République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Larbi Tébessi - Tébessa
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département : Mathématiques et Informatique

Mémoire de fin d'études
Pour l'obtention du diplôme de MASTER
Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Réseaux et sécurité informatique

Thème

Pour une dètection intelligente de télé intrusion

Présenté Par :
Melaoui taki eddine

Devant le jury :

Dr Sahraoui abdelatif	*MCB*	*Univérsité Larbi Tébessi*	*Président*
Dr Gahmousse Abdellatif	*MAA*	*Univérsité Larbi Tébessi*	*Examinateur*
Dr Souahi Med Salah	*MCB*	*Univérsité Larbi Tébessi*	*Encadreur*

Date de soutenance : *11/07/2021*


بسم الله الرحمن الرحيم

REMERCIEMENTS

Je voudrais tout d'abord exprimer mes plus profonds remerciements à mon encadreur ***Dr.Souahi Mohamed Salah*** *pour son accord d'être mon directeur de mémoire et de sa disponibilité et son aide pendant toute la préparation de ce travail.*

Je tiens aussi à remercier tous les membres du jury : ***Dr. Sahraoui Abdelatif*** *et* ***Dr.Gahmousse Abedelatif****, pour leur disponibilité et acceptation d'examiner et de rapporter mon travail.*

Mes plus profonds remerciements à ***Dr. Amroun Mohamed*** *pour ses conseils pour la préparation de ce travail.*

Je remercie ainsi tous les enseignements du département Mathématiques et informatique.

Je ne saurais oublier de remercier toute ma famille pour leurs soutien moral, leurs encouragements et leurs patience durant les étapes de réalisation de ce travail.

DÉDICACES

À tous ceux qui directement ou indirectement m'ont apporté leurs aide.

ملخص

تتطور الحوسبة السحابية بسرعة وأصبحت أكثر شيوعًا في حياتنا اليومية وتنمو يومًا بعد يوم ، فهي بحد ذاتها تنطوي على مخاطر. لذا فأنت بحاجة إلى النموذج الأكثر فاعلية لاكتشاف الأنشطة الضارة بأسرع ما يمكن وبدقة. في هذا العمل، استخدمنا شبكة عصبية عميقة لتحديد الهجمات في نظام سحابي.

لا يمكن بناء نظام كشف التسلل الذكي إلا إذا كانت هناك مجموعة بيانات فعالة. تم تقييم أداء الشبكة العصبية العميقة المصممة لتحديد الهجوم بشكل صحيح على عدة مجموعات بيانات هي الأحدث والأكثر استخدامًا. اختبرنا 5 طبقات مخفية مع 1024 و 512 و 64 و 32 خلية عصبية متتالية، ووظيفة التنشيط قراءة لجميع الطبقات ، وحجم الدُفعة ثابت عند 1024 وأيضًا عدد العصور 100 عصر ، من أجل تحسين أداء النموذج من حيث من حيث الوقت والكفاءة.

أظهرت نتائجنا التجريبية معدل دقة الطريقة المقترحة باستخدام شبكة عصبية عميقة. يظهر أن الدقة أكبر من 95٪ في كل مجموعة بيانات.

الكلمات الرئيسية: الحوسبة السحابية ، التعلم الآلي ، التعلم العميق ، الشبكة العصبية العميقة (DNN) ، NSl-KDD، UNSW-NB15 ، KDD Cup'99

ABSTRACT

Cloud computing is evolving rapidly and becoming more and more popular in our daily life and growing day by day, it itself carries a risk. So the most efficient model was needed to detect malicious activity as quickly and accurately as possible. In our brief, we looked at the Deep Neural Network (DNN) to identify attacks in the cloud. An intelligent intrusion detection system can only be built if there is an effective data set. The performance of DNN in correctly identifying the attack was evaluated on the most recent and most used NSL-KDD, KDD-Cup'99 and UNSW-NB15 datasets. Our experimental results showed the accuracy rate of the proposed method using DNN. It shows that the precision is greater than 95% in each data set.

Key words : Cloud Computing, machine learning, deep learning, Deep Neural Network (DNN),NSl-KDD,UNSW-NB15, KDD Cup'99, DataSets.

RÉSUMÉ

Le cloud computing évolue rapidement et devient de plus en plus populaire dans notre vie quotidienne et se développe de jour en jour, il comporte lui-même un risque. Il faut donc le modèle le plus efficace pour détecter les activités malveillantes aussi rapidement et précisément que possible. Dans ce travail, nous avons utilisé un réseau de neurones profond (DNN) pour identifier les attaques dans un système cloud.

Un système de détection d'intrusion intelligent ne peut être construit que s'il existe un ensemble de données efficace. Les performances de DNN conçu pour identifier correctement l'attaque ont été évaluées sur plusieurs ensembles de données qui sont les plus récents et les plus utilisés. Nous avons testé 5 couches cachées avec 1024, 512, 64 et 32 neurones successive, fonction d'activation reLu pour toutes les couche, la taille de lot (batch size) est fixé 1024 et aussi le nombre d'époques (epochs) 100 époques, afin d'améliorer les performances du modèle que ce soit en temps ou en efficacité.

Nos résultats expérimentaux ont montré le taux de précision de la méthode proposée en utilisant DNN. Il montre que la précision est supérieure à 95% dans chaque ensemble de données.
Mots clés : Cloud Computing, machine learning, deep learning, Deep Neural Network (DNN),NSl-KDD,UNSW-NB15, KDD Cup'99, Ensemble de données.

TABLE DES MATIÈRES

LISTE DES ABRÉVIATIONS

AI Artificial Intelligence.

CAIDA Center for Applied Internet Data Analysis.

CSV Comma-separated values.

DL Deep Learning.

IDS Intrusion Detection System.

ML Machine Learning.

OSI Open Systems Interconnection.

PCA Principal Component Analysis.

PNL Natural Language Processing.

SQL Structured Query Language.

SSL Semi-Supervised Learning.

TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

De nos jours, les technologies émergentes rendent la vie plus confortable, mais leur sécurité et leur confidentialité sont compromises, et c'est un facteur de préoccupation important. Le Cloud Computing soulève de nombreuses vulnérabilités en termes de réseau, d'infrastructure, d'objets, de communication, etc. Comme il existe des millions d'appareils, il est difficile de mettre en œuvre la sécurité sur chaque appareil. La surveillance des données peut être réalisée par une sécurité basée sur le réseau. Des solutions de sécurité basées sur le réseau peuvent être mises en œuvre sur le Cloud avec des modifications mineures requises car il est nécessaire de surveiller tout le trafic entrant et sortant de chaque objet. Désormais, si une valeur ne tombe pas dans la catégorie de comportement normal, elle est identifiée comme attaque et déclenche l'alarme en tant que signal au propriétaire des appareils .

L'IDS (Intrusion Detection System) peut être utile pour assurer la sécurité du réseau utilisé pour observer le comportement anormal du réseau. Maintenant, l'essentiel est de savoir où placer un IDS dans le système ? Si un IDS est placé sur les nœuds ou distribué de manière aléatoire, il est alors appelé IDS basé sur le réseau. Si un IDS est placé sur des postes de travail, il est alors appelé IDS basé sur l'hôte. Nous pouvons combiner IDS avec certaines technologies d'apprentissage automatique afin d'obtenir des résultats plus précis, l'apprentissage automatique (ML) est la partie importante de l'IA qui est utilisée pour analyser et construire le système sur la base des connaissances acquises à partir des ensembles de données.

Il existe principalement trois types de techniques d'apprentissage basées sur l'utilisation de données étiquetées, à savoir l'apprentissage supervisé, non supervisé et semi-supervisé. Les algorithmes d'apprentissage automatique courants sont : Support Vector Machine (SVM), la régression logistique, le classificateur naïf-bayes, la régression linéaire, le voisin le plus proche (KNN), le réseau de neurones artificiels (ANN), etc.

L'apprentissage profond est également reconnu comme un apprentissage structurel ou hiérarchique profond. C'est l'algorithme vital de l'apprentissage automatique (ML) en termes de structure complexe et de vitesse d'apprentissage des données. L'apprentissage profond nécessite une

grande quantité de données, contrairement à ML, pour trouver les modèles plus précis. Un réseau conçu à l'aide de l'apprentissage en profondeur s'appelle réseau de neurones profond (DNN [1]). La principale différence entre un réseau de neurones artificiels et un réseau de neurones profonds est que si ANN [2] contient deux ou plusieurs couches cachées, il est alors nommé structure profonde. La vitesse de traitement des données serait rapide et apprendrait les tâches plus en profondeur. Dans notre mémoire, nous utilisons un réseau de neurones profond pour former, valider et tester le réseau en utilisant trois ensembles de données.

Dans ce contexte et dans le cadre de notre projet de fin d'étude, nous avons organisé ce manuscrit de la façon suivante :

Le premier chapitre est consacré au contexte d'étude, nous allons présenter tout d'abord le cloud computing, sa définition, son architecture et ses avantages. Ensuite, nous allons démontrer les problèmes de sécurité du Cloud Computing en précisant ces modèles de déploiements et de livraison. Enfin nous allons terminer avec ces vulnérabilités générales et les Contre-mesures à concevoir pour atténuer les risques en fonction de leur évaluation.

Dans **le deuxième chapitre**, nous allons présenter tout d'abord Le processus général du système de détection d'intrusion, sa définition, ces types, ces fonctions et ces méthodes. Ensuite en démontre les différents ensembles de données utilisés par les systèmes de détection d'intrusion. Enfin, nous terminerons avec les pré-traitements des données qui est une étape essentielle.

Le troisième chapitre, nous définirons tout d'abord l'apprentissage automatique, ses différents types, ainsi que ses modèles les plus utilisés. Ensuite nous introduisons l'apprentissage profond en commençant par ses définitions, son histoire, ses différents modèles. Enfin nous conclurons avec ses métriques d'évaluation.

Le quatrième chapitre, nous présentons notre contribution, en commençant par l'architecture de notre système, dans cette partie nous détaillons l'architecture proposée. Ensuite nous définissons l'architecture de notre modèle, présentation des ensembles de données utilisés et les pré-traitements effectués. Enfin nous conclurons avec l'implémentation en mettant l'accent sur l'évaluation de notre modèle, l'environnement de développement et les outils utilisés lors de la création de notre système.

Enfin, Nous conclurons ce manuscrit en présentant quelques perspectives ouvertes par notre travail.

1. Deep Neural Network
2. Artificial Neural Network

CHAPITRE 1

LA SÉCURITÉ DANS LES PLATEFORMES DU CLOUD COMPUTING (CC)

1.1 Introduction

Le cloud computing en raison de sa nature perturbatrice, de son architecture complexe et de ses ressources à effet de levier posent un risque unique et grave. Il est essentiel que toutes les parties prenantes et acteurs comprennent le risque et l'atténuent de manière appropriée. Dans ce premier chapitre nous allons présenter tous d'abord le cloud computing, ça définition, son architecture et ces avantages. Ensuite, nous allons démontrer les problémes de sécurité du Cloud Computing en précisant ces modèles de déploiements et de livraison. Enfin nous allons terminer avec ces vulnérabilités générales et les Contre-mesures a concevoir pour atténuer les risques en fonction de leur évaluation.

1.2 Le Cloud Computing(CC)

Avant de plonger dans les problèmes de sécurité, il est important de comprendre la définition et l'architecture du cloud, Dans ce qui suit nous allons présenter la définition et l'architecture de ce dernier.

1.2.1 Définition :

Selon Sharma et Trivedi[46], le cloud computing est un ensemble de ressources qui peuvent évoluer à la demande. Il est disponible sur Internet dans un modèle de libre-service avec peu ou pas d'interaction requise avec le fournisseur de services. Le cloud permet de nouvelles façons d'offrir des produits et services avec des opportunités innovantes, techniques et tarifaires.

1.2.2 Architecture du cloud Computing :

Il y a cinq acteurs majeurs qui influencent et sont impactés par le cloud computing, ainsi que ses implications en matière de sécurité. Le Tableau suivant représente les acteurs de l'architecture de référence NIST Cloud Computing[17] :

Acteur	Définition
Consommateur cloud	Une personne ou une organisation qui entretient une relation commerciale avec un fournisseur de cloud et utilise le service de celui-ci.
Fournisseur cloud	Une personne, une organisation ou une entité chargée de mettre un service à la disposition des parties intéressées.
Auditeur Cloud	Une partie qui peut effectuer une évaluation indépendante des services cloud, des opérations du système d'information, des performances et de la sécurité de la mise en œuvre cloud
Courtier Cloud	Une entité qui gère l'utilisation, les performances et la fourniture de services cloud et négocie les relations entre les fournisseurs cloud et les consommateurs cloud
Carrier Cloud	Un intermédiaire qui fournit la connectivité et le transport des services cloud des fournisseurs cloud aux consommateurs cloud

TABLE 1.1 – les acteurs de l'architecture de référence NIST Cloud Computing[17].

La figure 1.1 est une architecture de référence complète pour le cloud computing. Il est important de noter que la figure représente une architecture de référence de bout en bout qui aborde les sept couches du modèle d'interconnexion des systèmes ouverts (OSI) et s'étend pour inclure les aspects commerciaux, le cloud computing est une solution complète et complexe avec de nombreux domaines de vulnérabilités.

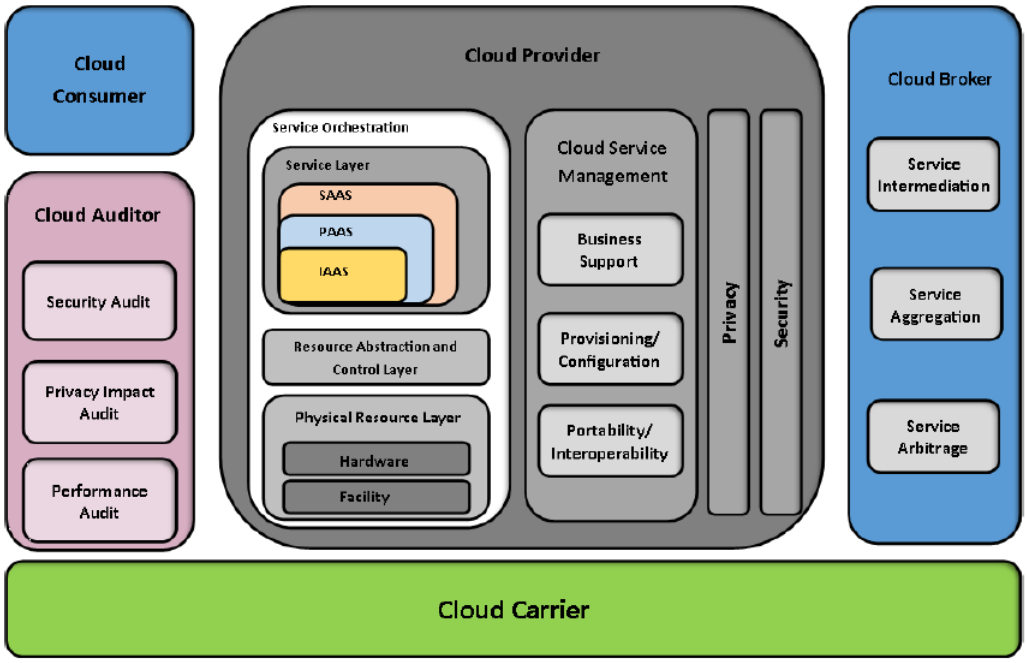


FIGURE 1.1 – Architecture de référence NIST Cloud Computing[17].

