REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université du 20 Août 1955-Skikda

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

En vue de l'obtention du diplôme de Master en Informatique

Option :

Réseaux et Systèmes Distribués

Thème :

Etude comparative des techniques De détection d'intrusion

Réalisé par :

- Merdaci Bouchra
- Taleb Mouna

Encadré par :

Mr. Benoudina Lazhar

Année universitaire **2021/2022**

Remerciement

Tout d'abord, nous remercions Dieu de nous aider et nous donner la force et la volonté pour achever ce modeste travail.

*Ensuite, nous tenons à exprimer nos plus vifs remerciements notre encadreur « **Mr Benoudina Lazhar** » pour son encadrement continu.*
On le remercie également pour la confiance qu'il nous a accordée et pour la grande liberté d'idées et de travail qu'il nous a donnée.

Nous tenons à remercier également les membres des jurys pour avoir bien voulu évaluation et juger ce travail.

Nous veut aussi adresser nous sincères remerciements à tous les enseignants de département de l'informatique qui ont contribué à nous formation.

Quelques personnes ont contribué à la réalisation de ce travail et méritent des remerciements.

Enfin et surtout, nous tenons à remercier vivement toute nous familles notamment nous parents, qui nous ont toujours encouragés dans la poursuite de nous études, ainsi que pour leur aider, leur soutenir sans oublier.

Dédicace

Je dédie ce modeste travail le fruit de plusieurs années d'études :

A mon très cher père « Saleh ».

Malgré les grandes responsabilités que vous assumez dans vos travaux, vous avez toujours été près de nous, pour nous écouté nous soutenir, nous suivre et nous encourager. Puisse ce travail diminuer v souffrances et vous porter bonheur.

A ma très chère mère « Fatima ».

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

A** ma très chère sœur **« Wassila »** et mon très cher frère **« Nabil ».

A** ma cousine **« Maissa »

A** mes grands-mères **« Aicha »** et **« khadidja ».

A** mon oncle **« Yousef »** et ses filles **« Maroua »** et **« Maram ».

A** tous les membres des familles **« Merdaci »** et **« Kedadra ».

***A** mon binôme, ma copine et ma meilleure amie **« Mouna Taleb »** d'être la à mes côtéss.*

*Puisse **Dieu** vous donne **santé**, **bonheur**, **courage** et surtout **réussite.***

Je vous aime.

Bouchra

Dédicace

Je dédie ce modeste travail le fruit de plusieurs années d'études :
En tout premier lieu, je remercie le bon ***Dieu****, tout puissant de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.*

A ***ma très chère mère***
Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit .ton affection me couvre, tu bienveillance me guide et ta présence à mes cotés a toujours été ma source de force pour affronter les différents obstacles.

A ***mon très cher père***
Tu as toujours été à mes cotés pour me soutenir et m'encourager.
Que ce travail traduit ma gratitude et mon affection.

A *mes chers frères* ***Mohammed*** *et* ***Ayoub****, et mes belles sœurs* ***Khaoula****,* ***Rayan*** *et* ***Maroua****.*
A *ma famille* ***TALEB*** *au sens large et à tout mon entourage.*

A *mon binôme, ma copine et ma meilleure amie* ***Bouchra Merdaci*** *d'être la à mes cotés.*
A *mes meilleures amies* ***Lamis, Maroua****,* ***Hadil*** *pour leur encouragement et leur soutien.*

Puisse ***Dieu*** *vous donne* ***santé****,* ***bonheur****,* ***courage*** *et surtout* ***réussite.***
Je vous aime.

Mouna

Résumé

Les réseaux informatiques sont exposés à plusieurs types d'attaques, qu'il faut protéger. Parmi ces moyens nous citons les systèmes de détection d'intrusion, cependant, avec le développement des techniques d'attaque, ces systèmes ne fonctionnent plus de bons résultats.

Dans cette mémoire, nous avons réalisé une étude expérimentale pour décidé qu'elle est l'algorithme de classification le plus approprié pour les données de la base de données **NSL-KDD**. Nous avons considéré trois algorithmes de classification à savoir, l'algorithme **KNN**, **Naïve Bayes** et **Random Forest**, et nous avons comparé leur précision.

Mots clés: sécurité informatique, les algorithmes d'apprentissages automatiques, système detection d'intrusion, NSL-KDD.

Abstract

Computer networks are exposed to several types of attacks, which must be protected. Among these means we cite intrusion detection systems; however, with the development of attack techniques, these systems no longer work with good results.

In this thesis, we carried out an experimental study to decide that it is the most appropriate classification algorithm for the data of **the NSL-KDD** database. We considered three classification algorithms, the **KNN algorithm**, **Naive Bayes** and Random Forest, and we compared their accuracy.

Keywords: computer security, machine learning algorithms, intrusion detection systems, NSL KDD.

Table de matières

Chapitre 3 : La base de données NSL-KDD & les techniques de classification

Liste des Figures

Liste des Tableaux

Liste des Abréviations

VPN :	**V**irtual **P**rivate **N**etwork
QOS :	**Q**ualité **O**f **S**ervice
IDS:	**I**ntrusion **D**etection **S**ystem
IDWG:	**I**ntrusion **D**etection **W**orking **G**roup
IDMEF :	**I**ntrusion **D**étection **M**essage **E**xchange **F**ormat
DMZ :	**D**e **M**ilitarized **Z**one
NIDS :	**N**etwork **I**ntrusion **D**étection **S**ystem
HIDS :	**H**ost **I**ntrusion **D**étection **S**ystem
IA :	**I**ntelligence **A**rtificielle
KNN :	**K N**earest **N**eighbor
NSL-KDD:	**N**etwork **S**ecurity **L**ayer-**K**nowledge **D**iscovery in **D**atabases
TCP:	**T**ransmission **C**ontrol **P**rotocol
DARPA:	**D**efense **A**dvanced **R**esearch **P**rojects **A**gency
TP :	**T**rue **P**ositif
FP :	**F**aux **P**ositif
TN :	**T**rue **N**égatif
FN:	**F**aux **N**égatif
CDDL :	**C**ommon **D**éveloppement and **D**istribution **L**icense
WEKA:	**W**aikato **E**nvironment for **K**nowledge **A**nalysis
http	**H**yper**T**ext **T**ransfer **P**rotocol
XML	**E**xtensible **M**arkup **L**anguage

IDE: **I**ntegrated **D**evelopment **E**nvironment

JDK: **J**ava **D**evelopment **K**it

JVM : **J**ava **V**irtual **M**achine

MSJVM: **M**icrosoft **J**ava **V**irtual **M**achine

J2ME : **J**ava 2 **M**icro **E**dition

Introduction générale

Vu le développement de la cybercriminalité, les failles de sécurité d'un système informatique peuvent avoir des conséquences désastreuses sur une organisation, qui peuvent aller de la perte financière aux atteintes à la réputation de cette dernière jusqu'a la faillite.

Les attaques informatiques ne concernent pas uniquement les entreprises et les sociétés mais touchent de plus en plus les gouvernements et les structures sensibles des pays et même les individus. A tel point que les services de sécurité ont clairement émis des menaces, en affirmant qu'ils envisageraient toutes les options possibles en cas de cyber-attaques.

Les systèmes de détection d'intrusion, portent une nouvelle voie, depuis les années quatre-vingt. Peu à peu les modèles mis en place dans ce contexte, évoluent, potentiellement avec l'évolution des réseaux. L'apparition de l'intelligence artificielle et l'apprentissage automatique a été un nouvel axe d'intérêts.

Les systèmes de détection d'intrusion réseaux sont l'un des mécanismes le plus utilisé aujourd'hui pour détecter les intrusions. Le but des systèmes de détection d'intrusion réseaux est de protéger les réseaux contre des attaques qui ne peuvent pas être identifiés par des firewalls. L'un des problèmes majeurs est le taux des faux positifs et la mauvaise classification de certaines actions.

Dans ce mémoire, nous présentons une étude comparative pour les techniques d'apprentissage automatique les plus utilisées ces derniers années pour la détection d'intrusion. Telque les K plus proche voisins Knn, le foret aléatoire. L'objectif est de montrer quel est l'algorithme de classification le mieux adapté aux données de la base de données NSL KDD.

Pour ce faire, nous allons tester les trois algorithmes de classification avec l'apprentissage supervisé, à savoir l'algorithme KNN, Naïve bayes et Random forest. Et calculer les résultats de chaque classifieur avec les métriques de performance.

Nous avons utilisé le logiciel Weka permettent d'instancier les classifieurs, et nous avons codée notre système en utilisent le langage java.

Notre mémoire est organisé comme suit :

- Le premier chapitre est consacré à la présentation des différents aspects de la sécurité Informatique et les systèmes de détection d'intrusion.
- Le deuxième chapitre est axé à la présentation de l'intelligence artificielle en général et l'apprentissage automatique avec ses différents types et ses algorithmes.

- Dans le chapitre trois nous allons présenter la base de données NSL-KDD avec son contenu et ses attribues et présenter également une étude détaillée des trois techniques de classification que nous utiliserons pour la détection d'intrusion, ainsi que de présenter les métriques de performance et le protocole expérimental approuvé.
- Dans le quatrième chapitre nous avons présenté tout d'abord les outils et le langage utilisé pour la réalisation du notre travail, ensuite nous passons aux résultats expérimentaux et à leur comparaison pour choisir le meilleur algorithme de classification des jeux de donnés NSL KDD.

Chapiter1 : sécurité informatique et détection d'intrusion

Introduction

Le progrès technologique, le développement des moyens de communications, l'ouverture du monde sur nouvelles technologies, et la transmission de divers types de données à travers les réseaux, ainsi que d'autre facteurs, apportent un danger d'accès et de manipulation des données par des personnes non autorisés, ou des concurrents. Donc la sécurité de l'information par une gamme de techniques et mécanismes d'authentification et de contrôle d'accès est devenue un besoin crucial afin de construire un système sécurisé déterminant et éliminant ces vulnérabilités.

Le système de détection des intrusions (IDS) est l'une de ces techniques qui offre un contrôle permanent des attaques ou suspectes et permettant ainsi de détecter toute tentative de violation de la politique de sécurité, c'est-à-dire toute intrusion.

Dans ce premier chapitre nous présentons deux parties, la première présente les principales notions de base de la sécurité informatique e, en commençant par les définitions des différentes notions de la sécurité informatique, puis les attaques informatiques et leurs classifications avec exemples. La deuxième partie présente la détection d'intrusions, définition, principe de fonctionnement, et les techniques de détection ...etc. A la fin de ce chapitre, nous citons les limites actuelles de la détection d'intrusions.

I. La sécurité informatique

I.1 Définition

La sécurité informatique protège l'intégrité des technologies de l'information comme les systèmes, les réseaux et les données informatiques contre les attaques, les dommages ou les accès non autorisés. Pour préserver leur compétitivité dans le contexte de la transformation numérique, les entreprises doivent comprendre comment adopter des solutions conception **[1].**

I.2 Les objectifs de la sécurité informatique

La sécurité informatique vise généralement les cinq principaux objectifs suivants :

- **Disponibilité :** demande que l'information sur le système soit disponible aux personnes autorisées.
- **Confidentialité :** permet d'assurer que l'information sur le système ne puisse être lue que par les personnes autorisées.

- **Intégrité :** L'intégrité permet de certifier que l'information sur le système ne puisse être modifiée que par les personnes autorisées (pas de divulgation à des tiers non autorisés).
- **Non-répudiation :** les acteurs impliqués dans la communication ne peuvent nier y avoir participé.
- **L'authentification :** L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

I.3 Soucis de la sécurité informatique

I.3.1 Vulnérabilité

Faute créée durant le développement du système, ou durant l'opération, pouvant être exploitée afin de créer une intrusion.

I.3.2 Intrusion

Faute malveillante externe résultant d'une attaque qui a réussi à exploiter une vulnérabilité.

I.3.3 Menace

Possibilités et probabilités d'attaque contre la sécurité. Une menace est définie par le processus d'attaque, par la cible et par le résultat (conséquences de la réussite d'une attaque).

I.3.4 Attaque

C'est n'importe quelle action qui a le but de menacer la sécurité des informations et de nuire au moins à l'une des propriétés de la sécurité informatique (disponibilité, Confidentialité, Intégrité, L'authentification). Il s'agit d'une tentative d'intrusion, nous abordons dans ce qui suit les différents buts et classes de ces attaques (tentatives d'intrusion).

I.3.4.1 Buts des attaques informatiques

Il existe plusieurs objectifs pour les attaques:

- **Interruption :** vise la disponibilité des informations (DoS, . . .).

- **Interception :** vise la confidentialité des informations (capture de contenu, analyse de trafic,. . .).
- **Modification :** vise l'intégrité des informations (modification, rejet,. .).
- **Fabrication :** vise l'authenticité des Informations (Masquerade) **[2].**

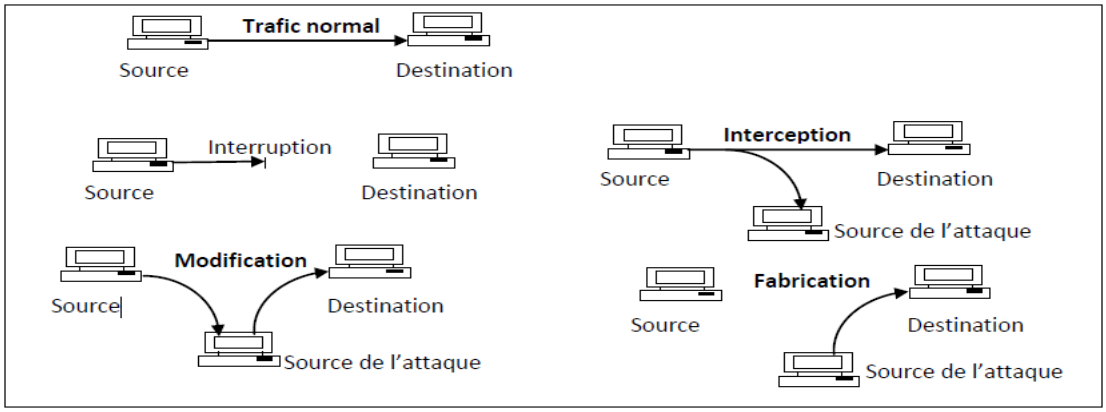

Figure 1.1. Buts des attaques Informatiques **[3].**

I.3.4.2 Schéma d'une attaque

Une attaque est une tentative de violation d'un des objectifs de la sécurité informatique alors que l'intrusion est une attaque réussie. Une attaque peut être schématisée en six points **[4] :**

- L'intrusion dans le système grâce à ces informations.
- La mise en place d'un système permettant un ré-intrusion future, tel que de code dans l'EEPROM.
- La recherche d'une propagation de l'intrusion dans un autre système et ainsi permettre des attaques distribuées.
- La paralysie du système.
- L'effacement des traces de l'attaquant.

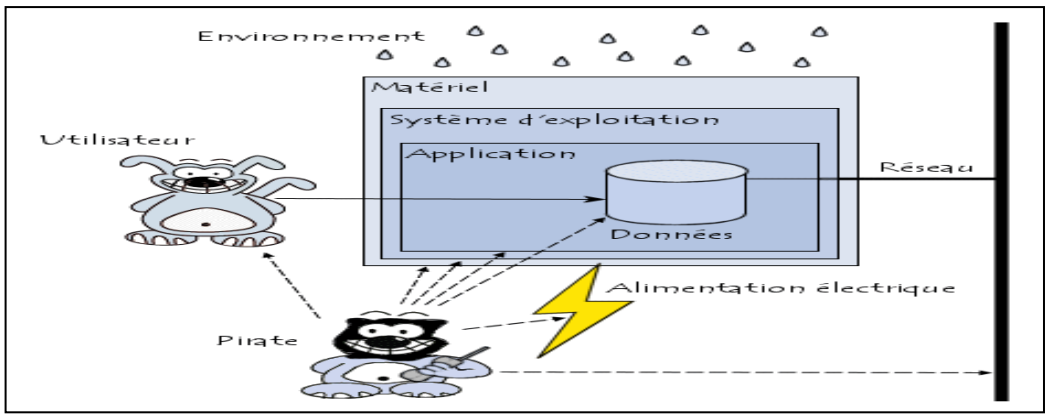

Figure 1.2. Schéma d'une attaque **[4]**.

