République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Abderrahmane Mira

Faculté de la Technologie

Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Réseau et Système de Télécommunication.

Thème

Détection d'intrusions et classification des attaques réseaux par les réseaux de neurones.

Préparé par :

BOUAICHI Katia

BENNAI Amine

Dirigé par :
M.DIIBOUNE

Examiné par :
M.TOUNSI

M.BELLAHCENE

Année universitaire : **2021/2022**

REMERCIEMENTS

En premier lieu, nos remerciements vont à notre encadrant, **Abdelhani DIBOUNE** qui a été d'une aide précieuse, ses remarques pertinentes et son sens du détail nous ont beaucoup aidé.

Nos remerciements vont aussi aux **membres du jury** pour avoir accepté d'examiner notre travail et de l'enrichir par leurs propositions.

Nous tenons aussi à remercier tous nos amis pour leur présence et leur soutien moral surtout **Youba BOUNCEUR et Rafik BENCHALLAL** .

Katia BOUAICHI et Amine BENNAI

DÉDICACE

Katia BOUAICHI

A mes chers parents
ceux qui m'ont doté d'une éducation digne, quoi que je fasse ou je dise. Votre affectation me couvre, votre bienveillance me guide, votre présence à mes cotés et votre amour a toujours été ma source de force pour affronter les différentes défis et faire de moi ce que je suis aujourd'hui.

A mon cher frère **AIMAD** et mes deux sœurs adorables que j'aime énormément **HANIFA** et **NASSIMA**, à l'effort qu'ils ont suscité en moi.

A Mon cher **YOUBA**, je te remercie infiniment, tu étais toujours à mes cotés pour m'aider et m'encourager. Ceci est ma profonde gratitude pour ton soutien.

A Mon binôme **AMINE** qui n'a jamais cessé de mettre ces efforts pour ce travail.

DÉDICACE

Amine BENNAI

A mes chers parents,
Qui n'ont ménagé aucun effort, qui ont toujours su m'écouter, me comprendre, partager mes hauts et mes bas.

A mes frères et soeurs,
AZIZ, YANIS, YOUBA, HANANE, SARAH
Merci pour tout ce que vous faites pour moi au quotidien, c'est grâce à vous que j'ai eu le courage de continuer.

A ma chère **INES**, ta présence et ton écoute ont été le meilleur des soutiens.

A ma binôme **KATIA**
C'était merveilleux de travailler à tes côtés et d'avoir partager cette experience avec toi.

GLOSSAIRE ALPHABÉTIQUE

C

CPU : *Central Proccessing Unit.*

D

DMZ : *Demilitarized Zone.*

DT : *Decision Tree.*

H

HIDS : *Host Intrusion Detection Systems.*

I

ICMP : *Internet Control Message Protocol.*

IDS : *Intrusion Detection System.*

IP : *Internet Protolcol.*

IPS : *Intrusion Prevention System.*

K

KDD : *Knowledge Discovery Databases.*

KNN : *K-Nearest Neighbors.*

N

NIDS : *Network Intrusion Detection System.*

NIPS : *Network Intrusion Prévention Système.*

O

OSI : *Open Systeme Interconnexion.*

T

SVM : *Support Vector Machine.*

T

TCP : *Transmission Control Protocol.*

U

UDP : *User Datagram Protocol.*

TABLE DES FIGURES

LISTE DES TABLEAUX

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE

De nos jours, l'utilisation des réseaux informatiques dans les entreprises est devenue une ressource essentielle et inévitable ce qui permet de faciliter la communication ainsi que le bon fonctionnement de l'entreprise, mais certains utilisateurs d'internet provoquent des risques importants dans le domaine de la sécurité informatique. Ces derniers peuvent exploiter les vulnérabilités des réseaux et systèmes pour réaliser leurs attaques d'où la protection des informations est devenue une nécessité. La sécurité informatique consiste généralement à s'assurer que les ressources matérielles ou logicielles d'une organisation ne sont utilisées que dans le cadre d'un réseau informatique. Plusieurs formes d'attaques ont récemment étaient découvertes , l'accès aux données confidentielles par des intrusions est un problème majeur et doit être résolu. Les attaques se viennent principalement de l'intérieur comme de l'extérieur (Internet). Pour cela, les administrateurs déploient des solutions de sécurité efficaces. Face à toutes ces menaces, la sécurité optimale des systèmes informatiques et des réseaux est devenue un enjeu stratégique et pour assurer cette sécurité, différents outils ont étaient utilisés, tels que les pare-feu et les anti-virus.

Dans la plupart de ces derniers temps, malheureusement les systèmes anti-virus et les pare-feu sont rendues très fragiles et ils ont échoué de faire face à ces nouvelles attaques ou menaces modernes. De ce fait, les systèmes de détection d'intrusion sont apparus, ils jouent un rôle indispensable dans la protection, l'atténuation et les préventions des agressions, dans ce cas l'IDS est un bon choix pour mieux protéger le réseau informatique.

Dans le premier chapitre on va définir qu'un IDS est une méthode de sécurisation des réseaux informatiques. En observant et en analysant le trafic, cette technologie permet aux utilisateurs du réseau de comprendre les différentes vulnérabilités et attaques dans les réseaux selon les couches TCP/IP. Ensuite, on à consacré a présenter l'architecture globale d'un IDS, le mode de fonctionnement de ce dernier. Ainsi que la classification des IDS et enfin la méthode de détection d'une intrusion.

Le deuxième chapitre sera une présentation et une explication global des deux méthodes de classification des attaques tels que la régression logistique et les réseaux de neurones en présentant son concept et la façon comment il fonctionne, de plus nous avons pu avoir un aperçu des composants que les rendent artificiellement intelligents, ainsi que les algorithmes que nous allons utiliser dans notre projet.
Dans le troisième chapitre on va simuler quelques tests sur certains méthodes de classifications, tels que les réseaux de neurones, arbres de décision et les k-plus proches voisins en utilisant une base de données pour faire des classifications afin d'établir un système de détection d'intrusions basé sur l'analyse de comportement de ces connexions TCP/IP.

CHAPITRE I

GÉNÉRALITÉS SUR LES SYSTÈMES DE DÉTECTION D'INTRUSION

I.1 Introduction

Aujourd'hui les réseaux et les systèmes informatiques sont devenus des dispositifs qui ont un rôle centrale dans les entreprises en améliorant l'efficacité du leurs fonctionnement interne. Par contre, l'augmentation des attaques sur ces réseaux ont devenu de plus en plus très compliqués.
Sur ce premier chapitre on va baser sur deux points essentiels, le premier contient les principaux notions de base de la sécurité informatique, dont les quels on va mettre des définitions de la sécurité informatique ainsi que les attaques sur les couches de réseau TCP/IP.
Le deuxième point, on va parler sur le système de détection d'intrusion, sa définition, son principe de fonctionnement, la classification des IDS et la méthode de détection d'une intrusion ainsi que ses avantages et inconvénients.
On va finir ce chapitre par les définitions des différentes méthodes de classification des attaques par apprentissage automatique.

I.2 Vulnérabilités et attaques sur les systèmes de détection d'intrusion

I.2.1 Définition d'une vulnérabilité

Dans la sécurité informatique on l'as définit comme une faille ou une faute crée durant le développement du système ou durant l'opération, ce qui facilite à l'attaquant de détruire ce système en touchant ces propriétés de sécurité : la confidentialité, la disponibilité ou encore plus toucher l'intégrité du système qui permet à l'attaquant de modifier ou supprimer ces données afin de créer une intrusion.

I.2.2 Définition d'une intrusion

On peut définir une intrusion comme un accès sans autorisation à un système informatique ou un réseau, c'est-à-dire toute sorte de dépassement des dispositifs sécuritaires en liaisons avec ces systèmes.

I.2.3 Définition d'une attaque réseau

Il s'agit de toute action visant à menacer la sécurité de l'information et à compromettre au moins un attribut de la sécurité informatique (disponibilité, confidentialité, intégrité,...). Il s'agit d'une tentative d'intrusion.

I.3 Les attaques des différentes couches

Il était nécessaire de développer une norme internationale afin de permettre l'interconnexion des réseaux. A cet effet, des efforts de modélisation ont été menés pour séparer les différents types de fonctions des systèmes de traitement de l'information numérique en plusieurs niveaux, notamment dans le cadre de la transmission en réseau.Il existe deux types modèles principaux tels que : le modèle OSI et TCP/IP.
Dans cette partie on s'intéresse beaucoup plus au TCP/IP qui englobe les couches de modèle OSI en quatre couches, mais malheureusement il existe plusieurs attaques qui interromprent son fonctionnement par exemple pour :

I.3.1 La couche Application

La couche applicaction est une couche de haut niveau.Elle correspond directement avec l'utilisateur,elle englobe les couches application, présentation et session de modéle OSI. Elle s'assure que les données soient correctement "empaquetées" pour qu'elles soient lisibles par la couche suivante[1]. Parmi les attaques qu'on peut trouver dans cette couche, on distingue les plus connues :

- **Password sniffing** : c'est une attaque qui est utilisée pour voler des noms d'utilisateur et des mots de passe sur un réseau. Cela se produit souvent sur le protocole http tellement il n'est pas sécurisé (il n'est pas crypté), ce qui facilite aux attaquants de détecter le mot de passe réseaux.La solution préventive pour éviter cette attaque c'est le cryptage, le certificat (IDS, https).
- **Password cracking** :Cette technique consiste à essayer plusieurs mots de passe afin de trouver le bon. Elle peut s'effectuer à l'aide d'un dictionnaire des mots de passe les plus courants (et de leur variantes), ou par la méthode de brute force (toutes les combinaisons sont essayées jusqu'à trouver la bonne). Cette technique longue et fastidieuse, souvent peu utilisée à moins de bénéficier de l'appui d'un très grand nombre de machines[2].
- **Buffer overflow** : Elle permet à l'attaquant d'occuper illégitimement une partie de la mémoire du système depuis l'espace mémoire alloué à une application légitime vulnérable.Cette attaque se fait en donnant des données qui sont supérieur au Buffer pour écraser le code source de service.

I.3.2 La couche Transport

Elle assure l'arrivé des paquets dans l'ordre et sans erreurs, en échangeant les accusés de réception de données et en retransmettant les paquets perdus. Cette communication est dite de type de bout en bout. Pour faciliter la communication, au lieu de définir les noms d'applications, nous

définissons les ports de communication spécifiques à chaque application. Elle gère deux protocoles de livraison d'informations : UDP "sans connexion" assure des services de bout en bout, et le TCP "avec connexion" assure des services de datagramme peu fiables[1]. Parmi les attaques qui peuvent être éxposées à cette couche, on peut trouver :

- **Scanning** : Est un processus dans lequel un utilisateur malveillant envoie des sondes à une machine victime pour déterminer les ports (applications) ouverts, le type de système d'exploitation et la version, les services exécutés sur la machine victime et les vulnérabilités.
- **DoS** : Denial of service est le terme anglophone du déni de service qui veut dire une attaque réseau restreint l'utilisation légale des ressources d'un serveur avec la surcharge de requêtes. Avec les limites des ressources des ordinateurs sur la puissance de calcul ou mémoire par exemple, le programme peut se rend lent ou au pire des cas va se bloquer, donc indisponible. En effet l'attaque DoS consiste des diverses techniques pour saturer ces ressources et faire en sorte qu'un serveur ou un réseau ne soit plus disponible pour ses utilisateurs légitimes, ou restreindre l'accès dans certains cas.

I.3.3 La couche Internet

Son rôle est de permettre l'injection des paquets dans n'import quel réseau et d'assurer leurs acheminement d'une manière indépendante les uns sur les autres jusqu'a la destination. Dans cette couche l'adresse IP est utilisée pour le routage des informations entre les réseaux et que le protocole de contrôle ICMP met à disposition des outils de dépistage d'erreurs et de signalisation[3]. Nous distinguons quelques attaques tels que :

- **IP spoofing** : une attaque au niveau de la couche 4, considérée comme étant une réservation d'identité. L'attaquant abolir les actions d'un administrateur puis il prend son adresse IP afin d'intercepter les données transmise par ce dernier.
- **ICMP tunneling** : c'est une technique d'attaque de commandement et de contrôle (c2) qui passe secrètement le trafic malveillant à travers des défenses périmétriques.