1.2.3 Les avantages du cloud

Le cloud computing présente des avantages uniques. Certains des principaux avantages sont[15] :

1. Coût d'entrée pour toutes les organisations, y compris les petites entreprises.
2. Accès quasi immédiat aux ressources.
3. La réduction des obstacles à l'innovation IT.
4. Facilité de mise à l'échelle des services.
5. Mettre en œuvre et / ou proposer une nouvelle classe d'applications et de services de livraison.

1.3 Problèmes de sécurité dans la plateforme du Cloud Computing

Les deux aspects les plus importants qui déterminent le niveau de vulnérabilité dans une plateforme de cloud computing sont le choix du modèle de déploiement et de livraison. Trois modèles de déploiement et trois modèles de livraison sont considérés comme des normes de l'industrie. Chacun de ces trois modèles a des implications de sécurité uniques. Dans ce qui suit nous allons présenter brièvement chacun de ces modèles et leurs implications en matière de sécurité.

1.3.1 Modèles de déploiements cloud

Le modèle Cloud Computing comporte trois modèles de déploiement principaux qui sont[47]

Cloud Privé :

Dans un cloud privé, le fournisseur de services cloud regroupe des ressources évolutives et des applications virtuelles et les met à la disposition des consommateurs du cloud. Dans ce modèle de déploiement, les ressources sont dédiées à une seule ou à un ensemble d'organisations et traitées comme une fonctionnalité intranet. La facturation est généralement basée sur un abonnement avec un consommateur de cloud prenant des engagements minimaux.

Cloud Publique :

Dans un cloud public, les ressources sont engagées de manière dynamique sur une base fine et en libre-service sur Internet ou un portail10. La facturation est généralement basée sur la consommation et est facturée sur un paiement à l'utilisation base.

Cloud Hybride :

Le cloud hybride est un modèle de déploiement dans lequel un cloud privé est lié à un ou plusieurs services cloud externes tout en étant géré de manière centralisée. Il fournit aux consommateurs du cloud une solution flexible et adaptée à l'usage avec une relative facilité d'opérations. Les Clouds hybrides ont un degré de complexité plus élevé en termes de facturation et de publicités.

1.3.2 Modèle de livraison cloud

Les trois principaux modèles de prestation de services cloud sont :

Infrastructure en tant que service (IaaS) :

L'infrastructure en tant que service est une couche cloud multi-locataire où les ressources dédiées du fournisseur de services cloud sont uniquement partagées avec les clients sous contrat moyennant des frais d'utilisation. Cela signifie généralement que le système d'exploitation est présenté au consommateur du cloud. La responsabilité du fournisseur de services cloud prend fin avec le système d'exploitation.

Plateforme en tant que service (PaaS) :

Plateforme en tant que service est l'un des services de livraison les plus populaires où le fournisseur de cloud fournit non seulement le système d'exploitation mais également une pile de développement. Il est courant pour les fournisseurs de ce modèle de fournir une administration

de base de données et d'applications avec des services de développement. Tout comme dans IaaS, PaaS est un modèle de paiement à l'utilisation.

Logiciel en tant que service (SaaS) :

Dans un modèle logiciel en tant que service, la pile d'applications complète est hébergée par le fournisseur de cloud, qui fournit des ressources de bout en bout, y compris les licences, les applications, la mise en réseau,...etc. les services dans un service Web ou une architecture orientée logiciel.

1.3.3 Vulnérabilités générales et contre-mesures

Le cloud computing, comme d'autres domaines de l'informatique, souffre d'un certain nombre de problèmes de sécurité, qui doivent être résolus. Ces risques concernent les risques liés aux politiques et à l'organisation, les risques techniques et les risques juridiques et autres.

A) **Vulnérabilités et problèmes ouverts** Le cloud présente également des vulnérabilités. Voici quelques-unes dans le cloud et des problèmes et menaces en suspens qui nécessitent une attention urgente[43][49]

(a) **Scripts intersites (XSS) :** Dans ce genre d'assaut, un script nuisible est injecté dans le Web. Il existe deux stratégies d'attaque XSS : Stored XSS et Reflected XSS. Dans XSS stocké, le code nuisible est stocké en permanence dans un actif traité par application web. En revanche, dans Reflected XSS, le code nuisible n'est pas stocké en permanence. En effet, il est instantanément renvoyé au client.Le fonctionnement de XSS est comme indiqué dans la figure[50] :

FIGURE 1.2 – Fonctionnement de l'attaque XSS[50].

(b) **Déni de service (DoS) :**Le but d'une attaque par déni de service est de refuser aux utilisateurs légitimes l'accès à une ressource particulière. Lorsque la charge de travail élevée sur les services inondés est notifiée par le système d'exploitation Cloud Computing, il commence à fournir plus de puissance de calcul pour faire face à la charge de travail supplémentaire. Ainsi, les limites matérielles du serveur pour une charge de

travail maximale à traiter ne sont plus valables. En ce sens, le système Cloud tente de travailler contre l'attaquant (en fournissant plus de puissance de calcul), mais dans une certaine mesure, cela l'aidera en lui permettant de faire le plus de dégâts possibles sur la disponibilité d'un service, à partir d'un seul point d'entrée d'attaque par inondation. . Ainsi, l'attaquant n'a pas à inonder tous les n serveurs qui fournissent un certain service dans la cible, mais peut simplement inonder une seule adresse basée sur le Cloud afin d'effectuer une perte totale de disponibilité sur le service prévu[50].

(c) **Déni de service distribué (DDoS) :** La forme étendue d'attaque DoS est l'attaque DDoS, comme le montre la figure 1.3. Les attaques DDoS utilisent de nombreuses machines et connexions Internet. L'attaquant utilise un groupe d'agents pour envoyer à plusieurs reprises des commandes d'attaque DDoS au système cible. Un trafic soudain peut conduire à charger le site Web très lentement pour les utilisateurs prévus. Parfois, ce trafic est si élevé qu'il ferme complètement le site[50].

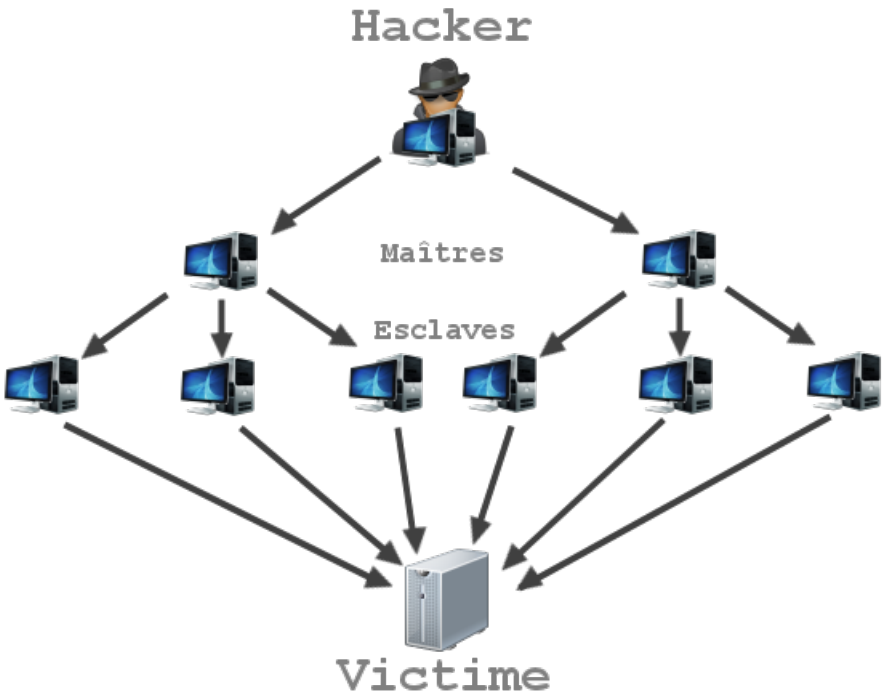


FIGURE 1.3 – Fonctionnement de DDOS[50].

(d) **Adresses IP réutilisées :** Dans ce cas, lorsqu'un utilisateur / client méticuleux s'éloigne de la couverture du réseau, alors l'adresse IP qui lui a été attribuée précédemment est attribuée à un nouvel utilisateur / client. Parfois, même si l'ancienne adresse IP est attribuée à un nouvel utilisateur, il existe encore des possibilités de récupérer les données par un autre utilisateur, car l'adresse toujours présente dans le cache DNS et les données appartenant à un utilisateur particulier peuvent devenir accessibles

à certains autre utilisateur enfreignant la vie privée de l'utilisateur précédent.

(e) **Man-in-the-Middle :** Dans une telle agression, un intrus tente de se mêler d'une discussion continue entre l'expéditeur et le destinataire pour infuser de fausses données et avoir connaissance de l'échange d'informations critiques entre eux, comme le montre la figure[50] :

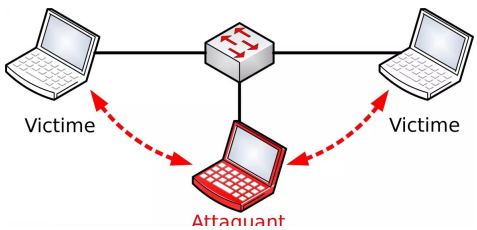


FIGURE 1.4 – Fonctionnement Man-in-the-Middle[50].

(f) **Attaques par injection SQL :**Un code SQL standard est rendu malveillant en injectant du code malveillant. Par conséquent, les attaquants peuvent accéder à des informations sensibles et obtenir un accès non autorisé à une base de données.

B) **Contre-mesures** Les vulnérabilités et les menaces dans le cloud sont bien documentées. Chaque fournisseur de services cloud et consommateur cloud doit concevoir des contre-mesures et des contrôles pour atténuer les risques en fonction de leur évaluation. Cependant, voici les contre-mesures pour les vulnérabilités décrites dans la sous section précédente qui peuvent être envisagées[49].

(a) **Attaques de script intersite (XSS) :** Diverses techniques telles que la technologie de détection de vulnérabilité des applications Web, le filtrage de contenu actif, la technologie de prévention des fuites de données basées sur le contenu, ont été proposées pour empêcher les attaques XSS. Ces techniques implémentent une variété de méthodologies pour identifier et corriger les failles de sécurité. Une approche basée sur des plans réduit la dépendance des navigateurs Web pour identifier le contenu non fiable sur le réseau.

(b) **Man-in-the-Middle :** La technologie de base du cloud computing est virtuelle, ce qui signifie que différentes données utilisateur peuvent être stockées dans un stockage physique partagé. Pour la mémoire partagée, la nécessité de parvenir à isoler les applications non approuvées en utilisant les ressources virtuelles pour limiter le comportement malveillant des applications non approuvées.

(c) **Cryptage des données :** Divers outils mettant en œuvre des technologies de cryptage fortes telles que Airjack, Ettercap, Dsniff, Cain, Wsniff, etc. ont été développés pour fournir une protection contre eux. Quelques-uns d'entre eux sont des processus distincts de sécurité des terminaux et des serveurs, évaluant le logiciel en tant que sécurité de service ; l'évaluation de la virtualisation au point final a été proposée pour lutter contre

cette attaque. Dans tous les cas, les pratiques de sécurité utilisées dans le réseau privé et le cloud privé de l'organisation. Cependant, dans le cas d'une implémentation de cloud public, la topologie du réseau doit être modifiée pour implémenter les fonctionnalités de sécurité.

(d) **Déni de service (DoS) :** L'utilisation d'un système de détection d'intrusion (IDS) est une méthode de protection courante contre ces attaques. Une fédération de défense est utilisée pour se prémunir contre de telles attaques. Chaque cloud est chargé avec des IDS distincts. L'échange d'informations est à la base du fonctionnement de différents systèmes de détection d'intrusion. L'ensemble du système est alerté en cas d'attaque d'un cloud particulier par l'IDS coopératif. Une décision sur la fiabilité d'un cloud est prise par vote et veille à ce que les performances globales du système ne soient pas entravées.

(e) **Déni de service distribué (DDoS) :** Une logique de groupe pour se protéger contre les attaques DDoS, l'utilisation de IIDS dans la machine virtuelle pour protéger le cloud contre les attaques DDoS. Un mécanisme de détection d'intrusion semblable à SNORT est implémenté sur la machine virtuelle pour renifler tous les trafics, qu'ils soient entrants ou sortants. Une autre méthode utilisée pour se prémunir contre les attaques DDoS consiste à implémenter des systèmes de détection d'intrusion sur toutes les machines physiques qui possèdent les machines virtuelles de l'utilisateur.

(f) **Adresses IP utilisées(Utilisation du gestionnaire de confidentialité basé sur le client) :** cela permet d'avoir plus de confidentialité des données sensibles et de réduire le risque de fuite de données et offre des avantages supplémentaires liés à la confidentialité dans le cloud. Les caractéristiques importantes du gestionnaire de confidentialité sont :

- Réglage des préférences, il s'agit d'une méthode permettant aux utilisateurs de définir leurs préférences concernant le changement de données personnelles.
- Accès aux données, il s'agit d'un module qui permet aux utilisateurs d'accéder aux informations personnelles dans le cloud, afin de voir ce qui est détenu à leur sujet et de vérifier leur exactitude.
- Le module de commentaires est utilisé pour gérer et afficher les commentaires à l'utilisateur concernant l'utilisation de ses informations personnelles, des personae qui permettent à l'utilisateur de choisir entre plusieurs personnes lors de l'interaction avec les services cloud.