I.3.4.3 Les motivations d'une attaque

Les motivations des attaques peuvent être liées à divers objectif **[5] :**

- Obtenir un accès au système.
- Voler des informations, tels que des secrets industriels ou des propriétés intellectuelles.
- Collectionner des informations personnelles sur un utilisateur.
- S'informer sur l'organisation.
- Récupérer des données bancaires.
- S'informer sur l'organisation (entreprise de l'utilisateur, etc.).
- Troubler le bon fonctionnement d'un service.
- Utiliser le système de l'utilisateur comme « rebond » pour une attaque.

I.3.4.4 Type d'attaques

Les hackers utilisent plusieurs techniques d'attaques. Ces attaques peuvent être regroupées en trois familles différentes **[6] :**

- **Les attaques directes :** C'est la plus simple des attaques. Le hacker attaque directement sa victime à partir de son ordinateur. La plus part des hackers utilisent cette technique. En effet, les programmes de hack qu'ils utilisent ne sont que faiblement paramétrable, et un grand nombre de ces logiciels envoient directement les paquets à la victime **[6].**

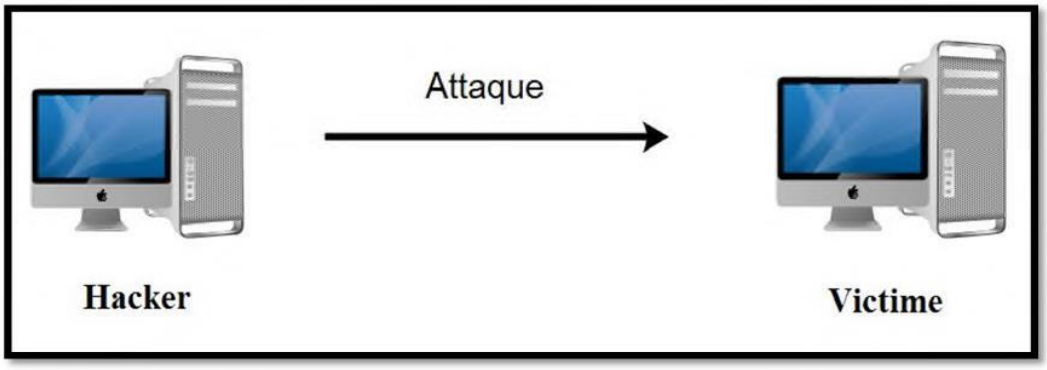

Figure 1.3. Attaque directe **[6].**

- **Les attaques indirectes par rebond :** Cette attaque est très prisée des hackers.
- En effet, le rebond a deux avantages :
- Masquer l'identité du hacker.
- Utiliser les ressources de l'ordinateur intermédiaire car il est plus puissant pour attaquer.

- Le principe en lui-même, est simple : Les paquets d'attaque sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime. D'où le terme par rebond.

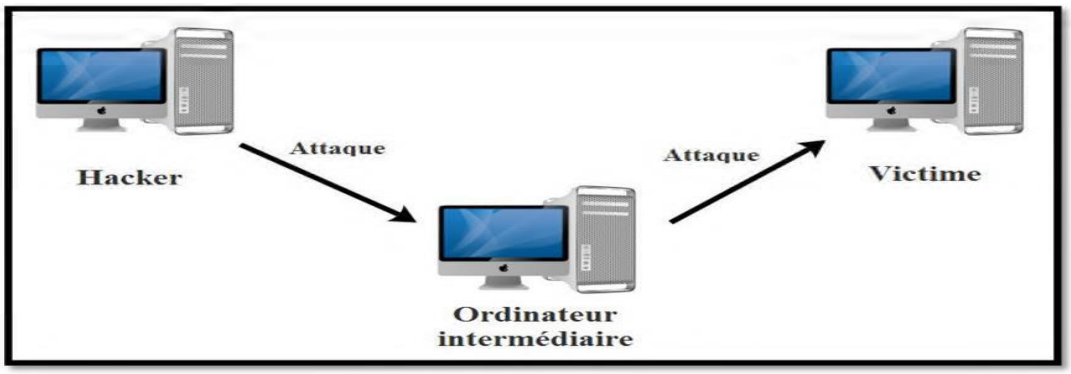

Figure 1.4. Attaque indirecte par rebond **[6].**

- **Les attaques indirectes par réponse :** Cette attaque est un dérivé de l'attaque par rebond. Elle offre les mêmes avantages, du point de vue du hacker. Mais au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête. Et c'est cette réponse à la requête qui va être envoyée à l'ordinatrice victime.

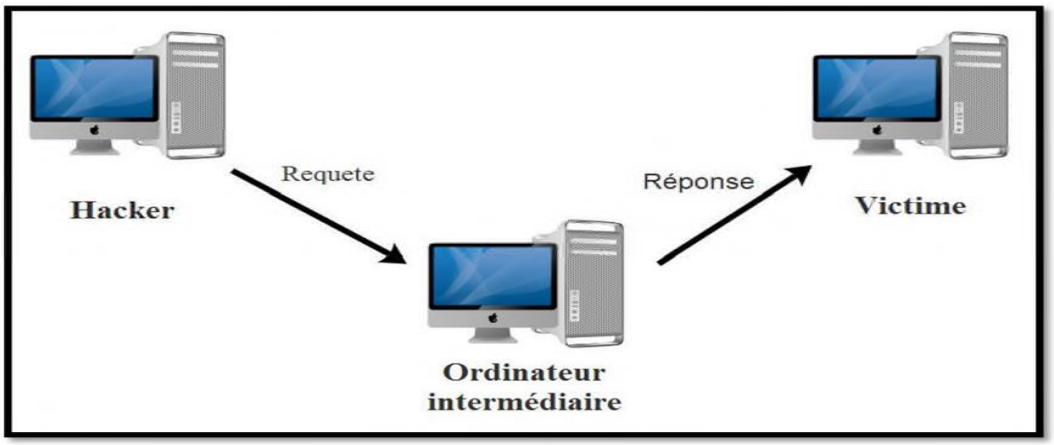

Figure 1.5. Attaque indirecte par réponse **[6].**

I.3.4.5 Catégorie des attaques

Il existe quatre catégories d'attaques **[7] :**

- **Attaques par interruption:** c'est une attaque protée à la disponibilité. La destruction d'une pièce matérielle (tel un disque dure), la coupure d'une ligne de communication, ou la mise hors service d'un système de. Gestion de fichiers en sont des exemples.

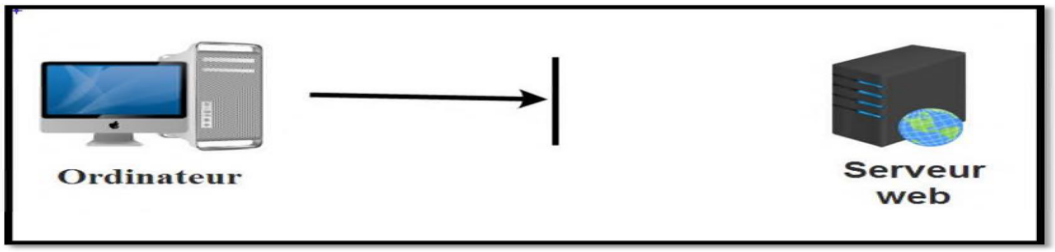

Figure 1.6. Attaque par interruption **[7].**

- **Attaque par interception :** C'est une attaque portée à la confidentialité. Il peut s'agir d'une personne, d'un programme ou d'un ordinateur. Une écoute téléphonique dans le but de capturer des données sur un réseau, ou la copie non autorisée de fichiers ou de programme en sont des exemples.

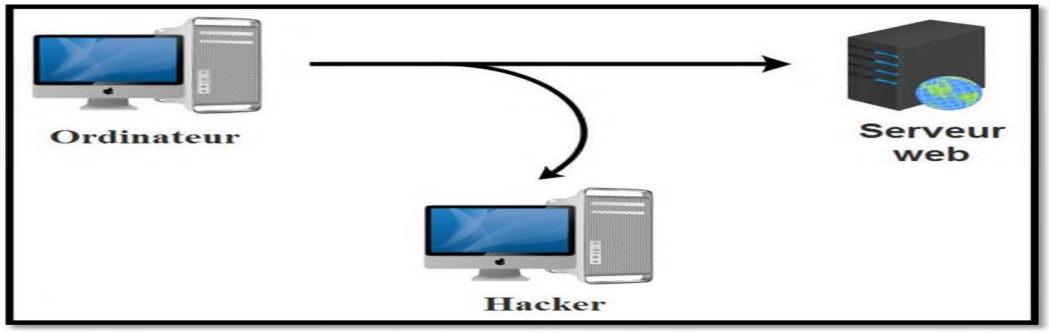

Figure 1.7. Attaque par interception **[7].**

- **Attaque par modification :** Il s'agit d'une attaque portée à l'intégrité. Changer des valeurs dans un fichier de données, altérer un programme de façon à bouleverser son comportement ou modifier le contenu de messages transmis sur un réseau sont des exemples de telles attaques.

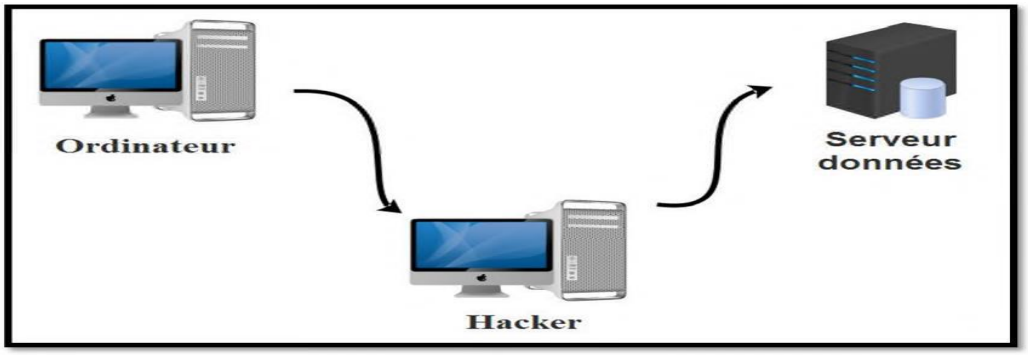

Figure 1.8. Attaque par modification **[7].**

- **Attaque par fabrication :** C'est une attaque portée à l'authenticité. Il peut s'agir de l'insertion de faux messages dans un réseau ou l'ajout d'enregistrements à un fichier.

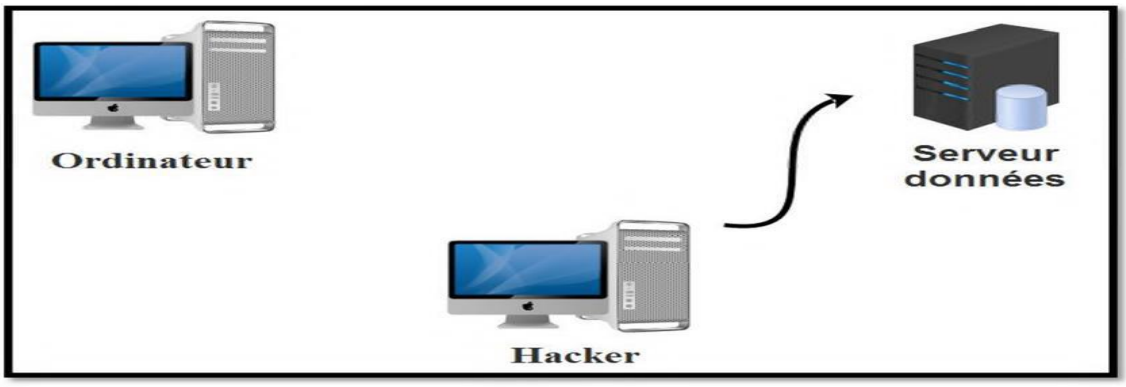

Figure 1.9. Attaque par fabrication **[7].**

I.3.4.6 Les différentes classes d'attaques informatiques

Du point de vue de la sécurité informatique, une menace est une violation potentielle de la sécurité. Cette menace peut être accidentelle, intentionnelle (attaque), active ou passive, Ils existent dans la littérature plusieurs classifications d'attaques informatique selon des critères différents, parmi lesquelles :

I.3.4.6.1 Classification selon l'effet de l'attaque

Selon les effets résultant de l'attaque on peut classifier les attaques en deux groupes principaux: les attaques passives et les attaques actives.

- **Les attaques passives :** consistent à accéder, utiliser ou à observer le système cible sans modifier les données ou dysfonctionné les ressources de ce dernier, elles sont généralement indétectables (ex: capture de contenu, analyse de trafic).
- **Les attaques actives :** consistent à effectuer des changements non autorisés sur les données des systèmes, à s'introduire dans des équipements réseau ou à perturber leurs fonctionnements, les attaques de ce type sont bien évidemment plus dangereuses.(ex: mascarade et déni de service).

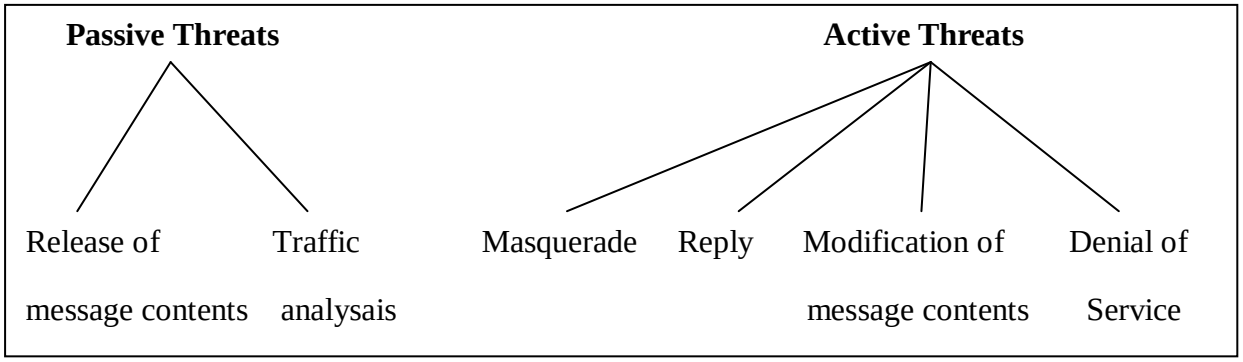

Figure 1.10. Attaques passives ou actives.

I.3.4.6.2 Classification selon la source de l'attaque

En termes de relation intrusion-victime, les attaques sont classées comme suit :

- **Les attaques internes :** provenant des employés de leur entreprise ou de leurs partenaires commerciaux ou clients.
- **Les attaques externes :** venant de l'extérieur, fréquemment via Internet.

I.3.4.6.3 Classification selon la cible de l'attaque

- **Les attaques réseaux :** Les attaques réseaux s'appuient sur des vulnérabilités liées directement aux protocoles ou à leur implémentation. Il existe un grand nombre. Néanmoins, la plupart d'entre elles ne sont que des variantes des cinq attaques réseaux les plus connues aujourd'hui.
- **Les attaques applicatives :** Les attaques applicatives s'appuient principalement sur des vulnérabilités spécifiques aux applications utilisées.

I.3.4.7 Exemples d'attaques

Il existe un nombre énorme d'attaques qui menacent les systèmes et les réseaux informatiques, néanmoins, la plupart d'entre elles ne sont que des variantes des autres. Voici des exemples d'attaques les plus connues aujourd'hui ciblant les réseaux informatiques.

- **Attaques de Dénis de Services :** (Denial Of Service [DOS]): est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser. Il peut s'agir de :
  - L'inondation d'un réseau afin d'empêcher son fonctionnement.
  - La perturbation des connexions entre deux machines, empêchant l'accès à un service particulier.
  - L'obstruction d'accès à un service à une personne en particulier.
  - Également le fait d'envoyer des milliards d'octets à un box internet.
  - L'attaque par déni de service peut ainsi bloquer un serveur de fichiers, rendre impossible l'accès à un serveur web ou empêcher la distribution de courriel dans une entreprise, les principales attaques qu'on peut trouver sont Apache2, Back, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Tear drop, Udp storm.
- **Probing (Sondage) :** L'attaquant de cette classe commence par un sondage de la future victime, ce que l'on appelle scan, ce sondage va balayer chaque port IP afin de

connaitre les services offerts par le système (OS, topologie du réseau, protections employées,..) une fois se achevé, la machine de l'intrus (celui qui réalise l'intrusion) tente alors d'identifier le système d'exploitation utilisé par cette victime et d'exploiter les informations qu'elle a récolté. Cette classe d'attaque est la plus étendue et qu'elle requit une expertise technique minime. Les exemples de ce type d'attaque sont: IPSec, Mscan, Nmap, Saint, Satan.