I.3.4 La couche Accès réseau

Cette couche regroupe les deux couches, physique et liaison de données du modèle OSI. Elle assure la bonne gestion du médium (détection de collisions), permet l'acheminement des informations entre l'émetteur etle destinataire au niveau des adresses MAC, L'accès au canal lorsque c'est le même canal pour plusieurs machines et finalement la délimitation de la trame[3]. Parmi les attaques par l'adresse MAC, on peut trouver :

- **MAC spoofing** : est une technique de piratage et d'usurpation d'identité consistet à changer l'adresse MAC d'un appareil réseau. La modification de l'adresse MAC permet de contourner, accéder aux listes de contrôle sur les serveurs ou les routeurs, soit en masquant un ordinateur sur un réseau ou lui permettant d'usurper l'identité d'un autre ordinateur. Chaque périphérique réseau est identifié par une adresse unique appelée adresse MAC qui est gravée dans sa carte d'interface réseau (NIC)[4].
- **MAC poisoning** : L'ARP ne peux pas vérifier si une machine est liée à plusieurs interfaces au même temps ou pas , ce qui facilite à l'attaquant de viser directement le swich afin de le rendre un Hub , car l'attaquant fait le croire qu'il y' a une adresse MAC appartient à plusieurs interfaces.
- **MAC flooding** : exploite la vulnérabilité résultante du fonctionnement de base du commutateur. Ce dernier place les entrées dans la table CAM où sont stockées l'adresse MAC et le mappage des ports des périphériques qui communiquent à travers celle là. Sur la base de cette table, le commutateur décide à quel port envoyer le trafic. La vulnérabilité réside dans le fait que la taille de cette table est limitée. Une fois cette table remplie, il n'y aura pas d'espace pour les adresses MAC des nouveaux appareils qui essaient de communiquer. Par la suite, le commutateur commencera à agir comme un HUB Ethernet, ce qui signifie qu'il s'agira de trafic de transfert vers tout les ports physiques. Un attaquant peut facilement capturer cette communication et analyser son contenu, par exemple, dans Wireshark.
- **Man in the middle** : cette attaque permet de détourner le trafic entre deux stations. Imaginons un client C communiquant avec un serveur S. Un pirate peut détourner le trafic du client en faisant passer les requêtes de C vers S par sa machine P, puis transmettre les requêtes de P vers S et inversement pour les réponses de S vers C. Totalement transparente pour le client, la machine P joue le rôle de proxy. Elle accédera ainsi à toutes les communications et pourra en obtenir les informations sans que l'utilisateur s'en rende compte[5].

I.4 Mesures préventives et méthodes de réparation

I.4.1 Mesures préventives

Le protocole 802.1X : Est un standard IEEE permet de contrôler l'accès aux périphériques d'infrastructures réseau. Il fournit une couche de sécurité pour l'utilisation des réseaux filaires et sans fil. Cette sécurité se traduit souvent par une authentification préalable à l'accès au réseau.il nécessite donc la présence d'un serveur d'authentification qui peut être un serveur RADIUS (Microsoft, Cisco,...) ou un produit libre comme FreeRADIUS ou encore un serveur TACACS dans le monde fermé des équipements Cisco.

Un pare-feu : Le pare-feu ou Firewall en anglais, c'est un mécanisme indispensable dans la sécurité informatique des entreprises et même dans des simples ordinateurs. Le Pare-feu propose donc un véritable contrôle sur le trafic réseau de l'entreprise. Il permet d'analyser, de sécuriser et de gérer le trafic réseau. Et ainsi d'utiliser le réseau de la façon pour laquelle il a été prévu et sans l'encombrer avec les activités inutiles, et d'empêcher une personne sans autorisation d'accéder à ce réseau de données. Il s'agit ainsi d'une passerelle filtrante comportant au minimum les interfaces réseaux[6].

Un proxy : Composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou contrôler leurs échanges. Par extension, nous appelons également des dispositifs "proxy", tels que des serveurs mis en place pour assurer le fonctionnement de tels services.
Le proxy est au niveau de la couche application. Une erreur courante consiste à utiliser la commande traceroute (ou tracert sous Windows) pour essayer de voir le proxy. Il n'apparaît pas car la commande utilise le protocole réseau IP de couche 4, il n'y a donc aucun moyen de connaître le proxy.

I.4.2 Détection réparation

Les IDS et les IPS font tout les deux partie de l'infrastructure réseau. Ces derniers font de références les paquets de réseau à une base de données de menaces réseau qui développent des signatures d'attaques connues et tous les paquets similaires à ces signatures[7].

✧ **Les IDS (Intrusion Detection Systems)** : Ils Analysent et surveillent le trafic réseau à la recherche des signes indiquant que des pirates utilisent des menaces réseau connues pour infiltrer ou voler vos données réseau. Pour pouvoir détecter divers types de comportement tels que les violations de la politique de sécurité, les logiciels malveillants et les analyseurs de port des systèmes. l'IDS compare l'activité réseau en cours à une base de données d'attaques connues[7].

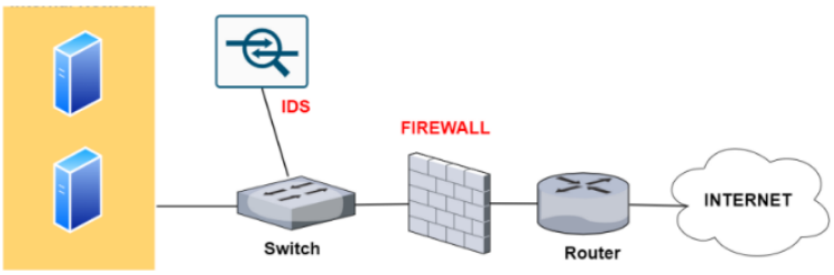


FIGURE I.1 – Emplacement d'un IDS dans un réseau[8].

Avantages d'un IDS

- Un IDS peut détecter toute activité suspecte, y compris l'activité provenant de l'intérieur du réseau.
- IDS peut également détecter les faux comptes et volés prendre des mesures immédiates sur ces intrusions.
- Un IDS peut aussi détecter quand les individus au sein de la mauvaise utilisation du réseau des ressources de réseau.
- La détectin de nouvelles attaques.

Inconvénients d'un IDS

- En raison des vitesses de transmission extrêmement élevées, il peut dépasser de manière significative les vitesses d'écriture des disques durs les plus rapides du marché et les vitesses de traitement des processeurs. Il n'est donc pas rare que des paquets ne soient pas reçus par l'IDS.
- Un IDS peut donner une fausse alarme.
- Un IDS consomme beaucoup de ressources CPU.
- Vulnérabilité de déni de service : un attaquant pourrait tenter de provoquer un déni de service au niveau du système de détection d'intrusion.

✧ **Les IPS (Intrusion Prevention Systems) :** Est un IDS, Par feu et antivirus. On le définit comme étant une forme de sécurité qui englobe à la fois l'analyse et la réponse. Comme les technologies IPS surveillent les flux de paquets en détectant les menaces identifiées, elles peuvent également servir à imposer l'utilisation de protocoles sécurisés et refuser l'utilisation de protocoles non sécurisés tels que les versions antérieures de SSL ou les protocoles utilisant des chiffrements faibles[7].

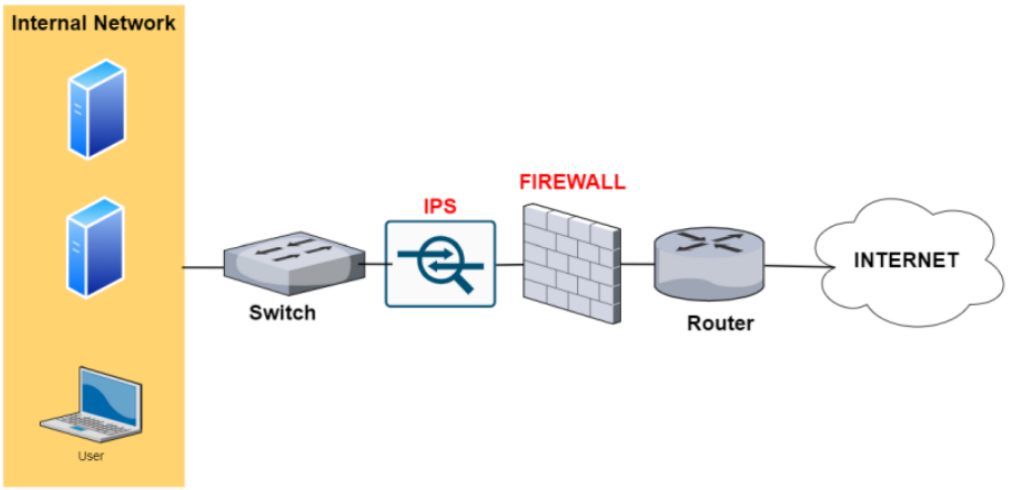


FIGURE I.2 – Emplacement d'un IPS dans un réseau[8].

Au temps où L'IDS ne modifie les paquets réseau en aucune façon. Tout comme un pare-feu qui bloque le trafic en se basant sur l'adresse IP, l'IPS empêche la transmission du paquet en fonction de son contenu. Donc la principale différence entre les deux tient au fait que l'IDS est un système de surveillance, alors que l'IPS est un système de contrôle.

La différence entre l'IDS et l'IPS :

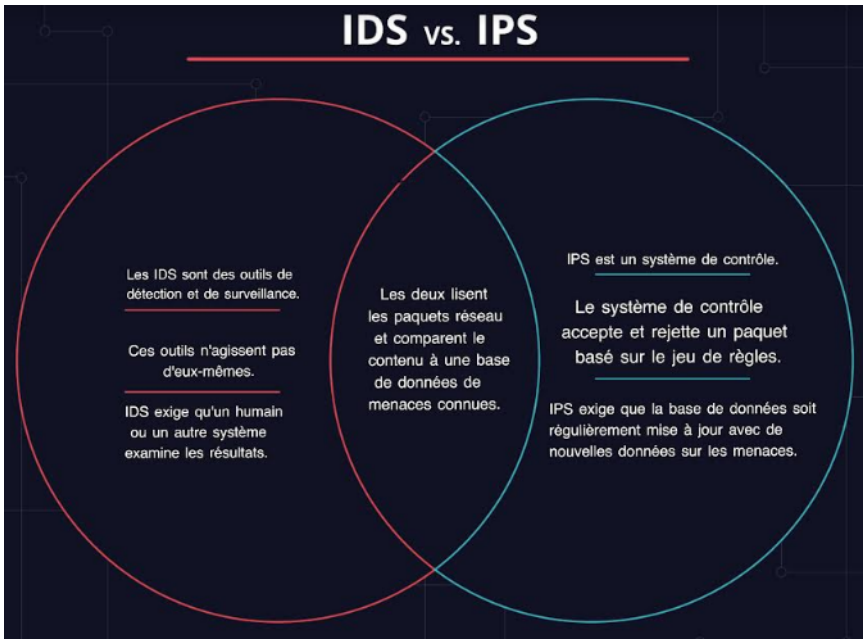


FIGURE I.3 – Différence entre l'IDS et l'IPS[8].

I.5 Les systèmes de détection d'intrusion

I.5.1 Architecture d'un IDS

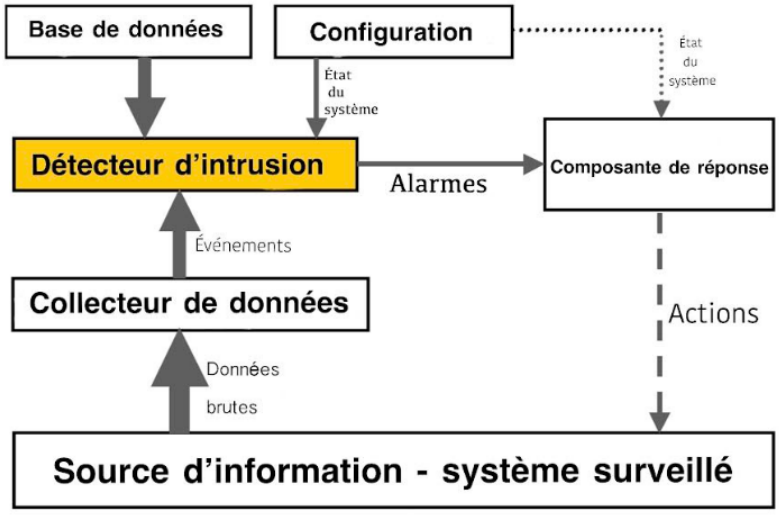


FIGURE I.4 – Architecture basique d'un système de détection d'intrusion.

- Le dispositif de collecte de données est responsable de la collecte des données à partir du système surveillé (source d'information).
- Le moteur d'analyse de détection d'intrusion traite les données collectées à partir des capteurs pour identifier les activités intrusives.
- La base de données ou base de connaissance contient des informations collectées par les capteurs, mais dans un format prétraité (par exemple, base de connaissances des attaques et de leurs signatures, données filtrées, profils de données, etc.). Ces informations sont généralement fournies par des experts en réseau et en sécurité.
- Le dispositif de configuration fournit des informations sur l'état actuel du système de détection d'intrusion (IDS).
- Le composant Réponse (Response component) initié des actions lorsqu'une intrusion est détectée. Ces réponses peuvent être automatisées (actives) ou impliquer une interaction humaine (inactive).