(g) **Attaques par injection SQL :** Pour vérifier les attaques par injection SQL, des techniques de filtrage, etc. peuvent être utilisées pour nettoyer l'entrée de l'utilisateur. Une architecture basée sur un proxy peut être utilisée pour empêcher les attaques par injection SQL qui détecte et extrait dynamiquement les entrées de l'utilisateur pour les séquences de contrôle SQL suspectées a été proposée.

1.4 Conclusion

Bien que le cloud computing puisse être considéré comme un nouveau phénomène qui est en passe de révolutionner la façon dont nous utilisons Internet, il y a de quoi être prudent. De nombreuses nouvelles technologies émergent à un rythme rapide, chacune avec des progrès technologiques et avec le potentiel de rendre la vie humaine plus facile. Cependant, il faut être très prudent pour comprendre les risques de sécurité et les défis posés par l'utilisation de ces technologies.

Dans ce chapitre, nous avons présenté le cloud, sa définition, son architecture et ses avantages. Ensuite, nous avons démontré les problèmes de sécurité du Cloud Computing en précisant ces modèles de déploiements et de livraison. Enfin nous avons terminé avec ces vulnérabilités générales et les Contre-mesures à concevoir pour atténuer les risques en fonction de leur évaluation.

CHAPITRE 2

LES APPROCHES DE DÉTECTION D'INTRUSION DANS LES CLOUD COMPUTING

2.1 Introduction

De nos jours, les systèmes d'informations des entreprises subissent des différentes attaques qui peuvent entraîner des pertes conséquentes, vu leurs évolutions sur les plans d'échange d'informations d'une part et l'ouverture sur le monde extérieur d'autre part. Alors les systèmes de détection d'intrusion sont largement répandus pour la sécurité de ces systèmes informatiques puisqu' ils permettent à la fois de détecter et de répondre à une attaque en temps réel ou en hors-ligne. Dans ce chapitre nous allons présenter tout d'abord Le processus général du système de détection d'intrusion, sa définition, ces types, ces fonctions et ces méthodes. Ensuite en démontre les diffirents ensembles de données utilisés par les systèmes de détection d'intrusion. Enfin, nous terminerons avec les pré-traitements des données qui est une étape essentielle.

2.2 Structure générale d'un système de détection d'intrusion

La détection d'intrusion est définie comme le processus de surveillance des événements survenant dans un système informatique ou un réseau et leur analyse pour détecter des signes d'intrusions, définis comme des tentatives de compromettre la confidentialité, la disponibilité de l'intégrité ou de contourner le mécanisme de sécurité de l'ordinateur ou du réseau[48]. Les principaux composants de l'IDS sont illustrés à la figure 2.1.

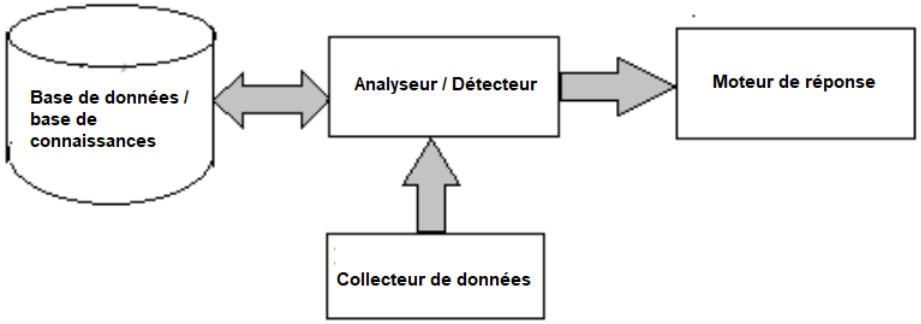


FIGURE 2.1 – Les principaux composants d'un IDS[48].

— **Collecte de données :** Le composant de collecte de données est chargé de collecter et de fournir les données d'audit qui seront utilisées par le prochain composant pour prendre des décisions. Les données utilisées pour détecter les intrusions vont du modèle d'accès utilisateur aux fonctionnalités au niveau des paquets réseau.

— **Analyseur (détecteur d'intrusion) :** L'analyseur ou le détecteur d'intrusion est le composant central qui analyse les modèles d'audit pour détecter les attaques. Il s'agit d'un élément essentiel et l'un des plus étudiés. Diverses techniques sont utilisées comme détecteurs d'intrusion.

— **Profil système (base de données ou base de connaissances) :** Le profil système est utilisé pour caractériser le comportement normal et anormal. C'est la base de connaissances pour les attaques, les informations de configuration sur l'état actuel du système et les informations d'audit décrivant les événements qui se produisent sur le système.

— **Moteur de réponse :** Le moteur de réponse contrôle le mécanisme de réaction et détermine comment réagir. Le système peut déclencher une alarme et signaler à l'administrateur ou peut bloquer la source de l'attaque.

2.2.1 Types de systèmes de détection d'intrusion (IDS)

La figure 2.2 montre les différents types de systèmes de détection d'intrusion.

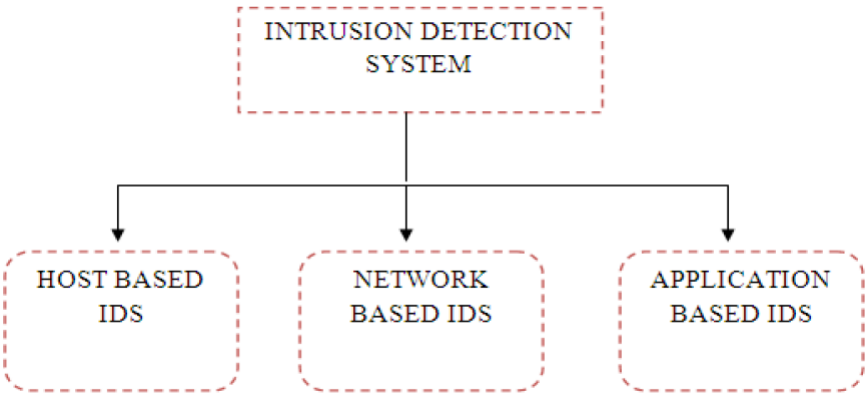


FIGURE 2.2 – Types de systèmes de détection d'intrusion (IDS)[28].

Système de détection d'intrusion basé sur l'hôte(HIDS)

HIDS a été le premier type développé de détection d'intrusion. HIDS surveille et analyse le système informatique interne ou les activités au niveau du système d'un hôte unique, telles que : configuration du système, activité de l'application, trafic du réseau sans fil (uniquement pour cet hôte) ou interface réseau, journaux système ou journal d'audit, exécution de processus utilisateur ou d'application, fichier accès et modification. Les capacités de HIDS incluent la vérification d'intégrité, la corrélation d'événements, l'analyse des journaux, l'application des politiques, la détection des rootkits, l'utilisation du processeur, de la mémoire, du disque dur et de la batterie, et les alertes[28].

Système de détection d'intrusion basé sur le réseau(NIDS)

les systèmes collectent les informations du réseau lui-même plutôt que de chaque hôte distinct. Le NIDS Détecte les attaques du réseau pendant que les paquets se déplacent sur le réseau. Les capteurs réseau sont équipés de signatures d'attaque qui sont des règles sur ce qui constituera une attaque et la plupart des systèmes basés sur le réseau permettent auHIDS a été le premier type développé de détection d'intrusion. HIDS surveille et analyse le système informatique interne ou les activités au niveau du système d'un hôte unique, telles que : configuration du système, activité de l'application, trafic du réseau sans fil (uniquement pour cet hôte) ou interface réseau, journaux système ou journal d'audit, exécution de processus utilisateur ou d'application, fichier accès et modification. Les capacités de HIDS incluent la vérification d'intégrité, la corrélation d'événements, l'analyse des journaux, l'application des politiques, la détection des rootkits, l'utilisation du processeur, de la mémoire, du disque dur et de la batterie, et les alertes [24, 25]x utilisateurs avancés de définir leurs propres signatures. L'attaque sur le capteur est basée sur la signature et elles proviennent des attaques précédentes et le fonctionnement des moniteurs sera transparent pour les utilisateurs et cela est également important[28].

Système de détection d'intrusion basé sur une application(APIDS)

(APIDS) vérifiera le comportement effectif et l'événement du protocole. Le système ou l'agent est placé entre un processus et un groupe de serveurs qui surveille et analyse le protocole d'application entre les périphériques. Les attaques intentionnelles sont les attaques malveillantes menées par des employés mécontents pour nuire à l'organisation et les attaques non intentionnelles causent des dommages financiers à l'organisation en supprimant le fichier de données important. De nombreuses attaques ont eu lieu dans la couche OSI[28].

2.2.2 Fonctions des IDS

L'IDS se compose de quatre fonctions clés à savoir, la collecte de données, la sélection des caractéristiques, l'analyse et l'action, qui sont données dans la figure 2.3.

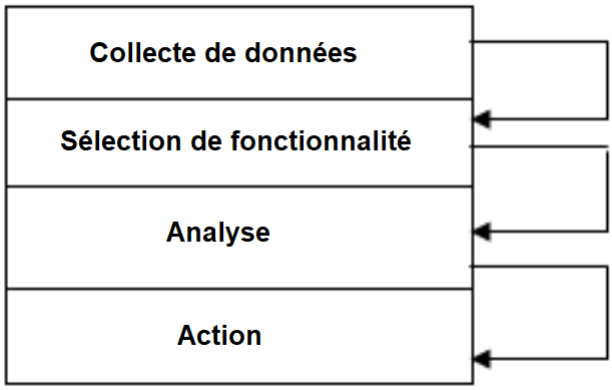


FIGURE 2.3 – Fonctions des IDS[28].

1. **La collecte de données :** Ce module transmet les données en entrée à l'IDS. Les données sont enregistrées dans un fichier puis analysées. L'IDS basé sur le réseau collecte et modifie les paquets de données et dans l'IDS basé sur l'hôte recueille des détails comme l'utilisation du disque et les processus du système[28].
2. **La sélection des caractéristiques :** Sélectionner la particularité des données volumineuses qui sont disponibles dans le réseau. Par exemple, l'adresse IP (Internet Protocol) du système source et de destination, le type de protocole, la longueur et la taille de l'en-tête peuvent être considérés comme une clé pour la sélection d'intrusion.[28].
3. **Analyse :** Les données sont analysées pour trouver l'exactitude. Les IDS basés sur des règles analysent les données où le trafic entrant est vérifié par rapport à une signature ou un modèle prédéfini[28].
4. **Action :** Il définit l'attaque et la réaction du système. Il peut soit informer l'administrateur système avec toutes les données requises via des icônes de courrier électronique / d'alarme,

soit jouer un rôle actif dans le système en abandonnant des paquets afin qu'il n'entre pas dans le système ou ne ferme pas les ports[28].

2.2.3 Les méthodes de détection d'intrusion

Diverses techniques sont en place pour la détection d'intrusions qui peuvent être classées de manière générale en détection basée sur : les signatures / modèles ,utilisation abusive, les connaissances et détection basée sur les anomalies et les comportements. Les deux méthodes ont leurs propres avantages et inconvénients.

Détection d'intrusion basée sur une signature

utilise un modèle bien défini d'attaques qui exploitent les faiblesses du système et du logiciel d'application pour identifier les intrusions. Ces systèmes détectent les intrusions sur la base d'un modèle d'activité malveillante. Il est très utile pour la détection des schémas d'attaque connus, des vulnérabilités connues du système. Ce système compare l'activité du réseau / système avec les signatures connues ou d'autres indicateurs de mauvaise utilisation pour produire des alarmes. Le taux de rapport manquant est élevé. Des mises à jour régulières des signatures sont nécessaires. Une signature partielle peut indiquer une tentative d'intrusion. Les exemples incluent Haystatck, Bro, IDES et Discovery, etc[11].

Détection d'intrusion basée sur les anomalies

Utilise les modèles de comportement d'utilisation normaux pour identifier les intrusions et est formé en utilisant le modèle comportemental normal du trafic. Détecte les activités malveillantes en fonction des écarts par rapport au comportement normal sont considérés comme des attaques. Il peut détecter des intrusions inconnues. Le taux de rapport manquant est faible. Ces systèmes créent un modèle basé sur les données normales, puis vérifient si les données s'inscrivent dans le modèle. Ce système peut détecter des attaques inconnues. Mais le taux et la précision des fausses alarmes sont faibles par rapport à l'approche basée sur la signature. Le système de détection d'anomalies peut utiliser des techniques d'apprentissage supervisé ou non supervisé. Les exemples sont IDES, NIDES et EMERALD etc. Le domaine et la nature des anomalies changent avec le temps et les intrus adaptent leurs attaques réseau pour échapper aux solutions de détection d'intrusion existantes[11].

Méthodes de détection	Avantages	Inconvénients
Détection basée sur les signatures	1) est capable de détecter avec précision. 2) Génère beaucoup moins de fausses alarmes.	1) Impossible de détecter des attaques nouvelles ou inconnues.
Détection d'anomalies	1) Est capable de détecter des attaques nouvelles / inconnues sur la base de l'audit 2) Moins dépendant des mécanismes spécifiques du système d'exploitation 3) Peut détecter les abus de privilèges 4) Taux de fausses alarmes élevé	1) Fausse alarme élevée et limitée par les données d'entraînement. 2) L'ensemble du champ du comportement n'est généralement pas couvert pendant la phase d'apprentissage. 3) Changement de comportement au fil du temps, entraînant de mauvaises performances du système.

TABLE 2.1 – Avantages et inconvénients des méthodes de détection d'intrusion[11].

2.3 Les ensembles de données et de test (DataSets)

Les ensembles de données d'évaluation jouent un rôle essentiel dans la validation de toute approche IDS, en nous permettant d'évaluer la capacité de la méthode proposée à détecter les comportements intrusifs. Les ensembles de données utilisés pour les paquets réseau et les analyses dans les produits commerciaux ne sont pas facilement disponibles en raison de problèmes de confidentialité. Cependant, il existe quelques ensembles de données accessibles au public tels que DARPA, KDD, NSL-KDD et ADFA-LD et ils sont largement utilisés comme références.

Les ensembles de données existants qui sont utilisés pour la construction et l'évaluation comparative de l'IDS sont traités dans cette section avec leurs caractéristiques et leurs limites.

2.3.1 DARPA / KDD Cup99

Le premier effort pour créer un ensemble de données IDS a été fait par la DARPA (Defense Advanced Research Project Agency) en 1998 et ils ont créé KDD98 (Knowledge Discovery and Data Mining (KDD)). En 1998, la DARPA a lancé un programme au MIT Lincoln Labs pour fournir un environnement d'analyse comparative IDS complet et réaliste. Bien que cet ensemble de données ait été une contribution importante à la recherche sur l'IDS, sa précision et sa capacité à

prendre en compte les conditions réelles ont été largement critiquées. Les paquets réseau collectés étaient d'environ quatre gigaoctets contenant environ 4 900 000 enregistrements. Les données de test de 2 semaines avaient environ 2 millions d'enregistrements de connexion, dont chacun avait 41 caractéristiques et était classé comme normal ou anormal[33].

