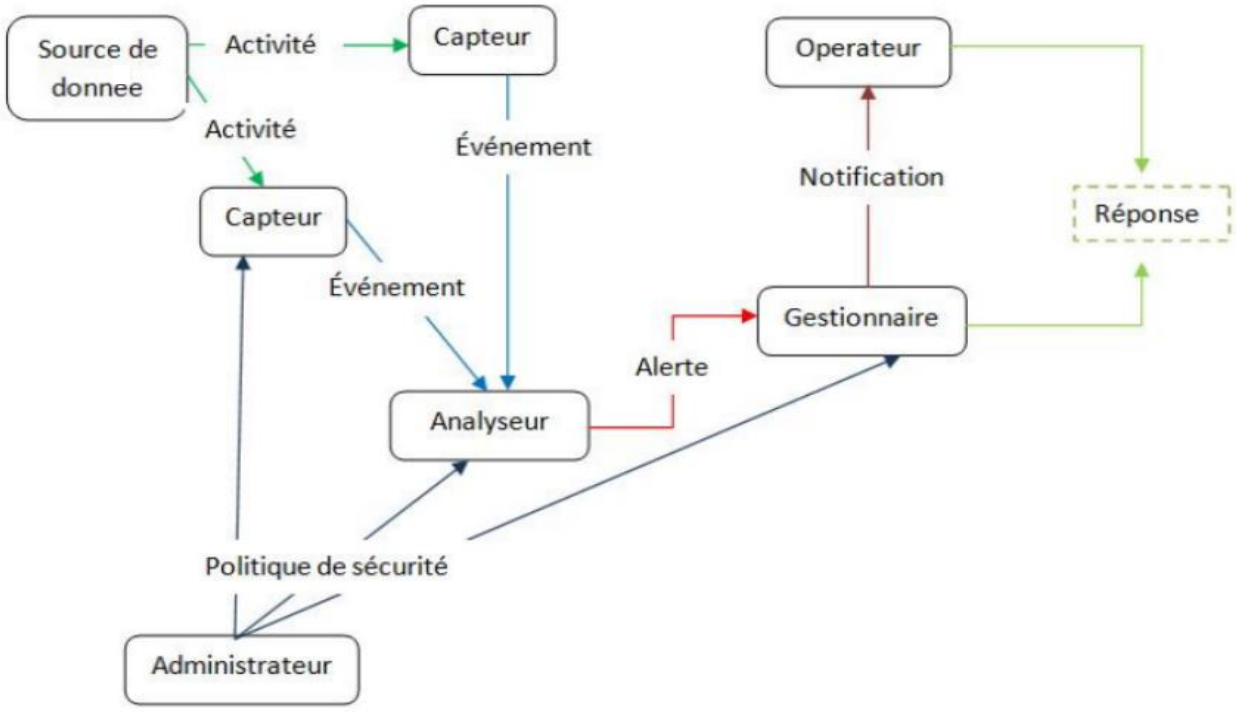


**Figure 1.2 – Un modèle fonctionnel du Système de détection d'intrusion proposé par l'IDWG** [3].

## 1.4 Taxonomie des IDSs

Il existe de nombreux types de technologies de systèmes de détection d'intrusion, caractérisés par différentes approches des architectures systèmes, des environnements de déploiement, des techniques de surveillance et des stratégies de détection.

Plusieurs taxonomies d'IDS ont été proposées dans la littérature. Une nouvelle perspective sur ces taxonomies d'IDS a été présentée par Liao et al [4]. Adoptée selon les normes. Il est représenté par quatre taxonomies principales, comme le montre la figure 1.2.

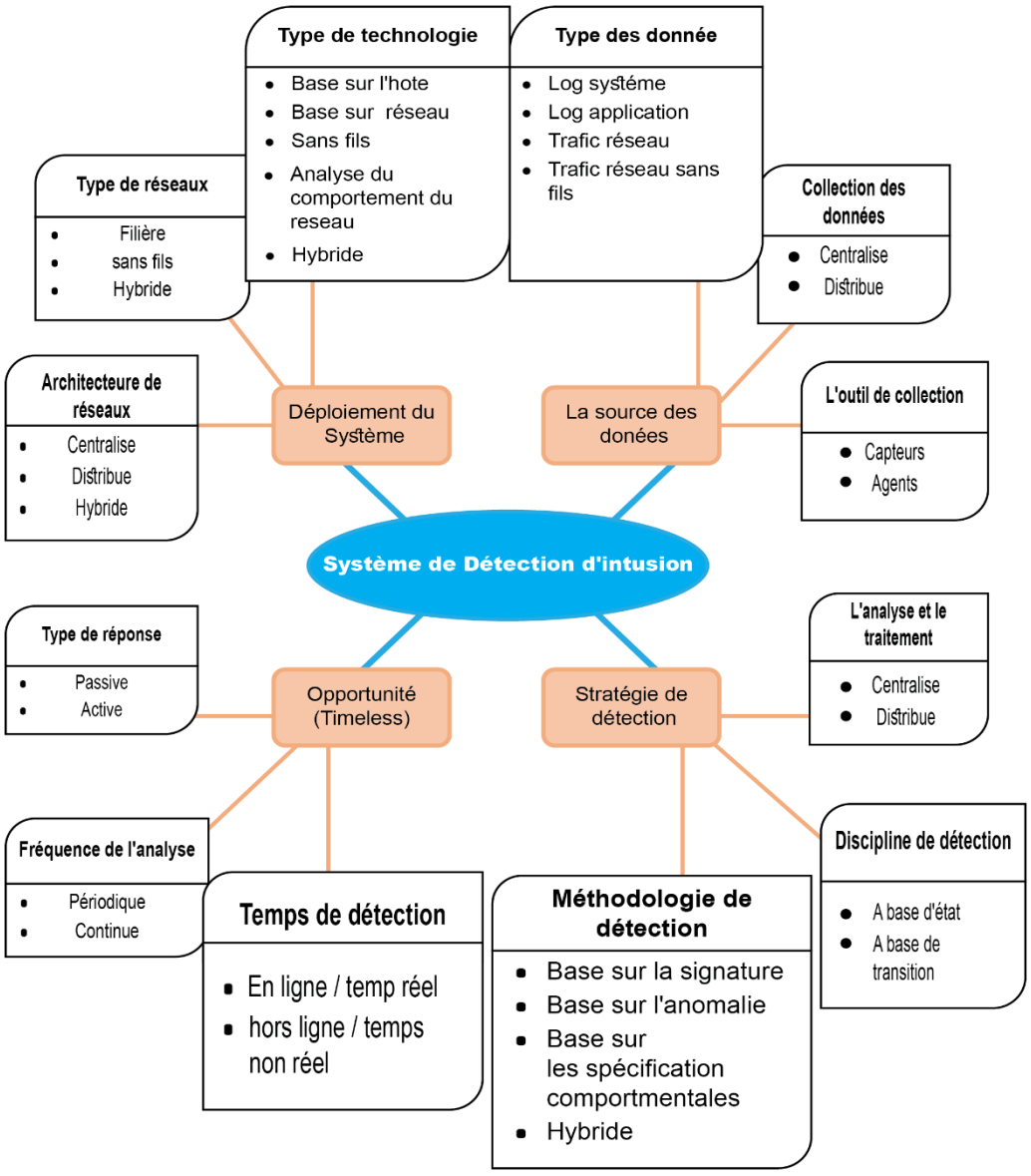


**Figure 1.3 – Taxonomie des systèmes de détection d'intrusion proposée par Liao et al** [4].

### 1.4.1 Les sources des données à analyser

**A. Type de données :** Les informations ou les données d'entrée à analyser par le système sont des caractéristiques essentielles des IDS avant d'entamer le processus de détection, les données proviennent des trafics réseau filière ou sans fil, les Systèmes IDS se base sur ces données sont

appelés Network-based IDS (NIDS). Elles peuvent aussi être des données des machines hôtes comme les log générées par le système d'exploitation ou par les applications, les Systèmes IDS basés sur ces données sont appelés Host-based IDS (HIDS). Un autre type d'IDS apparaît récemment, c'est le Cloud-based IDS basé sur les données du cloud computing [5].

**B. Collection des données :** Deux architectures de système différentes :

- ***Centralise :*** la source de données et le moteur de détection sont tous ensemble
- ***Distribue :*** les données sont collectées à l'aide de plusieurs capteurs situés à différents endroits.

**C. Outil de collection** : Leur rôle est d'accéder aux données brutes, de les filtrer et de ne renvoyer que les informations intéressantes à l'analyseur IDS. Cet outil peut fonctionner sur des capteurs externes ou des agents logiciels système.

### 1.4.2 La stratégie de détection

**A. L'analyse et le traitement :** c'est l'architecture de traitement utilisée "centralise" ou "distribues".

**B. Discipline de détection :** Les outils de détection peuvent être basés sur :

- ***L'état :*** détecter les intrusions en acquittant l'état sécurisé ou l'état non sécurisé.
- ***Transitions :*** Identifie les transitions spécifiques qui conduisent à des états dangereux.

**C. La Méthodologie de détection :**

- **La détection d'anomalies (*Anomalie-based détection* ) :** il définit des modèles de comportement normal (profils) pour les entités surveillées (trafic réseau, services, application, etc.), et les écarts importants entre les comportements et les modèles observés sont considérés comme potentiellement suspects et anormaux.
- **La reconnaissance de signature (*Signature-based détection* ) :** basé sur la technologie de correspondance du modèle pour détecter les intrus Une alarme est déclenchée lorsqu'une

signature d'intrusion correspond à une signature précédente déjà dans la base de données de signature.

- **Détection basé sur les spécifications (*Spécification-based détection)* :** Si le système a une connaissance préalable de la spécification du protocole, l'utilisation abusive de ce protocole sera signalée comme une activité malveillante.

- **Détection Hybride :** La première méthode a un taux élevé de fausses alarmes (false alarms) et la seconde méthode ne parvient pas à détecter de nouveaux intrus qui ne figurent pas dans la base de données. La dernière méthode ne parvient pas à détecter les attaques qui semblent utiliser le protocole de manière inoffensive. À cette fin, les systèmes IDS hybrides combinent plusieurs méthodes pour fournir une détection plus complète et plus précise.

Les IDS peuvent également être classés selon la méthode de détection utilisée. Elles se divisent en trois catégories : détection basée sur la signature, détection basée sur les anomalies et détection hybride [6].

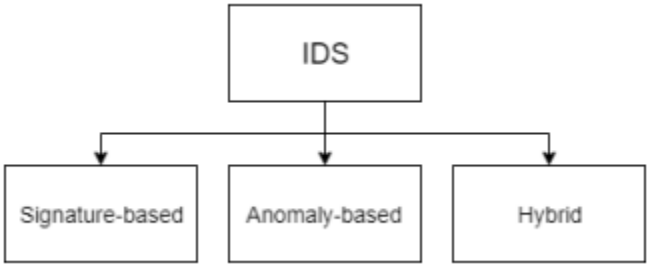


**Figure 1.4 – Classification of IDSs by detection method** [6]

### 1.4.3 L'Opportunité ( *Timeliness* )

**A. Le temps de détection :** Il s'agit du temps entre l'événement analysé et la détection, qui peut être une détection en temps réel (détection en ligne) ou non en temps réel (détection hors ligne).