I.5.2 Type des systèmes de détection d'intrusion

Les pirates utilisent une variété d'attaques, comme certains utilisent les failles de réseau(NIDS) et autres les failles de programmation (HIDS). Voilà pourquoi La détection d'intrusion doit être effectuée à plusieurs niveaux. Par conséquent, nous détaillerons les deux IDS ci-dessous :

✧ **Systéme de détection d'intrusion réseau (NIDS) :**

Un NIDS (Network Intrusion Detection System) est un IDS orienté réseau qui analyse le trafic qui circule au niveau IP afin d'y repérer des tentatives d'attaques. Il est composé de sondes qui capturent le trafic acheminées sur le réseau et d'un moteur pour analyser ce dernier[2].

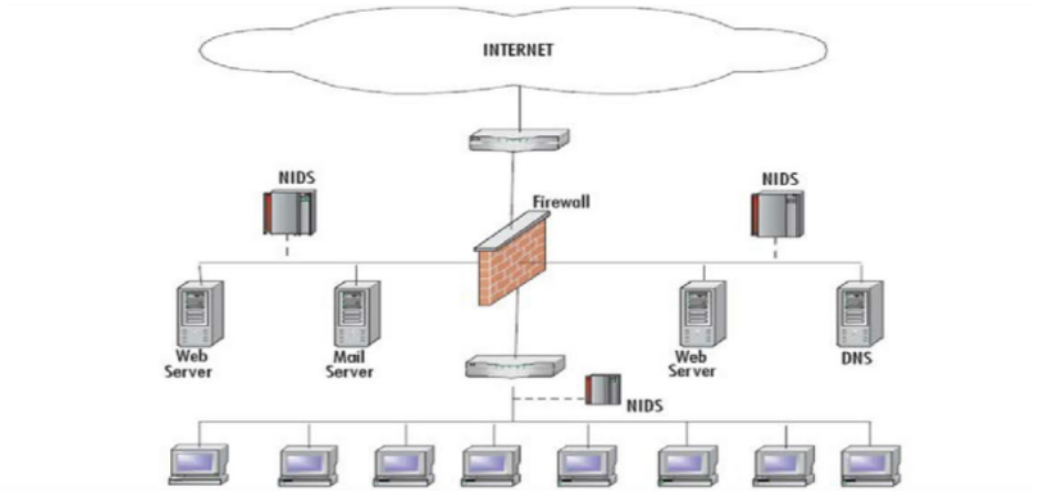


FIGURE I.5 – Architecture d'un NIDS proposé par le groupe IDWG.

IDWG : le Groupe de Travail Détection d'Intrusion à pour objectif de définir les données formats et procédures d'échange pour le partage d'informations d'intérêt pour systèmes de détection et de réponse aux intrusions, et aux systèmes de gestion qui peuvent avoir besoin d'interagir avec eux. Le fonctionnement de la détection d'intrusion le groupe coordonnera ses efforts avec d'autres groupes de travail de l'IETF.

Les avantages d'un NIDS

L'NIDS présente plusieurs avantages :

- Possibilité de filtrer le trafic permettant donc une surveillance discrète du réseau.
- NIDS est très sur contre l'attaque et peut être caché à plusieurs attaquants.
- Les NIDS permettent aux administrateurs de protéger les périphériques non informatiques, tels que les pare-feu, les serveurs d'impression,...
- Le déploiement de NIDS a peu d'impact sur un réseau existant. Les NIDS sont habituellement des dispositifs passifs qui écoutent sur un fil de réseau sans interférer l'opération normale de ce dernier. Ainsi, il est habituellement facile de monter en rattrapage un réseau pour inclure l'IDS avec l'effort minimal.

- Le NIDS offre l'avantage de la furtivité et n'ajoute aucune surcharge au réseau en termes de trafic.

Les avantages d'un NIDS

- NIDS ne peut pas analyser les informations crypté (cas d'utilisation des VPN).
- Il est difficile à traiter tous les paquets circulant sur un grand réseau. De plus il ne peut pas reconnaître des attaques pendant le temps de haut trafic.
- Quelques NIDS provoquent des paquets en fragments. Ces paquets mal formés font devenir l'IDS instable et l'accident.
- Plusieurs avantages de NIDS ne peuvent pas être appliqués pour les commutateurs modernes.

✧ **Système de détection d'intrusion hôte (HIDS) :**
Un HIDS (Host Intrusion Detection System) est un agent logiciel sur la machine à protéger afin d'analyser en temps réel les flux de trafic de cette machine ainsi que les fichiers journaux. Contrairement à un NIDS, un HIDS ne protège donc que le système local. Il est capable de détecter les changements dans les fichiers et dans le système d'exploitation de la machine hôte[2].

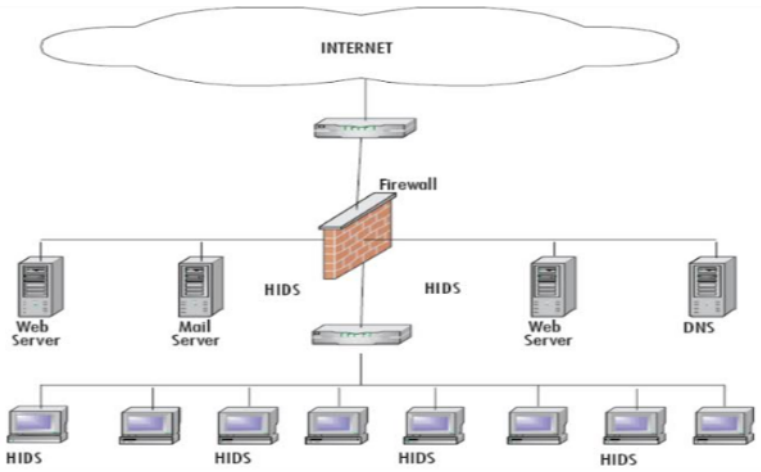


FIGURE I.6 – Architecture d'un réseau HIDS.

Les avantages d'un HIDS

- Pouvoir surveiller des événements locaux jusqu'au host, détecter des attaques qui ne sont pas vues par NIDS.

- Marcher dans un environnement dans lequel le trafic de réseau est encrypté, lorsque les sources des informations de host-based sont générées avant l'encrypte des données ou après le décrypte des données au host de la destination.
- les HIDS sont un outil de «dernière ligne de défense» utilisé pour parer aux attaques manquées par le NIDS.
- Les HIDS peuvent détecter le cheval de Troie ou les autres attaques relatives à la brèche intégrité de logiciel.

Les avantages d'un HIDS

- HIDS est difficile à gérer, et des informations doivent être configurées et gérées pour chaque host surveillé.
- Puisque au moins des sources de l'information pour HIDS se résident sur l'host de la destination par les attaques, l'IDS peut être attaqué et neutralisé comme une partie de l'attaque.
- HIDS n'est pas bon pour la surveillance qui s'adresse au réseau entier parce que le HIDS ne voit que les paquets du réseau reçus par ses hosts.
- HIDS peut être neutralisé par certaine attaque de DoS.

Les systèmes de détection d'intrusions machine et réseau ont leurs propres avantages et limites. La meilleure façon de protéger un réseau consiste à combiner les deux technologies (IDS hybride).

✧ **Emplacement d'un IDS dans un réseau :** Pour bien positionner un système de détection d'intrusion il faut d'abord identifier les ressources à protéger et ce qui est le plus susceptible d'être attaqué.Il existe pas mal de choix sur les endroits stratégiques où il convient positionner un IDS :

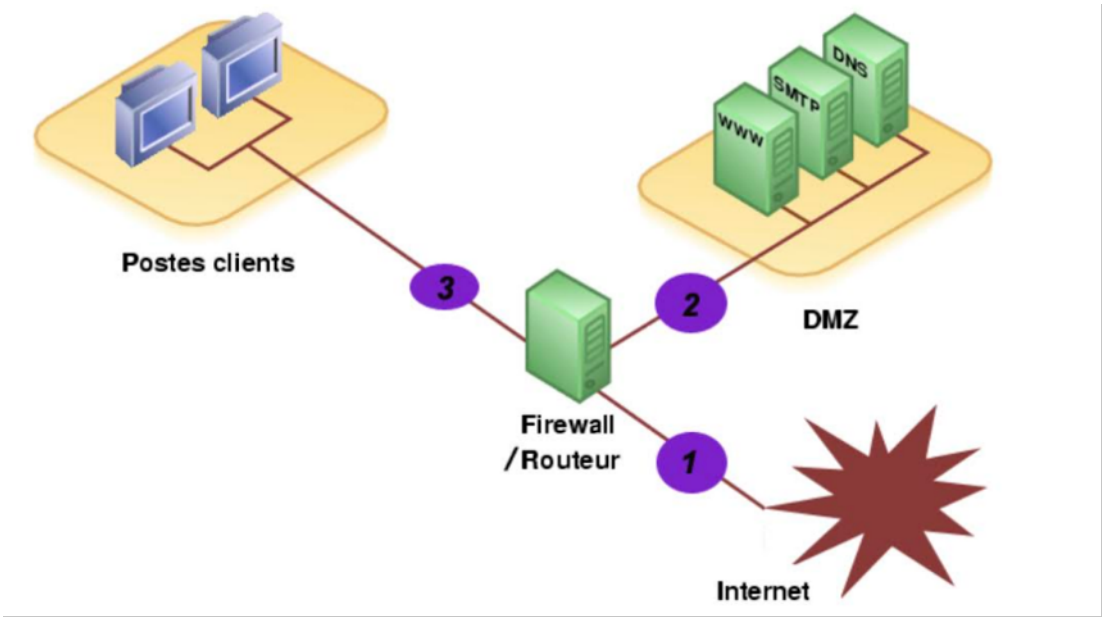


FIGURE I.7 – Emplacement d'un IDS dans un réseau[9].

- **Emplacement (1)** : À cette position, l'IDS peut détecter toutes les attaques frontales de l'extérieur devant le pare-feu. Ainsi, de nombreuses alertes sont générées et le logs sera difficile à consulter.
- **Emplacement (2)** : Si l'IDS est placé dans la DMZ, il détectera les attaques qui ne sont pas filtrées par le pare-feu et qui se situent dans un certain niveau de compétence. Il est plus facile de voir les logs ici car les attaques bénignes ne sont pas recensées.
- **Emplacement (3)** : Ici, l'IDS peut détecter les attaques internes à partir du réseau local de l'entreprise. Étant donné que 80% des attaques viennent de l'intérieur, c'est une bonne idée de placer les attaques là-bas. De plus, si des chevaux de troie ont pollué le parc informatique (navigation peu méfiante sur internet) ils peuvent être facilement identifiés ici puis éradiqués.

I.5.3 Caractéristiques d'un IDS

Parmi les caractéristique qui spécifie l'ids d'autres formes de sécurité, on trouve :

- **Source de données :**
  Les systèmes de détection d'intrusions conduisent leurs analyses sur des données qui proviennent de diverses sources. Nous pouvons établir deux catégories principales :

✧ Les données issues du trafic réseau : on mettant le IDS avant le par feu cella oblige le trafic de passé par eux même afin de lui faire connaitre les différentes attaques existantes et les signaler à l'administrateur pour mettre à jour la base de données.

✧ Les données issues des systèmes lorsque on a le IDS comme étant un logiciel (logs du noyau, appels système, alertes de sécurité, etc.).

La source des données à analyser est à mettre en parallèle avec la localisation de l'IDS lui-même : un HIDS se focalisera d'avantage sur l'analyse de données provenant du système alors qu'un NIDS se concentrera sur des données réseau. Avec l'apparition des IDS distribués, une troisième catégorie a émergé. En effet, les données à analyser ne sont plus directement les données brutes relevées par le système de sécurité lui même, mais des données récoltées par d'autres sondes. Ces sondes rapatrient leurs alertes vers des nœuds qui seront alors responsables de l'agrégation de l'ensemble des alertes.

- **Politique de détection** : on peut regroupés en juste deux catégories la plupart des méthodes de détection :

  Approche à scenario (approche par signature) : c'est en fonction du comportement actuel de l'utilisateur quelle détecte l'intrusion, sans prend en considération ses actions passées. Le terme « approche basée sur des scénarios » vient du fait que nous nous appuyons sur les connaissances techniques utilisées par les attaquants pour déduire des scénarios typiques.

  Approche comportementale : contrairement à l'approche par scénario, l'approche comportementale c'est en fonction du comportement passé d'un utilisateur quelle détecte l'intrusion. L'idée principale est d'essayer de crée un profil d'utilisateur en fonction de ses habitudes de travail et de déclencher une alarme lorsqu'un événement hors normes se produit. Ces IDS sont dans le but de détecter toute une action inhabituelle et non pas l'identification des attaques spécifiques.

- **Mode de fonctionnement :** afin de mesurer, d'évaluer et d'estimer la fréquence d'utilisations des systèmes de détection d'intrusions, on distingue deux façon tel que la surveillance continue et l'analyse périodique :

  La surveillance continue : La surveillance continue fait référence à la surveillance et à l'analyse continues, ininterrompues et en temps réel d'un système (observation du système basée sur le temps).
  La plupart de ces derniers systèmes de détection d'intrusion analysent d'une manière continu les données de la machine locale ou les paquets réseau pour fournir une détection en temps quasi réel. Ceci est nécessaire dans les environnements sensibles (confidentialité). Cependant, il s'agit d'un processus coûteux en calcul car tout ce qui se passe sur le système doit être analysé dynamiquement.