2.3.2 L'ensemble de données CAIDA

Cet ensemble de données contient des traces de trafic réseau provenant d'attaques par déni de service distribué (DDoS) et a été collecté en 2007. Ce type d'attaque par déni de service tente d'interrompre le trafic normal d'un ordinateur ou d'un réseau ciblé en submergeant la cible d'un flot de paquets réseau, empêchant le trafic régulier d'atteindre son ordinateur de destination légitime. L'un des inconvénients du CAIDA est qu'il ne contient pas une diversité d'attaques. De plus, les données collectées ne contiennent pas de caractéristiques de l'ensemble du réseau, ce qui rend difficile la distinction entre les flux de trafic anormaux et normaux[33].

2.3.3 NSL-KDD

NSL-KDD est un ensemble de données public, qui a été développé à partir du l'ensemble de données antérieur KDD cup99. Une analyse statistique effectuée sur l'ensemble de données cup99 a soulevé des problèmes importants qui influent fortement sur la précision de la détection des intrusions et aboutit à une évaluation trompeuse du SIDA. Le principal problème dans l'ensemble de données KDD est l'énorme quantité de paquets en double.

Cette énorme quantité d'instances dupliquées dans l'ensemble de données influencerait les méthodes d'apprentissage automatique pour qu'elles soient biaisées vers des instances normales et les empêcherait ainsi d'apprendre des instances irrégulières qui sont généralement plus dommageables pour le système informatique. Tavallaee et coll[33] ont construit l'ensemble de données NSL KDD en 2009 à partir de l'ensemble de données KDD-Cup99. L'ensemble de données de train NSL-KDD se compose de 125973 enregistrements et l'ensemble de données de test contient 22544 enregistrements. La taille de l'ensemble de données NSL-KDD est suffisante pour rendre pratique l'utilisation de l'ensemble de données NSL-KDD sans qu'il soit nécessaire d'échantillonner au hasard. Cela a produit des résultats cohérents et comparables à partir de divers travaux de recherche. L'ensemble de données NSL-KDD comprend 22 attaques d'intrusion d'entraînement et 41 attributs (c'est-à-dire des fonctionnalités). Dans cet ensemble de données, 21 attributs font référence à la connexion elle-même et 19 attributs décrivent la nature des connexions au sein du même hôte[33].

2.3.4 ADFA-LD et ADFA-WD

Les chercheurs de l'académie des forces de défense australiennes ont créé deux ensembles de données (ADFA-LD et ADFA-WD) en tant qu'ensembles de données publics qui représentent

la structure et la méthodologie des attaques modernes. Les ensembles de données contiennent des enregistrements des systèmes d'exploitation Linux et Windows ; ils sont créés à partir de l'évaluation du HIDS basé sur les appels système. Ubuntu Linux version 11.04 a été utilisé comme système d'exploitation hôte pour construire ADFA-LD.

les instances d'attaque dans ADFA-LD ont été dérivées de nouveaux logiciels malveillants zero-day [1], ce qui rend cet ensemble de données approprié pour mettre en évidence les différences entre les approches SIDS et SIDA en matière de détection d'intrusion. Il comprend trois catégories de données différentes, chaque groupe de données contenant des traces d'appels système bruts. Chaque ensemble de données de formation a été collecté auprès de l'hôte pour des activités normales, avec des comportements des utilisateurs allant de la navigation Web à la préparation de documents LATEX[33].

2.4 Les pré-traitements

Lorsqu'il s'agit de créer un modèle d'apprentissage machine ou profond, le pré-traitement des données est la première étape marquant le lancement du processus. En règle générale, les données du monde réel sont incomplètes, incohérentes, inexactes (contiennent des erreurs ou des valeurs aberrantes) et manquent souvent de valeurs / tendances d'attributs spécifiques. C'est là que le pré-traitement des données entre dans le scénario. Le pré-traitement des données est une étape cruciale qui permet d'améliorer la qualité des données afin de promouvoir l'extraction d'informations significatives à partir des données. Le pré-traitement des données fait référence à la technique de préparation (nettoyage et organisation) des données brutes pour les rendre adaptées à la formation de modèles d'apprentissage machine et profond. En termes simples, le pré-traitement des données est une technique d'exploration de données qui transforme les données brutes en un format compréhensible et lisible.

2.4.1 Étapes impliquées dans le pré-traitement des données

Pour garantir des données de haute qualité, il est essentiel de les pré-traiter. Pour faciliter le processus, le pré-traitement des données est divisé en trois étapes : intégration des données, réduction des données et transformation des données.

Intégration de données

Étant donné que les données sont collectées à partir de sources multiples, l'intégration des données est devenue une partie vitale du processus. Cela peut conduire à des données redondantes et incohérentes, ce qui peut entraîner une précision et une vitesse médiocres du modèle de données. Pour traiter ces problèmes et maintenir l'intégrité des données, des approches telles que

1. .zero-day ou 0-day (en anglais zero-day vulnerability) désigne une faille de sécurité informatique dont l'éditeur du logiciel ou le fournisseur de service n'a pas encore connaissance, ou qui n'a pas encore reçu de correctif.

la détection de duplication de tuple et la détection de conflits de données sont recherchées. Les approches les plus courantes pour intégrer les données sont expliquées ci-dessous[39] :

1. **Consolidation des données :** Les données sont physiquement collecté dans un seul ensemble de données. Cela implique généralement l'entreposage de données(Data Warehousing [2]).
2. **Propagation des données :** La copie de données d'un emplacement à un autre à l'aide d'applications est appelée propagation de données. Il peut être synchrone ou asynchrone et est piloté par les événements.
3. **Virtualisation des données :** Une interface est utilisée pour fournir une vue unifiée et en temps réel des données provenant de plusieurs sources. Les données peuvent être consultées à partir d'un point d'accès unique.

Réduction de donnée

Le but de la réduction des données est d'avoir une représentation condensée de l'ensemble de données qui est plus petit en volume, tout en conservant l'intégrité de l'original. Il en résulte des résultats efficaces mais similaires. Quelques méthodes pour réduire le volume de données sont[39] :

1. **rapport des valeurs manquantes :** Les attributs qui ont plus les valeurs manquantes à un seuil sont supprimés.
2. **Filtre à faible variance :** Les attributs normalisés dont la variance (distribution) est inférieure à un seuil sont également supprimés, car peu de changements dans les données signifient moins d'informations.
3. **Filtre de corrélation élevée :** Les attributs normalisés qui ont un coefficient de corrélation supérieur à un seuil sont également supprimés, car des tendances similaires signifient que des informations similaires sont transmises. Le coefficient de corrélation est généralement calculé à l'aide de méthodes statistiques telles que la valeur chi-carré de Pearson, etc.
4. **Analyse en composantes principales :** L'analyse en composantes principales, ou ACP, est une méthode statistique qui réduit le nombre d'attributs en regroupant des attributs hautement corrélés. A chaque itération, les caractéristiques initiales sont réduites à des composantes principales, avec une plus grande variance que l'ensemble d'origine à la condition qu'elles ne soient pas corrélées avec les composantes précédentes. Cette méthode, cependant, ne fonctionne que pour les entités avec des valeurs numériques.

Transformation des données

La dernière étape du pré-traitement des données consiste à transformer les données en une forme appropriée pour la modélisation des données, les stratégies qui permettent la transformation

2. est un processus de collecte et de gestion de données provenant de sources variées afin de fournir des informations commerciales significatives.

des données comprennent[39] :

1. **Construction d'attributs / d'entités :** Les nouveaux attributs sont construits à partir de l'ensemble d'attributs donné.
2. **Agrégation :** Les opérations de résumé et d'agrégation sont appliquées à l'ensemble d'attributs donné pour créer de nouveaux attributs.
3. **Normalisation :** Les données de chaque attribut sont mises à l'échelle entre une plage plus petite, par ex. 0 à 1 ou -1 à 1.
4. **Discrétisation :** Les valeurs brutes des attributs numériques sont remplacées par des intervalles discrets ou conceptuels, qui peuvent en retour être organisés en intervalles de niveau supérieur.

2.5 Conclusion

Le système de détection d'intrusion est l'un des considérations les plus essentielles de la cybersécurité qui permet de découvrir une intrusion avant et / ou après l'attaque. Il joue un rôle important en tant que mécanisme de défense des réseaux et des systèmes. Nous avons présenté dans ce chapitre le processus général du système de détection d'intrusion, sa définition, ces types, ces fonctions et ces méthodes. Ensuite nous avons démontré les diffirents ensembles de données utilisés par les systèmes de détection d'intrusion. Après, nous avons décrit les pré-traitements des données qui est une étape essentielle. Enfin en a terminé ce chapitre par l'apprentissage automatique et profond en précisant les types d'apprentissage automatique et les différents modèles d'apprentissage automatique et profond, les métriques d'évaluations de ces derniers. Dans le chapitre suivant nous allons proposer une nouvelle approche qui utilise l'une des techniques présenté ci-dessous d'apprentissage profond pour détecter les intrusions.

CHAPITRE 3

APPROCHES BASÉES SUR L'APPRENTISSAGE AUTOMATIQUE ET PROFOND

3.1 Introduction

L'Intelligence Artificielle est un domaine très vaste, où nous essayons d'imiter le comportement humain dans le but de rendre les machines si puissantes pour accomplir de nombreux types de tâches telles que la résolution de problèmes, la représentation des connaissances, la reconnaissance vocale, et d'autres. Grâce à ce domaine, et avec cette vague des techniques avancées d'apprentissage automatique et profond l'IA a fait un grand pas en avant.

Dans ce troisième chapitre nous définirons tout d'abord l'apprentissage automatique, ses différents types, ainsi que ses modèles les plus utilisés. Ensuite nous introduisons l'apprentissage profond en commençant par ses définitions, son histoire, ses différents modèles. Enfin nous conclurons avec ses métriques d'évaluation.

3.2 Les approches basées sur l'apprentissage automatique

3.2.1 Définitions

L'apprentissage automatique vu comme un domaine multidisciplinaire qui combine les statistiques et l'informatique. Plusieurs définitions sont données dans ce contexte :

L'apprentissage automatique a été inventé par Arthur Samuel,dés 1952 qui a créé le premier programme permettant de jouer et d'apprendre le jeu de dames. L'apprentissage automatique

est la discipline qui donne à l'ordinateur la capacité d'apprendre sans qu'il soit explicitement programmé[27].

Nous disons qu'un programme apprend d'une expérience E par rapport à une classe de taches T et à une mesure de performance P, si sa performance aux taches en T mesuré par P, s'améliore avec l'expérience E [27]. L'apprentissage automatique est un domaine de recherche mature et reconnu, qui s'intéresse principalement a générer des modèles de prédiction qui sont utilisés pour découvrir des informations, des connaissances et d'autres régularités dans les données.

Le processus d'apprentissage est définit comme une combinaison de trois activités[23] :

— **La représentation :** Le modèle doit être représenté dans un langage formel que l'ordinateur peut le gérer.

— **L'évaluation :** Une fonction (fonction objective ou de notation) est nécessaire pour distinguer entre les bons modèles et les mauvais.

— **L'optimisation :** Une technique d'optimisation est fondamentale pour rechercher parmi les modèles le plus performant.

3.2.2 Les différents types d'apprentissage automatique

Apprentissage supervisé

L'apprentissage supervisé repose sur des informations utiles contenues dans des données étiquetées. La classification est la tâche la plus courante dans l'apprentissage supervisé. Cependant, l'étiquetage manuel des données est coûteux et prend du temps. Par conséquent, le manque de données étiquetées suffisantes constitue le principal goulot d'étranglement de l'apprentissage supervisé[19].

Apprentissage non supervisé

un apprentissage non supervisé extrait des informations précieuses sur les fonctionnalités des données non étiquetées, ce qui facilite considérablement l'obtention des données de formation. Cependant, les performances de détection des méthodes d'apprentissage non supervisé sont généralement inférieures à celles des méthodes d'apprentissage supervisé.[19]

Apprentissage semi-supervisé

Certains algorithmes peuvent traiter des données d'apprentissage partiellement étiquetées, généralement beaucoup de données non étiquetées et un peu de données étiquetées. C'est ce qu'on appelle l'apprentissage semi-supervisé [26].

L'apprentissage semi-supervisé (SSL) est l'une des techniques Comment end d'apprentissage d'apprentissage automatique (ML). Il est à mi-chemin entre l'apprentissage supervisé et non supervisé. L'objectif principal de SSL est de surmonter les inconvénients de l'apprentissage supervisé

et non supervisé. L'apprentissage supervisé nécessite une énorme quantité de données de formation pour classer les données de test, ce qui est un processus long. D'un autre côté, l'apprentissage non supervisé ne nécessite aucune donnée étiquetée, qui regroupe les données en fonction de la similitude des points de données en utilisant une approche de clustering ou de probabilité maximale. Principal inconvénient de cette approche, elle ne peut pas regrouper avec précision des données inconnues. Pour surmonter ces problèmes, SSL a été proposé par la communauté des chercheurs, qui peut apprendre avec une petite quantité de données de formation peut étiquetée. SSL crée un modèle avec quelques modèles étiquetés comme données d'apprentissage et traite le reste des modèles comme des données de test [18].

La plupart des algorithmes d'apprentissage semi-supervisés sont des combinaisons d'algorithmes non supervisés et supervisés. Par exemple, les réseaux de croyances profondes (DBN) sont basés sur des composants non supervisés appelés machines Boltzmann restreintes (RBM) empilées les unes sur les autres. Les RBM sont formés séquentiellement de manière non supervisée, puis l'ensemble du système est affiné à l'aide de techniques d'apprentissage supervisé [26].

Apprentissage par renforcement

L'apprentissage par renforcement est une technique très différente. Le système d'apprentissage, appelé agent dans ce contexte, peut observer l'environnement, sélectionner et exécuter des actions et obtenir des valeurs scalaire, appelée récompenses en retour. Il doit ensuite apprendre par ces propres erreurs quelle est la meilleure stratégie, appelée politique, pour obtenir le plus de récompenses au fil du temps. Une politique définit l'action que l'agent doit choisir lorsqu'il se trouve dans une situation donnée(voir Figure 2.4). [26].