**B. Le type de réponse :**

- **Passive :** Une réponse d'intrusion passive est une alerte ou un message qui avertit un administrateur système (opérateur) lorsqu'une attaque est détectée.

- **Active :** La réponse active nécessite des mécanismes et des moyens supplémentaires de réponse passive pour permettre aux réponses automatisées d'arrêter une attaque en cours. et bloquer son adresse IP.

**C. La fréquence d'analyse :**

- **Périodique :** Il stocke une grande quantité de données pendant une certaine période de temps, puis démarre le processus de reconnaissance.

- **Continu :** La détection est Continu pour toutes les activités et tous les événements qui se produisent afin d'améliorer les niveaux de sécurité dans les contextes sensibles.

### 1.4.4 Déploiement du système

Différents types de technologie IDS, en fonction de l'emplacement et de l'environnement dans lequel elles sont déployées, pour enquêter sur les activités suspectes et les types d'événements pouvant être détectés:

**A. Architecture de réseaux :** En ce qui concerne le nombre de systèmes IDS et la corrélation entre eux, il existe trois architectures. :

- ***Systèmes centralisés :*** un système unique qui collecte les données et identifie les intrus.
- ***Systèmes Distribué :*** étant donné que les données sont collectées à partir de plusieurs IDS, la corrélation des données entre les IDS peut identifier les attaques distribuées.
- ***Systèmes hybrides***

**B. Type de réseaux :** Dépend de la connectivité IDS à votre système surveillance. Il peut s'agir d'une connexion filaire, sans fil ou hybride.

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

- ***IDS basé sur l'hôte :*** Surveiller et collecter les caractéristiques des hôtes contenant des informations sensibles, des serveurs exécutant des services publics et des activités suspectes.

- ***IDS basé sur réseaux :*** Capturer le trafic réseau sur les segments de réseau
  Analysez l'activité des applications et des journaux pour détecter les incidents suspects.

- ***IDS Sans fil :*** Si le système a une connaissance préalable de la spécification du protocole, l'utilisation abusive de ce protocole sera signalée comme une activité malveillante.

- ***Système basé sur l'analyse du comportement du réseau :*** Cette classe des systèmes se diffère à la classe précédente (IDS basés sur des réseaux), ce système inspecte le comportement du trafic réseau pour reconnaître les attaques avec des flux de trafic inattendus, comme DDos attaques, malware et des services AP inattendus.

- ***Système Hybride :*** adoptée Les technologies précédentes pour atteindre l'objectif d'une détection plus complète et plus précise.

## 1.5 Les techniques de la détection d'intrusion

Au cœur de tout système de détection d'intrusion se trouve un moteur de détection d'activités malveillantes. Le système de première génération s'appuie sur les connaissances des experts en sécurité pour identifier les attaques après que plusieurs approches de détection ont été développées pour construire des systèmes de détection très précis et efficaces.

### 1.5.1 Les approches de détection

Les approches de détection d'intrusion se répartissent en deux groupes principaux :

Détection d'anomalies et détection de signature. Cependant, pour voir les propriétés globales de mon approche de détection, il n'y a pas beaucoup de différence entre les propriétés de ces deux classes. Une classification de cinq sous clades a été proposée par (Liao et al) [4]. Il a une vue approfondie des propriétés statistiques basées sur des statistiques, des modèles, des règles, des états et des heuristiques.

Les approches basées sur les statistiques consistent principalement en des caractéristiques de données statistiques telles que des seuils prédéfinis, des moyennes, des écarts-types et des probabilités d'identification des intrus. Se concentrer sur la reconnaissance basée sur des motifs À propos des méthodes de classification des modèles par attaque connue. Les techniques basées sur des règles sont principalement appliquées sur la base de " if-then " ou " if-then-else " pour créer des modèles et des profils d'intrus connus. Les méthodes basées sur l'état utilisent des machines à états finis dérivé du comportement du réseau pour identifier les attaques telles que l'analyse de protocoles et les modèles de processus de Markov.

La finale est une approche heuristique, appliquant des techniques d'intelligence artificielle inspirées de concepts biologiques tels que le système immunitaire, les algorithmes génétiques et l'intelligence en essaim. Des recherches récentes combinent ces différentes approches de détection dans des approches sophistiquées pour obtenir une plus grande précision et efficacité [4].

### 1.5.2 Les mesures d'évaluation de l'IDS

Voici quelques façons d'évaluer l'efficacité globale de mon système de détection d'intrusion [7]:

**A. La précision :** Les systèmes IDS détectent avec précision les attaques sans générer de fausses alarmes. L'inexactitude se produit lorsque ma déclaration d'un comportement légitime dans mon environnement est anormale ou bénéfique.

**B. La performance de traitement :** mesuré par la vitesse à laquelle les événements sont traités. Un système IDS plus efficace permettrait une détection en temps réel.

**C. La complétude :** il s'agit d'une fonctionnalité IDS qui détecte toutes les attaques.

D. **La tolérance aux pannes :** La plupart des systèmes de détection d'attaque fonctionnent sur des systèmes d'exploitation ou du matériel connus pour être vulnérables aux attaques. Par conséquent, l'IDS doit être résistant à ces attaques, en particulier les attaques par déni de service.

E. **La rapidité :** Un IDS doit être rapide en termes d'analyse et d'exécution pour minimiser le temps de réponse et empêcher les attaques de modifier la source de vérification ou de perturber les opérations du système [7].

En général, les systèmes IDS nécessitent des taux de détection élevés pour empêcher les attaques avant qu'elles ne compromettent le système. [8]

Cette étape dessine le processus d'évaluation de la performance du modèle. Les performances du modèle peuvent être évaluées à l'aide de différentes métriques. pour calculer la précision avec les taux de vrais et faux positifs (VP, VN) et les taux de faux positifs et faux négatifs (FP, FN).

La précision, le rappel et la courbe caractéristique de fonctionnement du récepteur (ROC) sont utilisés pour montrer les performances du modèle à tous les seuils de classification. La formule de cette métrique est décrite dans ce qui suit [9]:

- **Vrai Positif (TP) :** nombre de tentatives d'intrusion détectées avec succès.
- **Vrai Négatif (TN) :** nombre de tentatives de non-intrusion détectées avec succès.
- **Faux Positif (FP) :** nombre de non-intrusions mal détectées.
- **Faux Négatif (FN) :** nombre d’intrusions mal détectées.

**Tableau 1.1 – Matrice de confusion** [9]**.**

| **Attaques** | **Classe prévue** | |
|---|---|---|
| | **Oui** | **Non** |
| **Oui** | **VP** | **VN** |
| **Non** | **FP** | **FN** |

## 1.6 Conclusion

On conclut que le système de détection d'intrusion a un modèle de base d'analyse de la source de données et de capture de l'intrusion afin qu'il puisse répondre avec le meilleur fonctionnement. J'ai conclu également que l'IDS a des mesures d'évaluation et des approches de détection pour lutter contre les intrusions.

Après avoir parlé du système de détection d'intrusion, il est maintenant nécessaire de se plonger dans l'outil qui a été développé à partir de la recherche et de la création de ce système Dans le prochain chapitre, j'ai fourni une analyse approfondie de l'apprentissage automatique, qui sert de base à cet outil.

# Chapitre 2 : Apprentissage Automatique

# Chapitre 2

# Apprentissage Automatique

## 2.1 Introduction

L'apprentissage automatique est une branche de l'intelligence artificielle (IA) qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmée. Les chercheurs dans ce domaine doivent comprendre comment fonctionne le cerveau humain et comment le traitement de l'information est observé dans le système nerveux biologique pour donner aux machines la capacité d'apprendre à partir des données, de les interpréter et de prendre des décisions éclairées. Communication autant que possible.

Les méthodes d'apprentissage automatique ont été appliquées avec succès dans plusieurs produits TIC (reconnaissance d'images, traduction automatique, diagnostics médicaux, etc.) et divers autres domaines techniques (voitures autonomes, robots intelligents, etc.) ces dernières années. Cependant, la performance de ce dernier dépend implicitement de la qualité des données d'apprentissage. Une étape importante appelée features ingénierie est nécessaire. Il est défini comme une méthode prescrite par les experts du domaine pour sélectionner les caractéristiques ou les propriétés de données importantes de chaque problème. Pour cette raison, avec la disponibilité des métadonnées, un nouveau processus d'apprentissage automatique appelé apprentissage en automatique a été utilisé pour apprendre des représentations et des caractéristiques implicitement abstraites [10].

Ce chapitre commençait par l'importance de l'apprentissage en profonde pour la détection d'intrusion. On a ensuite introduit la définition et le classement Emmy à DL méthode. J'ai également décrit trois approches de DL qui permettent cela le sujet de mon travail. À la fin du chapitre, j'ai cité divers ensembles de données utilisés pour évaluer les systèmes IDS basés sur l'apprentissage automatique.

## 2.2 Définition de l'apprentissage profonde

L'apprentissage profonde (Deep learning ou DL) appartient à une classe de techniques d'apprentissage automatique (machine learning ou ML), il obtient un grand succès dans de nombreuses tâches de l'intelligence artificielle (IA) par rapport aux algorithmes de ML classiques. Les architectures des modèles profondes sont relativement récentes où de nombreuses étapes de traitement non linéaire de l'information sont exploitées, dans lesquelles les informations sont traitées en couches hiérarchiques, chacune recevant et interprétant les informations de la couche précédente pour l'apprentissage des représentations de données [11].

### 2.2.1 Fonctionnement

Typiquement, pour chacun de ces types de réseaux, l'architecture d'un réseau profond est organisée en couches de neurones : une couche d'entrée (Input Layer), une ou plusieurs couches cachées (Hidden Layer) et une couche de sortie (Output Layer).

Chaque paire de couches adjacentes est connectée. Les liens entre eux sont appelés pieds. Les "neurones" d'une même couche sont souvent appelés "nœuds" et n'ont pas de connexion. La figure 2.1 illustre de l'architecture standard d'un modèle de réseau neuronal profond.