L'analyse périodique : Cette méthode nécessite au système de détection d'intrusion l'analyse des sources de données d'une façon périodique dans l'intention de chercher l'éventuelle intrusion ou anomalie passée. Dans des contextes peu sensibles cela peut être suffisant (on fera alors une analyse journalière, par exemple).

- **Architecture d'un IDS :**

  Centralisé : La première approche à émerger dans la littérature est l'approche centralisée, qui consiste en deux types de nœuds : les nœuds collecteurs et les nœuds d'analyse. Les nœuds collecteurs sont des IDS qui s'exécutent localement et signalent les intrusions à un nœud central, qui corrèle les informations.

  Distribué : L'inconvénient de l'approche précédente est l'existence d'un ou plusieurs points de défaillance. Si ces points sont désactivés, la corrélation ne peut plus avoir lieu. C'est pourquoi l'approche distribuée a été proposée. Celle-ci se repose sur une structure entièrement composée de nœuds à la fois collecteurs et analyseurs. Ces nœuds détectent localement les attaques et sont capables de corréler les informations des nœuds voisins pour détecter les attaques coordonnées.

- **Réponse à l'attaque :** Il existe deux types de réponses à l'attaque selon l'IDS utilisé. Les réponses informatives qui s'appliquent à tous les IDS, les réponses défensive plus ou moins qui sont mises en place.

  La réponse passive : Les réponses de défense enregistrent les détections d'intrusion dans des fichiers journaux (logs) que Security Manager analysera. Pour corriger une faille de sécurité afin d'empêcher que l'attaque enregistrée ne se reproduise, certains IDS peuvent enregistrer l'intégralité de la connexion identifiée comme malveillante. Mais cela n'empêche pas directement l'attaque de se produire.

  La réponse active : Le but d'une réponse active est d'arrêter une attaque lorsqu'elle est détectée. Deux techniques sont disponibles : reconfigurer le pare-feu et interrompre la connexion TCP.

I.6 Méthodes de classification des attaques par apprentissage automatique supervisée

I.6.1 Random forest :

Afin de résoudre les problèmes de classification et de régression, l'algorithme Random forest qui est un algorithme d'apprentissage automatique supervisé est utilisé pour construire des arbres de décision sur différents échantillons et prend leur vote majoritaire pour le classement et la moyenne en cas de régression.
Random Forest manipule aussi les données qui comportent les variables continues (cas de la régression) et des variables catégorielles (cas de la classification) dont les quelle donne les meilleurs résultats.

Random trees :

Est une méthode de classification et de prédiction basée sur la méthodologie de classification et d'arbre de régression, cette méthode de prédiction utilise le partitionnement récursif pour diviser les enregistrements d'entraînement en segments avec des valeurs de champ de sortie similaires. Le nœud commence par examiner les champs d'entrée à sa disposition pour trouver le meilleur fractionnement, qui est mesuré par la réduction d'un indice d'impuretés qui résulte de la scission. Cette dernière définit deux sous-groupes, dont chacun est ensuite divisé en deux autres sous-groupes, et ainsi de suite, jusqu'à ce que l'un des critères d'arrêt soit déclenché.

Random bagging :

C'est un algorithme d'ensemble qui s'adapte à plusieurs modèles sur différents sous-ensembles d'un ensemble de données d'entraînement, puis combine les prédictions de tous les modèles. La forêt aléatoire est une extension de bagging qui sélectionne également de manière aléatoire des sous-ensembles de caractéristiques utilisées dans chaque échantillon de données.

I.6.2 Séparateurs Vaste Marge SVM :

Les machines à vecteurs de support sont des modèles de machine learning supervisés qui a connu cette dernière décennie un grand développement en théorie et en application, centrés sur la résolution de problèmes de discrimination et de régression mathématiques. Elle repose sur un fondement théorique solide basé sur le principe de maximisation de la marge, ce qui lui confie une grande capacité de généralisation. Les SVMs ont été utilisées avec succès dans plusieurs domaines

tels que la reconnaissance des visages, des textes manuscrits, de la parole, ...etc. Ils sont appréciés pour leur simplicité d'usage [10].

I.6.3 K-nearest neighbors :

K plus proches voisins est une approche de classification supervisée intuitive les plus connus, souvent utilisée dans le cadre de la machine learning pour résoudre à la fois des problèmes de classification et de régression. Le principe est que les données connues sont disposées dans un espace défini par les caractéristiques sélectionnées. Lorsqu'une nouvelle donnée est fournie à l'algorithme, l'algorithme compare les classes des k données les plus proches pour déterminer la classe de la nouvelle données[10].

I.6.4 Régression logistique :

Par définition, la régression logistique est un modèle d'analyse statistique très souvent utilisé dans l'apprentissage automatique et l'intelligence artificielle qui étudie la relation entre un ensemble de variables prédictives et une variable binomiale. En effet, la régression logistique est un modèle linéaire généralisé qui utilise une fonction logistique comme fonction de lien. Elle est considéré comme l'un des modèles analytiques multi variés les plus faciles à analyser et à déchiffrer. Il peut prendre plusieurs formes, c'est-à-dire logique, binaire ou polynomiale. Les modèles de régression logistique utilisent également l'optimisation des coefficients pour prédire la probabilité qu'un événement puisse ou non se produire. Plus précisément, lorsque la valeur prédite est inférieure à un seuil prédéfini, il y a une forte probabilité que l'événement ne se produise pas. À l'inverse, si la valeur est supérieure au même seuil de départ, l'événement est susceptible de se produire dans ce cas. Il est important de préciser que le résultat de cette probabilité varie toujours entre 0 et 1[11].

I.6.5 Réseaux de neurones :

Les réseaux neuronaux, également connus sous le nom de réseaux de neurones artificiels (ANN) ou de réseaux de neurones à impulsions (SNN) constituent un sous-ensemble de l'apprentissage machine et sont au cœur des algorithmes de l'apprentissage en profondeur. Leur nom et leur structure sont inspirés par le cerveau humain. En effet, ces réseaux imitent la façon dont les neurones biologiques s'envoient mutuellement des signaux.
Les réseaux de neurones artificiels (ANN) sont constitués de différentes couches de nœud (ou neurone artificiel), contenant une couche en entrée, une ou plusieurs couches cachées et une couche en sortie. Chaque nœud, ou neurone artificiel, se connecte à un autre et possède un poids et un seuil associés. Si la sortie d'un nœud est supérieure à la valeur de seuil spécifiée, ce nœud est activé

et envoie des données à la couche suivante du réseau. Sinon, aucune donnée n'est transmise à la couche suivante du réseau[10].

I.7 Conclusion

La sécurité est toujours un sujet de débat car aucune solution fiable n'a été trouvée. Ce chapitre nous à amèner à découvrir quelques attaques réseaux et les vulnérabilités que les attaquants prennent comme une porte afin d'accéder aux systèmes. On a parler aussi sur les systèmes de détection d'intrusion, leurs capacités et leur fonctionnement, il nous a apparaît clairement que ces systèmes sont aujourd'hui indispensables aux entreprises pour sécuriser leurs machines, accomplir et aider les autres tâches des appareils. Nous venons de montrer que la détection d'intrusion réseau n'est pas en concurrence avec d'autres systèmes de sécurité, au contraire, ils sont complémentaires. Cependant, vous devez savoir qu'il n'y a pas des systèmes informatiques en matière de sécurité. Car cette technologie n'est pas encore arrivée à maturité et ces différents types de systèmes de détection d'intrusion ont des avantages, des inconvénients et surtout des limites.

CHAPITRE II

CLASSIFICATION DES ATTAQUES RÉSEAUX PAR LA RÉGRESSION LOGISTIQUE MULTICLASSE ET LES RÉSEAUX DE NEURONES

II.1 Introduction

On définit l'apprentissage automatique comme étant une application artificielle intelligente qui sert aux systèmes d'apprendre et de se développer d'une manière automatique, cela se fait selon ça propre expérience sans besoin d'être programmé. L'apprentissage machine vise le développement des programmes informatique qui peuvent rentrer à des données dans le but d'apprendre par eux même.

Plusieurs algorithmes de différentes classes sont proposés pour faciliter la machine learning.Dans ce chapitre on va baser sur deux, la régression logistique et les réseaux de neurones. Leurs principes est presque le même et débute par des observations ou des données comme des exemples, une expérience de première utilisation ou même des instructions afin de trouver des modèles et apprendre à bien choisir la bonne décision à l'avenir sur la base de données fournit.

Le but principal est de développer des machines qui peuvent apprendre automatiquement sans l'aide humaine et d'adapter les actions en conséquence.

II.2 Modéle d'apprentissage

Considérons un échantillon de n observations indépendantes : avec Y la variable qui nous cherchent à prédire ou à expliquer à l'aide d'une ou de plusieurs variables prévisionnelles $X = x_1, ..., x_m$ qui sont exclusivement continues ou binaires, dont m est caractéristiques. Elles participent de façon additive à l'équation de régression et la pondération de chaque variable prévisionnelle est évaluée par son coefficient de régression. Puisque nous sommes dans le cadre de la régression logistique binaire, Y représente la valeur de la variable dépendante dichotomique prenant soit la valeur 0 pour présenter une classe négative (activité normale) ou 1 pour présenter la classe positive (catégorie d'attaque)[11].
Soit $\Omega \subset \{0, 1\}$,Un ensemble de n observations de Y.

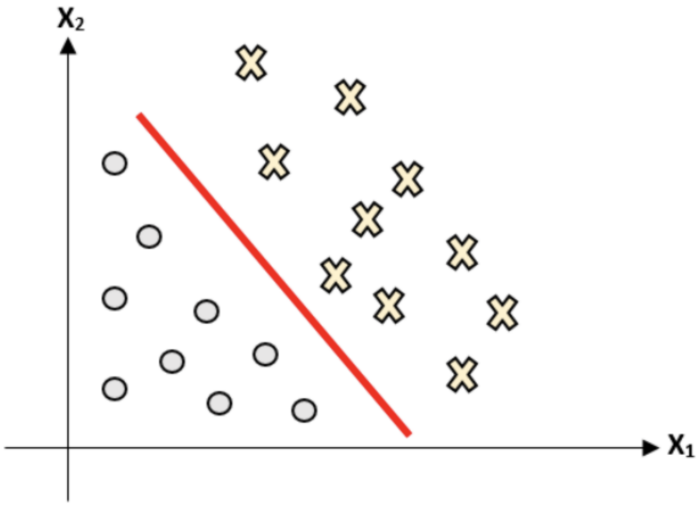
$$\mathbb{P}(y = 1) : \text{probabilité que } y = 1 \qquad \mathbb{P}(y = 0) : \text{probabilité que } y = 0$$

II.3 Frontières de décision

Une façon de classification qui sert à découper l'espace des caractéristiques en régions, pour chercher un seuil ou une frontière de décision $Z = \Theta^T X$ (qui peut être linéaire, polynomiale, etc.) de sorte que tout les points d'une région donnée soient destinés à recevoir la même étiquette. Les régions sont définies par leurs limites, c'est pourquoi nous voulons que la régression trouve des lignes de séparation au lieu d'un ajustement. Les figures ci-dessous montrent comment des

exemples d'apprentissage de la classification peuvent être considérés comme des points colorés dans l'espace de caractéristiques :

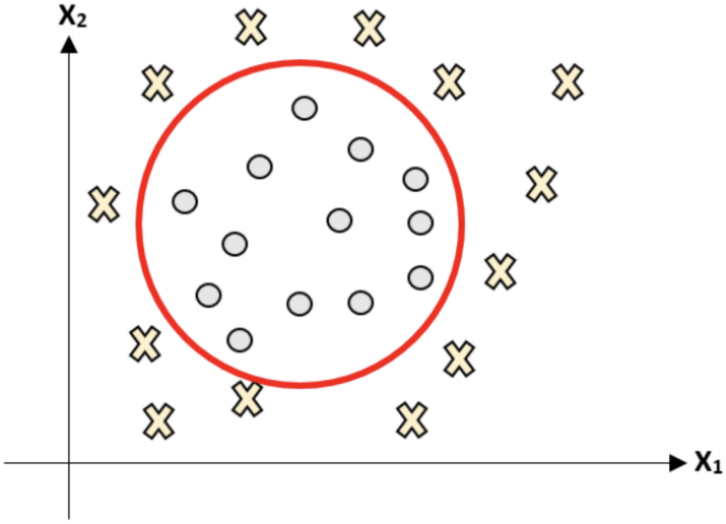
Exemple 1 :



$$z = \Theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

FIGURE II.1 – Séparateur linéaire entre deux classes.