- **Attaques User to Root :** L'objectif de cette classe d'attaques est d'obtenir la main de l'administrateur système (Root) à partir d'un simple compte utilisateur par l'exploitation des vulnérabilités, Les exploits les plus connus sont les débordements réguliers des Buffers (buffer overflows) dus aux erreurs de programmation, Les principales attaques de ce type sont: Eject, FF config, Fdformat, Load module, Perl, Ps, Xterm.
- **Attaque Remote to User :** Dans cette classe d'attaque, l'attaquant essaye d'exploiter les vulnérabilités d'une machine distante afin d'avoir un accès illégal à cette dernière, Pour réussir cette attaque, l'attaquant exploite les bugs des applications installées dans la machine cible, les mauvaises configurations de celles-ci et du système qui les héberge, etc.
- **L'usurpation d'adresse IP(IP Spoofing) :** Le principe de fonctionnement de cette attaque est d'envoyer des paquets IP en utilisant une adresse IP source qui n'a pas été allouée à l'ordinateur qui émet ces paquets pour le but de masquer l'identité de l'attaquant lors d'une attaque d'un serveur ou n'importe quel cible dans le réseau , ou d'usurper l'identité d'un autre équipement du réseau pour bénéficier des services auxquels il a accès.
- **Les analyseurs réseau (sniffer) :** Est un dispositif permettant d'« écouter » le trafic d'un réseau, c'est-à-dire de capturer les informations qui y circulent, vu que les données dans un réseau non commuté sont envoyées à toutes les machines du réseau et dans une utilisation normale les machines ignorent les paquets qui ne leur sont pas destinés. Le sniffer peut également servir cette propriété à une personne malveillante ayant un accès physique au réseau pour collecter des informations (ex: les mots de passes), Maisun sniffer peut aussi être utilisé comme un outil positif pour le but d'étudier et de capturer le trafic d'un réseau par les administrateurs réseaux et les détecteurs d'intrusion (IDS).

- **Balayage des ports :** (port scanning) est une des activités considérées comme suspectes servant par les pirates informatiques pour découvrir les faiblesses potentiellement exploitables et chercher les ports ouverts sur un serveur de réseau en balayant les ports disponibles de la victime qui est potentiellement exécute de nombreux services qui écoutent des ports connus, les balayages de ports se font habituellement sur le protocole TCP pour le but d'ouvrir des connexions pour effectuer une intrusion, la même technique de balayage des ports est aussi utilisée par les administrateurs des systèmes informatiques pour contrôler la sécurité des serveurs de leurs réseaux.
- **TCP Session Hijacking :** Le « **vol de session TCP** » est une technique consistant à intercepter une session TCP initiée entre deux machines afin de la détourner, dans la mesure où le contrôle d'authentification s'effectue uniquement à l'ouverture de la session, un pirate réussissant cette attaque parvient à prendre possession de la connexion pendant toute la durée de la session.
- **Les trappes (backdoors) :** C'est une fonction ou un programme permettant à un pirate de prendre le contrôle d'un ordinateur à distance. Il peut être placé dans un cheval de Troie1 ou un virus.
- **Attaque par virus :** Il s'agit d'un programme autoreproductible et généralement destructeur qui contamine le disque dur ainsi que tous autres supports de stockage utilisés et qui peut faire exécuter à l'ordinateur des actions non désirées, le virus informatique peut donc se propager à l'intérieur même de l'ordinateur, en infectant petit à petit tous les fichiers. Il est donc destiné à modifier à notre insu le fonctionnement de l'ordinateur, certains virus peuvent simplement faire «beeper» le PC, d'autres peuvent détruire les données (formater, effacer le secteur de démarrage, voir détruire le matériel) **[8].**

I.4 Les protocoles de sécurité

Un protocole est un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Sur Internet, les protocoles utilisées font partie d'une suite de protocoles TCP/IP, tel que la plus part de ces protocoles ne sont pas sécurisés lors de la transmission des données sur le réseau. Les protocoles sécurisés ont été mis au point, afin d'encapsuler les messages dans des paquets de données chiffrées. On cite parmi ces protocoles les suivants :

- **Protocole SSH (Secure Shell) :** c'est un protocole qui permet à des services TCP/IP d'accéder à une machine à travers une communication chiffrée appelée « tunnel ».
- **Protocole SSL (Secure Socket Layer) :** c'est un procédé de sécurisation des échanges, il a été conçu pour assurer la sécurité des transactions effectuées via Internet.
- **Protocole HTTPS :** HTTPS n'est rien d'autre que HTTP encapsulé dans la couche de chiffrement TLS (Transport Layer Security). En général le serveur est authentifié par un certificat X509, l'internaute peut s'authentifier par l'intermédiaire d'un serveur RADIUS, ou par un des autres procédés proposés par les logiciels serveur.
- **IPSec (IP Security) :** IPSec (*Internet Protocol Security*) est conçu pour sécuriser le protocole IPv6. La lenteur de déploiement de ce dernier a imposé une adaptation d'IPSec à l'actuel protocole IPv4. On établit un tunnel entre deux sites (voir figure 1.11), et IPSe c gère l'ensemble des paramètres de sécurité associés à la communication. Deux machines passerelles, situées à chaque extrémité du tunnel, négocient les conditions de l'échange des informations: quels algorithmes de chiffrement, quelles méthodes de signature numérique ainsi que les clés utilisées pour ces mécanismes. La protection est apportée à tous les trafics et elle est transparente aux différentes applications.

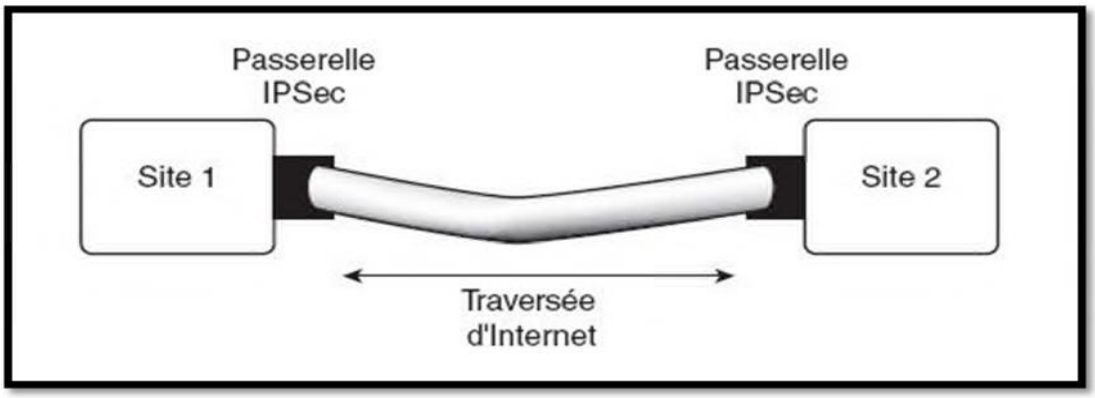

Figure 1.11. Un tunnel IPSec entre deux sites d'entreprise **[7].**

- **Les Algorithme de chiffrements :** Il existe deux grandes familles d'algorithmes de chiffrements, ceux à clés symétriques et ceux à clés asymétriques.
- **Algorithme de chiffrement symétrique :** il consiste à utiliser la même clé pour le chiffrement ainsi que pour le déchiffrement. Il est donc nécessaire que les deux interlocuteurs se soient mis d'accord sur une clé privée, ou ils doivent utiliser un canal sécurisé pour l'échanger.

- **Algorithme de chiffrement asymétrique :** c'est une méthode cryptographique faisant intervenir une paire de clés asymétrique (une clé publique et une clé privée). Elle utilise cette paire de clés pour le chiffrement et le déchiffrement. La clé publique est rendue publique et elle est distribuée librement, la clé privée quant à elle n'est jamais distribuée et doit être gardé secrète.

I.5 Mise en œuvre d'une politique de sécurité

La politique de sécurité des systèmes d'information est un plan d'actions définies pour maintenir un certain niveau de sécurité. Elle reflète la vision stratégique de la direction de l'entreprise en matière de sécurité des systèmes d'informations (SSI).

Une politique de sécurité s'élabore à plusieurs niveaux :

- Sécuriser l'accès aux données de façon logicielle (authentification, contrôle d'intégrité).
- Sécuriser l'accès physique aux données : serveurs placés dans des salles blindées avec badge d'accès…
- Un aspect très important pour assurer la sécurité des données d'une entreprise est de sensibiliser les utilisateurs aux notions de sécurité, de façon à limiter les comportements à risque: si tout le monde peut accéder aux salles de serveurs, peut imposte qu'elles soient sécurisées!
- De même, si les utilisateurs laissent leurs mots de passes écrit à côté de leur PC, son utilité est limitée…
- Enfin, il est essentiel pour un responsable de sécurité de s'informer continuellement, des nouvelles attaques existantes, des outils disponibles…de façon à pouvoir maintenir à jour son système de sécurité et à combler les brèches de sécurité qui pourraient exister.

I.6 Outils de sécurité

Dans ce qui suit nous allons présenter un ensemble non exhaustif d'outils de sécurité :

I.6.1 Antivirus

Un antivirus est une sorte de logiciel utilisé pour empêcher, analyse, détecter et supprimer les virus d'un ordinateur. Une fois installés, la plupart des logiciels antivirus s'exécutent automatiquement en arrière-plan pour fournir une protection en temps réel contre les attaques de virus.

Des programmes complets de protection antivirus aident à protéger vos fichiers et votre matériel contre les logiciels malveillants tels que les vers, les chevaux de Troie et les logiciels espions, et peuvent également offrir une protection supplémentaire telle que des pare-feu personnalisables et le blocage de sites Web **[9].**

I.6.2 Pare-feu (Firewall)

Un firewall (ou pare-feu) est outil informatique (matériel et/ou logiciels) conçu pour protéger les données d'un réseau (protection d'un ordinateur personne : relié à Internet par exemple, ou protection d'un réseau d'entreprises).

Il permet d'assurer la sécurité des informations d'un réseau en filtrant les entrées et en contrôlant les sorties selon des règles définies par son administrateur.

Le pare-feu est une passerelle filtrante qui protège un ordinateur ou un réseau des intrusions venues d'Internet. Il filtre en effet les paquets de données qui sont échangés. Il est parfois traduit comme coupe-feu, barrière de sécurité ou garde-barrière. Il est doté au moins de deux interfaces, l'un destiné au réseau interne et l'autre au réseau externe. Pour que le pare-feu s'intègre à un appareil, il importe que:

- Le système informatique soit protégé.
- Le système de filtrage des paquets soit unique.
- La machine soit puissante **[10].**

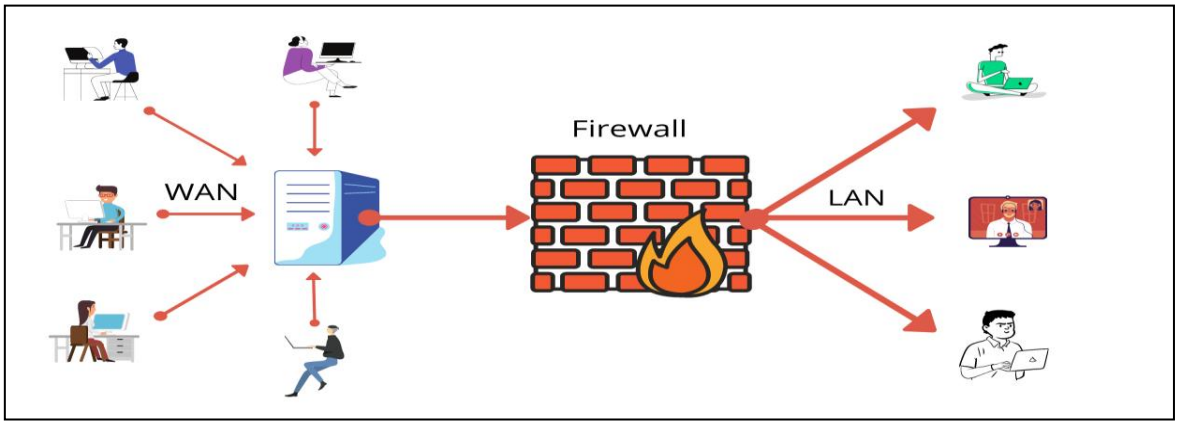

Figure 1.12. Pare-feu(Firewall) **[10].**

I.6.3 Cryptographie

La cryptographie est l'étude des méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible : c'est ce qu'on appelle le chiffrement, qui, à partir d'un texte en clair, donne un texte chiffré Inversement, le déchiffrement est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré en utilisant une clé particulière et un algorithme de déchiffrement **[11].**

I.6.4 Réseau privé virtuel (VPN)

Les réseaux privés virtuels (VPN : Virtual Private Network) permettent à l'utilisateur de créer un chemin virtuel sécurisé entre une source et une destination. Avec le développement d'internet, il est intéressant de permettre ce processus de transfert de données sécurisé et fiable. Grâce à un principe de tunnel (tunneling) dont chaque extrémité est identifiée, les données transitent après avoir été chiffrées.

Un des grands intérêts des VPN est de réaliser des réseaux privés à moindre coût. En chiffrant les données, tout se passe exactement comme si la connexion se faisait en dehors d'Internet. Il faut par contre tenir compte de la toile, dans le sens où aucune qualité de service (QoS) n'est garantie. Le principe du VPN est basé sur la technique dutunnelling. Cela consiste à construire un chemin virtuel après avoir identifié l'émetteur et le destinataire. Ensuite la source chiffre Les données et les achemine en empruntant ce chemin virtuel. Les données à transmettre peuvent appartenir à un protocole différent d'IP. Dans ce cas le protocole de tunneling encapsule les données en rajoutant un entête. Permettant le routage des trames dans le tunnel Le tunneling est l'ensemble des processus d'encapsulation, de transmission et d'encapsulation **[12].**

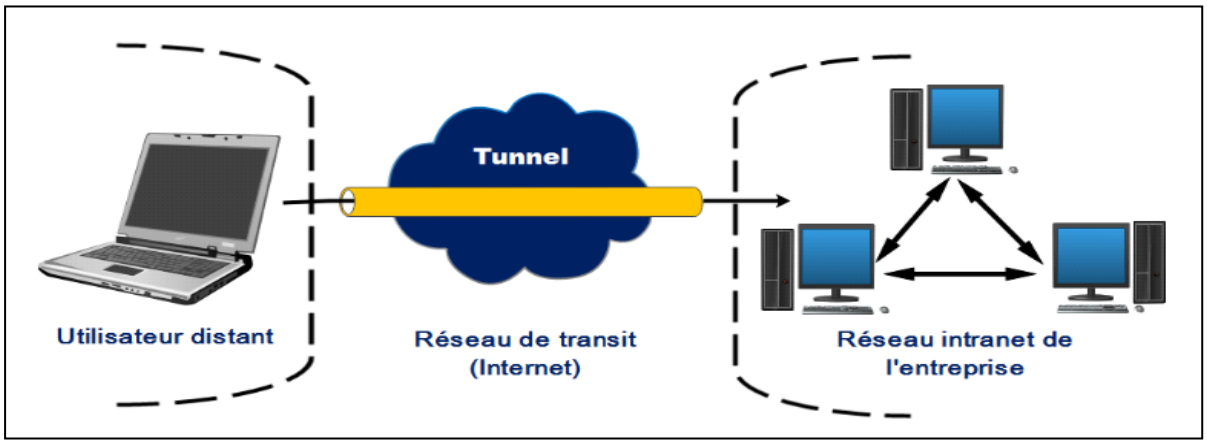

Figure 1.13. Réseau privé virtuel (VPN) **[12].**

I.6.5 Système de detection d'intrusion(IDS)

Un système de détection d'intrusion (ou IDS Intrusion Détection System) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions **[13].**

II. Détection d'intrusions

De nos jours, les attaques sont si rapides qu'avant, et tout le monde est exposé aux pertes des données essentielles, Malheureusement, les systèmes antivirus ou les pare-feux sont la plupart du temps inefficaces face à ces nouvelles menaces. C'est pour pallier ce manque que sont apparus récemment des nouveaux composants de sécurité appelés les systèmes de détection d'intrusions. Ces derniers permettent de déjouer les attaques attendues sur le réseau en générant des alertes, des avertissements là ou il existe des menaces externes ou internes, ce qui aide à réduire le temps et l'effort fournit par l'administrateur car la surveillance peut se faire sans supervision humaine. La protection des données des clients contre les intrusions permet d'avoir la confiance des clients et partenaires et garder une bonne réputation sur l'organisation.