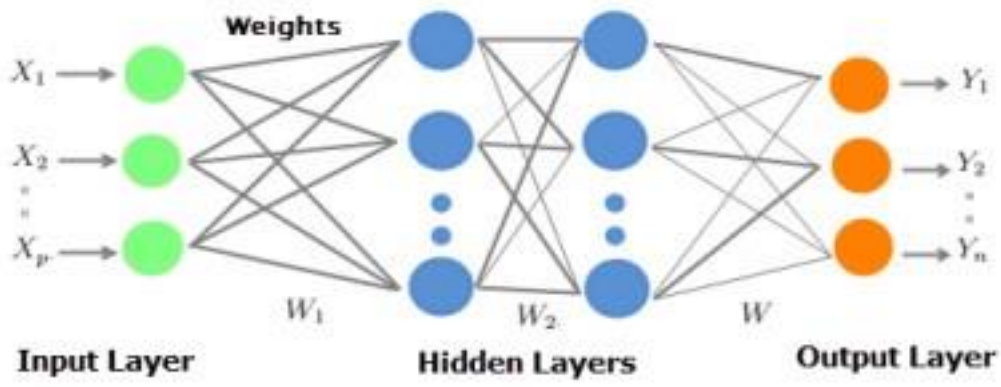


**Figure 2.1 – L'architecture d'un modèle Deep Learning** [12].

L'apprentissage en automatique lui-même est un système informatique avancé qui se compose de diverses techniques dans le domaine de l'apprentissage automatique qui utilisent un grand nombre de neurones non linéaires (nœuds) disposés en plusieurs couches de traitement pour extraire et transformer des valeurs variables d'entités à partir de vecteurs d'entrée afin de créer plusieurs niveaux, d'abstractions pour représenter les données [13].
L'apprentissage DNN (deep neural network) est l'optimisation des paramètres de pondération et de biais entre deux couches adjacentes pour évaluer l'exactitude du modèle et mieux s'adapter aux données d'apprentissage requises. Une fois que le modèle atteint une précision maximale avec des paramètres optimaux, il est généralisé aux données réelles. La quantité et la qualité des données d'apprentissage déterminent la qualité de l'apprentissage et déterminent la précision du modèle résultant.

### 2.2.2 Classification des méthodes DL

En pratique, toutes les approches d'apprentissage en profondeur sont des réseaux de neurones qui partagent certaines propriétés de base. Ils sont tous constitués de neurones interconnectés, organisés en couches. Ce qui les distingue, c'est l'architecture du réseau (comment les neurones sont organisés au sein du réseau) et parfois la façon dont ils se forment. Ferrag et al [14]. J'ai publié une étude analysant 10 approches d'apprentissage en profondeur différentes qui sont les plus couramment utilisées pour la détection des attaques de cybersécurité. Ces approches peuvent être classées en trois modèles en fonction de la façon dont elles sont formées et utilisées.

- **Deep Learning pour l'apprentissage supervisé :** Ceci est utilisé lorsque les données d'étiquetage ciblées sont disponibles. Il s'agit de modèles profondément différenciés tels que les réseaux de neurones profonds (DNN), les réseaux de neurones récurrents (RNN) et les réseaux de neurones convolutifs (CNN).
- **Deep Learning pour l'apprentissage non-supervisé :** Utiliser lorsque les données d'entrée ne sont pas étiquetées. Le but des modèles génératifs est de regrouper des données selon un certain critère de similarité à des fins de reconnaissance ou de synthèse de modèles.

Deep Belief Networks (DBN), Deep Autoencoder (DA), Machines de Boltzmann restreintes (RBM), Machines de Boltzmann profondes (DBM).

- **Deep Learning hybrides :** Une combinaison hybride de ces modèles ci-dessus. Les réseaux profonds non supervisés peuvent fournir une bonne initialisation pour étudier la discrimination (apprentissage supervisé).

## 2.3 Méthodes d'apprentissage automatique

### 2.3.1 Deep Neural Network (DNN)

Un réseau de neurones profonds (DNN) est un ensemble de neurones organisés en couches appelées perceptions multicouches (MLP). Il se distingue des réseaux de neurones classiques (réseaux de neurones artificiels) par la profondeur et le nombre de couches, nœuds (neurones) qui composent le réseau. Lorsqu'une ANN a deux couches cachées ou plus, on l'appelle un réseau neuronal profond. Ils tentent de combiner diverses transformations non linéaires pour modéliser des données dans des architectures complexes [15].

Le concept de base de la perception a été introduit par Rosenblatt en 1958 [16]. Perception calculant une sortie unique à partir de plusieurs entrées à valeur réelle (xi) en formant une combinaison linéaire basée sur les poids d'entrée (w) et en exécutant la sortie via une fonction d'activation non linéaire. Mathématiquement possible, Il s'écrit:

$$y = \delta\left(\sum_{n=1}^{n} W_i x_i + b\right) = \delta(W^T X + b) \qquad \textbf{(2-1)}$$

Avec :

- ***W :*** est le vecteur des poids.
- ***X :*** est le vecteur des entrées.
- ***b :*** désigne le biais.
- ***δ :*** représente la fonction d'activation.

Un réseau de reconnaissance multicouche (MLP) typique comprend un ensemble de nœuds source formant une couche d'entrée, une ou plusieurs couches cachées de nœuds de calcul et une couche de sortie de nœuds. Le signal d'entrée est étalé couche par couche sur le réseau. Le flux de signaux dans un tel réseau de couches cachées est illustré à la (Fig. 2.1). Les DNN sont couramment utilisés dans les problèmes d'apprentissage supervisé. Entraîner (apprendre) un modèle signifie ajuster tous les poids et toutes les précharges à des niveaux optimaux.

### 2.3.2 Convolutional neural networks (CNN)

Les réseaux de neurones convolutifs sont des types spécialisés de réseaux de neurones artificiels qui utilisent une opération mathématique appelée convolution à la place de la multiplication matricielle générale dans au moins une de leurs couches. Ils sont spécifiquement conçus pour traiter les données de pixels et sont utilisés dans la reconnaissance et le traitement d'images [17].

***On va discuter cet algorithme de Deep Learning avec plus de détails et plus d'explications dans le Chapitre 3.***

### 2.3.3 Recurrent neural networks (RNN)

Les réseaux de neurones s'inspirent du fonctionnement des neurones biologiques dans le cerveau humain. Ces neurones sont considérés comme des centres de réflexes et, avant de prendre des décisions, ils peuvent avoir besoin de mémoriser certains événements pour une utilisation ultérieure. Les réseaux de neurones traditionnels n'ont pas cette propriété. Ainsi, le comportement des réseaux de neurones récurrents (RNN) est motivé par le fait que les humains font des inférences basées sur les connaissances acquises et sur ce qu'ils ont précédemment mémorisé [18].

Les RNN ont un état interne (ou mémoires) qui prend en compte les données actuellement visualisées, plus tout ou partie des données précédemment visualisées (déjà fournies au réseau) pour ajuster les décisions. L'idée clé de base de ces réseaux est d'utiliser le calcul itératif à travers des boucles dans l'architecture du réseau. La sortie du réseau est la combinaison de son état interne (mémoire d'entrée) et de la dernière entrée, tandis que l'état interne change pour s'adapter à ses nouvelles données d'entrée. Cela permet aux informations de rester en mémoire (voir la figure 2.2).

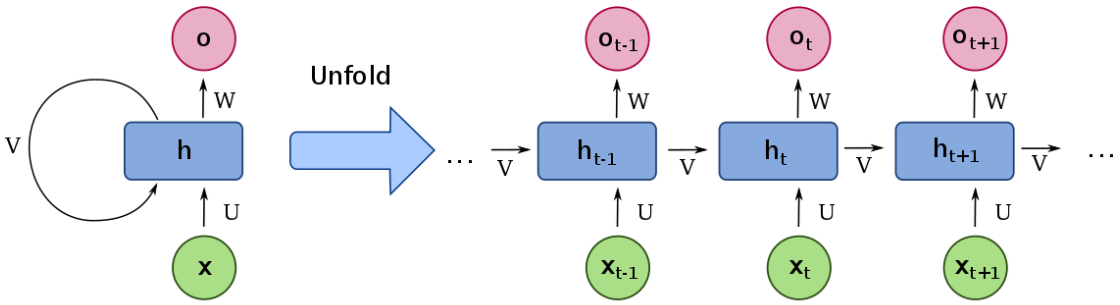


**Figure 2.2 – L'architecture d'un modèle RNN** [19]**.**

Ces propriétés rendent également les réseaux récurrents appropriés lorsque l'existence d'un formulaire n'est pas la seule information d'identification, mais, par exemple, l'ordre d'occurrences. Ils conviennent aux tâches qui traitent des données séquentielles, telles que : les données textuelles ou données avec des propriétés transitoires.

Une description mathématique du processus de transfert de mémoire est :

$$h_t = \delta(Ux_t + V\,h_{t-1} + b_h) \qquad \textbf{(2-2)}$$

$$O_t = \delta(W\,h_t + b_y) \qquad \textbf{(2-3)}$$

Un bloc de réseau neuronal, examine une entrée xt et génère une valeur mage O indice t. Une boucle de rétroaction se produit à chaque pas de temps, chaque état caché t contient des traces non seulement de l'état masqué précédent, mais également de tous ceux qui ont précédé h indice, t moins 1 fin indiquées longtemps que la mémoire peut persister.

Avec :

- $\boldsymbol{h_t}$ **:** est l'état caché au temps t.
- $\boldsymbol{x_t}$ **:** 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).

- $\boldsymbol{b_h}$ **:** est la valeur du biais de l'état caché.
- $\boldsymbol{b_y}$: est la valeur du biais de sortie.
- $\boldsymbol{O_t}$ **:** est la valeur de sortie au temps t.
- $\boldsymbol{\delta}$ **:** est une fonction de non-linéarités appelées fonctions d'activation. (soit une fonction sigmoïde logistique ou tanh) qui est un outil standard de changement d'échelle pour condenser des valeurs très grandes ou très petites dans un espace logistique, ainsi que pour rendre les gradients exploitables pour la rétro-propagation.

**Unités de mémoire à court terme (LSTM) :**

Les réseaux RNN ont des pas de temps longs. En effet, la mise à jour des pondérations pour tenir compte de l'état précédemment enregistré rend le gradient de plus en plus petit pendant l'apprentissage, et après quelques étapes, l'erreur ne peut pas se propager jusqu'à la fin du réseau. je ne peux pas mettre à jour les poids car il n'y a pas de différence significative dans les résultats. Ce problème RNN est appelé gradients de fuite. Pour résoudre ce problème, une architecture de mémoire à longue durée de vie (LSTM) pour les réseaux de neurones récurrents et une étape supplémentaire appelée unité récurrente fermée (GRU) ont été développées au milieu des années 1990 par les chercheurs allemands Sepp Hocheriez et Jürgen Schmid Huber. Ces étapes ont été utilisées pour améliorer les performances et la précision du RNN.

Une idée clé du schéma LSTM est l'état de la cellule. Il a la capacité de supprimer ou d'ajouter des informations sur l'état des cellules. Cette technologie est régulée par des structures appelées gâtés. Il peut s'agir d'une fonction sigmoïde, avec une valeur de 1 signifiant que toutes les informations sont transmises ayant une valeur de 0 signifiant l'inverse. Les architectures LSTM et GRU fonctionnent de manière similaire. Cependant, GRU utilise moins de paramètres d'entraînement, prend moins de mémoire et s'entraîne plus rapidement que LSTM. Bien que LSTM soit plus précis sur un ensemble de données de séquences plus longues. Les cellules LSTM sont les plus efficaces pour conserver les informations utiles pendant la rétro-propagation du gradient. Cela a permis de corriger les différences entre les prédictions sortantes et les catégories de référence en calculant le gradient d'erreur pour chaque neurone de la dernière couche à la première. La figure 2.3 montre quatre couches interactives (Sigmoïde et Tanh), trois portes et une opération ponctuelle agissant sur un vecteur x dans une cellule LSTM au temps t.