Exemple 2 :



$$z = \Theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

FIGURE II.2 – Séparateur circulaire entre deux calsses.

La régression logistique à pour objectif de trouver une fonction h_θ qui est définie sur $[0,1]$ tel que $h_\theta(X) = g\left(\theta^T X\right) = \mathbb{P}\left(y = 1|X,\theta\right)$.D'où θ est le paramétre de séparation (θ_0=1) et g généralement une fonction logistique. $g : x \longrightarrow \dfrac{\gamma}{1+\alpha e^{-\beta x}}$

On comprend donc qu'on attend de notre fonction h_θ qu'elle soit une probabilité comprise entre 0 et 1 et que le seuil que nous définissons correspond à notre critère de classification, généralement il est prit comme valant 0.5 :

$h_{\theta(X)} > 0.5 \longrightarrow y = 1$

$h_{\theta(X)} < 0.5 \longrightarrow y = 0$

La fonction qui remplit le mieux ces conditions est la fonction sigmoïde, définie sur $\mathbb{R}$ à valeur dans $[0,1]$. Elle s'écrit de la manière suivante : $g : x \longrightarrow \dfrac{1}{(1+e^{-x})}$; $\alpha = \beta = \gamma = 1$

II.4 Recherche de paramètres optimaux

L'optimisation est la meilleure méthode pour chercher la meilleure frontière, elle sert à minimiser la distance cumulative entre la prédiction et l'observation effective , c'est se qu'on appelle la fonction objective.

Soit un ensemble d'apprentissage de m échantillons :

$$\left\{\left(x^{(1)},\ y^{(1)}\right),\ \left(x^{(1)},\ y^{(1)}\right),.....,\left(x^{(m)},\ y^{(m)}\right)\right\}$$

d'où : $x^{(i)} = \begin{pmatrix} 1 \\ x_1^{(i)} \\ \cdots \\ x_n^{(i)} \end{pmatrix},\ y \in \{0,1\}$

$$h_\Theta(X) = g\left(\Theta^T X\right) = \frac{1}{1+e^{-\Theta^T X}}; \quad \Theta \in R^{n+1} \tag{II.1}$$

On définit la fonction du coût suivante :

$$D\left(h_\Theta\left(x^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(h_\Theta\left(x^{(i)}\right)\right) & \text{si } y^{(i)} = 1 \\ -\log\left(1 - h_\Theta\left(x^{(i)}\right)\right) & \text{si } y^{(i)} = 0 \end{cases} \tag{II.2}$$

$$\begin{cases} D\left(h_\Theta\left(x^{(i)}\right), y^{(i)}\right) \to 0 & \text{si } |h_\Theta\left(x^{(i)}\right) - y^{(i)}| \to 0 \\ D\left(h_\Theta\left(x^{(i)}\right), y^{(i)}\right) \to \infty & \text{si } |h_\Theta\left(x^{(i)}\right) - y^{(i)}| \to 1 \end{cases} \tag{II.3}$$

La fonction objective est alors :

$$J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} D\left(h_\Theta\left(x^{(i)}\right), y^{(i)}\right) \tag{II.4}$$

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \log\left(h_\Theta\left(x^{(i)}\right)\right) y^{(i)} + \log\left(1 - h_\Theta\left(x^{(i)}\right)\right)\left(1 - y^{(i)}\right) \tag{II.5}$$

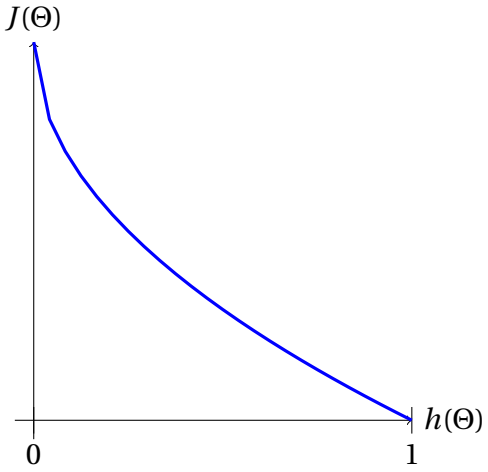
La fonction de coût en général est ce qui indique à quel point l'algorithme est proche de la prédiction ou de la classification correcte. Elle permet d'évaluer la performance d'un réseau de neurones au cours de l'apprentissage. On la calcule en fonction de la différence entre les valeurs attendues et réelles. On peut la résumée de la manière suivante :

— L'équation est la somme de –log(h_Θ) et 1–log(h_Θ) multipliée par $1/m$, d'où m est le nombre d'échantillons.

— Il faut minimiser la fonction objective J .

— L'équation est la somme des deux cas $y = 0$ et $y = 1$.

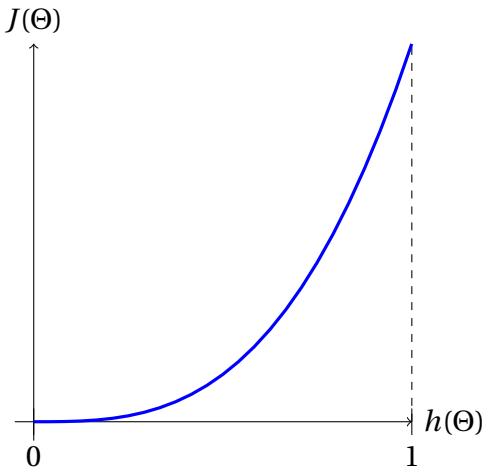
➢ **Le premier cas y=1 :**



La fonction objective en fonction de h(Θ)

- On remarque que la fonction objective (J) augmente lorsque la valeur de h (Θ) se diminue et se dirige vers 0 parce qu'elle est trop petite et loin de la valeur réelle 1.
- Au moment où la valeur de h (Θ) se rapproche de 1, le pourcentage de l'erreur se diminuer et sera égale à zéro sur la valeur de 1.

➢ **Le deuxième cas y=0 :**



La fonction objective en fonction de h(Θ)

- Lorsque $h(\Theta)$ diminue jusqu'à 0, la valeur de J sera faible parce que la valeur réelle est toujours à 1.
- La valeur de J sera trop forte tandis que $h(\Theta)$ se rapproche de 1.

Nous essayons de trouvés les valeurs de Θ, l'ajout d'un biais peut-être inutile avec les data sets de notre étude afin d'éviter le surapprentissage (overfiting) qui survient lorsque le modèle essaie de trop s'adapter aux données, donc il fonctionne bien sur les données d'apprentissage mais pas de validation.il effectue alors de mauvaises prédictions sur les données qu'il n'a pas vues auparavant lors des phases d'apprentissage. Cette situation augmente considérablement les marges d'erreur.Il existe Plusieurs méthodes auxquelles on peut avoir recours pour régler ce problème dont :

- La cross-validation : une méthode statistique permet de tirer plusieurs ensembles de validation d'une même base de données et ainsi d'obtenir une estimation plus robuste, avec biais et variance, de la performance de validation du modèle.
- L'ajout des données d'entrainement : pour une bonne variété de données et pour que la machine puisse mieux généraliser,il faudra ajouter des données d'entrainement et enlever tous ceux qui ne servent pas au modèle mais notamment ceux qui ont une variance trop faible mais peuvent juste fausser les résultats.
- La régularisation : l'objectif est de réduire les variances afin d'éviter que la machine étudie des modèles trop complexes et ne devienne trop flexible.Différentes techniques sont existées tel que : régularisation L1, régularisation L2, lasso.
- L'early stopping : consiste à stopper intuitivement l'entrainement en cas de risque d'overfiting, donc déterminer la bonne durée d'entrainement.

II.5 Algorithmes d'optimisation

II.5.1 Gradient descent

La descente de gradient (Gradient Descent) est l'un des algorithmes d'optimisation les plus importants de tout apprentissage automatique et de l'apprentissage en profondeur. Il permet de trouver le minimum de n'importe quelle fonction convexe.comme aussi il peut entraîner des modèles de régression linéaire, régression logistique et même des réseaux de neurones.
Nous utiliserons l'algorithme de descente de gradient dans un problème d'apprentissage supervisé pour minimiser la fonction de coût, qui est exactement une fonction convexe (l'erreur quadratique moyenne).
La fonction de gradient descente s'écrit de la manière suivante :

$$\{\theta_i \longleftarrow \theta_i - \alpha \frac{\partial}{\partial_i} J(\Theta)\}$$

- Il faut bien choisir la valeur de α, si il est trop petit on risque de mètre un temps infini pour atteindre notre objectif, par contre si on le choisissait plus grand afin d'apprendre (converger) plus rapidement donc on va osciller autour de notre objectif sans jamais l'atteindre.

- $\frac{\partial}{\partial \Theta_i} J(\Theta)$ est la dérivée partielle de la fonction objective.

- Les deux équations sont en parallèle et se répète en même temps pour tout les θ_i\$.

- On observe aussi que la mise à jour doit être comme celle de gauche et non pas celle de droite.

Gradient conjugué

La méthode du gradient conjugué permet de résoudre les systèmes linéaires dont la matrice est symétrique définie positive. Il s'agit d'une méthode qui consiste à partir d'un vecteur donné x^k et à déterminer à chaque étape un vecteur p^k et un scalaire α^k permettant de calculer x^{k+1} à partir de x^k par :

$$x^{k+1} = x^k + r^k p^k$$

Avec l'objectif de minimiser une fonction (descendre un potentiel).
Avant d'introduire la méthode du gradient conjugué pour résoudre un système linéaire, il est utile d'aborder sommairement les méthodes du gradient[12].

II.5.2 momentum

Le gradient descent a du mal dans les régions ou la surface est beaucoup plus courbée en une dimension plutôt que dans une autre [13], et ils sont communs autour des minimums locaux. Dans ces cas la, SGD oscille à travers les pentes de ces régions et n'achève que des progrès hésitant vers des optimums locaux. Une des méthodes qui peut aider le réseau à se sortir de ces pièges c'est d'utiliser le coefficient du momentum [14]. Lors de l'utilisation du momentum, on pousse une balle en bas d'une colline.

$$\{\theta_i \longleftarrow \theta_i - \alpha \frac{\partial}{\partial_i} J(\Theta)\} + momuntum$$

La balle accumule du momentum alors qu'elle roule vers le bas devenant de plus en plus rapide. La même chose se passe lors de la misa à jour des paramètres. Le momentum augmente pour les dimensions dont le gradient pointe vers la même direction et réduit les mises à jours dont le gradient change de direction. comme résultat, on a une convergence plus rapide et on réduit les oscillation.

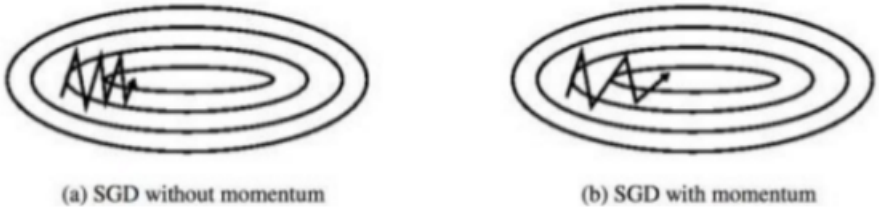


FIGURE II.3 – Accélération de SGD par la méthode du momentum et réduction des oscillation.

II.6 Réseaux de neurones

Les réseaux de neurones peuvent être appliqués à deux problèmes de classification : les problèmes de classification binaire et les problèmes de multi-classification.

- Dans les problèmes de classification binaire , y est égal à 0 ou 1, et le réseau neuronal n'a qu'une seule unité de sortie.

- Dans les problèmes multi-classes, y peut être n'importe quel nombre réel, et le réseau neuronal a plusieurs unités de sortie.

II.6.1 Les fonctions d'activation et le coefficient d'ajustement

La fonction d'activation

Il s'agit d'une fonction qui permet de transformer le signal entrant dans une unité (neurone) en signal de sortie (réponse). Généralement, elle ont un effet "d'aplatissement". Parmi les fonctions d'activation on trouve la fonction logistique sigmoïde et ReLU :

La fonction sigmoïde :

Les fonctions sigmoïdes sont souvent utilisés dans les réseaux de neurones,elle donne une valeur entre 0 et 1, une probabilité. Elle est donc très utilisée pour les classification binaire, lorsqu'un modèle doit déterminer seulement deux labels.
La fonction sigmoïde est la clé pour comprendre comment un réseau de neurones apprend des problèmes complexe [15].

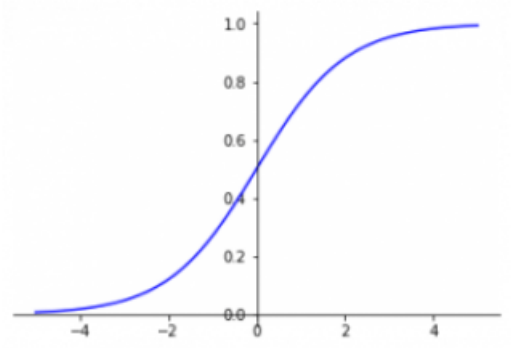


FIGURE II.4 – Fonction d'activation sigmoïde.