II.1 Définition de la Détection d'intrusions

La detection d'intrusion est définie comme étant le processus de surveillance et d'analyse des événements qui surviennent sur votre réseau afin d'identifier d'éventuels incidents, violations et menaces imminentes par rapport à vos stratégies de sécurité. La prévention d'intrusion consiste à détecter les intrusions afin de résoudre les incidents repérés. Ces mesures de sécurité sont disponibles sous la forme de systèmes de détection d'intrusion (IDS). Ces systèmes sont intégrés à votre réseau afin de détecter les incidents potentiels et d'mettre fin **[14].**

II.2 Architecture des IDS et Principes de fonctionnement

II.2.1 Architecture des IDS

Plusieurs architectures ont été proposées pour décrire les différents éléments intervenants dans un système de détection d'intrusion. L'architecture la plus simple est

composée de trois modules: la source de données, l'analyseur des données et le module des réponses.

L'architecture générale d'un IDS proposée par **IDWG (I**ntrusion **D**étection **W**orking **G**roup) est montrée dans la (Figure 1.14).

Dans l'architecture proposée par le groupe IDWG de l'IETF2on trouve les trois modules cités précédemment couplés avec d'autres composants, Dans cette architecture, l'objectif été la définition d'un standard de communication entre les composants du système de détection d'intrusion. Cette architecture définit un format d'échange de message pour les IDS: **I**ntrusion **D**étection **M**essage **E**xchange **F**ormat (**IDMEF**), qui contient implicitement un modèle de données. Proposée par IDWG.

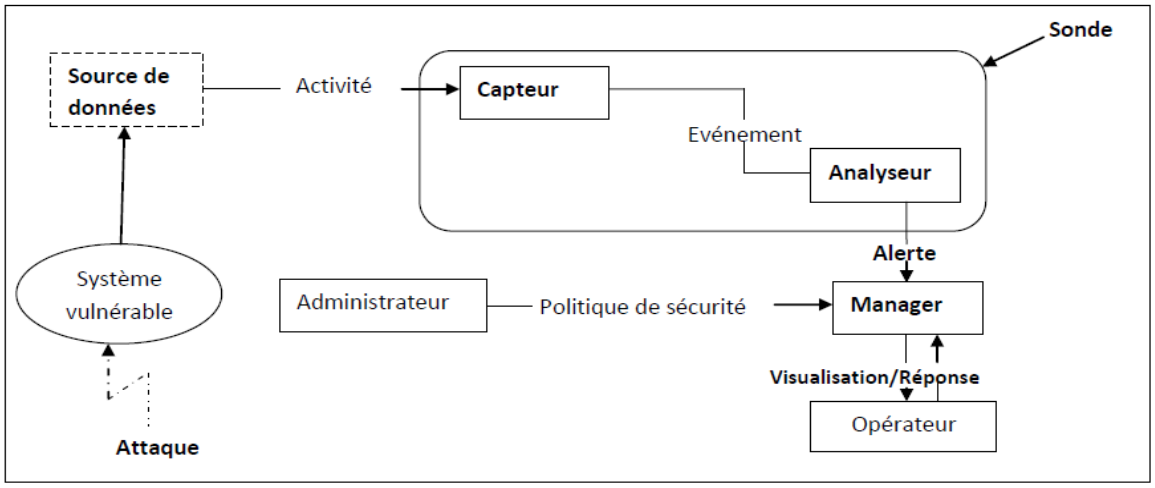

Figure 1.14. Architecture d'un IDS proposée par IDWG **[15].**

Cette architecture est composée des modules suivants :

- **Source de données:** C'est l'interface entre le système surveillé et l'IDS, elle fait la collecte d'informations sur les activités du système.
- **Capteur:** Chargé de filtrer et formater les informations brutes envoyées par la source de données. Le résultat de ce traitement sera un message formaté, appelé aussi événement, il représente l'unité de base dans un scenario d'attaque.
- **Analyseur:** Permet d'analyser les événements générés par le capteur. S'il détecte une activité intrusive il émet une alerte, qui est un message sous un format standard. Dans cette architecture, le capteur et l'analyseur forment ensemble une sonde.
- **Alertes:** Lorsqu'un IDS détecte une intrusion, il doit la signaler à l'administrateur à travers les alertes, Ces dernières générées par les IDS sont généralement stockées dans les journaux du système ou utilisés pour prendre des actions contre les

attaques (cela dépend du type d'IDS: à réaction active ou passive). Cependant il existe une norme qui permet d'en formaliser le contenu, afin de permettre à différents éléments de sécurité d'inter-opérer. Ce format s'appelle **IDMEF3** (**I**ntrusion **D**étection **M**essage **E**xchange **F**ormat), où il est possible de les visualiser ultérieurement par un expert de sécurité.

- **Manager:** En plus de la notification des alertes, il offre à l'administrateur la possibilité de configurer une sonde et de gérer les alertes envoyées par l'analyseur.

II.2.2 Principe de fonctionnement des IDS

La Figure 1.15 illustre le fonctionnement d'un IDS et l'enchainement de ses actions lors de la détection des intrusions**.**

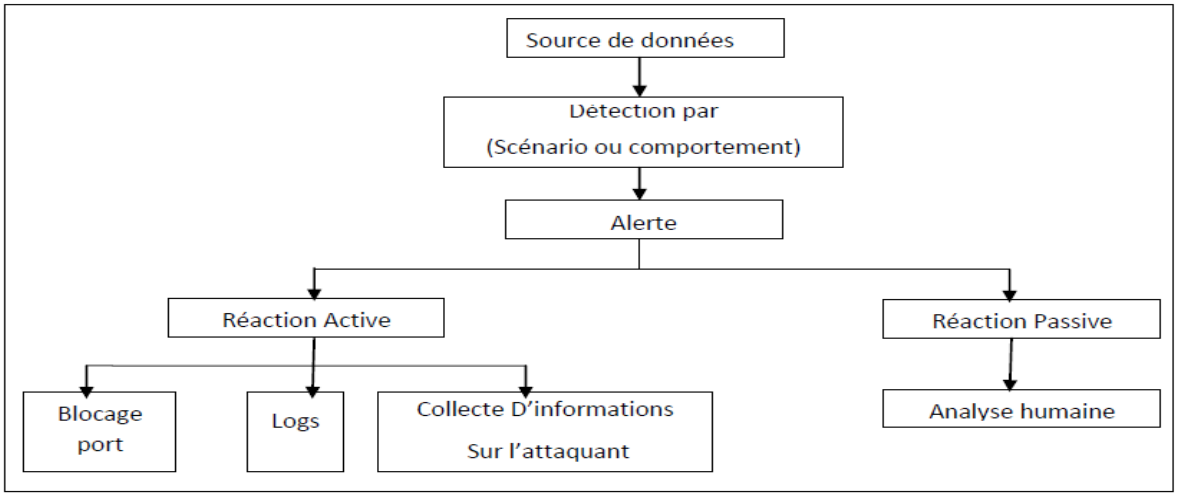

Figure 1.15. Fonctionnement d'un IDS **[16].**

II.3 Caractéristiques d'un système de détection d'intrusion

Les caractéristiques souhaitées d'un système de détection d'intrusion sont **[17] :**

- Capacité de fonctionner continuellement avec un minimum d'intervention humaine.
- Difficulté pour un attaquant de désactiver ou modifier sa configuration.
- Capacité de se contrôler lui-même et de détecter s'il vient de faire l'objet de manipulation de la part d'un attaquant.
- Utilisation minimale de ressources (de calcul, de stockage, etc.) sur le système sur lequel il est installé.
- Capacité d'accepter des mises à jour et des modifications de configuration pour rendre compte des nouvelles dispositions de la politique de sécurité et les

changements susceptibles de s'opérer dans l'organisation (nouvelles acquisitions, restructuration, etc.).

- Facilite et simplicité de déploiement : facilite d'installation et de configuration portabilité ...etc.
- Interopérabilité avec d'autres systèmes et outils de la sécurité informatique.
- Il doit être tolérant aux fautes c'est-à-dire qu'il doit être capable de retrouver son état initial de fonctionnement après un crash causé soit par une manipulation accidentelle soit par des activités n'émanant de personnes malintentionnées.
- Lorsque le nombre de systèmes à superviser augmente et donc que les attaques potentielles augmentent également, nous pouvons alors attendre de l'IDS les caractéristiques suivantes :
  - Il doit être capable de superviser un nombre important de stations tout en fournissant des résultats de manière rapide et précise.
  - Il doit fournir "un service minimum de crise" c'est-à-dire que si certains composants de l'IDS cessent de fonctionner, les autres composants doivent être affectés le moins possible par cet état de dégradation.

II.4 Emplacement de l'IDS

Il est très important de faire bien positionner le système de détection d'intrusion, cela nécessite de bien identifier les ressources à protéger et ce qui est le plus susceptible d'être attaqué, Il convient alors d'implémenter précautionneusement dans la zone convenable. Il existe plusieurs endroits stratégiques où il convient de placer un IDS. La Figure 1.16 illustre un réseau local ainsi que les trois positions que peut y prendre un IDS :

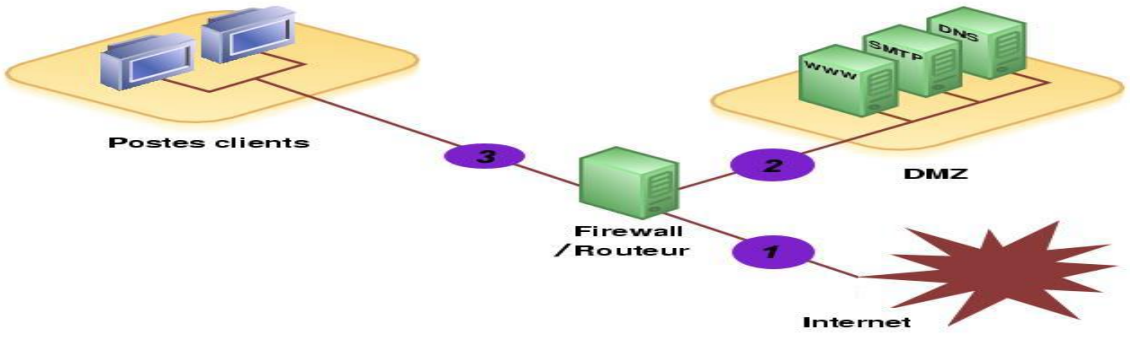

Figure 1.16. Emplacements des IDS **[15].**

- **Position (1) :** L'IDS Sur cette position sert à détecter l'ensemble des attaques frontales, provenant de l'extérieur, vers le firewall. Dans ce cas beaucoup d'alertes seront remontées ce qui rendra les logs difficilement consultables.
- **Position (2) :** L'IDS placé sur la DMZ4, utilisé pour détecter les attaques qui n'ont pas été filtrées par le firewall et qui relèvent d'un certain niveau de compétence. Les logs seront ici plus clairs à consulter puisque les attaques bénins ne seront pas recensées.
- **Position (3) :** L'IDS dans cette position a pour objectif de rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'un placer un à cet endroit étant donné le fait que 80% des attaques proviennent de l'intérieur. De plus, si des trojans ont contaminé le parc informatique (navigation peu méfiante sur internet) ils pourront être ici facilement identifiés pour être ensuite éradiqués.

II.5 Classification des IDS

Il existe plusieurs classifications des systèmes de détection des intrusions, nous avons choisi le modèle apparu dans la Figure 1.17 :

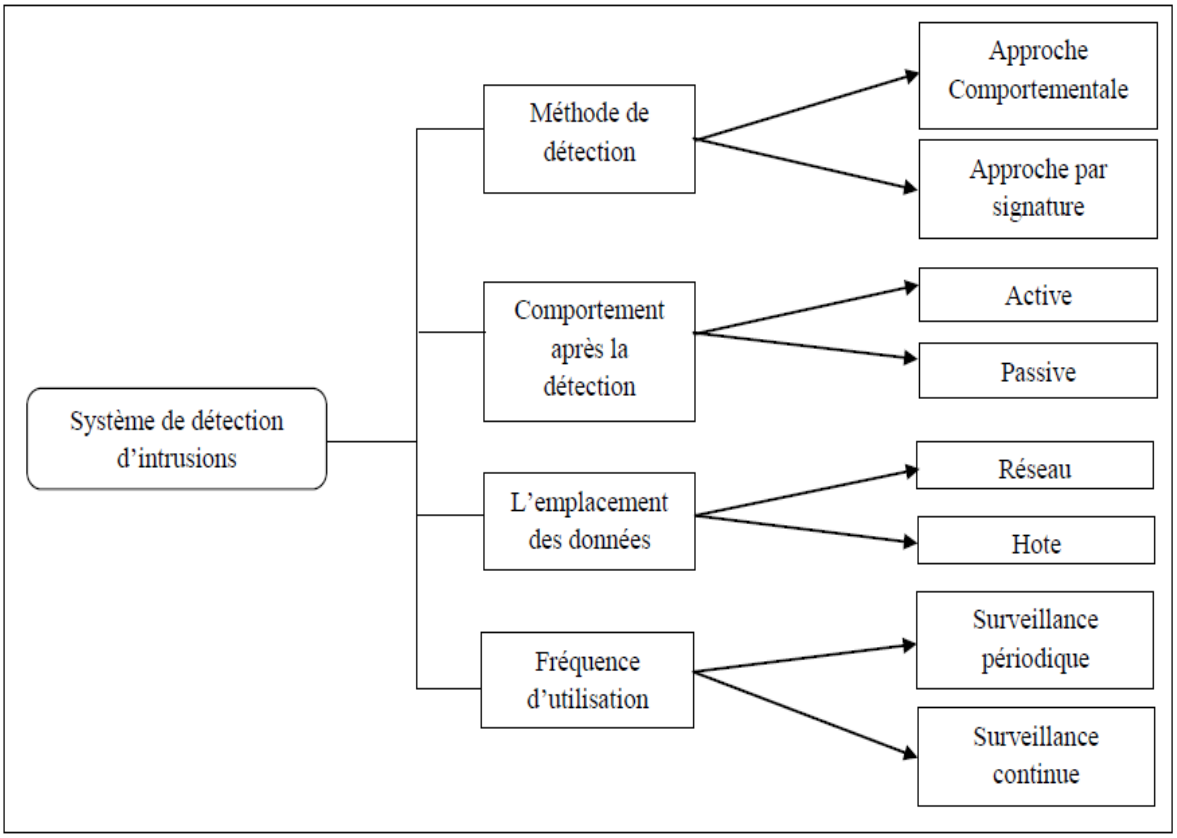

Figure 1.17. Classification d'IDS **[18].**

II.5.1 Les Techniques de détection

Comme le montre la Figure 1.18, il existe deux techniques principales de détection : par signatures (signature-based détection ou mésuse détection) ou par comportements (anomaly detection).

L'approche par signatures consiste à détecter des attaques en vérifiant si les observations correspondent à des attaques connues, tandis que l'approche par comportements (ou par détection d'anomalies) consiste à détecter une attaque en vérifiant que les observations ne correspondent pas à des comportements légitimes de référence. Certains IDS combinent les deux approches afin d'obtenir de meilleurs résultats.