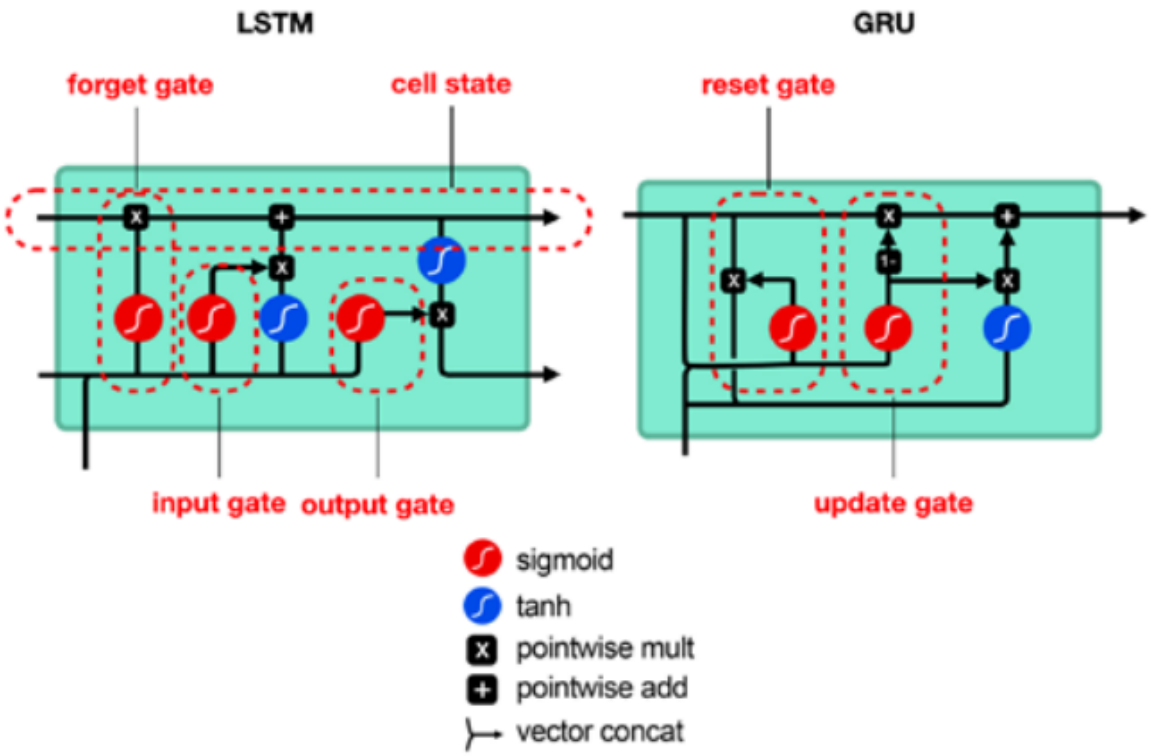


**Figure 2.3 – L'architecture d'un modèle LSTM_GRU** [20]**.**

## 2.4 Conclusion

Le domaine de l'apprentissage en automatique est très vaste et se développe rapidement, avec de nouveaux algorithmes, architectures ou variantes ajoutées chaque semaine. L'application de nouvelles méthodes DL et l'évaluation des performances de diverses architectures DL existantes constituent un axe et une direction de recherche majeurs pour les chercheurs en sécurité.

Maintenant qu'on a parlé de l'apprentissage automatique, il faut parler du choix des meilleures méthodes de l'apprentissage automatique que l'on souhaite utiliser sur l'étude et la création de ce système de détection d'intrusion, dans le chapitre 3, on expliquera la méthode choisie.

# Chapitre 3 : La détection d'intrusion basée sur l'Apprentissage Automatique

# Chapitre 3

# La détection d'intrusion base sur l'Apprentissage Automatique

## 3.1 Introduction

Récemment, différentes approches d'apprentissage en profondeur pour la détection d'intrusion ont été développées et examinées.

Les systèmes de détection basés sur les signatures identifient les intrusions en comparant le comportement surveillé avec des modèles prédéfinis d'intrusion. En revanche, les systèmes basés sur les anomalies se basent sur une connaissance du comportement normal pour détecter les déviations ou les activités suspectes.

Les techniques d'apprentissage en profondeur peuvent être appliquées à ces deux types de détection en exploitant leur capacité à extraire des relations non linéaires de niveau supérieur entre les données, ce qui permet d'identifier les comportements inhabituels par rapport aux activités bénignes.

Cependant, ces méthodes exigent de vastes quantités de données pour détecter et identifier différentes classes de modèles. Dans ce chapitre, j'ai exploré et analysé diverses capacités de détection d'intrusion basées sur l'apprentissage en profondeur, incluant les ensembles de données utilisés, l'architecture du réseau et les métriques des résultats obtenus.

## 3.2 Convolutional neural networks (CNN)

Un réseau neuronal convolutif (CNN) est une extension du réseau feed-forward traditionnel (FFN) associé à l'inspiration biologique. Ils ont été étudiés à l'origine pour le traitement d'images dans lesquelles des motifs répétitifs ont été trouvés. Par exemple, des images avec des bords répétés et d'autres motifs. CNN surpasse tous les autres algorithmes ML conventionnels, Ils ont très bien réussi dans les tâches de traitement de vision par ordinateur (tâches de vision par ordinateur) et ont de nombreuses applications dans le traitement d'images et de vidéos, le traitement du langage naturel (NLP) et les systèmes de recommandations, etc.
Les réseaux convolutifs sont particulièrement efficaces grâce à plusieurs types de couches spéciales : couches convolutions, couches de pooling et couches entièrement connectées [21]. La figure 3.1 montre un modèle de réseau conventionnel unidimensionnel (CNN).

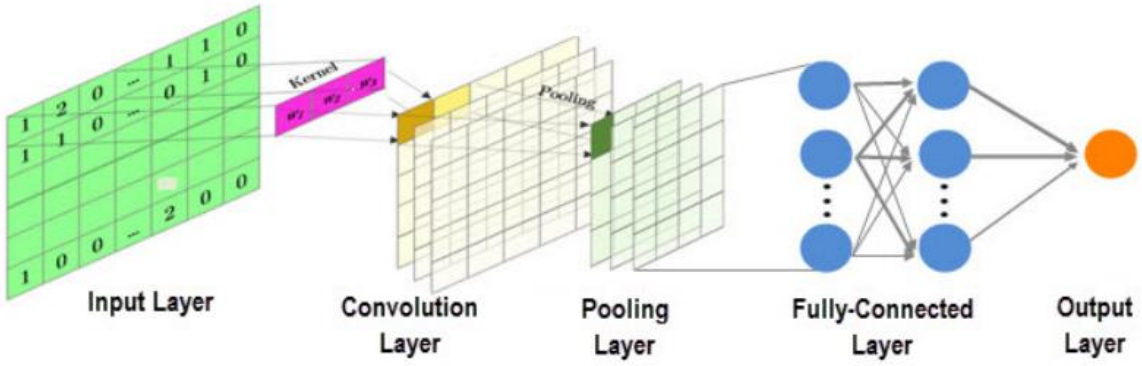


**Figure 3.1 – L'architecture d'un modèle de réseau neuronal convolutif** [22]**.**

### 3.2.1 Couche de convolution :

Le but de la convolution est d'extraire des caractéristiques de haut niveau. Il se compose d'un ensemble de filtres d'apprentissage (ou cœurs), chacun représentant une caractéristique indépendante spécifique avec un volume d'entrées. Ces filtres sont constitués de couches de poids de connexion et ont un petit champ de réception (la taille du noyau), mais lors de l'alimentation vers l'avant, chaque filtre est convolé avec la largeur et la hauteur du volume d'entrée, dont l'entrée et les points entre le filtre des valeurs produisent une nouvelle carte de caractéristiques qui représentent mieux les informations. Par conséquent, le réseau apprend des filtres qui s'activent

lorsqu'ils reconnaissent des types d'entités importants et spécifiques à un emplacement spatial particulier dans l'entrée. La figure 3.2 montre une opération de convolution 1D avec une entrée unidimensionnelle:

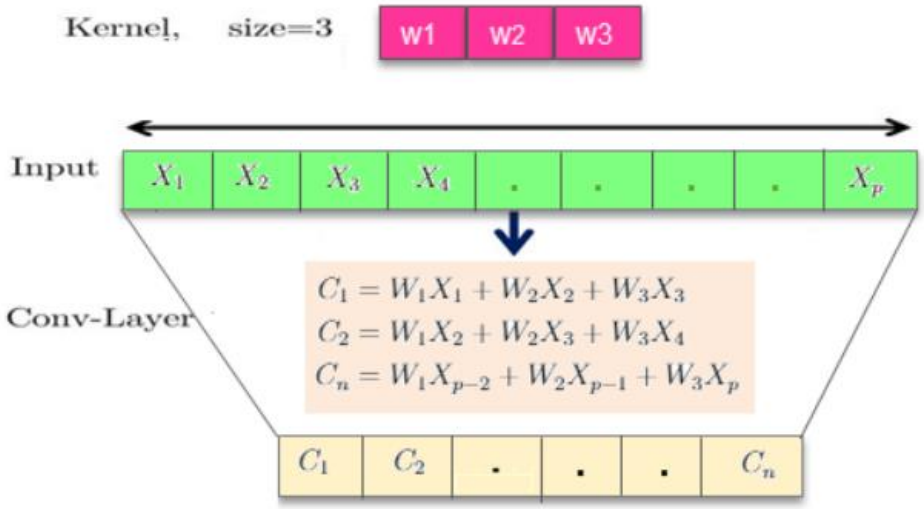


**Figure 3.2 – Convolution** [22]**.**

Les couches de convolution utilisent le même noyau de convolution, ce qui réduit considérablement le nombre de paramètres requis pour l'opération de convolution.

$$[f(x) = max(0, x)] \qquad \textbf{(3-1)}$$

Une fonction d'activation non linéaire est appliquée immédiatement après chaque couche convolutée. Deep CNN avec fonction d'activation "*Rectified Linear Units ReLU*", renvoie x pour toutes les valeurs où x > 0 et 0 pour toutes les valeurs où x <= 0. Utilisez des unités pour vous entraîner plusieurs fois plus vite que vos pairs "*Tanh Units*" [23].

### 3.2.2 Couche de pooling :

Après la transformation ReLU, l'opération de regroupement (Pooling) regroupe les activations des neurones d'une couche en neurone unique de la couche suivante. Une couche de pooling fonctionne indépendamment sur chaque unité d'entrée, réduisant progressivement la taille de la représentation pour réduire le nombre de paramètres ou de poids, réduisant la charge de calcul du réseau tout en préservant les informations les plus importantes. Il aide également à contrôler le sur-apprentissage. Il peut utiliser deux méthodes de mise en commun différentes :

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

La mise en commun est une forme de sous-échantillonnage non linéaire qui fonctionne de manière similaire à la convolution. Un noyau de regroupement convoque le volume d'entrée et le divise en une série de régions sans chevauchements. Chaque sous-région produit une seule valeur de sortie représentant la mise en commun maximale ou la mise en commun moyenne, dans la figure 3.3. L'opération de regroupement maximum avec entrée 1D et noyau de taille 2.
La couche de Pooling n'a aucun paramètre pouvant être appris. De ce fait, ces couches ne sont généralement pas incluses dans le nombre total de couches de réseau de convolution.