Le ReLU :

Les ReLUs laissent toutes les valeurs positives passer inchangées et attribuent simplement 0 aux valeurs négatives. Bien que les nouvelles fonctions d'activation gagnent du terrain, la plupart des réseaux neuronaux actuels utilisent le ReLU ou l'une de ses variantes proches.

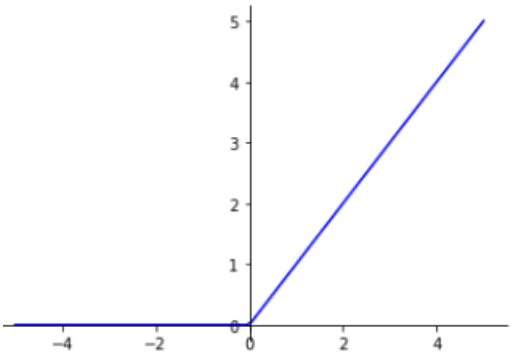


FIGURE II.5 – Fonction d'activation ReLU.

Le poids θ

Le poid est une variable du modèle (coefficient d'ajustement) qui est mis à jour pour améliorer la précision du réseau. Un poids est appliqué à l'entrée de chacun des neurones pour calculer une donnée de sortie. Les réseaux de neurones mettent à jour ces poids de manière continue.

II.6.2 Modélisation d'un neurone

Un neurone formel est une représentation d'une fonction mathématique et informatique d'un neurone biologique, dont la valeur de cette fonction dépend des paramètres appelés coefficients ou poids. Le neurone formel possède généralement plusieurs entrées et la valeur de la fonction est appelée sa "sortie". Les actions excitatrices et inhibitrices des synapses sont représentées la plupart du temps par des coefficients numérique qui sont associés aux entrées. Les valeurs numérique des ces coefficients sont ajustées dans une phase d'apprentissage.
Dans sa version la plus simple, un neurone formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation. La valeur finale obtenue est la sortie du neurone[16].

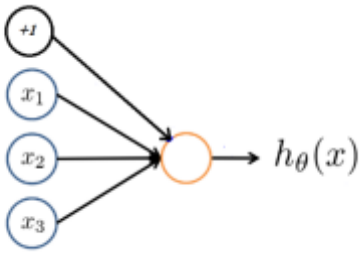


FIGURE II.6 – Schéma d'un systéme à un seul neurone

On a pris l'habitude de représenter graphiquement un neurone comme indiqué sur la FIGURE II.6, elle est faite comme une cellule neurone d'un cerveau humain et représentée d'un cercle orange dans ce schéma.Si on utilise la fonction logistique sigmoide on aura donc :

$$x = \begin{pmatrix} 1 \\ x1 \\ x2 \\ x3 \end{pmatrix}, \Theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = h_\Theta(x) = g(\Theta^T X) = \frac{1}{1 + e^{-\Theta^T x}} \tag{II.6}$$

Le principe est de donner les données x par l'intermédiaire d'une connexion avec certaine force connue sous le nom de poids θ, pour que la celleule peut calculer les valeurs approximatives de

sortie $h_\Theta(X)$ en utilisant la loi de la régression logistique qui est souvent utilisée dans les réseaux de neurones parce qu'elle est dérivable, ce qui est une contrainte pour l'algorithme de rétropropagation.
La valeur de 1 qu'on appelle le bias correspond au neurone dans lequel la fonction d'activation est en permanence égale à 1.Il est en quelque sorte une valeur d'auto apprentissage de notre réseau de neurone.

II.6.3 Modélisation d'un réseau de neurones

Ce type de réseau est dans la famille générale des réseaux à «propagation vers l'avant», c'est-a-dire qu'en mode normal d'utilisation, l'information se propage dans un sens unique, des entrées vers les sorties sans aucune réetroaction. Son apprentissage est de type supervisé, par correction des erreurs. Dans ce cas uniquement, le signal d'erreur est «retropropagé» vers les entrees pour mettre à jour les poids des neurones. Le perceptron multicouche est un des reseaux de neurones les plus utilisés pour des problémes d'approximation, de classification et de prediction. Il est habituellement constitué de deux ou trois couches de neurones totalement connectes[17].

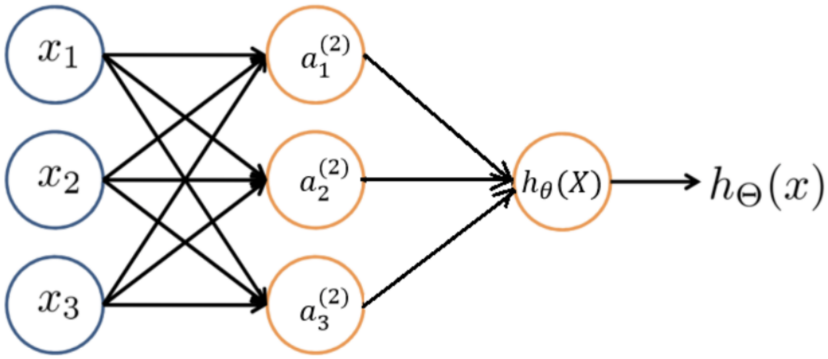


FIGURE II.7 – La perception d'un réseau de neurones

Les réseaux à une seule couche de neurone pouvaient résoudre que des problémes de classification linéairement separables. Les réseaux multicouches permettent de lever cette limitation. On peut méme demontrer qu'avec un réseau de trois couches (deux couches cachées + une couche de sortie), comme celui de la FIGURE II.7, on peut construire des frontieres de décision de complexite quelconque, ouvertes ou fermées, concaves ou convexes,à condition d'employer une fonction de transfert non lineaire et de disposer de suffisamment de neurones sur les couches cachées[17].
Un nombre d'équations mathématique comme cette suie présente le schéma :

$$a_1^{(2)} = g\left(z_1^{(2)}\right) = g\left(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3\right) \tag{II.7}$$

$$a_2^{(2)} = g\left(z_2^{(2)}\right) = g\left(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3\right) \tag{II.8}$$

$$a_3^{(2)} = g\left(z_3^{(2)}\right) = g\left(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3\right) \tag{II.9}$$

$$h_\Theta(x) = g\left(z^{(3)}\right) = g\left(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}\right) \tag{II.10}$$

On remarque que chaque cellule de la deuxième couche à une équation qui se verrouille avec ses entrées associées. Ensuite, il y a une équation pour la sortie qui se rapporte aux trois cellules de la couche cachée.

II.6.4 Forward propagation

La propagation vers l'avant (Forward proagation) fait référence au stockage et au calcul des données d'entrée qui sont transmises vers l'avant à travers le réseau pour générer une sortie. Chaque couche cachée accepte les données d'entrée, les traites conformément à la fonction d'activation et les transmettent à la couche de sortie ou aux couches successives. Les données circulent dans le sens direct afin d'éviter un flux de données de forme circulaire qui ne généra pas de sortie. La configuration qui aide à la propagation vers l'avant est connue sous le nom de réseau à anticipation.

Exemple :

On pose $z^{(2)} = \begin{pmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{pmatrix}, a^{(1)} = x$ alors

$z^{(2)} = \Theta^{(1)} a^{(1)}$

$a^{(2)} = g\left(z^{(2)}\right)$

On ajoute le biais $a_0^{(2)}$ alors

$z^{(3)} = \Theta^{(2)} a^{(2)}$

$h_\Theta(x) = g\left(z^{(3)}\right)$

$a^{(1)}$ c'est la valeur d'excitation de la première couche, qui est la couche d'entrée x, alors que $z^{(2)}$ égale à une matrice $\Theta^{(1)}$ multiplier par $a^{(1)}$, la valeur d'excitation de la deuxième couche $a^{(2)}$ est $g\left(z^{(2)}\right)$, il en va de même pour les dernières couches, où g est la fonction sigmoïde.

II.6.5 Optimisation

Pour faire adapter les paramètres d'un réseau de neurones à un ensemble d'apprentissage donné, nous commencerons d'abord par la fonction de coût. Dans la section précédente, nous avons obtenu l'expression de $J(\Theta)$ pour trouver les paramètres en utilisant la descente de gradient.

Par contre dans le cas des réseaux de neurones on utilise l'algorithme de rétropropagation (Back-propagation), qui permet de calculer rapidement et efficacement les gradients de l'erreur pour chaque neurone du réseau de la dernière couche (couche de sortie). Calcul couche par couche et calcul jusqu'à la deuxième couche, car la première couche est la couche d'entrée (il n'y a pas d'erreur). Ce qui permet ensuite d'ajuster les paramètres de ces réseaux de neurones pour mieux expliquer les données.

Pour optimiser $\Longrightarrow \dfrac{\partial}{\partial \theta_{ij}^{(l)}} j(\Theta) \Longrightarrow$ Backpropagation.

Pour les réseaux de neurones ,il n'y plus une seule unité de sortie, mais K à la place,alors sa fonction de coût s'exprime par :

$$j(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \log\left(h_\Theta\left(x^{(i)}\right)\right)_k y_k^{(i)} + \log\left(1 - h_\Theta\left(x^{(i)}\right)\right)_k \left(1 - y_k^{(i)}\right) \tag{II.11}$$

II.6.6 Rétro-propagation (back propagation)

L'algorithme de formation par rétro propagation a été décrit pour la première fois par Rumelhart et McClelland en 1986 ; il s'agissait de la première méthode pratique de formation des réseaux neuronaux. La procédure originale utilisait l'algorithme de descente de gradient pour ajuster les poids vers la convergence en utilisant le gradient. En raison de cette histoire, le terme "rétro propagation" ou "back-propagation" est souvent utilisé pour désigner un algorithme de formation de réseau neuronal utilisant la descente de gradient comme algorithme de base[18].

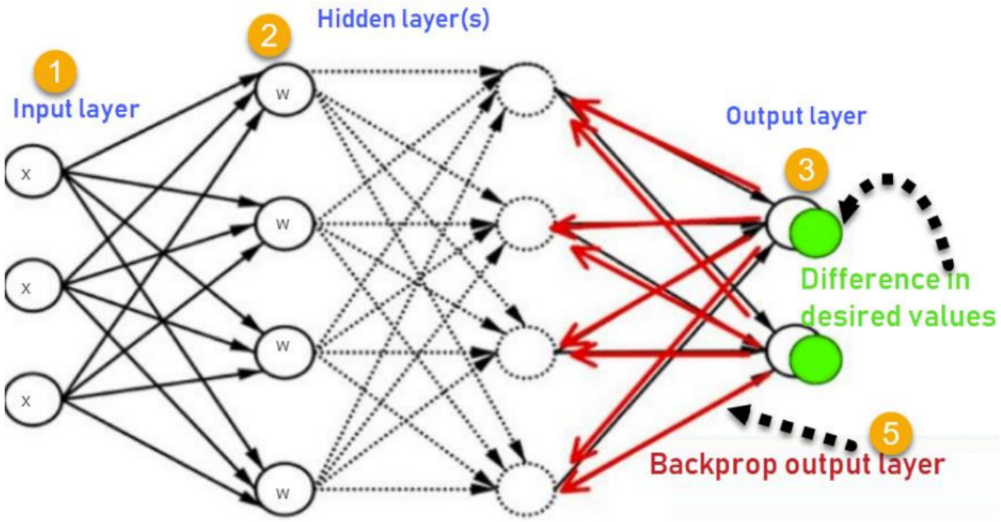


FIGURE II.8 – Rétropropagation.

- Les entrées X, arrivent par le chemin pré-connecté.
- Les entrées sont modélisées à l'aide de poids réels W. Les poids sont généralement choisis de manière aléatoire.
- Calculez la sortie de chaque neurone de la couche d'entrée, aux couches cachées, à la couche de sortie.
- Calculez l'erreur (fonction à minimiser) dans les sorties.
- Revenez de la couche de sortie à la couche cachée pour ajuster les poids de façon à réduire l'erreur.
- Répétez le processus jusqu'à ce que la sortie souhaitée soit atteinte.

Algorithme de rétro-propagation :

Initialiser $\boldsymbol{\Delta_{ij}^{(l)} = 0}$ **pour tout** $\boldsymbol{i, j, l}$
Pour i = 1 à m faire // parcourir le dataset –batch-
$\boldsymbol{a^{(1)} = x^{(i)}}$
Effectuer l'algorithme de forward propagation
$\boldsymbol{\delta^{(L)} = a^{(L)} - y^{(i)}}$.
Calculer $\boldsymbol{\delta^{(L-1)}, \delta^{(L-2)}, \dots, \delta^{(2)}}$ **tel que** $\boldsymbol{\delta^{(l)} = (\Theta^{(l)})^T \delta^{(l+1)}}$

$$\boldsymbol{\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}}$$

$\boldsymbol{D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)}}$ **si** $\boldsymbol{j = 0}$
$\boldsymbol{D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)} + \lambda \Theta_{ij}^{(l)}}$ **si** $\boldsymbol{j \neq 0}$
Finpour
$\boldsymbol{\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = D_{ij}^{(l)}}$

Nous avons maintenant un ensemble d'apprentissage $\left\{ \left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \dots, \left(x^{(m)}, y^{(m)}\right) \right\}$ dans lequel il y a m échantillons d'apprentissage, chacun contient un ensemble d'entrées et un ensemble de sorties, nous désignons le nombre total de couches du réseau de neurones (L) en désignant le nombre d'unités dans la première couche, c'est-à-dire le nombre de neurones (unité de biais non incluse).
Calculons chaque couche séquentiellement de l'arrière vers l'avant δ.En utilisant Δ qui représente l'erreur totale, chaque couche a un correspondant $\Delta^{(l)}$.
On introduise la fonction $J(\Theta)$ pour dériver le résultat des paramètres D, $j = 0$ correspond au terme de décalage.