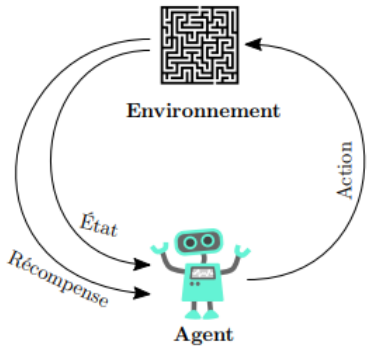


FIGURE 3.1 – Illustration du cadre général de l'apprentissage par renforcement[26].

L'apprentissage par renforcement, au sens général, est un cadre formel qui modélise des problèmes décisionnels séquentiels. Au sein de ce cadre, un agent apprend à prendre des décisions optimales en interagissant avec l'environnement (Narendra et Thathachar, 1974 ; Bush et Mosteller, 1955). Lorsqu'il effectue une action, l'état du système change et l'agent reçoit une valeur scalaire, appelée récompense, qui encode les informations sur la qualité de la transition

3.2.3 Les modèles d'apprentissage automatique

Nous présentons dans ce qui suit un résumé des modèles de classification les plus utilisés dans la détection des malwares Android, ainsi que les avantages et les inconvénients des modèles d'apprentissage supervisé et non supervisé(Tableau 3.1).

Réseau de neurones artificiels (ANN) :

L'idée de conception d'un ANN est d'imiter le fonctionnement du cerveau humain. Un ANN contient une couche d'entrée, plusieurs couches cachées et une couche de sortie. Un ANN contient un grand nombre d'unités et peut théoriquement approximer des fonctions arbitraires. Par conséquent, il a une forte capacité d'adaptation, en particulier pour les fonctions non linéaires. La formation des ANN prend du temps à cause de la structure complexe du modèle. Il est a noter que les modèles ANN sont formés par l'algorithme de rétropropagation qui ne peut pas être utilisé pour former des réseaux profonds. Ainsi, un ANN appartient à des modèles peu profonds et diffère des modèles d'apprentissage profond[19].

Machine à vecteurs de support (SVM) :

Une machine à vecteurs de support est un algorithme d'apprentissage supervisé qui trie les données en deux catégories. Il est formé avec une série de données déjà classées en deux catégories, construisant le modèle tel qu'il est initialement formé. La tâche d'un algorithme SVM est de déterminer à quelle catégorie appartient un nouveau point de données, cela fait de SVM une sorte de classificateur linéaire non binaire. Un algorithme SVM ne doit pas seulement placer les objets dans des catégories, mais avoir les marges entre eux sur un graphique aussi large que possible. Certaines applications de SVM incluent[37] :

— Classification de texte et d'hypertexte.

— Classification des images.

— Reconnaître les caractères manuscrits.

— Sciences biologiques, y compris la classification des protéines.

Les k plus proches voisins(KNN) :

Le K-plus proche voisin est un algorithme de classification de données qui tente de déterminer dans quel groupe se trouve un point de données en examinant les points de données qui l'entourent. Un algorithme, regardant un point sur une grille, essayant de déterminer si un point est dans le groupe A ou B, regarde les états des points qui sont proches de lui. La plage est déterminée arbitrairement, mais le but est de prélever un échantillon des données. Si la majorité des points sont dans le groupe A, alors il est probable que le point de données en question sera A plutôt que B.

Le k-plus proche-voisin est un exemple d'algorithme "paresseux" car il ne génère pas au préalable un modèle de l'ensemble de données. Les seuls calculs qu'il effectue sont lorsqu'il est demandé d'interroger les voisins du point de données. Cela rend knn très facile à implémenter pour l'exploration de données.[37].

Le classifieur naïf de Bayes :

Le classificateur Naïve Bayes est un modèle probabiliste qui tombe sous le coup de l'apprentissage bayésien. Il est basé sur le théorème de Bayes et cherche a calculer l'hypothèse avec une probabilité plus élevée d'un espace défini par des données d'entraînement. Ces modèles se caractérisent par leur simplicité mais aussi par leur capacité avérée a traiter des problèmes variés.[37]

Arbre de décision :

L'algorithme d'arbre de décision classe les données à l'aide d'une série de règles. Le modèle ressemble à un arbre, ce qui le rend interprétable. L'algorithme d'arbre de décision peut automatiquement exclure les fonctionnalités non pertinentes et redondantes. Le processus d'apprentissage comprend la sélection des fonctionnalités, la génération d'arbres et l'élagage des arbres. Lors de la formation d'un modèle d'arbre de décision, l'algorithme sélectionne individuellement les fonctionnalités les plus appropriées et génère des nœuds enfants à partir du nœud racine[37].

Clustering :

Le clusering est basé sur la théorie de la similitude, c'est-à-dire le regroupement de données hautement similaires dans les mêmes clusters et le regroupement de données moins similaires dans différents clusters. Différent de la classification, le clustering est un type d'apprentissage non supervisé, Par conséquent, les exigences relatives aux ensembles de données sont relativement faibles. Cependant, lors de l'utilisation d'algorithmes de clustering pour détecter des attaques, il est nécessaire de référencer des informations externes[19].

K-means :

K-means est un algorithme de clustering typique, où K est le nombre de clusters et la moyenne est la moyenne des attributs. L'algorithme K-means utilise la distance comme critère de mesure de similarité. Plus la distance entre deux objets de données est courte, plus ils sont susceptibles d'être placés dans le même cluster. L'algorithme K-means s'adapte bien aux données linéaires, mais ses résultats sur les données non convexes ne sont pas idéaux. De plus, l'algorithme K-means est sensible à la condition d'initialisation et au paramètre K. Par conséquent, de nombreuses expériences répétées doivent être exécutées pour définir la valeur du paramètre appropriée[19].

Comparaison de différents modèles d'apprentissage automatique :

Algorithmes	Avantages	Inconvénient
ANN	Capable de traiter des données non linéaires ; Forte capacité d'adaptation ;	Convient au sur-apprentissage ; A tendance a rester coincé dans un optimum local ; La formation des modèles prend du temps ;
KNN	Appliquer à des données massives ; Convient aux données non linéaires ; Entraînement rapide ; Robuste au bruit ;	Faible précision sur la classe minoritaire ; Temps de test lent ; Sensible au paramètre K
SVM	Apprenez des informations utiles à partir d'un petit train ; Forte capacité de génération ;	Ne fonctionne pas bien sur les méga données ou les tâches de classification multiples ; Sensible aux paramètres des fonctions du noyau ;
Naïve Bayes	Robuste au bruit ; Capable d'apprendre progressivement ;	Ne donne pas de bons résultats sur les données liées aux attributs ;
Arbre de décision	Sélectionnez automatiquement les fonctionnalités ; Interprétation forte ;	Tendances des résultats de la classification à la classe majoritaire ; Ignorer la corrélation des données ;
K-means	Simple, peut être formé rapidement ; Grande évolutivité ; Peut s'adapter aux méga données ;	Ne fonctionne pas bien sur des données non convexes ; Sensible à l'initialisation ; Sensible au paramètre K ;

TABLE 3.1 – Les avantages et les inconvénients des algorithmes d'apprentissage automatique[37].

3.3 L'apprentissage profond (DL)

3.3.1 Définitions

L'apprentissage profond vu comme l'une des principales technologies d'apprentissage automatique et d'intelligence artificielle.la figure 3.2 démontre la relation entre ces derniers. Plusieurs définitions sont données pour l'apprentissage profond :

Définition 01 : L'apprentissage profond est un ensemble d'algorithmes d'apprentissage automatique qui utilise plusieurs couches qui correspondent à différents niveaux d'abstraction à chaque niveau. Il se compose d'une couche d'entrée, d'une couche de sortie et de plusieurs couches ca-

chées. Il est utilisé pour la synthèse vocale, le traitement d'image, la reconnaissance de l'écriture manuscrite, la détection d'objets, l'analyse de prédiction et la prise de décision [20].

Définition 02 : L'apprentissage profond permet aux modèles de calcul composés de plusieurs couches de traitement d'apprendre des représentations de données avec plusieurs niveaux d'abstraction. Ces méthodes ont considérablement amélioré l'état de l'art en matière de reconnaissance vocale, de reconnaissance visuelle d'objets, de détection d'objets et de nombreux autres domaines tels que la découverte de médicaments et les logiciels malveillants. [35].

Définition 03 : L'apprentissage profond est une branche d'apprentissage automatique qui dépend de l'étude de différents niveaux de représentations, analogues a un classement de caractéristiques ou de notions, où les notions de haut niveau sont déterminées à partir de celles de niveau inférieur, et des notions similaires de niveau inférieur pourraient aider a déterminer de nombreuses notions de haut niveau [41].

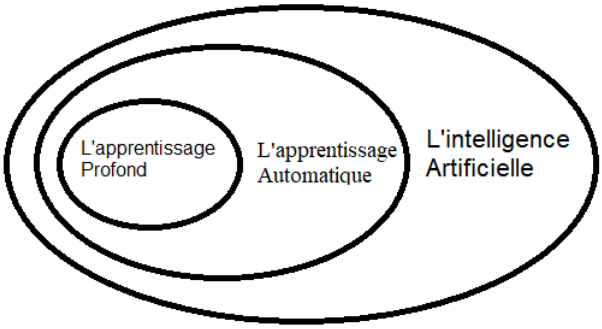


FIGURE 3.2 – La relation entre l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond [1].

3.3.2 Les différents modèles d'apprentissage profond

Machine Boltzmann restreinte (RBM) :

Un RBM est un réseau de neurones aléatoire dans lequel les unités obéissent à la distribution de Boltzmann. Un RBM est composé d'une couche visible et d'une couche cachée. Les unités d'une même couche ne sont pas connectées. Cependant, les unités dans les différentes couches sont entièrement connectées, les RBM ne font pas de distinction entre les directions avant et arrière. Ainsi, les poids dans les deux directions sont les mêmes. Les RBM sont des modèles d'apprentissage non supervisés formés par l'algorithme de divergence contrastive, et ils sont généralement appliqués pour l'extraction de caractéristiques [37].

Réseau de croyances profondes (DBN) :

Un DBN se compose de plusieurs couches RBM et d'une couche de classification softmax [1]. La formation d'un DBN comprend deux étapes : une pré-formation non supervisée et un réglage fin supervisé. Tout d'abord, chaque RBM est formé à l'aide d'un prétraitement gourmand en couches [2]. Ensuite, le poids de la couche softmax est appris par des données étiquetées. Dans la détection d'attaque, les DBN sont utilisés pour l'extraction et la classification des fonctionnalités [37].

Réseau de neurones profonds (DNN) :

Une stratégie de pré-formation et de réglage fin au niveau des couches permet de construire des DNN avec plusieurs couches. Lors de la formation d'un DNN, les paramètres sont d'abord appris à l'aide de données non étiquetées, qui est une étape d'apprentissage des fonctionnalités non supervisées. Ensuite, le réseau est réglé via les données étiquetées, qui est une étape d'apprentissage supervisé. Les réalisations étonnantes des DNN sont principalement dues à l'étape d'apprentissage des fonctionnalités non supervisées [37].

Réseau neuronal convolutif (CNN) :

Les CNN sont conçus pour imiter le système visuel humain (HVS). Par conséquent, les CNN ont atteint de grandes réalisations dans le domaine de la vision par ordinateur. Un CNN est empilé avec des couches de convolution d'une dimension, deux dimensions et trois dimensions (Conv1D,Conv2D et Conv3D) et de mise en commun alternatives (Maxpooling1D,Maxpooling2D et Maxpooling3D).Les couches convolutifs sont utilisées pour extraire des entités et les couches de mise en commun sont utilisées pour améliorer la généralisabilité des entités [37]. Lorsque ces couches sont empilées, une architecture CNN a été formée.

Réseau de neurones récurrents (RNN) :

Les RNN sont des réseaux conçus pour des données séquentielles et sont largement utilisés dans le traitement du langage naturel (PNL). Les caractéristiques des données séquentielles sont contextuelles. L'analyse de données isolées de la séquence n'a aucun sens. Pour obtenir des informations contextuelles, chaque unité d'un RNN reçoit non seulement l'état actuel mais également les états précédents. Cette caractéristique fait que les RNN souffrent souvent de gradients qui disparaissent ou explosent. En réalité, les RNN standard ne traitent que des séquences de longueur limitée. Pour résoudre le problème de dépendance à long terme, de nombreuses variantes de RNN

1. La fonction softmax est une fonction qui transforme un vecteur de K valeurs réelles en un vecteur de K valeurs réelles dont la somme est égale à 1.

2. un prétraitement gourmand ou glouton (Greedy en anglais) en couches fournit un moyen de développer des réseaux de neurones multicouches profonds tout en ne formant que des réseaux peu profonds.

ont été proposées, telles que la mémoire à court terme à long terme (LSTM), l'unité récurrente fermée (GRU) [37].

— **LSTM(Long Short Term Memory) :** Les LSTM sont un type particulier de RNN, capable d'apprendre des dépendances à long terme. Les LSTM sont explicitement conçus pour éviter le problème de dépendance à long terme. Se souvenir des informations pendant de longues périodes. Ils fonctionnent extrêmement bien sur une grande variété de problèmes et sont maintenant largement utilisés [3].

— **GRU(unité récurrente fermée) :** L'unité récurrente fermée est un autre type de cellule RNN, vise a résoudre le problème du gradient de fuite qui vient avec un réseau de neurones récurrent standard. GRU est une modification de la couche cachée RNN qui permet de mieux capturer les connexions à longue portée. GRU est similaire au LSTM mais sa structure est simplifiée. Comme LSTM, pour résoudre le problème de gradient de fuite d'un RNN standard, GRU utilise, soi-disant, la porte de mise à jour et la porte de réinitialisation. Fondamentalement, ce sont deux vecteurs qui décident quelles informations doivent être transmises à la sortie[3].