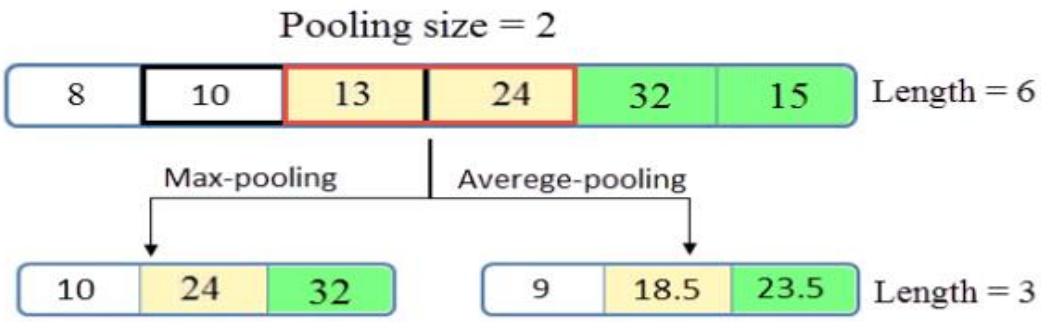


**Figure 3.3 – Pooling** [22]

### 3.2.3 Couche de Full connected :

À la fin d'un CNN se trouve une ou plusieurs couches entièrement connectées (tous les nœuds de la première couche sont connectés à tous les nœuds de la couche suivante). Ils consistent à effectuer une classification basée sur les caractéristiques extraites de la convolution. La couche finale contient une fonction d'activation softmax qui renvoie une valeur de probabilité comprise entre 0 et 1 pour chaque étiquette de classe que le modèle tente de prédire. Dans certaines architectures de réseau CNN modernes, les couches entièrement connectées peuvent être remplacées par plusieurs couches de mise en commun moyennes (average-pooling). En conséquence, ces réseaux sont c'est le nombre total de paramètres et peut mieux éviter le sur-apprentissage [24].

## 3.3 Les Data-set d'évaluation des IDSs base sur Deep Learning :

Les ensembles de données utilisés dans les études publiées pour l'application de l'apprentissage en automatique dans la cybersécurité jouent un rôle important dans la validation de toutes les approches DL proposées. Certains de ces ensembles de données ne sont pas facilement accessibles en raison de problèmes de confidentialité. Les ensembles de données dédiés à la détection des attaques de cybersécurité comprennent [25]:

- **Benchmark Data-sets :** Ces ensembles de données sont accessibles au public à des fins de recherche et pour l'évaluation des performances des algorithmes proposés.
- **Data-sets Privé :** Les données sont collectées à partir de sources publiques et en temps réel sont considérés comme des ensembles de données privés. Ces ensembles de données ne sont pas accessibles au public à des fins de recherche.
- **En temps réel :** Les données sont collectées dans un environnement en temps réel et sont considérés comme real-time data-sets.
- **Collectés à partir de sources accessibles au public :** Ces données sont collectées à partir de diverses sources accessibles au public. Dans la plupart des cas, ces Data-sets ne sont pas accessibles au public à des fins de recherche.

Selon [26] [14] [25] Parmi les ensembles de données accessibles au public qui sont largement utilisés comme des benchmark il y a : DARPA, le KDD99, le NSL-KDD et l'ADFA-LD.

**Tableau 3.1 – Ensembles de données public relatives à la cybersécurité.**

| Data-set public | Type | Etiqueté | Année | [ Réf ] |
|---|---|---|---|---|
| **KDD99** | **trafic du réseau** | **oui** | **1999** | [27] |
| **NSL-KDD** | **trafic du réseau** | **oui** | **2009** | [28] |
| **MAWI** | **trafic de l'internet** | **oui** | **2011** | [29] |
| **ISCX dataset** | **trafic du réseau** | **oui** | **2012** | [28] |
| **CIC DoS dataset** | **trafic du réseau** | **oui** | **2017** | [28] |
| **Bot-IoT dataset** | **trafic IoT** | **oui** | **2018** | [30] |
| **CIC DDoS** | **trafic du réseau** | **oui** | **2019** | [28] |

## 3.4 Travaux connexes pour la détection d'intrusion base sur l'Apprentissage automatique

En ce qui concerne les travaux de détection d'intrusion basés sur l'apprentissage automatique, je dois trouver l'algorithme optimal parmi les 3 méthodes d'apprentissage en profondeur en comparant leur précision et en mesurant leurs performances et en fonction des données que j'obtiens de l'étude, je peux trouver le meilleur algorithme parmi 3 méthodes et choisir celui avec lequel je veux travailler dans mon étude du système de détection d'intrusion réseau utilisant l'apprentissage automatique.

**Tableau 3.2 – Travaux antérieurs connexes pour la détection d'intrusion basé sur l'apprentissage automatique.**

| Méthode | Dataset | Messeuses de performances | Année | [ Réf ] |
|---|---|---|---|---|
| **CNN** | **NSL-KDD** | **ACC = 97.88% - 99.46%**<br>**DR = 68.66%**<br>**FPR = 27.9%** | **2018** | [31] |
| **RNN** | **DARPA** | **DR = 95.37%**<br>**FPR = 2.1%** | **2015** | [32] |
| **DNN** | **NSL-KDD** | **ACC = 75.75%**<br>**PR = 83%**<br>**RC = 76%**<br>**F1 = 75%**<br>**ROC_AUC = 86%** | **2016** | [33] |

## 3.5 Conclusion

Du tableau 3.2, on conclut que l'algorithme CNN est le plus précis et le plus efficace pour détecter les intrusions simplement en comparant la performance de la masseuse pour chaque algorithme, et pour cela, on doit utiliser l'algorithme CNN comme algorithme principal pour mon étude sur la détection d'intrusion à l'aide de l'apprentissage en profondeur.

# Chapitre 4 : Conception et réalisation

# Chapitre 4

# Conception et réalisation

## 4.1 Introduction

Les systèmes de détection d'intrusion basés sur l'apprentissage en profondeur ont été largement étudiés dans la recherche. Dans le chapitre précédent, j'ai discuté des différentes méthodes d'apprentissage en profondeur qui ont été appliquées avec succès à la détection d'intrusion en utilisant divers ensembles de données dédiés à la cybersécurité. Cependant, les performances de ces systèmes dépendent fortement de l'ensemble de données utilisées, et il n'existe pas de modèle de référence universel pour la détection d'intrusion.

Dans cette étude, j'ai choisi un ensemble de données récent fournies par le département américain d'information et d'informatique de l'Université de Californie (UCI), appelé KDD99, pour l'utiliser dans le cadre du troisième concours international d'outils de découverte de connaissances et d'exploration de données.

Comme conclu dans le chapitre 3, j'ai utilisé des modèles d'apprentissage en profondeur (CNN) en cybersécurité pour permettre aux IDS de faire face à différents types d'attaques réseau, en particulier pour identifier les activités DDoS malveillantes dans le trafic réseau réel. J'ai commencé par un CNN simple pour classifier ce type d'attaque réseau afin de résoudre le problème de détection des différentes attaques DDoS possible avec un taux de détection très élevé et un taux d'alarme négligeable. J'ai évalué les performances des approches de détection proposées en utilisant différentes mesures d'évaluation des algorithmes d'apprentissage profonds, telles que la précision, le rappel, le score F1, le taux de détection et le taux de fausse alarme.

## 4.2 Environnement d'exécution

L'apprentissage en profondeur est un domaine avec des exigences de disponibilité des ressources matérielles (en particulier les GPU) capables d'effectuer des calculs intensifs. Tout d'abord, on a commencé à configurer un environnement de développement Python local à l'aide de la plate-forme Anaconda.

### 4.2.1 Anaconda :

L'apprentissage en profondeur est un domaine avec des exigences de disponibilité des ressources matérielles (en particulier les GPU) capables d'effectuer des calculs intensifs. Tout d'abord, on a commencé à configurer un environnement de développement Python local à l'aide de la plate-forme Anaconda [34].

Afin de déplacer notre travail vers l'apprentissage en profondeur, j'ai dû passer au cloud. Ces derniers fournissent des ressources de calcul et de stockage importantes au-delà des ordinateurs. Selon les besoins, le cloud peut donner accès à des processeurs graphiques GPU entièrement gratuits. Google Colab est l'un des outils cloud d'apprentissage automatique les plus utilisés [35].

### 4.2.2 Google Colab :

Un service cloud basé sur Jupyter Notebook vous permet de développer des applications de l'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.

Comme langage de développement, j'ai choisi Python, un langage de programmation interprété multiparadigme et multiplateforme. Python est également un langage plus populaire et populaire pour l'apprentissage automatique et l'intelligence artificielle en raison de sa flexibilité et du grand nombre de bibliothèques de logiciels open source disponibles. .Activez les bibliothèques utilisées dans le projet : pandas, numpy, scikit-learn. . etc. Panda et Numpy sont utilisés pour la manipulation des données (chargement, réorganisation et traitement des données), et Scikit-Learn vous permet

d'expérimenter rapidement et facilement diverses techniques et algorithmes prédéfinis pour l'apprentissage automatique et l'analyse des données.

Les frameworks TensorFlow et Keras ont été choisis pour implémenter la méthode d'apprentissage en profondeur proposée. TensorFlow est une bibliothèque d'apprentissage en profondeur open source développée par Google et utilisée pour effectuer des opérations mathématiques complexes et diverses autres tâches de modélisation d'architecture d'apprentissage en profondeur. Les calculs peuvent être facilement répartis sur plusieurs plates-formes telles que les CPU et les GPU.

Keras est une API de haut niveau pour la création et la formation de modèles d'apprentissage en profondeur basés sur Python. Il a été montré pendant le but de digérer une vérification rapide. Parmi ces avantages:

- J’ai pu passer de l'idée au résultat dans les délais les plus courts. Et c'est la clé d'une recherche efficace.

- Prend en charge les réseaux convolutifs (CNN) et les réseaux récurrents (RNN), ainsi que les combinaisons des deux éléments Il n'y a pas de fichier de configuration de modèle séparé, tout est déclaré dans le code.

- Fonctionne sur CPU et GPU.

Keras fournit toutes les fonctionnalités générales nécessaires pour créer des modèles d'apprentissage en profondeur, mais pas autant que TensorFlow, qui propose des opérations plus avancées pour mieux contrôler le développement de types de modèles spécifiques. Cela m'a aidé également à mieux comprendre ce qui se passe à l'intérieur du réseau DL. Pour tirer le meilleur parti des deux mondes, j'ai utilisé TensorFlow comme backend pour Keras. RAM.