Une fois calculé $D_{ij}^{(l)}$, vous pouvez obtenir la dérivée partielle de la fonction de coût par rapport à chaque paramètre, car on peut prouver que :
$\frac{\partial}{\partial \theta_{ij}^{(l)}} j(\Theta) = D_{ij}^{(l)}$,

$D_{ij}^{(l)}$ est utilisé pour accumuler la valeur d'erreur et nous aide enfin à calculer la différentielle.

II.6.7 Classification à classes multiples

La classification multi-classes est la technique de classification qui nous permet de catégoriser les données de test en plusieurs étiquettes de classe présentes dans les données formées en tant que prédiction de modèle.Il existe principalement deux types de techniques de classication multi-classes :

– **Un contre Un :**

Dans le classement un contre un, pour le Classe N jeu de données d'instances, nous devons générer le N * (N-1) / 2 modèles de classificateurs binaires. En utilisant cette approche de classification, nous avons divisé le jeu de données principal en un jeu de données pour chaque classe opposée à toutes les autres classes.

– **Un contre tous (un contre reste) :**

Dans la classification un contre tous, pour le jeu de données d'instances de classe N, nous devons générer les modèles de classificateur N-binaires. Le nombre d'étiquettes de classe présentes dans l'ensemble de données et le nombre de classificateurs binaires générés doivent être identiques.

Au lieu d'avoir une unité dans la couche de sortie, on aura plusieurs (K) unités (une par classe)

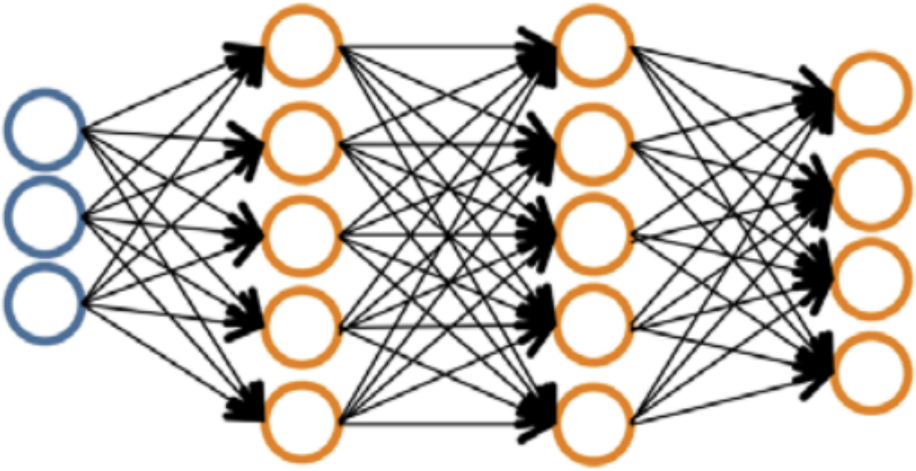


FIGURE II.9 – Unités de sortie multiples.

- Plusieurs étiquettes de classe sont présentes dans l'ensemble de données.
- Le nombre de modèles de classificateur dépend de la technique de classification à laquelle nous appliquons.
- Dans la classification multi-classes un contre un, nous divisons l'ensemble de données principal en un ensemble de données de classification binaire pour chaque paire de classes.

II.7 Conclusion

Dans ce chapitre nous avons abordé dans ca premiére partie la régression logistique qui est une méthode d'analyse multi variée puissante permettant d'obtenir une quantification de l'association entre un phénomène étudiée et chacun des facteurs l'influençant, tout en tenant compte de l'effet simultané des autres facteurs. Elle permet ainsi de contrôler de possibles biais de confusion. Son emploi est rendu aisé par l'utilisation de logiciels statistiques.
Dans la deuxiéme partie nous avons entamer les détails des réseaux de neurones artificiels, inspirés du comportement du cerveau humain. Notamment appliqués en datamining principalement à travers l'apprentissage non supervisé, ils servent à prédire, à identifier et à classifier les données. L'apprentissage, moteur essentiel du système, le permet d'assimiler un traitement d'information à travers une fonction et de le reproduire pour les données qui lui seront ensuite présentées.

CHAPITRE III

TESTS ET RÉSULTATS

III.1 Introduction

Dans ce chapitre nous nous sommes basé sur les réseaux de neurones déjà décris dans le chapitre précédent pour faire une classification supervisée des connexions TCP/IP de la base KDD afin d'établir un système de détection d'intrusions basé sur l'analyse du comportement de ces connexions et permet la distinction entre les "mauvaises" connexions, appelées intrusions ou attaques, et les "bonnes" connexions normales. On commence notre chapitre par une description de la base de données KDD et les étapes de prétraitement que nous avons fait sur cette dernière, ensuite nous etudions les performences des algorithmes de classification pour trouver a la fin le meilleure classifieur.

III.2 Méthodes utilisées

III.2.1 Réseaux de neurones

Le réseau de neurones est composé de plusieurs neurones artificiels, chacun de ces neurones reçoit des données en entrée, puis effectue des calculs sur ces données d'entrée afin de calculer le résultat à la sortie de ce neurone. Les neurones d'un réseau de neurones sont répartis entre différentes couches. Ici, on voit que nous avons utilisé de différentes architectures (jusqu'à 3 couches) en fonction de nombre de neurones par couche (45,50 et 60). Nous changeons la valeur de λ afin de calculer la précision pour les déffirentes manipulation.

III.2.2 K plus proches voisins

KNN est l'algorithme le plus couramment utilisé et l'un des plus simples pour trouver des modèles dans les problèmes de classification et de régression.
Nous exécutons l'algorithme KNN plusieurs fois avec différentes valeurs de K (1,2,3,4 et 5) et avec les mesures de distance les plus populaires (minkowski,cosine,hamming et euclidean) pour choisir le K et la distance qui réduit le nombre d'erreurs que nous rencontrons tout en maintenant la capacité de l'algorithme à faire des prédictions avec précision lorsqu'il reçoit des données qu'il n'a pas vu avant.

III.2.3 Arbre de décision

Les arbres de décision (DT) sont une méthode d'apprentissage supervisé non paramétrique utilisée pour la classification et la régression. L'objectif de l'utilisation d'un arbre de décision est de créer un modèle de formation qui peut être utilisé pour prédire la classe ou la valeur de la variable

cible en apprenant des règles de décision simples déduites de données antérieures (données de formation). Dans les arbres de décision, pour prédire une étiquette de classe pour un enregistrement, nous partons de la racine de l'arbre. Nous comparons les valeurs de l'attribut racine avec l'attribut de l'enregistrement. Sur la base de la comparaison, nous suivons la branche correspondant à cette valeur et sautons au nœud suivant. Plus l'arbre est profond, plus les règles de décision sont complexes et plus le modèle est adapté.

III.2.4 Les métriques

L'Accurancy :

L'accurancy est la mesure de performance la plus intuitive et il s'agit simplement d'un rapport entre l'observation correctement prédite et le nombre total d'observations. On peut penser que, si nous avons une grande accurancy, notre modèle est le meilleur. Oui, l'accurancy est une excellente mesure, mais uniquement lorsque vous avez des ensembles de données symétriques où les valeurs des faux positifs et des faux négatifs sont presque identiques[19].

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

Où le :
TP = vrai positifs : la prédiction et la valeur réelle sont positives.
TN =vrai négatifs : la prédiction et la valeur réelle sont négatives.
FP = faux positifs : la prédiction est positive alors que la valeur réelle est négative.
FN = faux négatifs : la prédiction est négative alors que la valeur réelle est positive.

Précision :

La précision est une mesure permettant d'évaluer les modèles de classification. De manière informelle, la précision est la fraction des prédictions que notre modèle à eu raison.Elle fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire[19].

$$Précision = \frac{TP}{TP + FP}$$

Recall :

Le recall est le rapport entre les observations positives correctement prédites et toutes les observations de la classe réelle[19].

Le recall devrait idéalement être de 1 (élevé) pour un bon classificateur. Il devient 1 uniquement lorsque le numérateur et le dénominateur sont égaux, c'est-à-dire TP = TP + FN, cela signifie également que FN est égal à zéro.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score :

Le F1 score est la moyenne pondérée de la précision et du recall. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. Le F1 score devient 1 uniquement lorsque la précision et le recall sont tous deux à 1. D'autre part,il devient élevé lorsque la précision et le recall sont élevés. Le score F1 est la moyenne harmonique de la précision et du recall et constitue une meilleure mesure que la précision.

$$F1Score = 2 * \frac{Présicion * Recall}{Présicion + Recall}$$

LOSS :

Loss ou bien fonction de perte dans un réseau de neurones quantifie la déférence entre le résultat attendu et le résultat produit par le modèle d'apprentissage automatique. A partir de la fonction de perte, nous pouvons dériver les gradients qui sont utilisés pour mettre à jour les poids. La moyenne de toutes les pertes constitue le coût.

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

III.3 La base de donné utilisée

KDD :

Le KDD (Knowledge Discovery in Databases) est une base de données qui contient des connexions TCP /IP extraites de l'ensemble de données d'évaluation des systèmes de détection d'intrusions. Elle été réalisées en 1998 par l'agence de L'armé américain DARPA (Défense Advanced Research Projects Agency) et AFRL (Laboratoire de recherche de l'armée de l'air), ensuite MIT Lincoln Labs8 a collecté et distribué les ensembles de données pour l'évaluation du système de détection d'intrusions de réseau informatique. Chaque paquet de l'ensemble des données de KDDcup est constitué de 41 champs et est labellisés comme paquet normal ou paquet anormal avec les types d'attaques. Parmi ces champs, 37 sont des champs de type numérique et 4 sont des champs de type non numérique. KDD99 regroupe 37 types d'attaque. Ces attaques sont divisées en quatre grandes classes : DOS, U2R, R2L et Probest. Ces attaques réussies dans les réseaux ont pour conséquence immédiate le blocage du trafic réseaux[18].

Attaques étudiées : Pour notre manipulation, nous avons utilisé environ dix milles instances en regroupant 3 attaques qui sont :

Spyware(1343 instances) :le spyware est un logiciel espion malwares.Il est d'autant plus menaçant qu'il surveille vos données sans se faire surprendre et sans en avoir eu l'autorisation.le spyware enregistre secrètement des informations,copier tout ce que vous saisissez,téléchargez et stockez dans le but de suivre vos activités en ligne sur vos appareils.

Satan(2967 instances) : c'estle malware qui transforme un vulgaire câble en mouchard. Présenter par des chércheurs d'une univérsité Israélien,ils sont parvenus à transformer un câble SATA tout ce qu'il y a de plus générique en un véritable transmetteur radio, et sans devoir procéder à la moindre modification physique du matériel.

Normal avec 4860 instances

– **Etape 01 : Relation entre lambda (évitement de l'overfitting) et la précision :**
Vous retrouvez ici la formule de coût que l'on a utilisé auparavant auquel on a rajouté la somme des carrés des poids du réseau pondérés par le rapport $\frac{\lambda}{m}$. Le facteur λ est appelé le paramètre de régularisation (λ est positif). Le paramétre m est toujours le nombre d'échantillons d'apprentissage de notre modèle :

$$J(\Theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\Theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2 \tag{III.1}$$

λ	0	0.01	0.02	0.04	0.08	0.16	0.32	0.64	1.28	2.56	5.12	10.24
Précision (100%)	99.93	99.93	99.95	99.92	99.92	99.94	99.91	99.92	99.90	99.93	99.96	99.98

TABLE III.1 – la précision d'un réseau de neurone en fonction de lambda.

D'après les résultats obtenus on remarque que la précision augmente a chaque fois qu'on augmente la valeur du paramètre de régularisation λ, ce qui veut dire qu'il ya une relation proportionnelle directe entre eux.
L'augmentation de lambda annule plusieurs thêtas, par conséquent un séparateur plus rigoureux (plus lisse) va être produit pour bien classer les datasets.