Comparaison des différents modèles d'apprentissage profond

Algorithmes	Aventages	Inconvénients
RBM	- Permet de produire des échantillons comme s'ils provenaient de la distribution des données - Il peut être utilisé comme extracteur de fonctionnalités pour entraîner d'autres modèles par-dessus	- Difficile de bien s'entraîner - Le calcul de la probabilité prend du temps
DBN	Offrir une approche d'apprentissage couche par couche pour initialiser le réseau	La phase de formation consomme des ressources système en raison du processus d'initialisation et d'échantillonnage.
CNN	- Moins de connexions neuronales nécessaires par rapport à un NN standard. - De nombreuses variantes de CNN ont été développées	- Habituellement, il a besoin de plusieurs couches pour découvrir une hiérarchie complète des fonctionnalités visuelles - Il a généralement besoin d'un grand ensemble de données d'images balisées
RNN	- Modélisation des dépendances temporelles - Capable de se souvenir des événements en série	Le processus d'apprentissage souffre d'un problème de gradient qui disparaît (un grand changement dans la valeur des paramètres pour les premières couches n'a pas un grand effet sur la sortie)
DAE	- Appliqué à la fonction d'extraction / réduction de dimensionnalité. - De nombreuses variantes de DAE ont été proposées	- Il nécessite une étape de pré-formation - Il n'a pas la capacité de déterminer quelles données sont pertinentes
DNN	Succès accompli dans différentes applications	Le processus d'apprentissage pourrait prendre du temps

TABLE 3.2 – Les avantages et les inconvénients des algorithmes d'apprentissage profond [40].

3.3.3 Métriques d'évaluation

De nombreuses métriques sont utilisées pour évaluer les méthodes d'apprentissage automatique et d'apprentissage profond. Les modèles optimaux sont sélectionnés à l'aide de ces métriques [37] :

Accuracy :

Est une mesure appropriée lorsque l'ensemble de données est équilibré. Dans des environnements réseau réels ; cependant, les échantillons normaux sont beaucoup plus abondants que les échantillons anormaux ; par conséquent, Accuracy peut ne pas être une mesure appropriée.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

Précision (P) :

Est défini comme le rapport des échantillons positifs réels aux échantillons positifs prédits, il représente la confiance de la détection d'attaque.

$$P = \frac{TP}{TP + FP} \tag{3.2}$$

Rappel (R) :

Est défini comme le rapport des vrais échantillons positifs au total des échantillons positifs et est également appelé le taux de détection. Le taux de détection reflète la capacité du modèle à reconnaître les attaques, qui est une mesure importante dans les systèmes de détection d'intrusions.

$$R = \frac{TP}{TP + FN} \tag{3.3}$$

Mesure-F (F) :

Est définie comme la moyenne pondérée de la précision et du rappel.

$$F = \frac{2 * P * R}{P + R} \tag{3.4}$$

Le taux de faux négatifs (FNR) :

est défini comme le rapport des échantillons faussement négatifs au total des échantillons positifs. Dans la détection d'attaque, le FNR est également appelé le taux d'alarme manquée.

$$FNR = \frac{FN}{TP + FN} \tag{3.5}$$

Le taux de faux positifs (FPR) :

est défini comme le rapport des échantillons faux positifs aux échantillons positifs prédits. Dans la détection d'attaque, le FPR est également appelé le taux de fausse alarme, et il est calculé comme suit :

$$FPR = \frac{FP}{TN + FP} \tag{3.6}$$

Où TP est le vrai positif(Nombre d'échantillons correctement distingué comme malveillant), FP est le faux positif(Nombre d'échantillons identifiés à tort comme malveillants), TN est le vrai négatif-(Nombre d'échantillons correctement distingués comme bénins), FN est le faux négatif(Nombre d'échantillons identifiés à tort comme bénins).

3.4 Conclusion

Nous avons présenté en détails les différentes techniques d'apprentissage automatique et d'apprentissage profond ainsi que ces métriques d'évaluation. Le chapitre suivant sera consacrer sur la contribution et les résultats obtenu en utilisant l'une des techniques d'apprentissage profond décrite précédemment.

CHAPITRE 4

VERS UNE APPROCHE BASÉE DEEP LEARNING POUR LA DÉTECTION D'INTRUSIONS

4.1 Introduction

Après avoir étudié l'état de l'art des différents outils d'apprentissage automatique et profond pour les systèmes de détection d'intrusions, nous présentons notre contribution, en commençant par l'architecture de notre système, dans cette partie nous détaillons l'architecture proposée. Ensuite nous définissons l'architecture de notre modèle, présentation des ensembles de données utilisés et les pré-traitements effectués. Enfin nous conclurons avec l'implémentation en mettant l'accent sur l'évaluation de notre modèle, l'environnement de développement et les outils utilisés lors de la création de notre système.

4.2 Présentation de l'architecture de la solution proposée

4.2.1 Architecture générale du modèle

Notre projet porte sur la tâche d'analyse des intrusions et de détecter ces derniers, pour atteindre cet objectif et pour obtenir les meilleures résultats possible nous proposant l'architecture générale du modèle suivant :

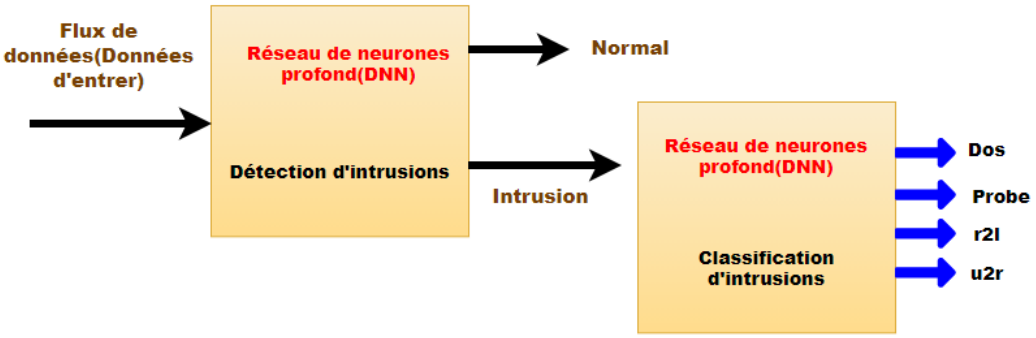


FIGURE 4.1 – Architecture générale du modèle.

1. **Flux de données :** Les flux de données sont les paquets a quel en va les classer comme Bénignes ou malveillants, ces paquets contiennent des caractéristiques(fonctionnalités) qui sont des données d'entrée pour le modèle proposé. Voici un exemple sur les fonctionnalité d'un paquet dans l'ensemble de données NSL-KDD : Fonctionalités d'un echontillon ="duration","protocol-type","service","flag","src-bytes", "dst-bytes"...
2. **Détection d'intrusions :** C'est l'étape ou le modèle prend les données d'entrée, effectue la détection de ces derniers et retourne comme résultat si le paquet est bénigne ou contient des caractéristiques malveillantes.
3. **Classification d'intrusions :** Les caractéristiques du paquet qui ont été détecté comme malveillantes vont passer à l'étape de classification, ces derniers vont être classés par type, Exemple des types d'attaques de l'ensemble de données NSL-KDD : DoS,Probing,r2l,u2r.

4.2.2 Architecture du réseau de neurones profond (DNN) proposée

Au cours de nos expériences, nous avons effectués plusieurs test des différentes méthodes d'apprentissage profond, où nous avons testé 5 couches cachées avec 1024, 512, 64 et 32 neurones successive, fonction d'activation reLu pour toutes les couche, la taille de lot (batch size) est fixé 1024 et aussi le nombre d'époques (epochs) 100 épauques, afin d'améliorer les performances du modèle que ce soit en temps ou en efficacité. La figure suivante représente l'architecture de notre modèle proposée :

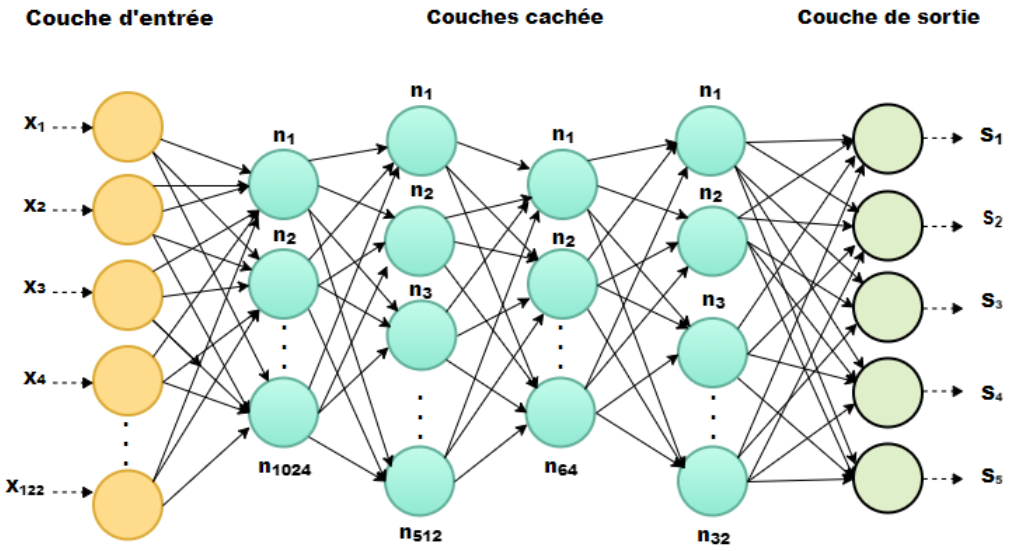


FIGURE 4.2 – Architecture générale du modèle DNN proposée.

Chaque modèle a une couche d'entrer, des couches cachées et une couche de sortie, ces couches sont détaillé comme suit :

- **Couche d'entrer :** Contient les données extraites du vecteur de caractéristiques(ou vecteur d'entrer) avec le total des échantillons, le vecteur contient 122 caractéristiques.
- **Couches cachées :** Nous avons utilisé plusieurs couches cachées, Dense et Dropout :
  - **Couche Dense :**La couche dense est une couche de réseau neuronal qui est profondément connectée, ce qui signifie que chaque neurone de la couche dense reçoit des entrées de tous les neurones de sa couche précédente. Une couche dense peut être définie comme :

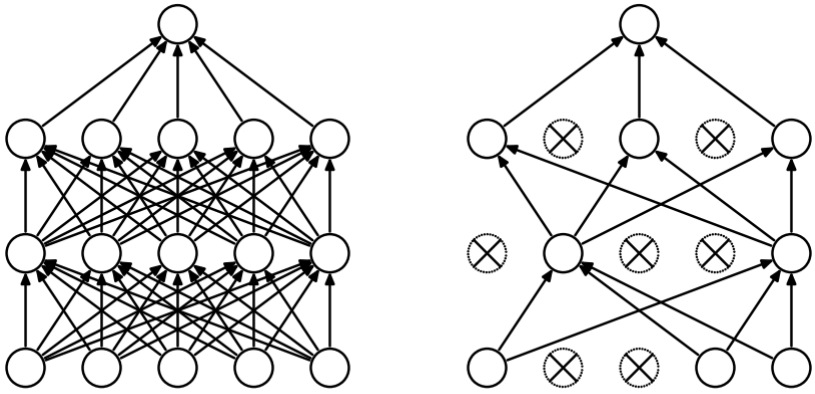
    $$y = activation(W * x + b) \tag{4.1}$$

    où W est le poids, b est un biais, x est l'entrée et y est la sortie, * est la multiplication matricielle et une fonction d'activation.

    Les fonctions d'activation sont des fonctions utilisées dans les réseaux de neurones pour calculer la somme pondérée des entrées et des biais, qui sert à décider si un neurone peut être déclenché ou non. Il manipule les données présentées à travers un traitement de gradient généralement descente de gradient et produit ensuite une sortie pour le réseau neuronal, qui contient les paramètres dans les données [42].

    Nous avons utilisé quatre couches Dense de taille 1024,512,64 et 32 suivi d'une fonction d'activation relu pour chaque couche.
  - **Couche Dropout :** Le Dropoute est une technique populaire et efficace contre le surapprentissage dans les réseaux de neurones. L'idée initiale du Dropout est de supprimer au hasard des unités et des connexions pertinentes des réseaux de neurones pendant

l'entraînement. Cela empêche les unités de trop s'adapter [53], Comme le montre la figure 4.3.



(a) Standard Neural Net (b) After applying dropout.

FIGURE 4.3 – Illustration du dropout lors de l'apprentissage (à droite) et lors du test (à gauche)[32].

Nous avons appliqué trois couches Dropout après chaque couche DNN pour la régularisation.

- **Ccouche de sortie :**Nous avons ajouté une autre couche dense celle de la sortie, elle contient 5 neurones pour prédire la sortie suivi d'une fonction d'activation softmax.

  Nous avons utilisé la fonction d'activation softmax pour les 3 ensembles de données par ce qu'elle produit une sortie qui est une plage de valeurs entre 0 et 1, (c.à.d elle est utilisée lors de la classification à plusieurs classes ou sortie). La figure suivante présente un exemple d'une fonction softmax :

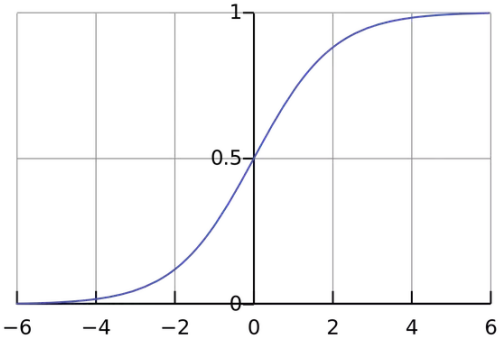


FIGURE 4.4 – Fonction Softmax[42].

Pour la procédure d'apprentissage finale du modèle, une taille de lot(batch size) de 1024 a été fixée et 100 époques(epochs), avec l'optimiseur [1] «Adam» a été utilisé.

1. Permet de réduire le poids des erreurs

Adam est un algorithme d'optimisation qui peut être utilisé pour mettre à jour les poids du réseau de manière itérative en fonction des données d'apprentissage.

4.2.3 Présentation des ensembles de données utilisés

Un ensemble de données est une collection de données. En d'autres termes, un ensemble de données correspond au contenu d'un seul tableau de base de données, ou d'une seule matrice de données statistiques, où chaque colonne du tableau représente une variable particulière et chaque ligne correspond à un membre donné de l'ensemble de données en question. Dans les projets d'apprentissage profond, nous avons besoin d'un ensemble de données d'entraînement. Il s'agit de l'ensemble de données réel utilisé pour entraîner le modèle afin d'effectuer diverses actions[29].

L'apprentissage profond dépend fortement des données, sans données, il est impossible pour l'IA d'apprendre. C'est l'aspect le plus crucial qui rend possible la formation d'algorithmes. Peu importe la taille de l'ensemble de données, si l'ensemble de données n'est pas assez bon, tout le projet d'IA échouera[29] !