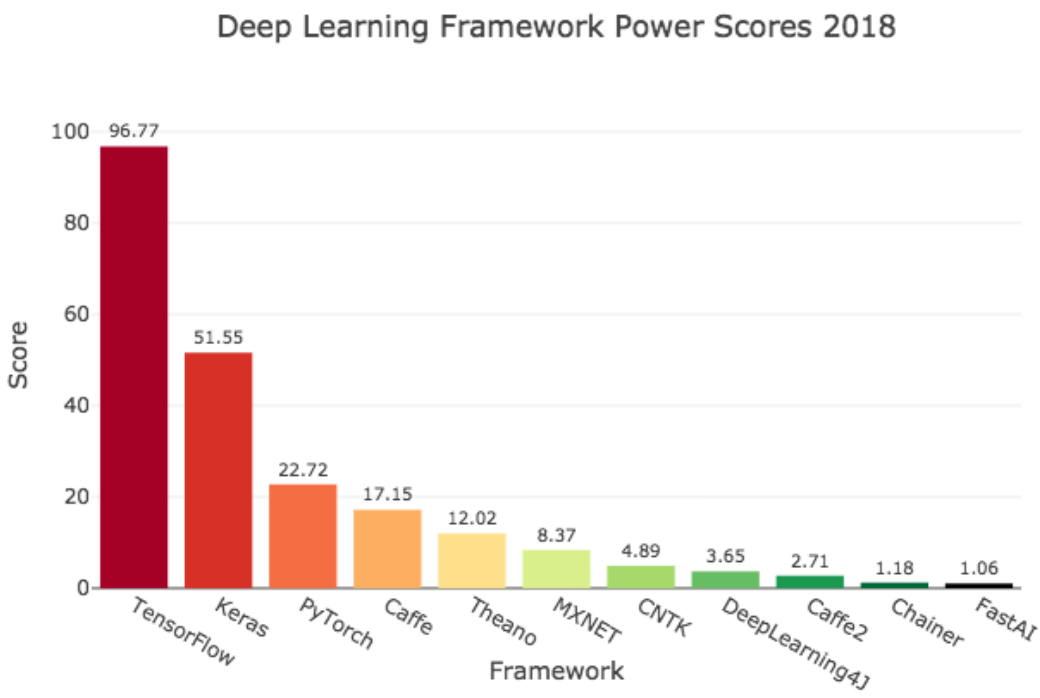


**Figure 4.1 – Les 10 frameworks Deep learning les plus Populaire** [36].

## 4.3 Dataset

La base de données KDD99 est une base de données de référence bien connue dans le domaine des systèmes de détection d'intrusion (IDS). Fourni par le département américain d'information et d'informatique de l'Université de Californie (UCI), L'ensemble de données a été créé par l'armée américaine en 1999 et a été largement utilisé dans la recherche pour évaluer les performances de différents algorithmes d'apprentissage automatique et technique d'exploration de données pour détecter les intrusions sur le réseau.

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 311 029 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 [37] .

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 [38].

### 4.3.1 Caractéristiques du Dataset:

**Tableau 4.1 – Caractéristiques de base des connexions TCP individuelles** [39]**.**

| Nom de la fonction | Description | Type |
|---|---|---|
| **Duration** | **length (number of seconds) of the connection** | **Continu** |
| **Protocol_type** | **type of the protocol, e.g. tcp, udp, etc.** | **discrète** |
| **service** | **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** | **1 if connection is from/to the same host/port; 0 otherwise** | **discrète** |
| **Wrong_fragment** | **number of ``wrong'' fragments** | **Continu** |
| **Urgent** | **number of urgent packets** | **Continu** |

**Tableau 4.2 – Fonctionnalités de contenu dans une connexion suggérée par la connaissance du domaine** [39]**.**

| Nom de la fonction | 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 | 1 if root shell is obtained; 0 otherwise | discret |
| Su_attempted | 1 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_files | 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 | discret |
| Is_guest_login | 1 if the login is a ``guest''login; 0 otherwise | 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 4.3 – Caractéristiques du trafic calculées à l'aide d'une fenêtre de temps de deux secondes** [39]**.**

| 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 following features refer to these same-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 different services** | **Continu** |
| **Srv_count** | **number of connections to the same service as the current connection in the past two seconds** | **Continu** |
| | ***Note: The following features refer to these same-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 connections to 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).Prétraitement du dataset:

Pour retraiter notre base de données KDD99, je dois passer à 5 étapes très importantes, car elles sont essentielles pour préparer notre base de données d'abord pour y travailler et appliquer des fonctions pour atteindre notre objectif.

### A. Supprimez les fonctionnalités qui ne sont pas nécessaires :

L'ensemble de données KDD99 est un ensemble de données largement utilisées pour la détection d'intrusion dans la sécurité du réseau. Il contient un grand nombre de fonctionnalités liées aux données de trafic réseau. Cependant, toutes les fonctionnalités ne sont pas pertinentes pour le problème de la détection d'intrusion. Par conséquent, la première étape du prétraitement du jeu de données KDD99 consiste à supprimer toutes les fonctionnalités qui ne sont pas requises. Cela peut inclure des colonnes qui ne présentent aucune variation, contiennent des informations en double ou redondantes, ou ne sont pas pertinentes pour l'analyse spécifique ou la tâche de modélisation en cours.

### B. Séparez les caractéristiques et les étiquettes:

Dans un problème d'apprentissage supervisé, l'ensemble de données KDD99 est composé d'entités (également appelées prédicteurs ou entrées) et d'étiquettes (également appelées cibles ou sorties). Les caractéristiques sont les attributs du trafic réseau et les étiquettes indiquant si chaque connexion réseau est normale ou une instance de l'un des nombreux types d'attaque. Par conséquent, la deuxième étape du prétraitement du jeu de données KDD99 consiste à séparer les entités et les étiquettes dans des tableaux ou des dataframes séparés.

### C. Encoder les étiquettes catégorielles en nombres entiers:

Les étiquettes de l'ensemble de données KDD99 sont catégorielles, ce qui signifie qu'elles représentent des classes ou des catégories discrètes (c'est-à-dire normales ou l'un des nombreux types d'attaques). Cependant, la plupart des algorithmes d'apprentissage automatique cependant, la plupart des algorithmes d'apprentissage automatique nécessitent des étiquettes numériques comme entrées. Par conséquent, la troisième étape du prétraitement de l'ensemble de données KDD99 consiste à encoder les étiquettes catégorielles en nombre entier à l'aide de techniques telles que l'encodage d'étiquettes ou l'encodage ordinal.

### D. Codage à chaud des étiquettes:

Dans certains cas, l'encodage d'étiquettes catégorielles sous forme d'entier peut créer une fausse hiérarchie ou relation entre les classes, ce qui peut nuire aux performances des modèles d'apprentissage automatiques. L'encodage à chaud est une technique qui peut être utilisée pour représenter des variables catégorielles sous forme de vecteurs binaires, où chaque vecteur représente une seule classe et a une valeur de 1 dans la position correspondante et de 0 ailleurs. Par conséquent, la quatrième étape du prétraitement de l'ensemble de données KDD99 consiste à coder à chaud les étiquettes catégorielles.

### E. Fonctionnalité de mise à l'échelle des données d'entrée :

Le jeu de données KDD99 contient des entités avec différentes plages des valeurs. Certains algorithmes d'apprentissage automatique sont sensibles à l'échelle des caractéristiques d'entrée et peuvent produire des résultats sous-optimaux si les caractéristiques ne sont pas mises à l'échelle ou normalisées. Par conséquent, la dernière étape du prétraitement du jeu de données KDD99 consiste à mettre à l'échelle les entités d'entrée à l'aide de techniques telles que la mise à l'échelle min-max ou la normalisation, ce qui peut amener toutes les entités à une échelle similaire et aider les algorithmes à converger plus rapidement et à produire de meilleurs résultats.

## 4.3.2 Fractionnement de l’ensemble de donnée :

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

**X_train :** X irrévocable utilisé pour s'adapter au modèle d'apprentissage automatique.

**X_test :** X irrévocable utilisé 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.

## 4.4 Réalisation :

### 4.4.1 Création du modèle CNN :

Créer un modèle CNN avec deux couches convolutives 1D, deux couches de regroupement (pooling) maximum, une couche entièrement connectée, une couche d'abandon et une couche de sortie. Le modèle est compilé avec l'optimiseur Adam et la fonction de perte d'entropie croisée catégorielle et est évaluée à l'aide de la métrique de précision.

**A. Création d'une instance de la classe Séquentiel, qui est un modèle Keras.:**

La classe séquentielle est un type de modèle Keras qui fournit un moyen simple et intuitif de construire et de former des réseaux de neurones, composés de nœuds interconnectés qui transforment les données d'entrée en sortie utile.

**B. Ajout d'une première couche Convolutive 1D (1D convolution layer):**

La première étape de la création d'un modèle de réseau neuronal convolutif (CNN) consiste à ajouter la couche convolutive 1D initiale, qui est responsable de l'apprentissage et de l'extraction des caractéristiques des données d'entrée.

**C. Ajout d'une première couche Pooling :**

Les couches Pooling sont utilisées pour réduire la dimensionnalité de la sortie des couches convolutives et empêcher le surajustement. La mise en pool maximale est un type courant de couches de mise en pool où la valeur maximale est conservée tandis que les autres valeurs sont ignorées, ce qui permet de conserver des fonctionnalités importantes tout en réduisant la taille des données.

**D. Ajout d'une deuxième couche Convolutive 1D (1D convolution layer):**

L'ajout d'une deuxième couche convolutive 1D à un modèle CNN peut améliorer ses performances en appliquant des filtres convolutifs supplémentaires à la sortie de la première couche. Cependant, trop de couches peuvent entraîner un surajustement, il est donc important d'équilibrer le nombre de couches avec la taille de l'ensemble de données et la complexité du problème.

### E. Ajout d'une deuxième couche Pooling :

L'ajout d'une deuxième couche de regroupement à un modèle CNN peut aider à extraire les fonctionnalités les plus pertinentes et à réduire les calculs. Cependant, trop de couches de regroupement peuvent entraîner la perte de fonctionnalités importantes, il est donc important d'équilibrer le nombre avec la complexité du problème et la taille de l'ensemble de données. La mise en commun moyenne peut être utilisée à la place de la mise en commun maximale.

### F. Aplatit la sortie des couches convolutives en un tableau unidimensionnel:

Dans un modèle CNN, après les couches de convolution et de regroupement, la sortie doit être aplatie dans un tableau unidimensionnel afin de pouvoir être transmise à une couche entièrement connectée. L'opération d'aplatissement prend la sortie multidimensionnelle des couches de convolution et de mise en commun et la convertit en une seule dimension afin que la couche entièrement connectée puisse la traiter. Cette étape est essentielle pour transmettre la sortie du CNN aux couches entièrement connectées pour les tâches de classification ou de régression.

### G. Ajout d'une couche entièrement connectée (fully connected):

L'ajout d'une couche entièrement connectée à un modèle CNN permet d'effectuer la tâche finale de classification ou de régression en apprenant les relations complexes entre les caractéristiques et la sortie. La couche contient un ensemble de neurones entièrement connectés à la couche précédente, et le nombre de neurones dépend de la complexité du problème et de la taille de l'ensemble de données.

### H. Ajout d'une couche d'abandon (dropout layer):

Une couche d'abandon dans un modèle CNN abandonne au hasard certains neurones pendant l'entraînement, réduisant la dépendance à l'égard d'un seul neurone et encourageant le modèle à apprendre des fonctionnalités plus robustes. Il peut aider à prévenir le surajustement et est généralement ajouté après les couches entièrement connectées.