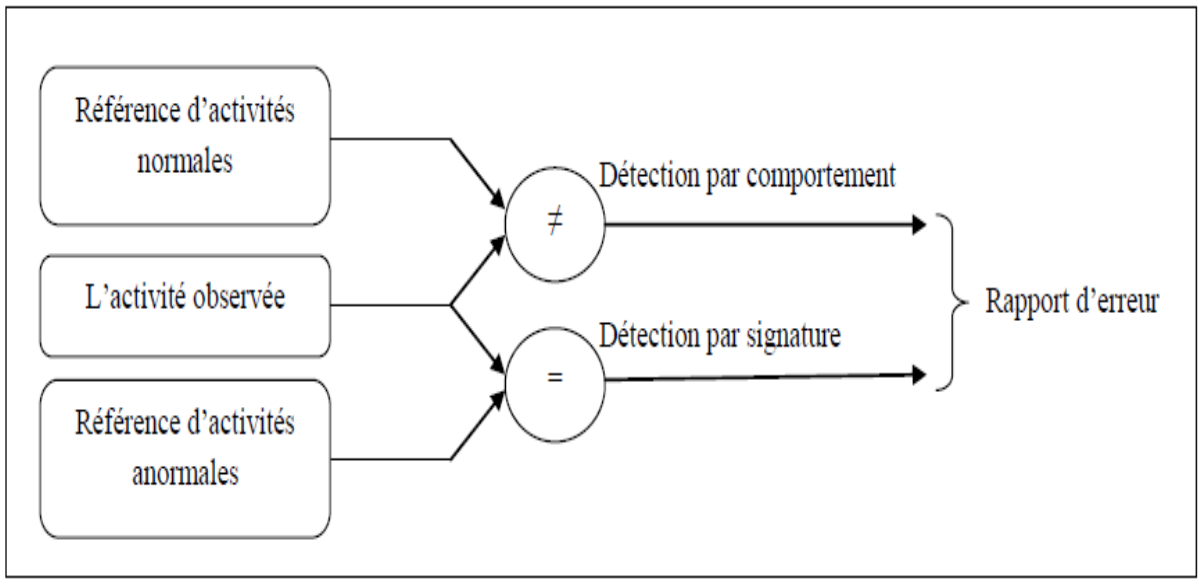

Figure 1.18. Techniques de détection d'intrusions **[19].**

II.5.1.1 Approche comportementale

Les modèles comportementaux sont apparus bien plus tard que les IDS à signatures. Ils ont pour principe la détection d'anomalies. Leur mise en œuvre comprend toujours une phase d'apprentissage au cours de laquelle ils vont "découvrir" le fonctionnement "normal" des éléments surveillés. Une fois cet apprentissage effectué ces IDS signaleront les divergences par rapport au fonctionnement de référence, les modèles comportementaux peuvent être élaborés à partir d'analyses statistiques ou de techniques proches de l'intelligence artificielle.

La principale caractéristique des IDS comportementaux est la détection des nouveaux types d'attaque, en effet ces IDS ne sont pas programmés pour reconnaître des attaques spécifiques mais signalent toute activité "anormale". De ce fait une attaque ne doit pas nécessairement être connue d'avance ; dès lors qu'elle représente une activité anormale elle peut être détectée par l'IDS comportemental. Du fait même de leur conception ces IDS sont

incapables de qualifier la criticité des attaques. De plus, ces IDS signaleront par exemple tout changement dans le comportement d'un utilisateur qu'il soit hostile ou non. De fréquents ajustements sont nécessaires afin de faire évoluer le modèle de référence de sorte qu'il reflète l'activité normale des utilisateurs et réduire le nombre de fausses alertes générées.

II.5.1.2 Approche par signature

Le concept de bibliothèque de signatures d'attaque est l'approche la plus basique et la plus ancienne. Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Cette démarche appliquée à la détection d'intrusion, est très similaire à celle des outils antivirus et présente les mêmes inconvénients que celle-ci. Il est aisé de comprendre que ce type d'IDS est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour quotidiennes. De plus, ce système de détection est aussi bon qu'il est la base de signature. Si les signatures sont erronées ou incorrectement conçues, l'ensemble du système est inefficace.

C'est pourquoi ces systèmes sont souvent contournés par les pirates qui utilisent des techniques dites "d'évasion" qui consistent à maquiller les attaques utilisées.

C'es techniques de maquillage tendent à faire varier les signatures des attaques qui ainsi ne sont plus reconnues par l'IDS. Ce modèle est par contre très aisé à implémenter et à optimiser

Il permet la séparation du moteur logiciel de la base de signature qui peut ainsi être mise à jour indépendamment. Il permet également une classification relativement facile de la criticité des attaques signalées.

Les avantages et les Inconvénients des deux approches (comportementale, par signature) sont montrés dans le Tableau 1.1:

L'approche	Avantages	Inconvénients
Comportementale	Détection d'intrusion inconnue possible.	□ □ Choix délicat des mesures à retenir pour un système cible donné. □ □ Pour un utilisateur au comportement erratique, toute activité est « normale ». □En cas de profonde modification de l'environnement du système cible, déclenchement d'un flot ininterrompu d'alarmes. □ □ L'utilisateur pouvant changer lentement de comportement dans le but d'habituer le système a un comportement intrusif.
par signature	Prise en compte des comportements exacts des attaques potentiels.	□ □ Base de règles délicates à construire. □ □ Seules les attaques contenues dans la base sont détectés.

Tableau 1.1. Comparaison entre les deux approches (comportementale et par signature).

II.5.2 Comportement après la détection

Ils existent deux types d'IDS ; actifs et passifs :

II.5.2.1 IDS à réponse passive

La réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par le responsable de sécurité ou générer des alarmes, envoi d'un e-mail à un ou plusieurs utilisateurs, etc. Ceci permet de remédier aux failles de sécurité

pour empêcher les attaques enregistrées de se reproduire, mais elle n'empêche pas directement une attaque de se produire.

II.5.2.2 IDS à réponse active

La réponse active au contraire a pour but de stopper une attaque au moment de sa détection sans attendre l'intervention humaine, Pour cela on dispose de deux techniques: la reconfiguration du firewall et l'interruption d'une connexion TCP, La reconfiguration du firewall permet de bloquer le trafic malveillant au niveau du firewall, en fermant le port utilisé ou en interdisant l'adresse de l'attaquant. Cette fonctionnalité dépend du modèle de firewall utilisé, tous les modèles ne permettant pas la reconfiguration par un IDS. De plus, cette reconfiguration ne peut se faire qu'en fonction des capacités du firewall.

L'IDS peut également interrompre une session établie entre un attaquant et sa machine cible, de façon à empêcher le transfert de données ou la modification du système attaqué **[20]**.

II.5.3 Emplacement de données

Il existe des IDS qui surveillent l'état de la sécurité au niveau du réseau par la capture et l'analyse des paquets qui circulent à travers le réseaux (NIDS: Network Intrusion Détection System) , et d'autres surveillent l'état de la sécurité au niveau des hôtes et analysent les informations produites par le système d'exploitation ou par des applications installées dans les machines locales (HIDS: Host Intrusion Détection System), quelques IDS hybrides utilisent les NIDS et HIDS pour avoir des alertes plus pertinentes **[21].**

II.5.3.1 Systèmes de détection d'intrusion réseaux (NIDS)

L'IDS réseau ou (NIDS: Network Intrusion Détection System) surveille le trafic réseau. Il se place sur un segment réseau et "écoute" le trafic. Ce trafic sera ensuite analysé afin de détecter les signatures d'attaques ou les différences avec le fonctionnement de référence,

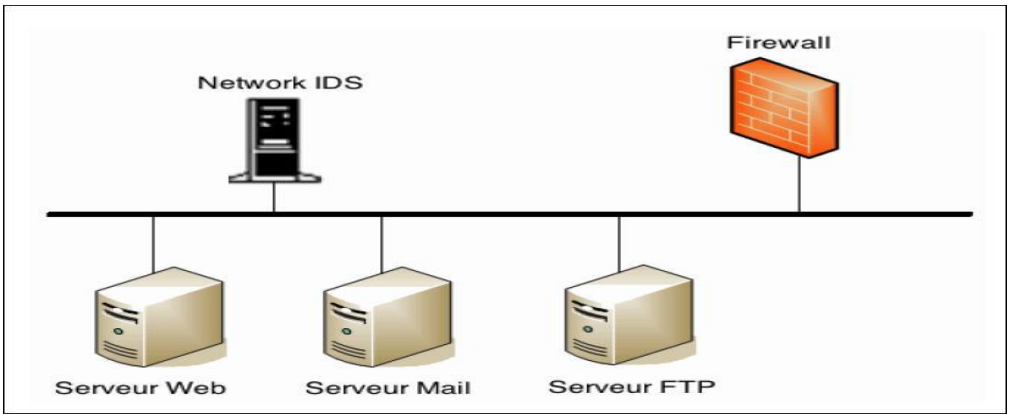

Figure 1.19. Exemple d'un IDS dans un réseau (NIDS) **[21].**

II.5.3.1.1 Les avantages des NIDS

- Le NIDS peut surveiller un grand réseau (un grand nombre d'hôte).
- 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 IDS avec l'effort minimal.
- NIDS peut être très sûr contre l'attaque et être même caché à beaucoup d'attaquants.

II.5.3.1.2 Les inconvénients des NIDS

- 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.
- NIDS ne peuvent pas analyser des informations chiffrées (dans le cas d'utilisation des VPN).
- Quelques NIDS provoquent des paquets en fragments. Ces paquets mal formés font devenir l'IDS instable.

II.5.3.2 Systèmes de détection d'intrusion sur hôte (HIDS)

HIDS : Host Intrusion Détection System ou L'IDS Système: accomplir le travail de surveillance du trafic sur une machine locale par l'analyse des journaux, les appels systèmes, analyse de la base de registre et des logs en provenance de firewalls hétérogènes et vérifie l'intégrité des systèmes de fichiers. Le principe de fonctionnement des HIDS dépend du système sur lequel ils sont installés. L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées par nature.

HIDS peut aussi observer les paquets réseaux de l'hôte (de la machine locale) pour la découverte des signaux d'intrusions (Déni de Services, Backdoors, chevaux de Troie, etc.).

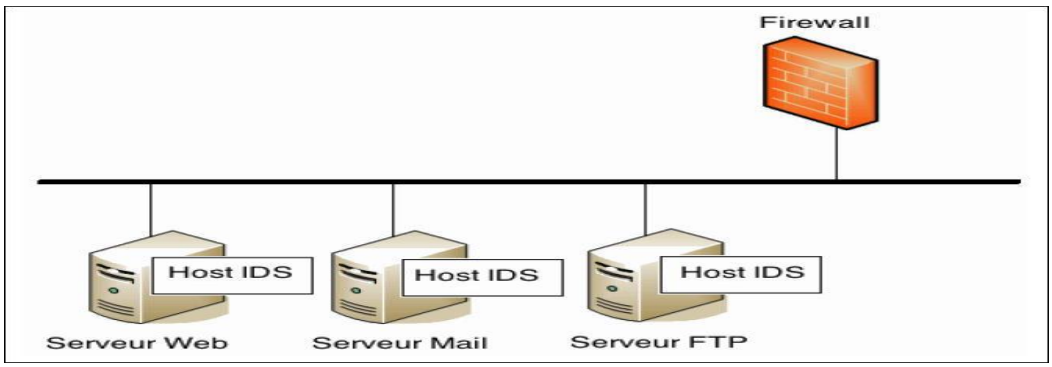

Figure 1.20. Système de détection d'intrusion hôte **[21].**

II.5.3.2.1 Les avantages des HIDS

- Pouvoir surveiller des événements locaux jusqu'au host, détecter des attaques qui ne sont pas vues par NIDS.
- Analyse des flux cryptés (ce que ne peut réaliser un NIDS). 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 peuvent détecter le cheval de Troie ou les autres attaques relatives à la brèche intégrité de logiciel.

II.5.3.2.2 Les inconvénients des HIDS

- HIDS est difficile à gérer, et des informations doivent être configurées et gérées pour chaque host surveillé.
- 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.

II.5.3.3 Les IDS hybrides

Les IDS hybrides rassemblent les caractéristiques de plusieurs IDS différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil, de surveiller les réseaux et les terminaux. Les sondes sont placées en des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout (Figure 1.21), et agréger/lier les informations d'origines multiples **[22].**

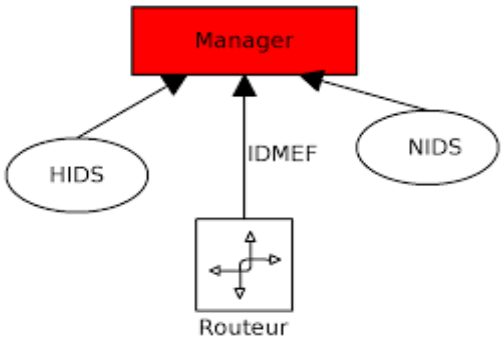

Figure 1.21. IDS hybride **[22].**

Les IDS hybrides sont donc basés sur une architecture distribuée, où chaque composant unifie son format d'envoi d'alerte (Exemple typique: IDMEF: Intrusion Detection Message

Exchange Format) permettant à des composants divers de communiquer et d'extraire des alertes plus pertinentes.

II.5.4 Fréquence d'utilisation

La dernière caractéristique des systèmes de détection d'intrusions est leur fréquence de surveillance : **périodique** ou **continue**.

II.5.4.1 Surveillance périodique

Dans ce cas les systèmes de détection d'intrusions analysent périodiquement les sources de données à la recherche d'une éventuelle intrusion ou anomalie passée. Cela peut être suffisant dans des contextes peu sensibles (on fera alors une analyse journalière, par exemple).

II.5.4.2 Surveillance continue

La plupart des systèmes de détection d'intrusions récents effectuent leur analyse des données sur la machine locale ou des paquets réseau de manière continue afin de proposer une détection en quasi temps-réel. Cela est nécessaire dans des contextes sensibles (confidentialité). C'est toutefois un processus coûteux en temps de calcul car il faut analyser à la volée tout ce qui se passe sur le système.

II.6 Les avantages d'utilisation des IDS

- **Déjouer les attaques attendues sur le réseau:** Les IDS protègent les systèmes contre les attaques réseaux par : détection de porte dérobée, détection d'usurpation d'adresse IP, DoS, les vers, les chevaux de Troie, virus, Botnet, rootkit, Spyware, et autres menaces qui pourraient nuire au réseau, Les IDS actifs prennent des mesures automatiques contre les menaces de sécurité et les risques auxquels font face.
- **Avertis Administrateur réseau d'alerte pour les événements de sécurité potentiels:** la fonction de base des systèmes de détection d'intrusions est de générer des avertissements là où existent des menaces externes, internes ou de violations de la politique de sécurité réseau, et aussi de fournir à l'administrateur des informations détaillées sur le mouvement des données au sein du réseau.
- **Gagnez du temps:** L'utilisation des IDS fournit beaucoup de temps et d'effort pour connaitre de ce qui se passe dans le réseau, peut aussi tourner en permanence sans superviseur humain.

- **Contrôle des Programmes utilisés par les employés pour surveiller l'Internet:** IDS peut aider à découvrir les programmes qui traitent de l'internet, cela permet de mieux contrôler et de protéger le réseau,
- **Avoir la confiance des clients:** Les IDS aident les organisations de protéger les données de ses clients contre le vol et la violation de la sécurité, Cela permet d'avoir la confiance des clients et partenaires et garder une bonne réputation sur l'organisation.
- **Économisez de l'argent:** Grace aux IDS les organisations peuvent déterminer les mouvements suspects dans le réseau et signaler les responsables pour prendre des mesures proactives en protéger le réseau et gagner l'argent qui sera dépensé si la violation de la sécurité est arrivé dans le réseau ou si le vol de renseignements personnels a eu lieu.

II.7 Les limites actuelles de la détection d'intrusions

- Les systèmes de détection d'intrusions actuels peuvent être mis en défaut, soit parce qu'ils sont incapables de détecter certains types d'attaques, soit parce qu'ils sont eux-mêmes attaquables.
- Les systèmes de détection d'intrusions actuels sont trop fermés, ce qui limite, les possibilités de comparaison de performances, et de coopération et de rendre difficile d'établir des standards d'interopérabilité entre outils pour résoudre ce problème.
- L'inadéquation entre les preuves exigées par les tribunaux et celles fournies par les outils de détection d'intrusions, cela nécessite des travaux supplémentaires pour extraire des preuves (des fichiers de logs du système, une partie du trafic réseau capturé durant l'attaque, des adresses IP incriminées et divers autres fichiers) lorsque des poursuites en justice sont envisagées, Le problème qui se pose est comment prouver que tout cela n'a pas été altéré.
- Les systèmes de détection d'intrusions génèrent trop de faux positifs.