Au cours de notre réalisation du système de détection d'intrusion nous avons appliqué plusieurs ensemble de données **NSL-KDD, KDD-Cup99 et unsw-nb15**, nous avons commancer par **NSL-KDD** parce qu'il est le plus récent (2021) et contient beaucoup de caractéristiques et de type d'intrusions **(Dos,Probe,r2l et u2r)**

L'ensemble de données NSL-KDD

L'ensemble de données NSL-KDD [25] est une version raffinée de son prédécesseur KDD-Cup99[21], il contient des caractéristiques essentielles de l'ensemble de données KDD complet. Une collection de fichiers téléchargeables est à la disposition des chercheurs. Ils sont répertoriés dans le tableau 3.1.

Dans chaque enregistrement, il y a 41 attributs dépliant différentes caractéristiques du flux et une étiquette attribuée à chacun en tant que type d'attaque ou en tant que normal. Le **42éme** attribut contient des données sur les différentes 5 classes de vecteurs de connexion réseau et ils sont classés en une classe normale et quatre classes d'attaque, les 4 classes d'attaque sont regroupées en **DoS, Probe, R2L et U2R**.

Les classes d'attaques présentes dans l'ensemble de données NSL-KDD sont regroupées en quatre catégories :

1. **DoS :** Le déni de service est une catégorie d'attaque, qui épuise les ressources de la victime, la rendant ainsi incapable de traiter les demandes légitimes.
2. **Probe(Sondage) :** L'objectif de surveillance et d'autres attaques de sondage est d'obtenir des informations sur la victime distante.
3. **U2R :** L'accès non autorisé aux privilèges locaux de super utilisateur (root) est un type d'attaque par lequel un attaquant utilise un compte normal pour se connecter à un système

victime et essaie d'obtenir des privilèges root/administrateur en exploitant une vulnérabilité de la victime.

4. **R2L :** Accès non autorisé depuis une machine distante, l'attaquant s'introduit dans une machine distante et obtient un accès local de la machine victime.

Catégorie.	Données d'entrainement	données de tests
Normal	67,343	9,710
DoS	45,927	5,741
Probe	11,656	1106
r2l	995	2,199
u2r	52	37

TABLE 4.1 – Ensembles de formation et de tests de l'ensemble de données NSL-KDD. [22].

L'ensemble de données UNSW-NB15

L'ensemble de données UNSW-NB15 est nouveau et a été publié en 2015. Il comprend les attaques modernes (neuf types d'attaques contre 4 types d'attaques dans l'ensemble de données KDD'99). Il a 45 fonctionnalités et une variété d'activités normales et attaquées, y compris avec des étiquettes de classe de 2540044 enregistrements au total. Il y a 221,876 enregistrements normaux et 321,283 enregistrements attaqués dans le nombre total d'enregistrements. Les fonctionnalités de l'ensemble de données UNSW-NB15 sont classées en six groupes, à savoir les fonctionnalités de base, les fonctionnalités de flux, les fonctionnalités de temps, les fonctionnalités de contenu, les fonctionnalités générées supplémentaires et les fonctionnalités étiquetées. Les fonctionnalités comptant de 36 à 40 sont appelées fonctionnalités à usage général. Les caractéristiques comptant de 41 à 45 sont appelées caractéristiques de connexion. En outre, l'ensemble de données UNSW-NB15 comporte neuf types de catégories d'attaques appelées Analyse, Fuzzers, Backdoors, DoS Exploits, Reconnaissance, Generic, Shellcode et Worms[22][21]. Les types d'attaques peuvent être classés en neuf groupes :

1. **Fuzzers :** Une attaque dans laquelle l'attaquant tente de découvrir des failles de sécurité dans un programme, un système d'exploitation ou un réseau en l'alimentant avec la saisie massive de données aléatoires pour le faire planter.
2. **Analyse :** Un type d'intrusions variées qui pénètrent dans les applications Web via des ports (par exemple, des analyses de ports), des e-mails (par exemple, du spam) et des scripts Web (par exemple, des fichiers HTML).
3. **Porte dérobée :** Une technique consistant à contourner une authentification normale furtive, à sécuriser l'accès à distance non autorisé à un appareil et à localiser l'entrée du texte en clair alors qu'il a du mal à rester inaperçu.

4. **DoS :** Une intrusion qui perturbe les ressources de l'ordinateur via la mémoire, pour être extrêmement occupée afin d'empêcher les requêtes autorisées d'accéder à un appareil.
5. **Exploit :** Une séquence d'instructions qui tire parti d'un problème, d'un bogue ou d'une vulnérabilité causé par un comportement non intentionnel ou insoupçonné sur un hôte ou un réseau.
6. **Générique :** Une technique qui établit contre chaque bloc-chiffrement en utilisant une fonction de hachage à collision sans égard à la configuration du bloc-chiffrement.
7. **Reconnaissance :** Peut être défini comme une sonde ; une attaque qui recueille des informations sur un réseau informatique pour échapper à ses contrôles de sécurité.
8. **Shellcode :** Une attaque dans laquelle l'attaquant pénètre un petit morceau de code à partir d'un shell pour contrôler la machine compromise.
9. **Ver :** Attaque par laquelle l'attaquant se réplique afin de se propager sur d'autres ordinateurs. Souvent, il utilise un réseau informatique pour se propager, en fonction des défaillances de sécurité sur l'ordinateur cible pour y accéder.

Catégorie.	Données d'entrainement	données de tests
Normal	56,000	37,000
Analysis	2,000	677
Backdoor	1,746	583
DoS	12,264	4089
Exploits	33,393	11,132
Fuzzers	18,184	6,062
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shellcode	1,133	378
Worms	130	44

TABLE 4.2 – Ensembles de formation et de tests de l'ensemble de données UNSW-NB15. [22].

L'ensemble de données KDD Cup99

L'ensemble de données KDD'99 a été créé par la DARPA en 1999 en utilisant le trafic réseau enregistré à partir de l'ensemble de données de 1998. Il est pré-traité en 118 fonctionnalités par connexion réseau. Les fonctionnalités de l'ensemble de données KDD'99 sont classées en quatre groupes. KDD'99 se compose de 4,898,430 enregistrements, ce qui est plus volumineux que les autres ensembles de données. Il existe quatre catégories principales d'attaques, à savoir DoS, R2L (accès non autorisé à partir d'une machine distante), U2R (accès non autorisé à la racine) et sonde(Probing). De nombreuses techniques d'exploration de données ont été appliquées à l'ensemble de données KDD'99 pour détecter les intrusions dans le trafic réseau[21][31].

KDD Cup'99 est un ensemble de données principalement utilisé pour créer un système de détection d'intrusion (IDS). L'ensemble de données KDD présente deux problèmes critiques conclus par l'analyse statistique, qui affectent profondément les performances du système. Le problème le plus important dans l'ensemble de données KDD est qu'il contient un grand nombre d'enregistrements répliqués. Il s'avère qu'environ 78% et 75% des enregistrements sont dupliqués dans l'ensemble de données de train et de test, respectivement. Un grand nombre d'enregistrements répliqués peut conduire les algorithmes d'apprentissage à être partiels au lieu de nombreux enregistrements. Ainsi, l'algorithme arrêtera d'apprendre les enregistrements peu fréquents. Ces enregistrements peuvent être nocifs pour les réseaux tels que U2R, R2L, etc[21][31].

Catégorie.	Données d'entrainement	données de tests
Normal	97,278	60,593
DoS	391,458	229,853
Probe	4,107	4,166
r2l	1,126	16,189
u2r	52	228

TABLE 4.3 – Ensembles de formation et de tests de l'ensemble de données KDD-Cup99. [21].

4.2.4 Présentation des pré-traitements effectués

Lorsqu'il s'agit de créer un modèle d'apprentissage profond, le prétraitement des données est la première étape marquant le lancement du processus.Le prétraitement des données prépare les données brutes pour un traitement ultérieur. Les données passent par une série d'étapes au cours du prétraitement :

Nous avons appliqué plusieurs méthodes pour l'ensemble de données utilisée afin d'améliorer la qualité des données et les rendre adaptées à la formation de notre modèle d'apprentissage profond.

pré-traitements effectués pour NSL-KDD et KDD-Cup 99

1. **des colonnes de chaque attribut du l'ensemble de données :**

   Comme première étape du prétraitement, l'ensemble de données utilisé n'a pas les noms ou l'indication de chaque attribut, donc nous avons ajouté ces colonnes afin de nous aider à mieux détecter et de classer les types d'intrusion indiqué par le modèle. Voici un extrait du code pour les colonnes ajouté à l'ensemble de données(Figure3.3.). Après nous avons ajouté les colonnes à l'ensemble de données(Figure3.4).

```
# définir les colonnes de chaque attribut du dataset:
col_names = ["duration","protocol_type","service","flag","src_bytes",
    "dst_bytes","land","wrong_fragment","urgent","hot","num_failed_logins",
    "logged_in","num_compromised","root_shell","su_attempted","num_root",
    "num_file_creations","num_shells","num_access_files","num_outbound_cmds",
    "is_host_login","is_guest_login","count","srv_count","serror_rate",
    "srv_serror_rate","rerror_rate","srv_rerror_rate","same_srv_rate",
    "diff_srv_rate","srv_diff_host_rate","dst_host_count","dst_host_srv_count",
    "dst_host_same_srv_rate","dst_host_diff_srv_rate","dst_host_same_src_port_rate",
    "dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_host_srv_serror_rate",
    "dst_host_rerror_rate","dst_host_srv_rerror_rate","class","dificulty"]
```

FIGURE 4.5 – Extrait du code pour les colonnes manquantes.

```
# importer dataset et attribuer pour chaque dataset ces collonnes:
# NSL-KDD:
dataset_train = pd.read_csv("/content/KDDTrain+.txt", header=None, names = col_names)
dataset_test = pd.read_csv("/content/KDDTest+.txt", header=None, names = col_names)
```

FIGURE 4.6 – Ajouter les colonnes à l'ensemble de données.

2. **Redimensionnement des données :**

   Le redimensionnement des données est nécessaire lorsque l'ensemble de données à des fonctionnalités pas de même type où le modèle ne peut lire ces derniers et renvoie un erreur. Afin d'éviter ce problème et avoir des meilleurs résultats possible nous utilisant la méthodes **MinMaxScaler**[45].

   Cet estimateur met à l'échelle et traduit chaque caractéristique individuellement de telle sorte qu'elle se situe dans la plage donnée sur l'ensemble d'apprentissage,dont notre cas est entre 0 et 1, voici un extrait du code utilisé par MinMaxScaler :

```
# Nous redimensionnons les fonctionnalités à [0, 1] :
min_max_scaler = MinMaxScaler()
train = min_max_scaler.fit_transform(train)
test = min_max_scaler.transform(test)
```

FIGURE 4.7 – Extrait du code de MinMaxScaler.

pré-traitements effectués pour UNSW-NB15

1. **Encodage des caractéristiques catégorielles :**

   Souvent, les caractéristiques ne sont pas données sous forme de valeurs continues mais catégoriques. Par exemple, une personne pourrait avoir des fonctionnalités [**"homme", "femme"**], [**"d'Europe", "des États-Unis", "d'Asie"**], [**"utilise Firefox", "utilise Chrome", "utilise Safari", "utilise Internet Explorer"**]. De telles caractéristiques peuvent être efficacement codées sous forme d'entiers, comme dans notre ensemble de données ["DoS","Normal", "Probe", "r2l", "u2r"] pourrait être exprimé comme[1, 2, 3, 4, 5][44].

Pour convertir des caractéristiques catégorielles en de tels codes entiers, nous pouvons utiliser **l'OrdinalEncoder**. Cet estimateur transforme chaque caractéristique catégorique en une nouvelle caractéristique d'entiers (0 à n categories - 1)[44] :

```
>>> enc = preprocessing.OrdinalEncoder()
>>> X = [['male', 'from US', 'uses Safari']]
>>> enc.fit(X)
OrdinalEncoder()
>>> enc.transform([['female', 'from US', 'uses Safari']])
array([[0., 1., 1.]])
```

2. **Normalisation des données :**

   La normalisation des données est le processus de structuration d'une base de données, généralement un ensemble de données relationnelles, conformément à une série de formes dites normales afin de réduire la redondance des données et d'améliorer l'intégrité des données[44].

   Cette hypothèse est la base du modèle d'espace vectoriel souvent utilisé dans les contextes de classification et de regroupement de textes.

   La fonction "normalize" fournit un moyen rapide et facile d'effectuer cette opération sur un seul ensemble de données de type tableau[44] :

```
>>> normalizer = preprocessing.Normalizer().fit(X)
>>> normalizer
Normalizer()
```

4.3 Simulations et résultats

Dans cette section, nous adaptons notre modèle proposé DNN. Ensuite, nous comparons les résultats obtenus pour les 3 ensembles de données utilisés. Dans cette étude comparative, nous avons utilisé comme ensemble de données de d'apprentissage 80% de la totalité des données et comme données de test 20% de l'ensemble de données pour les 3 ensembles.

4.3.1 Évaluation de notre modèle

L'évaluation de notre modèle se base sur quatre mesures : vrai positif, vrai négatif, faux positif et faux négatif. Ces valeurs sont utilisées pour calculer les mesures de performance pour le modèle proposé. Nous considérons cinq paramètres pour l'évaluation :F-mesure, Taux d'exactitude(ACC), précision(P), taux de détection(DR), Taux de fausses alarmes(FAR). Les tableaux suivants présentent la précision, le taux de détection et le taux de fausses alarmes pour les 3 ensembles de données utilisées avec leurs classes d'attaques.

-	Normal	Dos	U2R	Probing	R2L
P	99.75%	99.94%	82.92%	99.52%	98.34%
DR	99.86%	100%	65.38%	99.6%	89.84%
FAR	0.29%	0.028%	0.05%	0.98%	0.01%

TABLE 4.4 – Taux de détection, Taux de précision et Taux de fausses alarmes pour l'ensemble de données NSL-KDD.

-	Normal	Dos	U2R	Probing	R2L
P	70.07%	90.13%	91.91%	84.52%	0.83% .
DR	92.70%	85.64%	8.35%	83.14%	66% .
FAR	29%	4.66%	0.05%	1.78%	0.1% .

TABLE 4.5 – Taux de détection, Taux de précision et Taux de fausses alarmes pour l'ensemble de données KDD-Cup 99.

-	Normal	Fuzzers	Analysis	Backdoor	DoS	Exploits	Generic	Rec	Shell	Worms
P	100%	98.5%	100%	98.88%	99.9%	99.85%	99.52%	100%	96.91%	99.94%.
DR	100%	100%	90.9%	99.58%	99.85%	99.98%	0.99%	6.81%	99.73%	100%.
FAR	0%	0.067%	0%	0.058%	0.001%	0.001%	0.038%	0%	0.014%	0.001%.