### I. Ajout d'une couche de sortie (output layer):

La couche de sortie dans un modèle CNN est la couche finale qui prend la sortie de la couche précédente et applique une fonction d'activation pour la mapper à une distribution de probabilité sur des classes possibles ou des cibles de régression. La fonction d'activation et le nombre de neurones dans la couche de sortie dépendent du problème à résoudre.

### J. Compiler le modèle :

Compiler un modèle dans Keras implique de spécifier l'optimiseur, la fonction de perte et la métrique d'évaluation utilisées pendant la formation. L'optimiseur contrôle la manière dont le modèle est mis à jour en fonction de la fonction de perte et des gradients calculés lors de la rétro propagation. La fonction de perte est la fonction objective que le modèle essaie de minimiser pendant l'apprentissage. La métrique d'évaluation est la précision des problèmes de classification.

### K. Imprime un résumé:

Imprime un résumé de l'architecture du modèle, y compris les couches, le nombre de paramètres et les formes de sortie.

- On crée 3 modèles CNN avec chacun d'eux et des paramètres différents dans chaque couche, tels que (la convolution, la mise en commun, la fonction d'activation dense de la connexion complète, le taux d'abandon, la fonction d'activation dense de la sortie et enfin l'optimiseur).

## 4.4.2 Former le modèle CNN (Train the model):

Le processus de formation d'un réseau de neurones implique l'ajustement itératif des poids et des biais du modèle afin de minimiser la fonction de perte. La fonction de perte est une mesure de la capacité du modèle à prédire les valeurs de sortie pour une entrée donnée.

Pendant la formation, le modèle est présenté avec un lot de données de formation et les prévisions de sortie sont comparées aux valeurs de sortie réelles à l'aide de la fonction de perte. L'optimiseur ajuste ensuite les poids et les biais du modèle afin de réduire la perte. Ce processus est répété pour plusieurs époques, le modèle voyant les données d'apprentissage plusieurs fois.

Voici quelques concepts et techniques clés impliquées dans la formation d'un modèle CNN :

- **Rétropropagation :** consiste à calculer les gradients de la fonction de perte pour mettre à jour les poids et les biais du modèle pendant l'apprentissage.
- **La taille du lot :** détermine le nombre d'échantillons utilisés dans chaque lot d'apprentissage.
- **Taux d'apprentissage :** contrôle la taille des pas de l'optimiseur pendant l'entraînement.
- **Dropout :** il s'agit d'une technique qui fait tomber aléatoirement certains neurones pendant l'entraînement pour éviter le surajustement.
- **Arrêt précoce :** il s'agit d'une technique pour arrêter l'entraînement plus tôt si la perte de validation ne s'améliore plus.
- **Augmentation des données :** peut être utilisée pour augmenter artificiellement la taille de l'ensemble de données d'apprentissage.
- **Apprentissage par transfert :** implique l'utilisation d'un modèle préformé comme point de départ pour la formation d'un nouveau modèle.

Bien que le processus de formation d'un réseau de neurones puisse être difficile, un réglage et une optimisation minutieux peuvent aboutir à des modèles très précis pour une variété d'applications.

**Le résultat de la formation du modèle CNN :**

Selon les paramètres qu'on a choisis dans chaque modèle CNN, on a créé selon les performances de la machine d'exécution qu'on utilise, je peux voir que le temps d'exécution pour former chaque modèle CNN est différent, le 3ème modèle étant le plus rapide suivi du 1er modèle et je voulais que le 2ème modèle a pris le plus de temps pour s'entraîner.

**Tableau 4.4 – Le temps d'exécution du processus de formation de chaque modèle CNN.**

| Nom de modèle | Temp d'Exécution de formation |
|---|---|
| **CNN Model 1** | **45.85479235649109 seconds** |
| **CNN Model 2** | **142.68514251708984 seconds** |
| **CNN Model 3** | **30.53381061553955 seconds** |

### 4.4.3 Évaluation du modèle :

La sortie de la méthode "evaluate" est stockée dans la variable "score". La variable "score" contient une valeur scalaire qui représente la performance globale du modèle sur les données de test. La métrique spécifique utilisée pour l'évaluation dépend du type de modèle et du problème à résoudre. Par exemple, si le modèle est un classificateur, la métrique peut être l'exactitude, la précision ou le rappel. Si le modèle est un régressé, la métrique peut être l'erreur quadratique moyenne ou l'erreur absolue moyenne, Si la précision est plus élevée que la perte alors notre système est précis, par contre Si notre perte est plus élevée que la précision alors notre système n'est pas précis, tel que c'est indiqué dans la figure 4.2.

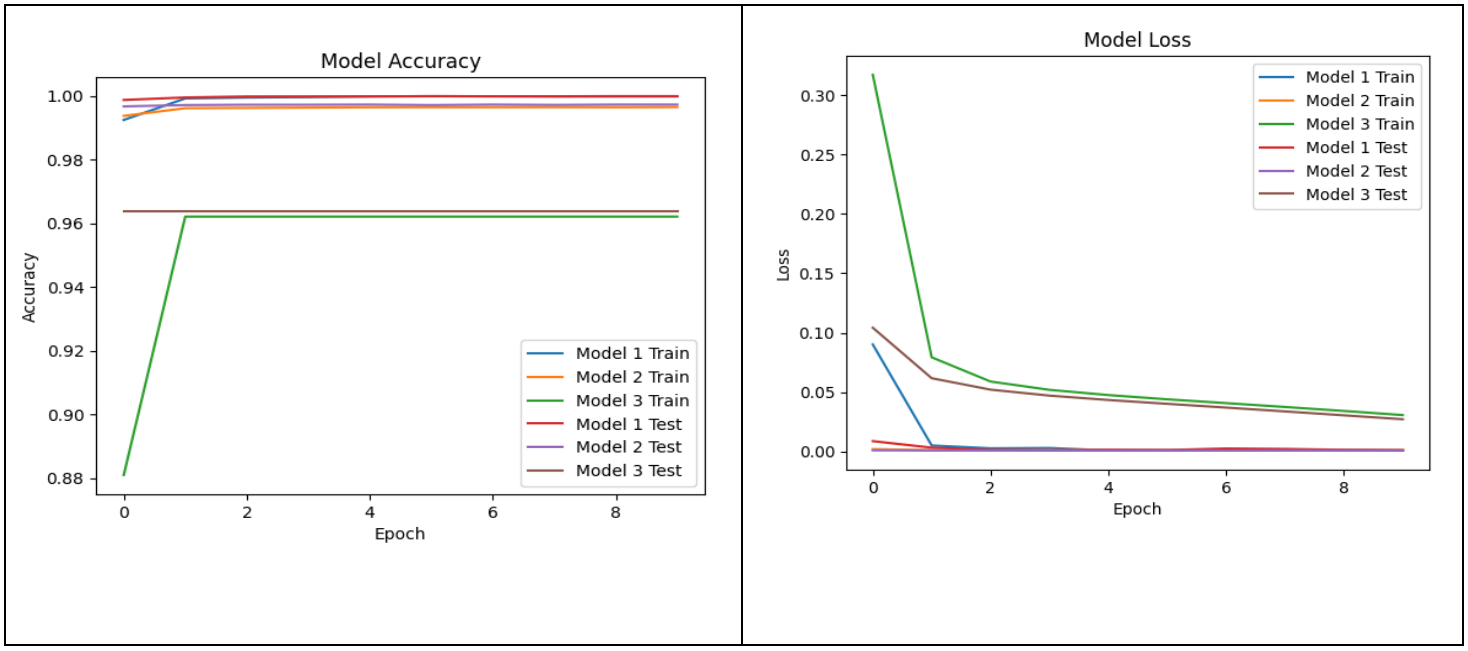


**Figure 4.2 – Le résultat de la précision et la perte par époque du modèle CNN.**

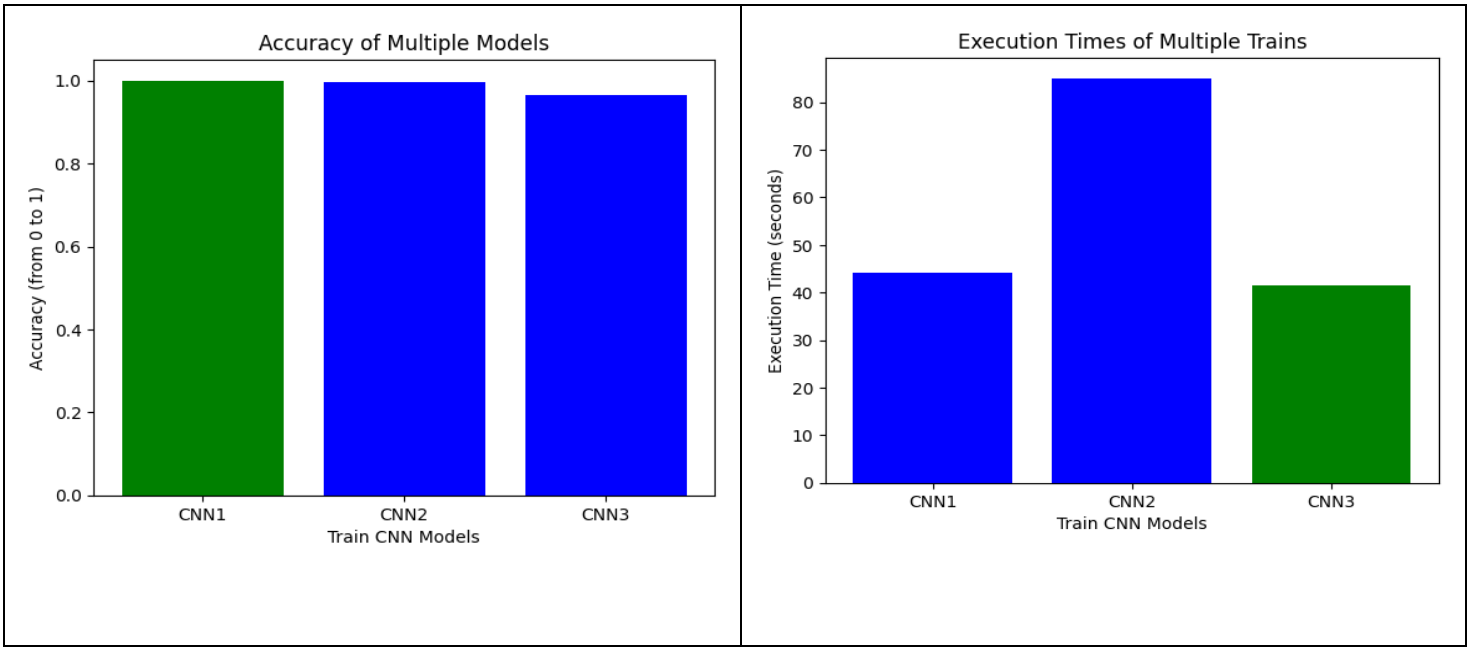


**Figure 4.3 – La précision de chaque model et le temps d'exécution.**

D'après les résultats, la perte de test représente la valeur de perte moyenne des prédictions du modèle sur l'ensemble de données de test. Des valeurs inférieures indiquent de meilleures performances, car cela signifie que les prédictions du modèle sont plus proches des valeurs réelles.