– **Étape 2 : Relation entre lambda (évitement de l'overfitting) et le coût :**

A partir des résultats intermédiaires obtenus dans l'étape 1, nous avons tracé ce graphe pour les valeurs 0.01, 0.04, 0.16, 0.64 et 2.56 :

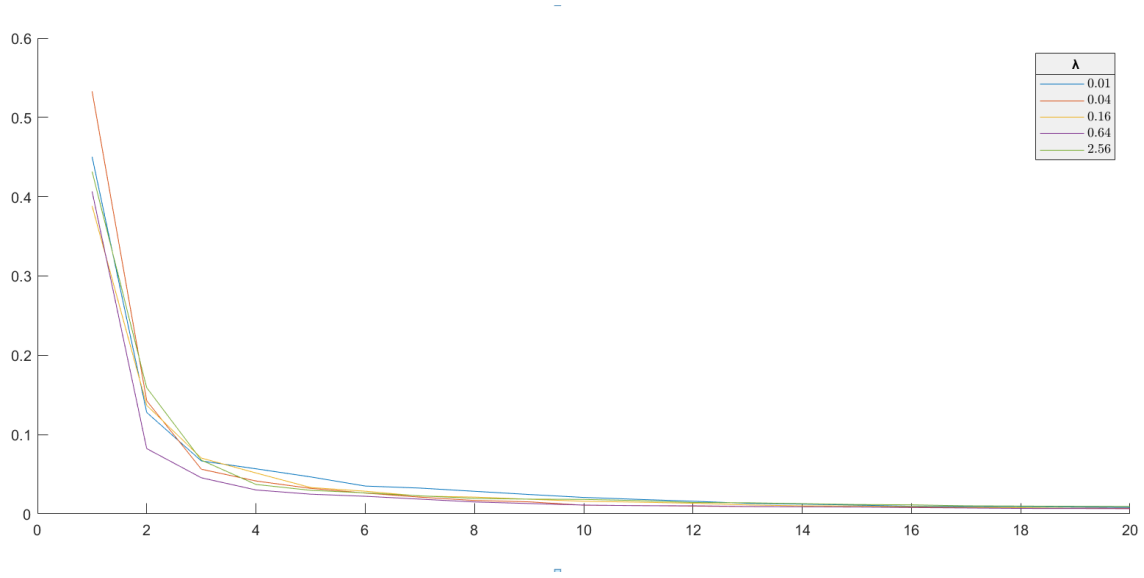


FIGURE III.1 – La relation entre λ et le coût.

L'overfitting est l'un des pires ennemis du Data Scientist. Il s'agit d'un problème fréquemment rencontré en Machine Learning, Cette situation augmente considérablement les marges d'erreur (fonction de coût), alors que la précision est ce que l'on recherche le plus en matière d'analyse de données.

D'aprés la figure ci-dessous, nous remarquons que la valeur de coût diminue en augmentant la valeur de λ. D'où on constate que si λ permet d'éviter l'overfitting ce qui diminue le coût, donc le modèle étudié ne fait pas de faux prédiction (bonne précision).

– **Étape 3 : Relation entre la taille du réseau et la précision :**

– **Nombre de neurones par couche = 45 :**

	01 couche cachée	02 couche cachées	03 couche cachées
$\lambda = 0.01$	99.926	99.792	98.289
$\lambda = 0.04$	99.954	99.858	98.354
$\lambda = 0.16$	99.951	99.914	98.428
$\lambda = 0.64$	99.962	99.916	98.377

TABLE III.2 – La précision d'un réseau de 45 neurones par couche.

– **Nombre de neurones par couche = 50 :**

	01 couche cachée	02 couche cachées	03 couche cachées
$\lambda = 0.01$	99.956	99.891	98.332
$\lambda = 0.04$	99.929	99.901	98.449
$\lambda = 0.16$	99.918	99.883	98.456
$\lambda = 0.64$	99.916	99.873	98.498

TABLE III.3 – La précision d'un réseau de 50 neurones par couche.

– **Nombre de neurones par couche = 60 :**

	01 couche cachée	02 couche cachées	03 couche cachées
$\lambda = 0.01$	98.88	99.903	99.928
$\lambda = 0.04$	98.954	99.878	99.942
$\lambda = 0.16$	99.541	99.902	99.940
$\lambda = 0.64$	99.916	99.924	99.960

TABLE III.4 – La précision d'un réseau de 60 neurones par couche.

D'après les résultats des tableaux et quand on compare entre eux, on remarque que la précision augmente en augmentant les trois paramètres suivants : le paramètre de régularisation λ, le nombre de couches cachées et le nombre de neurones par couche cachée.
De ce fait, on distingue que c'est le nombre de couches cachées qui correspond à une aptitude à traiter des problèmes de non-linéarité, le nombre de neurones par couche cachée. Ces deux choix conditionnent directement le nombre de paramètres (de poids) à estimer et donc la complexité du modèle. Ils participent à la recherche d'un bon compromis biais/variance, c'est-à-dire à l'équilibre entre la qualité d'apprentissage et la qualité de prévision donc la précision d'un réseau de neurones.

III.4 Etude de performance des classifieurs KNN et DT

– **Étape 1 : Relation entre la profondeur de l'arbre et la précision :**

MaxNumSplits	**4**	**8**	**16**	**32**	**40**
%	98.320971	98.459204	98.391773	98.451200	98.540512

TABLE III.5 – la précision en fonction de la profondeur de l'arbre de décision.

Une arbre de décision est un moyen de classer visuellement et logiquement les informations et de prendre des décisions. Grâces aux calculs qu'on a fais et d'aprés les résultas obtenues, on remarque qu'il y a une relation proportionnelle entre la profondeur de l'arbre et la précision. On conclu que plus l'arbre est profond, plus les règles de décision sont complexes et le modèle est adapté.

– **Étape 2 : La relation entre le nombre de calsses et la distance avec la précision de KNN :**

	K				
Distance		2	3	4	5
	minkowski	98.337829	98.465947	98.449090	98.465947
	cosine	98.341200	98.361430	98.395145	98.368173
	hamming	98.320971	98.395145	98.438975	98.408631
	euclidean	98.385030	98.428860	98.540121	98.582461

TABLE III.6 – la précision de KNN en fonction des calsses et de distance.

Pour que l'algorithme fonctionne au mieux sur un ensemble de données particulier, nous devons choisir la métrique de distance et le K les plus appropriées.
Aprés l'analyse des résultats qu'on a on distingue que quand on calcul la distance Euclidean pour k=5, on obtient une meilleur précision et que la marge d'erreur diminue,Et ça ce qui méne l'algorithme à faire des prédictions avec precision lorsqu'il reçoit des données qu'il n'a pas reçu avant.

Comparaison entre les méthodes utilisés :

Le temps de prédiction pour le KNN est très long puisqu'on doit calculer la distance pour toutes les manipulations, contrairement à l'arbre de décision, le résultats s'affiche rapidement, et sa précision est plus élevées par rapport à KNN.
D'autre part, les réseaux de neurones donne en général de meilleurs résultats que les autres algorithmes k-plus proches voisins, arbres de décision.
Pour les calculs de régression, le Machine Learning avec les réseaux de neurones permettent de prédire des tendances sur les paramètres sortant sensiblement du domaine d'apprentissage et réalise des performances assez impressionnantes, ce que peuvent réaliser plus difficilement les autres algorithmes .
On ne peut pas affirmer qu'un tel algorithme est meilleur par rapport à un autre, le choix dépend du problème où il va être appliqué et de ses caractéristiques.

III.5 Conclusion

Nous avons abordé dans ce dernier chapitre quelques attaques de la base de données KDDcup qui a été utilisée pour l'apprentissage et test du modèle de détection d'intrusions généré basé sur les

réseaux de neurones artificiels on utilisants les déffirentes algorithmes de classification tel que les réseaux de neurones, k- plus proche et l'arbre de décision dans le but d'évaluer leurs perfermances.

Aprés avoir testé plusieurs méthodes, nous avons constaté que l'implémentation des réseaux de neurones est la meilleure méthode vu sa rapidité et sa bonne précision .

CONCLUSION GÉNÉRALE

Les attaques informatiques sont en forte hausse ces dernières années et représentent aujourd'hui un risque réel qui menace les réseaux informatiques, les applicatifs et les systèmes d'information des entreprises. En d'autres termes, La découverte périodique et permanente de ces attaques, en temps opportun peut contribuer à les réduire en prenant les moyens de protection nécessaires.
Cela nous a dirigés, dans ce mémoire, vers le développement d'un modèle du système de détection d'intrusions comme moyen d'identification des attaques, dans le but d'éviter leurs dommages. Pour réaliser ce modèle nous nous sommes basé sur les réseaux de neurones et la régression logistique multiclasse.

Nous avons aussi utilisé quelques attaques de la base de données KDDcup comme source de données, et nous avons fait un modèle de classification binaire (deux classes : normale et attaque) pour classifier les connexions TCP/IP en deux classes : attaque ou normal.

En effet, le réglage des paramètres du réseau de neurones a une grande importance pour obtenir de bons résultats. Pour cette raison, Nous avons effectué plusieurs expériences afin de choisir les meilleurs paramètres (nombre d'itérations, nombre de couches cachées et nombre de neurones dans chaque couche) qui nous donnent les meilleurs résultats en termes de taux de réussite. Avec une comparaison entre deux autres méthodes (KNN et DT) pour savoir la quelle donne un taux de réussite élevé, d'où nous avons constaté que les réseaux de neurones reste la méthode efficace pour ces calculs.
Le choix d'une méthode appropriée dépend fortement de l'application, la nature des données et les ressources disponibles. Une analyse attentive des données aide à bien choisir le meilleur algorithme. Il n'existe pas un algorithme qui peut répondre à toutes les demandes.
La majorité des objectifs tracés dans de ce travail ont été atteints, mais il reste toujours des perspectives et des améliorations possibles qui peuvent encore être réalisr dans le future.

BIBLIOGRAPHIE

[1] ZOUAOUI, S. (2012). Implémentation et validation de la pile TCP/IP de Microchip sur un dsPIC (Doctoral dissertation, UNIVERSITE MOHAMED BOUDIAF M'SILA : FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE : Département d'Informatique).

[2] BAUDOIN, Nicolas et KARLE, Marion. NT Réseaux : IPS et IDS. Université de Marne la Vallée, France, 2004.

[3] PARZIALE, Lydia, LIU, Wei, MATTHEWS, Carolyn, et al. TCP/IP tutorial and technical overview. 2006.

[4] SODAGUDI, Suhasini, KOTHA, Sita Kumari, et RAJU, M. David. Novel Approaches to Identify and Prevent Cyber Attacks in Web. In : 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2019. p. 833-839.

[5] BURGERMEISTER, David et KRIER, Jonathan. Les systèmes de détection d'intrusions. 2006.

[6] Hamzata Guey : « Mise en place d'un IDS en utilisant Snort ». Licence en informatique et reseau.2010

[7] Alicherry, M., Muthuprasanna, M., Kumar, V. (2006, November). High speed pattern matching for network IDS/IPS. In Proceedings of the 2006 IEEE International Conference on Network Protocols (pp. 187-196). IEEE.

[8] https ://forum.huawei.com/enterprise/fr/comparaison-et-diffC3A9rences-entre-ips-vs-ids-vs-firewall-vs-waf/thread/778991-100371

[9] http ://www-igm.univ-mlv.fr/ dr/XPOSE2004/IDS/IDSSnort.html.

[10] Kaulanjan, K., et al. "Approche par intelligence artificielle de la prédiction de la récidive biologique après prostatectomie dans une population d'ascendance africaine." Progrès en Urologie-FMC 32.3 (2022) : S55-S56.

[11] LAROUSSI, KARIM, et al. "MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES." (2015).

[12] https ://wwceremade.dauphine.fr/~bey/enseignement/Enseignement/All_enseignement_dauphine/Module_analyse_numerique/Base_ENIT_Analyse20numérique/Analyse20Numérique20Matricielle/chapitre4.pdf

[13] Sutton, R. (1986). Two problems with back propagation and other steepest descent learning procedures for networks. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986 (pp. 823-832).

[14] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. Neural networks, 12(1), 145-151.

[15] SAGAR SHARMA. (Sep 6, 2017). Activation Functions in Neural Networks.

[16] G. DREYFUS,LES RÉSEAUX DE NEURONES :École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris (ESPCI),Laboratoire d'Électronique.

[17] Marc Parizeau :RESEAUX DE NEURONES,GIF-21140 et GIF-64326

[18] Héritier, Nsenge Mpia, and Inipaivudu Baelani Nephtali. "L'Algorithme de rétro-propagation de gradient dans le perceptron multicouche : Bases et étude de cas." International Journal of Innovation and Applied Studies 32.2 (2021) : 271-290.

[19] SOKOLOVA, Marina, JAPKOWICZ, Nathalie, et SZPAKOWICZ, Stan. Beyond accuracy, F-score and ROC : a family of discriminant measures for performance evaluation. In : Australasian joint conference on artificial intelligence. Springer, Berlin, Heidelberg, 2006. p. 1015-1021.