TABLE 4.6 – Taux de détection, Taux de précision et Taux de fausses alarmes pour l'ensemble de données UNSW-NB15 .

On remarque que notre modèle (DNN) donne un des bons résultats pour les 3 ensembles de données utilisés et on trouve que l'ensemble UNSW-NB15 est plus performant que les 2 autres ensembles 2 données, vu qu'il contient plus de types d'attaques que les 2 autres.

La figure 3.8 nous montre que notre modèle donne le taux de d'exactitude le plus élevé avec 99.79% pour NSL-KDD, 97.55% pour KDD-Cup99 et 99.75% pour UNSW-NB15.

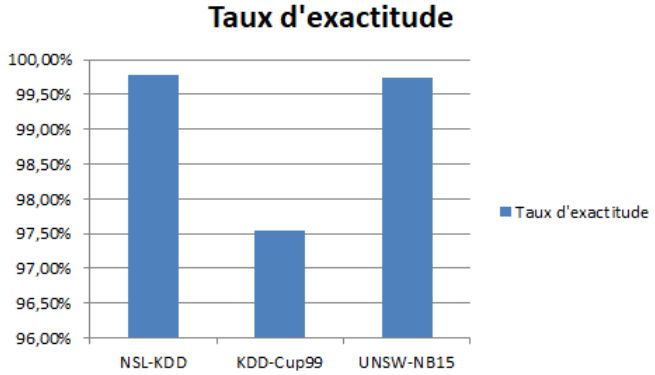


FIGURE 4.8 – Le taux de détection(DR), la précision(ACC) de notre modèle comparé à d'autre modèles d'apprentissage automatiques et profond.

4.3.2 Environnement et technologies logicielles

1. **Ressources matérielles**

CPU	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
GPU	AMD Radeon(TM) R7 M440
RAM	8.00Go

TABLE 4.7 – Ressources matérielles

2. **Ressources logicielles**

SE	Windows 10 Professionnel
HARDWARE	Jupyter NoteBook, Colab

TABLE 4.8 – Ressources logicielles

4.3.3 Outils utilisé

- **TensorFlow**
  TensorFlow est une bibliothèque d'apprentissage automatique, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques complexes plus facilement. Elle permet de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels[34].
  TensorFlow regroupe un grand nombre de modèles et d'algorithmes d'apprentissage automatique et profond. Son API front-end de développement d'applications repose sur le langage Python, tandis que l'exécution de ces applications s'effectue en C++ haute-performance. Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification d'images, la la détection des malwares, classification des objets, etc.[34].
- **Keras**
  Keras est une bibliothèque open-source de composants de réseau de neurones écrite en Python et capable de fonctionner au-dessus de tensorflow , CNTK ou Théano. Il a été développé pour but de lui permettre une expérimentation rapide [4].
  Utilisation de Keras dans le besoin d'une bibliothèque d'apprentissage profond qui [4] :
  — Permet un prototypage facile et rapide (grâce à la convivialité, la modularité et l'extensibilité).
  — Prend en charge les modèles d'apprentissage profond récents.
- **Scikit-learn**
  Scikit-learn est une bibliothèque en Python qui fournit de nombreux algorithmes d'apprentissage non supervisés et supervisés. Il s'appuie sur certaines des technologies, comme NumPy, pandas et Matplotlib. Les fonctionnalités fournies par scikit-learn incluent [8] :

— **Régression**, y compris régression linéaire et logistique.

— **Classification**, y compris K-voisins les plus proches.

— **Clustering**,y compris K-Means et K-Means ++.

— **Sélection du modèle**.

— **Prétraitement**, y compris la normalisation Min-Max.

Autres outils

- **Matplotlib [6] :** est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python.
- **pandas [7] :** est une bibliothèque open source qui fournit des structures de données hautes performances et faciles a utiliser et des outils d'analyse de données pour le langage de programmation Python.
- **Numpy [5] :** est une bibliothèque en Python qui fournit de nombreux algorithmes d'apprentissage non supervisés et supervisés. Il s'appuie sur certaines des technologies, comme NumPy, pandas et Matplotlib.

4.4 conclusion

Ce chapitre nous a permis de découvrir et comparer les différents ensembles de données pour la détection d'intrusions avec un modèle d'apprentissage profond DNN. D'après les résultats on constate que le modéle proposé est puissant et efficace en taux de précisions pour les trois ensembles.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Le Cloud Computing est une technologie en pleine croissance qui a changé notre vie de bonne à intelligente. Les appareils dans le Cloud sont connectés sur Internet, il y a donc plus d'insécurité face aux attaques. Pour découvrir très efficacement les intrusions, nous devons construire des solutions. Le développement de méthodes intelligentes est nécessaire pour lutter contre ces attaques.

Dans ce travail nous avons proposé et évalué notre nouvelle méthode afin de caractériser, classifier et détecter les intrusions avec leur type en utilisant trois différents ensembles de données.

Le produit final résultant de cette étude est un modèle pour la détection d'intrusions qui se constitue : D'une procédure d'extraction de caractéristiques à partir de chaque ensemble de données basées sur l'analyse des échantillons . Nous avons conçu un réseau de neurones profond(DNN), qui a donné de bons résultats, taux d'exactitude 99.79%, 97.55%, 99.75% pour les ensembles de données NSL-KDD, KDD-Cup 99, UNSW-NB15, ce qui signifie qu'il est capable de discriminer presque tous les cas d'intrusion existant dans chaque ensemble de données considéré.

Comme perspective, nous envisageons de raffiner notre étude à travers plusieurs critères, où nous essayons de faire Classification et détection des intrusions par plusieurs classes de famille ou type d'attaques qui ont pas été cité dans les ensembles de données utilisé(Ransomware,rootkit,...etc), de combiné d'autre solution qui ont été proposés pour mieux détecter des nouvel attaques, extraire d'autres caractéristiques(fonctionnalités) que les attaquant ont trouvé la faille pour y accéder à l'appareil de la victime, et pourquoi pas de créer un nouvel ensemble de données qui contient plusieurs échantillons.

BIBLIOGRAPHIE

[1] Atelier d'intelligence artificielle. https://github.com/KamiKeys/TalksBlast/blob/master/talentwoman2018/ia.md. consulté le Janury 15,2020.

[2] Intents and intent filters. https://developer.android.com/guide/components/intents-filters. consulté le Janury 22,2020.

[3] Compréhension des rÉseaux neuraux rÉcurrents (lstm, gru). https://mc.ai/understanding-of-recurrent-neural-networks-lstm-gru/, 2018. consulté le Avril 21,2020.

[4] Keras : la bibliothèque python deep learning. https://keras.io/, 2018. consulté le Mars 21,2020.

[5] Numpy. https://numpy.org/, 2019. consulté le Mars 21,2020.

[6] Matplotlib : Visualisation avec python. https://matplotlib.org/, urldate = 2019, 2020. consulté le Mars 21,2020.

[7] pandas python. https://pandas.pydata.org/docs/, 2020. consulté le Mars 21,2020.

[8] Qu'est-ce que scikit-learn ? https://www.codecademy.com/articles/scikit-learn, 2020. consulté le Mars 21,2020.

[9] Tensorflow lite guide. https://www.tensorflow.org/lite/guide, urldate = 2020-03-31, 2020. consulté le May 20,2020.

[10] Mohammed K. Alzaylaee, Suleiman Y. Yerima, and Sakir Sezer. DL-Droid : Deep learning based android malware detection using real devices. *Computers and Security*, 89, 2020.

[11] D. Ashok and Kumar. Intrusion Detection Systems : a Review. *International Journal of Advanced Research in Computer Science*, 8(8) :356–370, 2017.

[12] Radoniaina Andriatsimandefitra and Ratsisahanana Caract. Caracterisation et detection de malware Android basees sur les flux d'information . Radoniaina Andriatsimandefitra Ratsisahanana. 2015.

[13] Daniel Arp, Michael Spreitzenbarth, Malte Hübner, Hugo Gascon, and Konrad Rieck. Drebin : Effective and Explainable Detection of Android Malware in Your Pocket. (February), 2014.

[14] Saba Arshad, Munam A. Shah, Abdul Wahid, Amjad Mehmood, Houbing Song, and Hongnian Yu. SAMADroid : A Novel 3-Level Hybrid Malware Detection Model for Android Operating System. *IEEE Access*, 6 :4321–4339, 2018.

[15] M.G. Avram. Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective. *Procedia Technology*, 12 :529–534, 2014.

[16] Elmouatez Billah, Mourad Debbabi, Abdelouahid Derhab, and Djedjiga Mouheb. MalDozer : Automatic framework for android malware detection using deep learning. *Digital Investigation*, 24 :S48–S59, 2018.

[17] Winston Bumpus. NIST Cloud Computing Standards Roadmap. *NIST Cloud Computing Standards*, pages 1–3, 2013.

[18] Y C A Padmanabha Reddy, P Viswanath, and B Eswara Reddy. Semi-supervised learning : a brief review. *International Journal of Engineering & Technology*, 7(1.8) :81, 2018.

[19] D Raúl Lara Cabrera. Machine learning techniques for Android malware detection and classification. (March), 2019.

[20] Ayushi Chahal and Preeti Gulia. Machine learning and deep learning. *International Journal of Innovative Technology and Exploring Engineering*, 8(12) :4910–4914, 2019.

[21] Sarika Choudhary and Nishtha Kesswani. Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT. *Procedia Computer Science*, 167(2019) :1561–1573, 2020.

[22] cloudstor. Unsw-nb15 csv files. `https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys?path=%2FUNSW-NB15%20-%20CSV%20Files`, 2020. consulté le Juin 30,2021.

[23] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10) :78–87, 2012.

[24] Umer Farooq. Android Operating System Architecture. (July) :2–8, 2018.

[25] Canadian Institute for Cybersecurity. Nsl-kdd dataset. `https://www.unb.ca/cic/datasets/nsl.html`, 2020. consulté le Juin 25,2021.

[26] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media.

[27] Aurélien Géron. Machine Learning avec Scikit-Learn. page 256, 2017.

[28] Edmund A. Gleason. Plant Intrusion Detection Systems. *Plant Engineering (Barrington, Illinois)*, 35(11) :250–252, 1981.

[29] Alexandre Gonfalonieri. How to build a data set for your machine learning project. `https://towardsdatascience.com/how-to-build-a-data-set-for-your-machine-learning-project-5b3b871881ac`, 2019. consulté le Juin 30,2021.

[30] Abhilash Hota and Paul Irolla. Deep neural networks for android malware detection. *ICISSP 2019 - Proceedings of the 5th International Conference on Information Systems Security and Privacy*, pages 657–663, 2019.

[31] Information and Irvine Computer Science University of California. Kdd cup 1999 data. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`, 1999. consulté le Juin 30,2021.

[32] A. G. Khachaturyan and G. A. Shatalov. Elastic Energy of Heterophase Systems of Lamellar Inclusions. *Physics of Metals and Metallography*, 31(6) :1–5, 1971.

[33] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems : techniques, datasets and challenges. *Cybersecurity*, 2(1), 2019.

[34] Bastien L. Tensorflow : tout savoir sur la bibliothèque machine learning open source. `https://www.lebigdata.fr/tensorflow-definition-tout-savoir`, 2018. consulté le Mars 21,2020.

[35] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553) :436–444, 2015.

[36] Solène LIMOUSIN. Android et ios. `https://www.supinfo.com/articles/single/9003-android-ios`, 2019. consulté le Janury 29,2020.

[37] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems : A survey. *Applied Sciences (Switzerland)*, 9(20), 2019.

[38] A. Martín, R. Lara-Cabrera, and D. Camacho. A new tool for static and dynamic Android malware analysis. (September) :509–516, 2018.

[39] Sana Mushtaq. Data preprocessing in detail. `https://developer.ibm.com/technologies/data-science/articles/data-preprocessing-in-detail/`, 20119. consulté le April 4,2021.

[40] Abdelmonim Naway and Yuancheng Li. International Journal of Computer Science and Mobile Computing A Review on The Use of Deep Learning in Android Malware Detection. *International Journal of Computer Science and Mobile Computing*, 7(12) :42–58, 2018.

[41] Abdelmonim Naway and Yuancheng Li. International Journal of Computer Science and Mobile Computing A Review on The Use of Deep Learning in Android Malware Detection. *International Journal of Computer Science and Mobile Computing*, 7(12) :42–58, 2018.

[42] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation Functions : Comparison of trends in Practice and Research for Deep Learning. pages 1–20, 2018.

[43] Gururaj Ramachandra, Mohsin Iftikhar, and Farrukh Aslam Khan. A Comprehensive Survey on Security in Cloud Computing. *Procedia Computer Science*, 110(2012) :465–472, 2017.

[44] scikit-learn developers. Normalization. `https://scikit-learn.org/stable/modules/preprocessing.html#normalization`, 2020. consulté le Juin 28,2021.

[45] scikit-learn developers. sklearn.preprocessing.minmaxscaler. `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html`, 2020. consulté le Juin 25,2021.

[46] Rajani Sharma and Rajender Kumar Trivedi. Literature review : Cloud Computing –Security Issues, Solution and Technologies. *International Journal of Engineering Research*, 3(4) :221–225, 2014.

[47] Nirupam Sutradhar, Madhuwesh Kumar Sharma, and G. Sai Krishna. Cloud Computing : Security Issues and Challenges. *Lecture Notes in Electrical Engineering*, 692(3) :25–32, 2021.

[48] Rajni Tewatia, Asha Mishra, and Mark Embrechts Presented. Introduction To Intrusion Detection System Review. *International Journal of Scientific & Technology Research*, 4(5) :219–223, 2015.

[49] Varun Krishna Veeramachaneni. Security Issues and Countermeasures in Cloud Computing Environment. *International Journal of Engineering Science and Innovative Technology (IJESIT)*, 4(5) :82–93, 2015.

[50] Sarang V.Hatwar and R. K. Chavan. Cloud Computing Security Aspects, Vulnerabilities and Countermeasures. *International Journal of Computer Applications*, 119(17) :46–53, 2015.

[51] Haohan Wang and Bhiksha Raj. On the Origin of Deep Learning. pages 1–72, 2017.

[52] Suleiman Y. Yerima, Sakir Sezer, and Igor Muttik. Android malware detection using parallel machine learning classifiers. *Proceedings - 2014 8th International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST 2014*, (Ngmast) :37–42, 2014.

[53] Xue Ying. An Overview of Overfitting and its Solutions. *Journal of Physics : Conference Series*, 1168(2), 2019.