Conclusion

Dans ce chapitre nous avons abordé différentes notions de la sécurité informatique, on a expliqué les attaques informatiques et les classifications attaques, parmi ce mécanisme on a détaillé les systèmes de détection des intrusions vu que c'est notre objectif dans ce mémoire,

qui joue un rôle complémentaire aux mécanismes de sécurité traditionnels. Nous avons présenté aussi le principe de fonctionnement des IDS ainsi leurs classifications selon différents critères avec leurs avantages et inconvénients, parmi les critères de classification abordés la méthode de détection qui divise les IDS en deux types, les IDS comportementaux et à base de signatures, le principe de ce dernier consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues d'une façon similaire à celle des antivirus, ce type d'IDS est simple d'implémenter mais il présente des difficultés comme la nécessité de mettre à jour d'une façon quotidienne la base de signature, en plus seuls les attaques contenues dans la base sont détectables, cela peut baissé l'efficacité des systèmes de détection des intrusions à base des signature surtout avec l'énorme développement des attaques réseaux d'aujourd'hui. Dans le chapitre suivent, nous passons en revue les techniques de classification relavant de l'apprentissage automatique.

Chapitre2 : L'apprentissage automatique

Introduction

Suite au développement rapide des différentes méthodes et techniques de piratage et avec le nombre très important d'attaques effectuées dans le monde entier, la protection des données cyber structurelle a connu une très grande vulnérabilité causée par l'incapacité de suivre le rythme d'évolution de la cybercriminalité, pour cela, les chercheurs en sécurité informatique ont utilisé les différentes techniques d'apprentissage automatique, de statistique et de data mining, afin de relever les défis de la cyber sécurité.

Dans ce deuxième chapitre, nous allons présenter l'intelligence artificielle, les types d'apprentissage automatique et l'apprentissage basé sur la détection d'intrusion.

I. L'intelligence artificielle (IA)

I.1 Définition de l'intelligence artificielle (IA)

Selon le Larousse, l'intelligence Artificielle se définirait comme étant « l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence. » Ce serait, de ce fait, des ordinateurs ou des machines dotées de programmes capables de performances similaires à l'intelligence humaine, ou même, amplifiées par la technologie.

Ces machines sont en mesure de:

- Raisonner.
- Traiter de grandes quantités de données.
- Discerner des modèles indétectables par l'œil d'un humain.
- Comprendre et analyser ces modèles.
- Interagir avec l'Homme.
- Apprendre progressivement.
- Améliorer continuellement ses performances **[23].**

Depuis 1950, année à laquelle l'IA a été créée, cette dernière est en mutation. Elle a, même en janvier 2018, franchi l'étape selon laquelle elle dépasserait l'intelligence humaine.

I.2 Ou'est-ce que l'intelligence artificielle?

Qu'est-ce que l'intelligence ? Est-ce la capacité à percevoir le monde, à prédire le futur immédiat ou lointain, ou à planifier une série d'actions pour atteindre un but ? Est-ce la capacité d'apprendre, ou celle d'appliquer son savoir à bon escient ? La définition est difficile à cerner.

O pourrait dire que l'intelligence artificielle (IA) est un ensemble de techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés aux humains et à certains animaux.

Les tâches relevant de l'IA sont parfois très simples pour les humains, comme par exemple Reconnaitre et localiser les objets dans une image, planifier les mouvements d'un robot pour attraper in objet, ou conduire une voiture Elles requièrent parfois de la planification complexe, comme par exemple pour jouer aux échecs ou au Go. Les tâches les plus compliquées requièrent beaucoup de connaissances et de sens commun, par exemple pour traduire un texte ou conduire un dialogue.

Depuis quelques années, on associe presque toujours l'intelligence aux capacités d'apprentissage. C'est grâce à l'apprentissage qu'un système intelligent capable d'exécuter une tâche peut améliorer ses performances avec l'expérience. C'est grâce à l'apprentissage qu'il pourra apprendre à exécuter de nouvelles tâches et acquérir de nouvelles compétences.

Le domaine de l'IA n'a pas toujours considéré l'apprentissage comme essentiel à l'intelligence. Par le passé, construire un système intelligent consistait à écrire un programme à la main » pour jouer aux échecs (par recherche arborescente), reconnaitre des caractères imprimés (par comparaison avec des images prototypes), ou faire un diagnostic médical à partir des symptômes (par déduction logique à partir de règles écrites par des experts) Mais cette approche manuelle » a ses limites **[24].**

I.3 Comment fonctionne l'intelligence artificielle ?

Selon Harry Shum, Président Exécutif de Microsoft, I'IA fonctionne seulement s'il y a présence «d'une vaste quantité de data : d'une puissance informatique extraordinaire, notamment grâce au Cloud : et des algorithmes révolutionnaires, basés sur le Deep Learning ».

L'IA s'applique aujourd'hui dans des domaines variés tels que :

- Les jeux de réflexion.
- La recherche mathématique.

- La finance.
- La médecine.
- Les assistants personnels et la domotique.
- La reconnaissance faciale et la compréhension des langues.
- La robotique.

I.4 L'intelligence artificielle aujourd'hui ses enjeux

Les opportunités sont telles que l'IA, particulièrement l'apprentissage profond, est vue comme des technologies d'importance stratégique pour l'avenir. Les progrès en vision par ordinateur ouvrent la voie aux voitures sans chauffeur, et à des systèmes automatisés d'analyse d'imagerie médicale.

Des systèmes d'analyse d'images médicales détectent des mélanomes et autres tumeurs de manière plus fiable que des radiologues expérimentés. Chez Facebook. Google et Microsoft, des systèmes de reconnaissance d'image permettent la recherche et L'organisation des photos et le filtrage d'images violentes ou pornographiques. Depuis plusieurs années déjà, tous les moteurs de reconnaissance vocale sur smart phone utilisent l'apprentissage profond.

Des efforts considérables de R&D sont consacrés au traitement du langage naturel: la compréhension de texte, les systèmes de question-réponse, les systèmes de dialogue pour les agents virtuels, et la traduction automatique. Dans ce domaine, la révolution de l'apprentissage profond a été annoncée, mais n'est pas encore achevée. Néanmoins, on assiste à des progrès rapides. Dans la dernière compétition internationale de traduction automatique, le gagnant utilisait un réseau récurrent **[24].**

I.5 La recherche en intelligence artificielle et les obstacles au progrès

Malgré tous ces progrès, nous sommes encore bien loin de produire des machines aussi intelligentes que l'humain, ni même aussi intelligentes qu'un rat. Bien sûr, nous avons des systèmes qui peuvent conduire une voiture, jouer aux échecs et au Go, et accomplir d'autres tâches difficiles de manière plus fiable et rapide que la plupart des humains (sans parler des rats). Mais ces systèmes sont très spécialisés. Un gadget à 30 euros nous bat à plate couture aux échecs, mais il ne peut faire rien d'autre.

Ce qui manque aussi aux machines, c'est la capacité à apprendre des tâches qui impliquent non seulement d'apprendre à représenter le monde, mais aussi à se remémorer, à raisonner, à prédire et à planifier. Beaucoup de travaux actuels à Facebook AI Research et à

DeepMind sont focalisés sur cette question. Une nouvelle classe de réseaux neuronaux, les Memory Augmented Recurrent Neural Nets (réseaux récurrents à mémoire) est utilisée de manière expérimentale pour la traduction, la production de légendes pour les images, et les systèmes de dialogues. Mais ce qui manque principalement aux machines, c'est le sens commun, et la capacité à l'intelligence générale qui permet d'acquérir de nouvelles compétences, quel qu'en soit le domaine. Mon opinion, qui n'est partagée que par certains de mes collègues, est que l'acquisition du sens commun passe par l'apprentissage non supervisé. Qu'il soit naturel ou artificiel, il y a trois formes principales d'apprentissage. Nous avons déjà parlé de l'apprentissage supervisé. Les deux autres formes sont l'apprentissage par renforcement, et l'apprentissage non supervisé.

L'apprentissage par renforcement désigne la situation où la machine ne reçoit qu'un simple signal, une sorte de récompense, indiquant si la réponse produite était correcte ou pas.

Le scénario est similaire à l'entrainement d'un animal de cirque à qui l'on donne une friandise lorsqu'il exécute l'action désirée. Cette forme d'apprentissage nécessite de très nombreux essais, et est utilisée principalement pour entraîner les machines à jouer à des jeux (par exemple les jeux vidéo ou le jeu de Go), ou à opérer dans des environnements simulés.

On a assisté à un succès éclatant de l'apprentissage par renforcement combiné à l'apprentissage profond lors de la victoire récente du programme de Go AlphaGo de DeepMind face au champion européen. L'apprentissage non supervisé, quant à lui, est le mode principal d'apprentissage des animaux et des humains. C'est l'apprentissage que nous faisons par nous même en observant le monde et en agissant. C'est en observant le monde que nous apprenons qu'il a trois dimensions, que des objets peuvent en cacher d'autres, que certains objets peuvent être déplacés, qu'un objet sans support tombe, qu'un objet ne peut pas être à deux endroits en même temps, etc.

C'est grâce à l'apprentissage non supervisé que nous pouvons interpréter une phrase simple comme «Jean prend son portable et sort de la pièce». On peut inférer que Jean et son portable ne sont plus dans la pièce, que le portable en question est un téléphone, que Jean s'est levé, qu'il a étendu sa main pour attraper son portable, qu'il a marché vers la porte. Il n'a pas volé, il n'est pas passé à travers le mur. Nous pouvons faire cette inférence, car nous savons comment le mode fonctionne. C'est le sens commun.

Comment acquérir ce sens commun ? Une hypothèse possible est l'apprentissage prédictif.

Si l'on entraîne une machine à prédire le futur, elle ne peut y arriver qu'en élaborant une bonne représentation du monde et de ses contraintes physiques. Dans un scénario

d'apprentissage prédictif, on montre à la machine un segment de vidéo, et on lui demande de prédire quelques images suivantes. Malheureusement, le futur est impossible à prédire exactement et la machine s'en tient à produire une image floue, une mixture de tous les futurs possibles. Si l'intelligence est un gâteau au chocolat, le gâteau lui-même est l'apprentissage non supervisé, le glaçage est l'apprentissage supervisé, et la cerise sur le gâteau est l'apprentissage par renforcement. Les chercheurs en IA sont dans la même situation embarrassante que les physiciens: 95 % de la masse de l'univers est de nature complètement inconnue matière noire et énergie noire. La matière noire de l'AI est la génoise au chocolat de l'apprentissage non supervisé.

Tant que le problème de l'apprentissage non supervise ne sera pas résolu, nous n'aurons pas de machine vraiment intelligente. C'est une question fondamentale scientifique et mathématique, pas une question de technologie. Résoudre ce problème pourra prendre de nombreuses années ou plusieurs décennies. En vérité, nous n'en savons rien **[24].**

I.6 Faut-il avoir peur de l'intelligence artificielle ?

L'IA n'éliminera donc pas l'humanité de sa propre initiative. Mais comme toute technologie puissante, I'LA peut être utilisée pour le bénéfice de l'humanité entière ou pour le bénéfice d'un petit nombre aux dépens du plus grand nombre.

L'émergence de l'AI va sans doute déplacer des métiers. Mais elle va aussi sauver des vies (par la sécurité routière et la médecine). Elle va très probablement s'accompagner d'une croissance de la production de richesses par habitant. La question pour les instances dirigeantes est comment distribuer ces nouvelles richesses. Et comment former les travailleurs déplacés aux nouveaux métiers créés par le progrès technologique. C'est une question politique et non technologique. C'est une question qui n'est pas nouvelle : l'effet du progrès technologique sur le marché du travail existe depuis la révolution industrielle. L'émergence de l'IA n'est qu'un symptôme de l'accélération du progrès technologique.

II. L'apprentissage automatique

II.1 Définition d'apprentissage automatique

L'apprentissage automatique ou statistique, aussi appelé «machine Learning» est un domaine à la jonction des statistiques et de l'intelligence artificielle qui a pour but la résolution automatique de problèmes complexes à partir d'exemples. La démarche de

conception d'un modèle par apprentissage nécessite de postuler une fonction, dont les variables sont susceptibles d'avoir une influence sur la grandeur à modéliser. Cette fonction dépend des paramètres ajustables.

L'apprentissage statistique consiste en l'ajustement de ces paramètres de telle manière que le modèle ainsi obtenu présente les qualités requises d'apprentissage et de généralisation **[25].**

II.2 Types d'apprentissage automatique

Il existe de nombreux types de systèmes d'apprentissage automatique (Machine Learning). Dans ce qui suit, nous les classons selon qu'ils nécessitent ou non une supervision humaine (supervisés, non supervisés, apprentissage de renforcement).

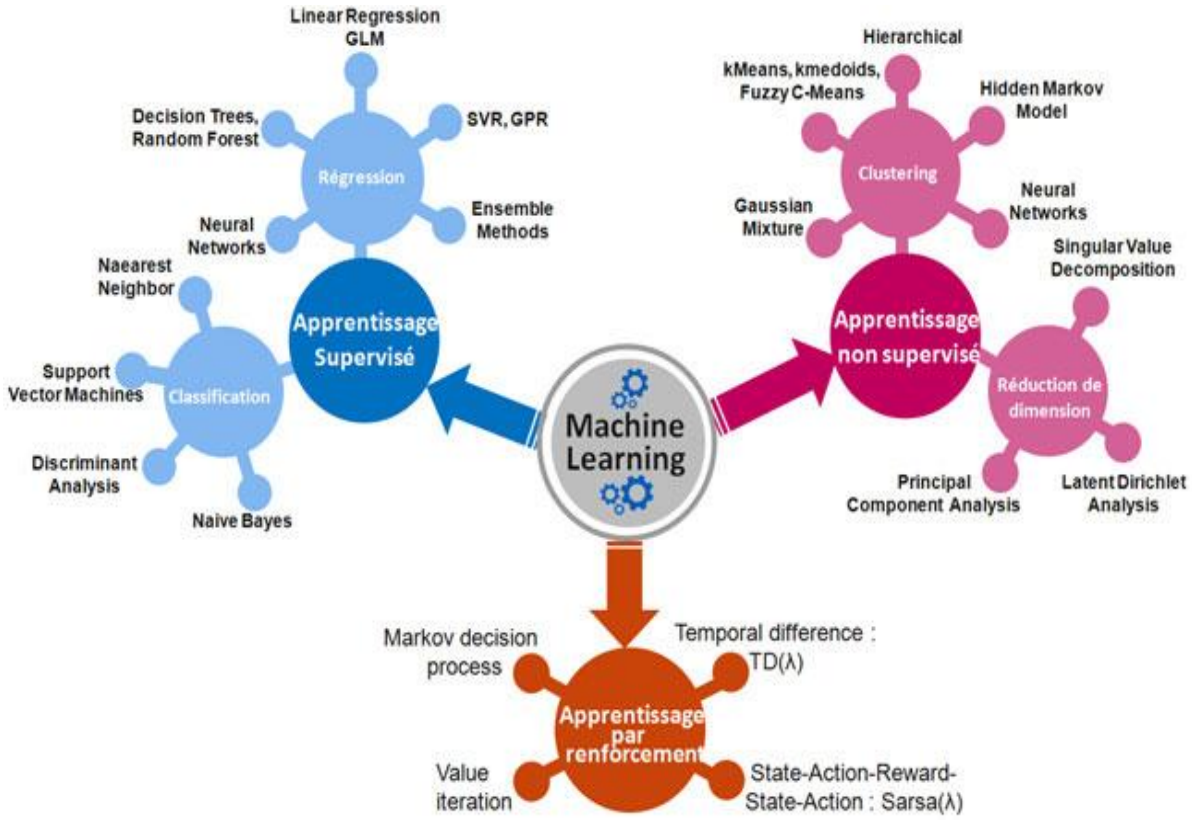

Figure 2.1. Quelques algorithmes des 3 types d'apprentissage du Machine Learning: supervisé ou non supervisé et par renforcement **[26].**

II.2.1 Apprentissage supervisé

Dans l'apprentissage supervisé, les données fournies sont des paires : une entrée et une étiquette. On parle alors d'entrées étiquetées. Le but de l'apprentissage est d'inférer la valeur de l'étiquette étant donnée la valeur de l'entrée. On peut distinguer deux grands types d'apprentissage supervisé: la classification et la régression **[25].**

II.2.1.1 Classification

Lorsqu'on fait de la classification, l'entrée est l'instance d'une classe et l'étiquette est la classe correspondante. La classification consiste donc à apprendre une fonction f class de X= Rd dans Y= N qui associe à un vecteur sa classe. Si le nombre de classe est égal à 2, on parle alors de la classification binaire **[25].**

II.2.1.2 Régression

Dans le cas de la régression, l'entrée n'est pas associée à une classe mais à une ou plusieurs quantités continues. Ainsi, l'entrée pourrait être les caractéristiques d'une personne (son âge, son sexe, son niveau d'études) et l'étiquette son revenu.la régression consiste donc à apprendre une fonction fregr de X =Rd dans Y=Rk qui associe à un vecteur sa valeur associée **[27].**

Voici quelques-uns des algorithmes d'apprentissage supervisé:

- K-plus proches voisins (k-NN).
- Machines à vecteurs de support (Support Vector Machine).
- Arbres de décision.
- Régression logistique.
- Naïve Bayésien.
- Random Forest.
- Régression linéaire.