La précision du test représente le pourcentage d'échantillons correctement classés dans l'ensemble de données de test. Des valeurs de précision plus élevées indiquent de meilleures performances, car cela signifie que le modèle fait des prédictions plus précises. Sur la base de ces résultats, il semble que le modèle 1 présente les meilleures performances, avec une perte de test très faible et une précision de test élevée. Le modèle 2 fonctionne également bien, mais a une précision légèrement inférieure à celle du modèle 1. Le modèle 3 a une perte de test plus élevée et une précision inférieure par rapport aux deux autres modèles, ce qui indique qu'il peut ne peut pas fonctionner aussi bien que les autres.

Il est important de noter que l'évaluation des performances d'un modèle uniquement basée sur la perte et la précision des tests peut ne pas fournir une compréhension complète de ses capacités. Il est recommandé de considérer d'autres métriques d'évaluation et également d'effectuer une validation croisée ou d'évaluer les performances du modèle sur des données invisibles pour obtenir une évaluation plus complète.

**Tableau 4.5 – La précision et perte du processus d'évaluation de chaque modèle CNN**

| Nom de modèle | Précision | Perte |
|---|---|---|
| CNN Model 1 | ~99.99% | ~0.10% |
| CNN Model 2 | ~99.73% | ~0.08% |
| CNN Model 3 | ~96.37% | ~2.71% |

### 4.4.4 Tester le meilleur modèle CNN :

La phase de test du modèle IDS basée sur CNN a joué un rôle déterminant dans l'évaluation de ses performances et la confirmation de son efficacité dans la détection des intrusions à l'aide de l'ensemble de données KDD99. Après une comparaison et une analyse minutieuses, le meilleur modèle parmi les trois candidats ont été sélectionnés en fonction de sa plus grande précision.

Par la suite, le modèle choisi a subi des tests rigoureux pour valider ses capacités. Trois tests spécifiques ont été menés pour évaluer l'efficacité du modèle dans la détection de divers types d'intrusion sur le réseau. Le premier test consistait à simuler une attaque "schtroumpf", qui a abouti à une détection réussie de l'intrusion par le modèle CNN. Dans le deuxième test, une attaque "Neptune" a été utilisée et le modèle a identifié avec précision l'intrusion, soulignant sa robustesse dans la détection de différents types d'intrusion.

Enfin, un test avec un trafic réseau normal a été effectué et le modèle CNN l'a reconnu avec succès comme non intrusif. Ces tests complets ont confirmé la fiabilité et l'efficacité du modèle IDS basé sur CNN, fournissant des preuves convaincantes de son inclusion dans notre étude en tant que solution précieuse pour la détection des intrusions à l'aide de l'ensemble de données KDD99.

**Tableau 4.6 – Les Test de meilleur modèle de CNN.**

| Nom de Test | Résultat détecté | VP | VN | FP | FN |
|---|---|---|---|---|---|
| **Test 1** | **Smurf** | **1** | **0** | **0** | **0** |
| **Test 2** | **Neptune** | **1** | **0** | **0** | **0** |
| **Test 3** | **Normal** | **0** | **1** | **0** | **0** |

## 4.5 Conclusion:

En conclusion, le chapitre "Conception et réalisation" s'est concentré sur le développement d'un IDS basé sur CNN en utilisant l'ensemble de données KDD99. Les principaux sujets abordés comprenaient l'établissement de l'environnement d'exécution, la sélection et la préparation des ensembles de données, la création de modèles CNN, la formation, l'évaluation et les tests.

Le chapitre a démontré l'efficacité du modèle CNN pour détecter avec précision les intrusions et différencier le trafic réseau normal. Ces résultats fournissent des informations précieuses pour de nouvelles recherches sur la détection des intrusions et contribuent aux progrès de la cybersécurité.

# Conclusion Générale

En conclusion, cette recherche sur les systèmes de détection d'intrusion réseau (IDS) a fourni des informations précieuses dans le domaine. L'étude a porté sur les concepts fondamentaux, les principes de l'apprentissage automatique et la conception, la mise en œuvre et l'évaluation d'un outil IDS.

Les résultats mettent en évidence le rôle essentiel que joue IDS dans l'atténuation et la réponse aux menaces potentielles dans les environnements réseau. En utilisant des techniques d'apprentissage automatiques, un outil IDS robuste a été développé, présentant des performances fiables dans la détection et l'alerte contre les intrusions.

Les recherches futures devraient se concentrer sur 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.

# Bibliographie

[1] Ni Gao, Ling Gao, Quanli Gao et Hai Wang, «An Intrusion Detection Model Based on Deep Belief Networks,» *In 2014 Second International Conference on Advanced Cloud and Big Data,* vol. IEEE, p. 247–252, 2014.

[2] L. R. TechTarge, «LeMagIT,» [En ligne]. Available: https://www.lemagit.fr/definition/Systeme-de-detection-dintrusions#:~:text=Un%20syst%C3%A8me%20de%20d%C3%A9tection%20d,de%20compromettre%20son%20r%C3%A9seau%20informatique.. [Accès le 16 07 2023].

[3] Maxime Labonne, «Anomaly-based network intrusion detection using machine learning, Cryptography and Security [cs.CR],» 2020. [En ligne]. Available: https://theses.hal.science/tel-02988296. [Accès le 16 07 2023].

[4] Wood , Mark et Erlinger, Michael, «Intrusion Detection Message Exchange Requirements,» Harvey Mudd College, 03 2007. [En ligne]. Available: https://tools.ietf.org/html/rfc4766. [Accès le 27 12 2022].

[5] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Li, «Intrusion detection system : A comprehensive review,» *Jornal of Network and Computer Applications,* pp. 16-24, 2013.

[6] Chiba Zouhair, Noreddine Abghour, Khalid Moussaid, Amina El Omri et Mohamed Rida, *A review of intrusion detection systems in cloud computing. In Security and Privacy in Smart Sensor Networks,* Casablanca,: IGI Global, 2018, pp. 253-283.

[7] Hervé Debar, Marc Dacier et Andreas Wespi, «A revised taxonomy for intrusion detection systems,» *In Annales des télécommunications,* vol. 55, pp. 361-378 Springer, 2000.

[8] Liran Lerman, Olivier Markowitch et Gianluca Bontempi, *Les systèmes de détection d'intrusion basés sur du machine learning,* 2008.

[9] Nassima Chaibi, Baghdad Atmani et Mostéfa Mokaddem, «A Convolutional Neural Network With Feature Selection Based Network Intrusion Detection,» *1st National Conference on Applied Computing and Smart Technologies ACST'21,* pp. 12-13, 2021.

[10] Nicholas Carlini et David Wagner, «Towards evaluating the robustness of neural,» *In 2017 ieee symposium on security and privacy (sp),* pp. 39-57, 2017.

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

[12] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu et Fuad E Alsaadi, «A survey of deep neural network architectures and their applications,» chez *Neurocomputing*, 234, 2017, pp. 11-26.

[13] Li Deng, Dong Yu et al, «Deep learning : methods and applications.,» *Foundations and Trends in Signal Processing,* pp. 197-387, 2014.

[14] Mohamed Amine Ferrag, Leandros Maglaras, Sotiris Moschoyiannis et Helge Janicke, «Deep learning for cyber security intrusion detection : Approaches, datasets, andc omparative study,» *Journal of Information Security and Applications,* p. 50, 2020.

[15] Hassan Hadi, Al-Maksousy, Michele C Weigle et Cong Wang, «Nids : Neural network based intrusion detection system,» *In 2018 IEEE International Symposium on Technologies for Homeland Security (HST),* vol. IEEE, pp. 1-6, 2018.

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

[17] Ian Goodfellow, Yoshua Bengio et Aaron Courville, «Deep Learning,» *MIT Press,* p. 326, 2016.

[18] N. G. a. V. C. Clément Dalloux, « Détection de la négation : corpus français et apprentissage supervisé,» *Revue des Sciences et Technologies de l'Information-Série TSI : Technique et Science Informatiques,* pp. 1-21, 2019.

[19] Pradeep BV, «What is the fundamental difference between CNN and RNN?,» [En ligne]. Available: https://ai.stackexchange.com/questions/4683/what-is-the-fundamental-difference-between-cnn-and-rnn. [Accès le 15 06 2023].

[20] Lukman Zaman, Surya Sumpeno et Mochamad Hariadi, «Analisis Kinerja LSTM dan GRU sebagai Model Generatif,» vol. 8, p. 2, 2019.

[21] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai et al, «Recent advances in convolutional neural networks,» *Pattern Recognition,* vol. 77, p. 354–377, 2018.

[22] Djallel Hamouda, «Intrusion Detection System for Cyber Security,» [En ligne]. Available: https://www.researchgate.net/figure/Larchitecture-dun-modele-de-reseau-neuronal-convolutif_fig4_367053422. [Accès le 15 06 2023].

[23] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, «Imagenet classification with deep convolutional neural networks,» *In Advances in neural information processing systems,* pp. 1097-1105, 2012.

[24] Kwangjo Kim, Muhamad Erza Aminanto et Harry Chandra Tanuwidjaja, *Network Intrusion Detection Using Deep Learning : A Feature Learning Approach. Springer,* 2018.

[25] Soman KP, Mamoun Alazab et al., «A comprehensive tutorial and survey of applications of deep learning for cyber security,» 2020.

[26] Ansam Khraisat, Iqbal Gondal, Peter Vamplew et Joarder Kamruzzaman, «Survey of intrusion detection systems : techniques, datasets and challenges,» *Cybersecurity,* p. 2, 2019.

[27] «kddcup99 dataset,» Irvine Univ of California, [En ligne]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Accès le 27 12 2022].

[28] «Cic datasets,» Univ of new Brunswick, [En ligne]. Available: https://www.unb.ca/cic/datasets/.html,. [Accès le 22 12 2022].

[29] «MAWI dataset,» MAWILab, [En ligne]. Available: http://www.fukuda-lab.org/mawilab/data.html. [Accès le 22 12 2022].

[30] «Bot iot dataset,» CIC, [En ligne]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php. [Accès le 22 12 2022].

[31] Kehe Wu, Zuge Chen et Wei Li, «A novel intrusion detection model for a massive network using convolutional neural networks,» *IEEE Access, 6,* p. 50850–50859, 2018.

[32] Jihyun Kim et Howon Kim, «Applying recurrent neural network to intrusion detection with hessian free optimization,» *In International Workshop on Information Security Applications,* vol. Springer, p. 357–369, 2015.

[33] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi et Mounir Ghogho, «Deep learning approach for network intrusion detection in software defined networking,» *In 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM),* vol. IEEE, p. 258–263, 2016.

[34] « The World's Most Popular Data Science Platform,» Anaconda , [En ligne]. Available: https://www.anaconda.com/. [Accès le 01 05 2023].

[35] E. Bisong, «Google colaboratory,» chez *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Springer, 2019, pp. 59-64.

[36] J. Hale, «Deep Learning Framework Power Scores 2018,» [En ligne]. Available: https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a. [Accès le 2 5 2023].

[37] 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.

[38] 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].

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

[40] Karen Scarfone et Peter Mell, «Guide to intrusion detection and prevention systems,» *Technical report, National Institute of Standards and Technology,* 2012.