II.2.2 Apprentissage non-supervisé

On dit que l'apprentissage est non supervisé lorsqu'on ne connaît pas les valeurs en sortie et que l'algorithme doit travailler sur l'ensemble des aspects x où il doit reconnaître les structures communes entre ce²s derniers pour prédire la cible y. Dans l'apprentissage non supervisé, l'agent apprend des structures dans les données d'entrée, même s'il ne dispose pas de feedback explicite sur ses actions **[25].**

On peut ainsi regrouper des données dans des Clusters (c'est le Clustering), détecter des anomalies, ou encore réduire la dimension de données très riches en compilant les dimensions ensemble **[28].**

II.2.2.1 Clustering

Par exemple, disons que nous avons beaucoup de données sur les visiteurs de notre blog. Nous pouvons utiliser un algorithme de Clustering pour essayer de détecter des groupes

de visiteurs similaires. À aucun moment on dit à l'algorithme de Clustering à quel groupe appartient un visiteur : il trouve ces connexions sans notre aide. Par exemple, l'algorithme pourrait remarquer que 40 % de vos visiteurs sont des hommes qui aiment les bandes dessinées et qui lisent généralement votre blog le soir, tandis que 20 % sont de jeunes amateurs de science-fiction qui visite le site durant le week-end, etc. Si nous utilisons un algorithme de Clustering hiérarchique, celui ci peut subdiviser chaque groupe en plus petits groupes. Cela peut aider à cibler des messages pour chaque groupe.

Dans le graphique suivant « figure 2.2 », nous montrons comment un ensemble de points peut être classé pour former trois sous-ensembles :

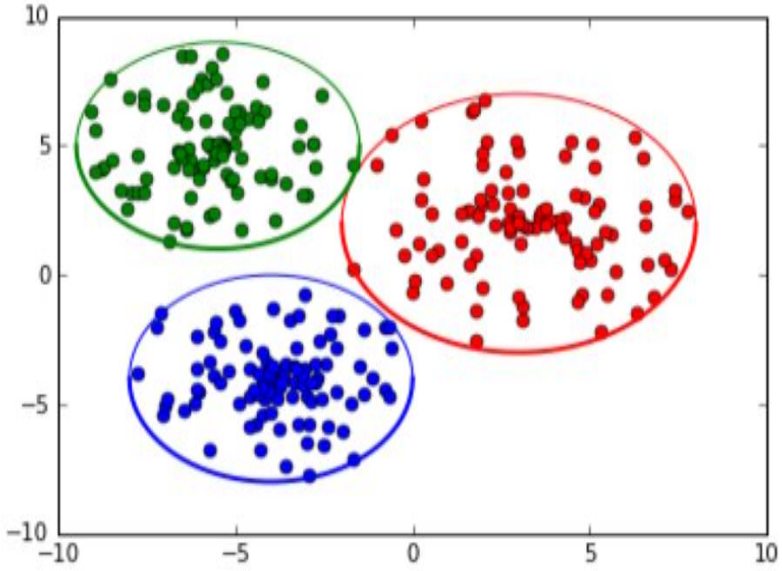

Figure 2.2. Clustering **[29].**

II.2.2.2 Réduction de la dimensionnalité

L'objectif est de simplifier les données sans perdre trop d'informations. Une façon d'y parvenir est de fusionner plusieurs caractéristiques corrélées en une seule. Par exemple, le kilométrage d'une voiture peut être très corrélé avec son âge, de sorte que l'algorithme de réduction de la dimensionnalité les fusionnera en une seule caractéristique qui représente l'usure de la voiture. C'est ce qu'on appelle l'extraction de caractéristiques.

II.2.2.3 La détection des anomalies

La détection des anomalies est un domaine passionnant, qui vise à identifier les objets éloignés qui sont déviants de la distribution générale des données. La détection des valeurs aberrantes s'est avérée essentielle dans de nombreux domaines, par exemple, la détection des transactions inhabituelles par carte de crédit pour éviter la fraude, la détection des défauts de

fabrication ou la suppression automatique des valeurs aberrantes d'un ensemble de données avant de les transmettre à un autre algorithme d'apprentissage. Le système est entraîné avec des instances normales et, lorsqu'il voit une nouvelle instance, il peut dire si elle ressemble à une instance normale ou s'il s'agit probablement d'une anomalie, comme le montre le graphe suivant « figure 2.3 » :

Figure 2.3. Détection d'anomalies **[25].**

II.2.2.4 Apprentissage des règles d'association

L'objectif est de fouiller dans de grandes quantités de données et de découvrir des relations intéressantes entre les attributs. Par exemple, supposons que nous sommes propriétaire d'un supermarché.

L'application d'une règle d'association sur les registres de vente peut révéler que les personnes qui achètent de la sauce barbecue et des pommes de terre ont également tendance à acheter du steak. Par conséquent, on pourrait peut-être placer ces articles à proximité les uns des autres.

Voici quelques-uns des algorithmes d'apprentissage non supervisé :

- K moyennes (K-Means).
- Réseaux de neurones artificiels.
- Apriori.

- La comparaison entre les deux apprentissages supervisé et non supervisé :

	Apprentissage supervisé	**Apprentissage non supervisé**
Données d'entrée	Utilise les données connues et étiquetées comme entrées.	Données inconnues en entrées.
Complexité informatique	Très complexe.	Moins de complexité informatique.
Temps réel	Utilise l'analyse hors ligne.	Utilise l'analyse en temps réel des données.
Sous-domaines	Classification et régression.	Exploitions de règles de clustering et d'association.
Précision	Produit des résultats précis.	Génère des résultats modérés.
Nombre de classes	Nombre de classes connues.	Le nombre de classes n'est pas connu.

Tableau 2.1. Comparaison entre les deux apprentissages supervisé et non supervisé.

II.2.3 Apprentissage par renforcement

L'apprentissage par renforcement, c'est apprendre à agir par essais et erreur. Dans ce paradigme, un agent peut percevoir son état et effectuer des actions. Après chaque action, une récompense numérique est donnée. Le but de l'agent est de maximiser la récompense totale qu'il reçoit au cours du temps **[30].**

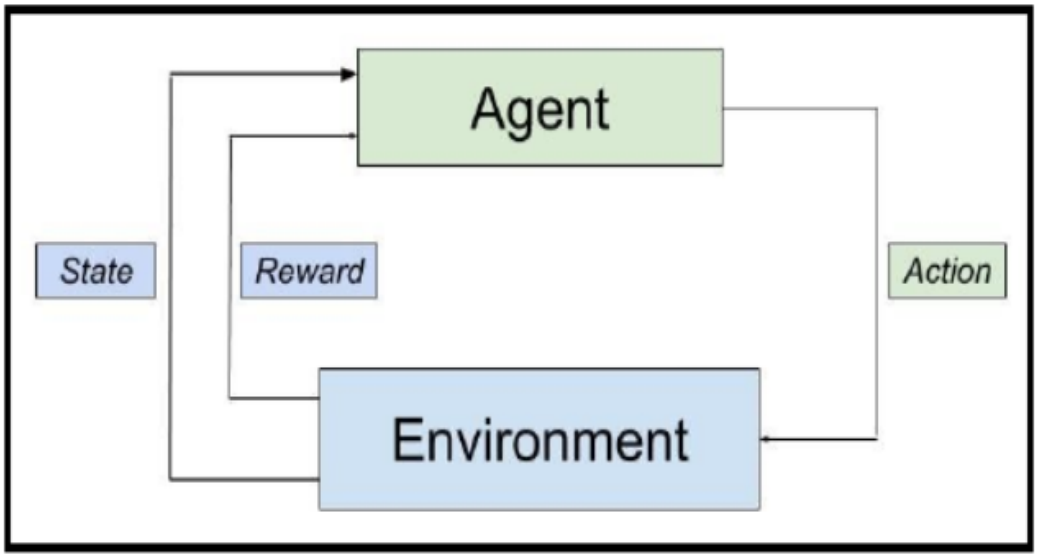

Figure 2.4. Apprentissage par renforcement **[30].**

II.3 Processus de Machine Learning

La figure suivante permet une explication des différentes phases du processus de Machine Learning:

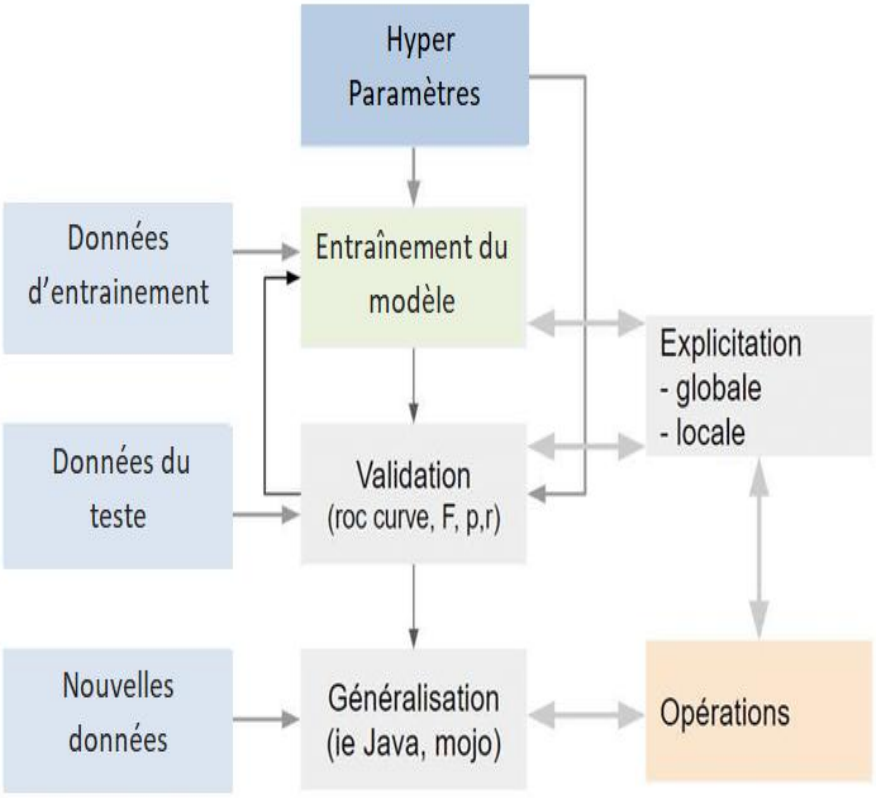

Figure 2.5. Le processus de Machine Learning **[31].**

Afin de mieux expliquer le processus de Machine Learning, nous commençons par la définition de quelques concepts de base :

- **Dataset**

C'est un ensemble de données qu'on fournit à une machine sous forme d'un couple d'exemples

(x, y) dans l'apprentissage supervisé où x représente les questions et y les réponses au problème que la machine doit résoudre. Dans l'apprentissage non supervisé, le dataset contient que des questions x. On ne peut pas démarrer un projet sans avoir de dataset.

- **Modèle et ses paramètres**

C'est une fonction mathématique qu'on développe à partir du dataset fournit et qui peut être linéaire ou bien non-linéaire. Les coefficients de cette fonction sont les paramètres du modèle et ils sont les prédicteurs x et l'annotation y que l'on veut généraliser.

- **Les hyper-paramètres**

Les hyper-paramètres sont les valeurs de réglage du modèle: le nombre d'itérations, les valeurs de seed (valeur aléatoire initiales), la solution initiale et les autres paramètres spécifiques des différents modèles testés **[31].**

- **Fonction Coût**

La phase de validation établit la performance du modèle en termes de taux de faux positifs (les fausses alertes) et de faux négatifs (les ratés) que l'on doit réduire simultanément **[31].**

Ceci se fait par une fonction coût qui représente un ensemble d'erreurs qu'un modèle nous retourne par apport à notre dataset. La validité du modèle dépend de la fonction coût à partir de laquelle la machine distingue les paramètres de notre modèle qui minimisent cette dernière.

- **La généralisation**

Consiste à intégrer le modèle dans un processus de big data, et dépasse l'horizon méthodologique concernant l'industrialisation des processus de plus en plus souvent assurée par une distribution du calcul **[31].**

Le processus de Machine Learning ce résume comme suit: Premièrement, on obtient l'échantillon d'entraînement qui représente les prédicteurs x, ce sont les données d'entraînement, et l'annotation y que l'on veut généraliser (ensemble de dataset). Puis on associe les hyper paramètres à notre modèle. Un entraînement du modèle sera effectué et passera par la phase de validation où l'efficacité du modèle s'étudiera et un calcule de nombre d'erreurs se réalisera par la fonction coût pour valider le modèle. Sachant qu'un algorithme d'apprentissage doit être choisit, une fonction issue d'un ensemble de fonction défini. Au préalable, réalise l'erreur moyenne la plus faible sur les exemples de la base d'entraînement.

Enfin, on passe à la phase de généralisation où un test du modèle se déroulera sur les données globales.

III. Systèmes de détection d'intrusion basés sur l'apprentissage automatique

III.1 Obtention des données

La première étape est l'obtention des données. Lors de la phase d'apprentissage, ces informations permettent de connaître les habitudes des utilisateurs ou les différents types d'attaques de manière générale. Lors de la phase d'exécution, ils permettent de détecter une attaque.

Néanmoins, la question est de savoir s'il existe une mesure de la quantité de données nécessaire pour avoir un modèle correct, c'est-à-dire qui respecte un certain taux d'erreur accepté. Cette question est encore ouverte à ce jour. Toutefois, le choix des informations à prendre en compte dépendra du type d'apprentissage.

III.2 Supervisé ou non supervisé

Les apprentissages supervisés utilisent généralement des data sets représentant des attaques connues. Les apprentissages non supervisés se basent uniquement sur les comportements des utilisateurs pour détecter une variation par rapport aux comportements normaux qui représentent une attaque. Selon l'article les algorithmes supervisés réalisent d'excellents résultats pour des intrusions connues, ils sont meilleurs que les algorithmes non supervisés. Inversement, pour des agressions inconnues, les algorithmes supervisés voient une réduction drastique de leur efficacité contrairement aux algorithmes non supervisés. Ceci peut s'expliquer par le fait que, puisque les algorithmes non supervisés ne font que partitionner des données, les attaques leur sont toujours inconnues.

En effet, étant donné que ces derniers ne font que regrouper des données proches, ils ne savent pas quand elles représentent une attaque. Bien que les algorithmes supervisés et non supervisés obtiennent des résultats semblables pour des attaques inconnues, les modèles non supervisés sont préférés pour leur robustesse. En effet, ceux-ci ne changent pas drastiquement leur taux de réussite selon que l'intrusion est connue ou non. De plus, ils n'ont pas d'oracle leur disant à quelle classe appartient telle donnée. Ils réalisent solitairement les classes. Ils sont donc plus indépendants que les techniques supervisées qui ont besoin d'un oracle. Leur point fort est la classification d'attaques appartenant à un label inconnu. En effet, on ne doit pas leur présenter l'ensemble des labels. De plus, il est parfois très dur de classifier

univoquement une information dans une classe. C'est pourquoi cette méthode est parfois préférée pour les IDS **[32].**

La thèse parle d'une implémentation du modèle SOM sur les IDS. Les résultats de cette thèse sont néanmoins peu encourageants pour ce type de structure. Par ailleurs, les algorithmes supervisés et non supervisés peuvent être combinés pour obtenir un apprentissage semi-supervisé où on utilise les avantages des deux techniques précédentes pour réaliser un IDS. Cependant, une hypothèse sur les IDS supervisés reste **[32].**

III.3 Hypothèse

Dans le cas des algorithmes supervisés, une hypothèse difficile à respecter lors de l'apprentissage est qu'il n'y a pas d'attaque contenue dans les informations modélisant le comportement normal des programmes qui peuvent pourtant avoir beaucoup d'irrégularités.

En effet, dans le cas contraire, on insérerait des propriétés d'une attaque comme comportement normal.

Par conséquent, cette dernière ne sera pas détectée. Inversement, un algorithme non supervisé peut lever cette hypothèse en nettoyant les données pour enlever les informations d'attaque. Pour cela, on recherche un ensemble d'enchaînement, appelé motif, de système call fortement présent dans le système et on les range suivant leur dangerosité d'attaquer le système **[33].**

Par après, on regarde l'ensemble des séquences des systèmes calls présentes pour les comparer aux motifs découverts précédemment, réalisant ainsi une sorte d'empreinte des différentes séquences de système calls par rapport aux motifs qu'ils contiennent. Par après, on regarde la distance de toutes les empreintes par rapport à une autre afin d'avoir un graphique unique pour l'ensemble.

Ceci nous permet de mieux visualiser celles de même nature (qui sont donc très proches les unes des autres) par rapport à celles très différentes. Ainsi, un espacement très grand entre deux groupes d'empreintes où l'un en contient beaucoup et l'autre très peu permet de supposer que l'un en est une attaque alors que l'autre ne l'est pas. De plus, comme on suppose qu'il y a moins d'empreintes d'attaques que de non-attaques, on retrouve facilement l'ensemble de celles d'attaques. C'est pourquoi on utilise un algorithme non supervisé offline pour détecter les anomalies dans les données. Par après, un algorithme supervisé peut utiliser ces données nettoyées pour concevoir et implémenter le modèle.

III.4 Implémentation

Une méthode généralement utilisée, qui est basée sur l'approche comportementale, est la prise d'empreintes des utilisateurs, c'est-à-dire de leur comportement, et de regarder quand elle ne lui correspond pas sur le système. Ainsi, on peut détecter un comportement anormal et donc une attaque éventuelle. Inversement, on peut prendre l'empreinte de certains pirates connus pour les détecter.

Cette dernière peut être apprise par la machine Learning. Toute fois, une autre méthode, basée sur l'approche par scénario, est l'utilisation de données représentant des attaques. Pour que cela soit réaliste, il faut que l'IDS soit suffisamment rapide, efficace et flexible aux petits changements des utilisateurs, sans toutefois permettre de dévier vers une situation d'attaque.

Dans le cas contraire, soit elle ne sera pas détectée soit le nombre de faux positifs pourrait exponentiellement augmenter. De manière globale, on réalise un tel IDS en trois étapes: on modélise tous les comportements normaux de chaque utilisateur ou les signatures des attaques, on le donne au machine Learning pour qu'il l'apprenne et ensuite on regarde si le comportement dévie de l'habituel ou s'approche d'une situation offensive. Dans certains cas, il est intéressant de savoir le type de l'attaque et non seulement si elle a eu lieu **[34].**

Une dernière implémentation est la gestion d'un grand nombre d'alertes venant des IDS par du machine Learning. Ainsi, cette méthode est une sorte de filtre de ces dernières permettant de se focaliser sur les alarmes les plus importantes. En effet, un IDS peut générer un nombre volumineux de fausses alertes, ce qui rend la tâche des administrateurs système impossible.

Ceci est donc un complément aux IDS et non un remplacement de ceux-ci. Comme vu précédemment, il existe des HIDS et des NIDS.

HIDS : Pour modéliser le comportement d'un utilisateur, on peut regarder l'ensemble des commandes qu'il a utilisées durant une période, ce qui donne un HIDS offline. Cette méthode est justifiable puisque la plupart des personnes n'utilisent pas le système dans le même but ni de la même manière. Pour un HIDS semblable mais online, la machine apprend à reconnaître les commandes futures selon les k dernières utilisées. Néanmoins, leur ordre n'est pas révélateur pour savoir si une attaque a lieu ou pas. Il semble plus significatif de regarder l'ensemble des commandes utilisées durant une période. L'inconvénient majeur est la non-prise en compte des arguments des system calls. Ainsi, il est intéressant de prendre en compte les valeurs de retour, les statuts d'erreur et d'autres arguments pour détecter des attaques. En effet, prenons l'exemple des system calls suivants exécutés par un simple utilisateur : open,

read, write. Ces trois systèmes calls peuvent sembler inoffensifs puisque c'est une simple ouverture, lecture et écriture dans un _chier.

Néanmoins, la situation change si on regarde l'argument de ces system calls et que celui-ci est le fichier paswd. Une autre méthode analysée est l'apprentissage du profil des programmes et non des utilisateurs. Ainsi, la machine apprend le fonctionnement normal des logiciels sur une machine **[33].**

NIDS : Pour ce qui est du NIDS, il est nécessaire de bien comprendre l'ensemble des variables d'un paquet, ainsi que du protocole, pour comprendre le fonctionnement normal du système.

Essayons de partitionner les paquets TCP. Pour cela, on peut d'abord voir les attributs qui ne changeront sans doute jamais entre ceux-ci : Version du protocole + les flags réservés.

D'autres attributs permettent de les partitionner : adresse source/destination + protocole utilisé. Ces deux propriétés sont généralement utilisées par les, firewalls pour filtrer les paquets. Enfin, certains attributs pourraient être différents dans une même partition : taille du header, identificateur, TTL.

Ce sont ceux qui sont généralement utilisés pour détecter une anomalie **[32],** regarder les valeurs de ces attributs pour déterminer une anomalie n'est pas une bonne manière de faire, ce qui pourrait expliquer les faibles résultats **[34]**, détaillés précédemment. Néanmoins, regarder les changements de ces attributs au cours du temps semble être une meilleure manière de faire. Ainsi, on regarde la moyenne de certaines valeurs, le pourcentage d'événements selon la valeur d'un attribut, le pourcentage de paquets ayant telle valeur, Il faut donc s'assurer d'avoir suffisamment de paquets lors de la phase d'apprentissage pour garantir qu'aucun comportement normal non présent dans ces paquets ne soit oublié. Néanmoins, beaucoup d'attaques connues surpassent cette méthode :

- Utilisation d'un proxy qui va mapper les ports non légitimes vers des ports légitimes pour que les IDS le considèrent comme un paquet légitime.
- Modification d'un logiciel qui fonctionnera d'une telle manière pour l'attaquant et d'une autre manière pour le reste des personnes du système. Après avoir obtenu toutes les informations, il faut classifier les paquets selon le serveur à qui ils appartiennent. Pour cela, il existe deux manières de faire. La première méthode est d'utiliser un apprentissage supervisé pour pouvoir classifier les différents services proposés grâce à des paquets connus par le système, et par la suite définir à quel service appartient tel nouveau flux de paquets.

Ceci peut être fait avec un arbre de décision. La deuxième méthode est d'utiliser la détection d'anomalies où on définit le comportement normal de chaque service. Ainsi, avec un nouveau flux de paquets, on détermine si celui-ci est conforme au comportement normal du service. Ceci est très semblable à la méthode qui regarde le comportement normal d'un utilisateur. Pour connaître le service utilisé à tel instant, on peut construire un arbre de décision qui apprendra les attributs des paquets, par exemple: le nombre de paquets avec le flag FIN selon tel service, et qui classifiera les nouveaux paquets du réseau. Pour détecter une attaque, on regarde vers où l'arbre de décision nous amène. Si par exemple celui-ci nous dit que c'est un service ftp et qu'on voit que le port utilisé est 80, alors on peut se demander si un nouveau service non autorisé est présent dans le système.

III.5 Optimisation

Il y a une forte volonté de réduire le temps d'apprentissage pour pouvoir mettre en œuvre une solution commerciale. Propose une manière de réduire le temps d'apprentissage ainsi que la taille de la structure. Pour être encore plus performant, on peut définir la structure de l'ensemble des réseaux neuronaux utilisés dans l'IDS. En voici quelques-unes :

- Une première boîte contenant un filtre d'information suivie par une autre contenant le réseau neuronal.
- Une première boîte contenant un filtre d'information suivie par n boites de réseaux neuronaux suivies par une dernière qui va jouer le rôle d'arbitre en déterminant ce qui se passe sur le réseau selon les informations reçues par les n boîtes de réseaux neuronaux.

  Chacune d'entre elles reconnaît un type d'attaque (DOS, U2R, R2L, scan,. . .).

  Ceci est aussi appelé la méthode Boo Sting **[35].**
- Trois premières boîtes suivies d'une dernière. L'entraînement se fait ainsi :
  - On entraîne la première boîte avec un certain nombre d'informations.
  - On prend au hasard des nouvelles informations et on entraîne une deuxième.
  - On prend des nouvelles informations et on regarde la réaction des deux premières. Si ces deux boîtes ne convergent pas vers la même idée, on prend cette information et on la met comme entraînement pour la troisième.

Conclusion

Dans ce chapitre nous avons passé en revue les principaux algorithmes de l'apprentissage automatique qui tend à résoudre des problèmes complexes à partir d'exemples en conjuguant les statistiques et l'intelligence artificielle. Nous avons aussi présenté des techniques de détection d'anomalies qui sont basées sur l'apprentissage supervisé et non-supervisé, ce domaine est en perpétuelle évolution en effet de nombreux travaux et recherches sont actuellement entrepris afin d'accroitre les performances des systèmes de détection d'anomalies. Au chapitre suivant, nous présentons notre contribution via ce mémoire de master, et qui consiste à la sélection des bons classifieurs, bien appropriés aux données de la base KDD.

Chapitre 3 : La base de données NSL-KDD & les techniques de classification

Introduction

Notre application se base sur les algorithmes d'apprentissage supervisé pour faire une classification supervisée des connexions de la base NSL-KDD afin d'établir la détection d'intrusions basé sur l'analyse du comportement de ces connexions et permet de les classifier en deux types (anomalie et normale).

Pour cela nous avons montré dans ce chapitre premièrement la description de la base de données NSL-KDD, ensuite nous présentons une étude conceptuelle concernant les techniques de classification en détection d'intrusions et dont le but de comparer classifieur afin d'adapté à la détection d'intrusion, enfin nous présentons les mesures de performance et le protocole expérimentale.

I. La Base de données NSL-KDD

I.1. Description de la base NSL-KDD

La base NSL-KDD (**N**etwork **S**ecurity **L**ayer-**K**nowledge **D**iscovery in **D**atabases) a été fondé sur l'ensemble de données KDD99, Cette dernière 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. KDD99 été réalisées en 1998 par l'agence de L'armé américain DARPA (Défense Advanced Research Projets 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. La base NSL-KDD est un ensemble de données qui représente une version réduite de l'originale KDD 99, proposé en 2010 par les chercheurs dans le domaine de détection d'intrusions réseaux afin de résoudre certains problèmes qui ont apparu dans la base KDD 99. NSL-KDD considérée comme un ensemble de données de référence pour aider les chercheurs à comparer les différentes méthodes de détection d'intrusions. Le NSL-KDD présente les différences suivantes par rapport à l'originale KDD 99 :

- Il n'inclut pas les enregistrements redondants dans les données d'apprentissage, ce qui améliore la performance de classification.
- Il n'y a pas d'enregistrements en double dans les ensembles de test proposés, ce qui aide à l'obtention de meilleurs taux de détection.

- Le nombre d'enregistrements dans les données d'apprentissage et les ensembles d'essais sont raisonnables, ce qui il est abordable d'exécuter les expériences sur l'ensemble complet sans la nécessité de choisissez au hasard une petite portion. Par conséquent, les résultats d'évaluation des travaux de recherche seront cohérents et comparables.

Enregistrement est constitué de 41 attributs caractérisant la connexion, et un attribut représente 5 classes qui sont: normales et 4 types d'attaques. Les principaux types d'attaques de l'ensemble de données NSL-KDD sont (Probing, Dénis de Services, User to Root, Remote to User), Ces attaques ont été abordés en détail dans le premier chapitre de ce mémoire **[36] [37].**

Classes	**Normal**	**Probe**	**Dos**	**R2L**	**U2L**	**Total**
Nombre	97278	4107	391458	1126	52	494021
Pourcentage	19.69%	0.8313%	79.24%	0.2279%	0.0105%	100%

Tableau 3.1. Répartition des attaques dans l'ensemble d'apprentissage KDD99.

Classes	**Normal**	**Probe**	**Dos**	**R2L**	**U2L**	**Total**
Nombre	60593	4166	229853	16189	228	311029
Pourcentage	19.48%	1.34%	73.90%	5.20%	0.0733%	100%

Tableau 3.2. Répartition des attaques dans l'ensemble de Test KDD99.

I.2 Le contenue de l'ensemble de données NSL-KDD

KDDTrain+.ARFF : Ensemble complet NSL-KDD avec étiquettes binaires en format ARFF.

KDDTrain + .TXT : Ensemble complet de trains NSL-KDD incluant les étiquettes d'attaque et le niveau de difficulté au format CSV.

KDDTrain+20Percent._ARFF : Un sous-ensemble de 20% du fichier KDDTrain + .arff.

KDDTrain +_20Percent.TXT: Un sous-ensemble de 20% du fichier KDDTrain + .txt.

KDDTest + .ARFF : Le test complet NSL-KDD avec des étiquettes binaires au format ARFF.

KDDTest + .TXT : Ensemble de test complet NSL-KDD incluant les étiquettes d'attaque et le niveau de difficulté au format CSV.

KDDTest-21.ARFF : Un sous-ensemble du fichier KDDTest + .arff qui n'inclut pas les enregistrements avec un niveau de difficulté de 21 sur 21.

KDDTest-21.TXT: Sous-ensemble du fichier KDDTest+ .txt qui n'inclut pas les enregistrements ayant un niveau de difficulté de 21 sur 21.

I.3 Distribution des connexions réseau de NSL KDDTest+, et NSL- _20%

Catégorie	Nombre d'enregistrement en KDDTrain_20%		Nombre d'enregistrement en KDDTest+	
Normal	13499	53.39%	9711	43.08%
Dos	9234	36.65%	7458	33.08%
Probe	2289	9.09%	2421	10.74%
R2L	209	0.83%	2754	12.22%
U2R	11	0.04%	200	0.88%
Nombre total des enregistrements	25192		22544	

Tableau 3.3. Distribution des connexions réseau de NSL KDDTest+, et NSL KDDTrain_20%.

I.4 Attributs de la base NSL-KDD

Le tableau suivent représenté les 41 attributs de la base NSL-KDD et leurs types de données. Ces attributs peuvent être classés en trois groupes :

Attribut	Désignation	Description
A1	la durée	Longueur (nombre de secondes) de la connexion
A2	type de protocole	Type du protocole, par exemple tcp, udp, etc...
A3	Service	Service réseau sur la destination, par exemple http, telnet, etc...
A4	src_bytes	Nombre d'octets de donnes de la source à la destination
A5	dst_bayes	Nombre d'octets de donnes de la destination à la source
A6	Drapeau	Etat normal ou d'erreur de la connexion
A7	Laterre	1si la connexion est de /vers le même hôte/port ; 0 sinon
A8	incorrect_Fraagment	Nombre de «mauvais »fragments

A9	Urgent	Nombre de colis urgents
A10	Chaud	Nombre d'indicateurs 'chauds'
A11	num_failed_logins	Nombre tentatives de connexion infructueuses
A12	log_in	1si connecté avec succès sinon 0
A13	num_compromised	Nombre de conditions 'compromises'
A14	root_shell	1 si la racine est obtenue 0 sinon
A15	su_attentée	1 si la commande ''su root'' à été tentée ; 0 sinon
A16	num_root	Nombre d'accès ''root ''
A17	num_files_creations	Nombre d'opération de création de fichier
A18	num_shells	Nombre d'invites de Shell
A19	num_acces_files	Nombre d'opération sur les fichiers de contrôles d'accès
A20	num_outbound_cmd	Nombre de commandes sortants dans une session ftp
A21	is_hot_login	1si login appartient à la liste ''chaude'' 0 sinon
A22	is_guest_login	1 si login est un ''invité'' ; 0 sinon
A23	Compter	Nombre de connexions au même hôte que la connexion actuelles au cours des deux derniers secondes
A24	serror_rate	% de connexion comportant des erreurs ''SYN''
A25	reeror_rate	% de connexion comportant des erreurs ''REJ''
A26	same_srv_rate	% de connexions au même service
A27	diff_srv_rate	% de connexions à différents service
A28	srv_count	Nombre de connexion au même service que la connexion actuelle au cours de deux dernières secondes
A29	srv_serror_rate	% de connexion comportant des erreurs ''SYN''
A30	srv_serror_rate	% de connexion comportant des erreurs ''REJ''
A31	srv_diff_host_rate	%de connexions à différents hôtes
A32	dst_host_count	Nombre de connexion pour le même hôte
A33	dst_host_svr_count	Nombre de connexion pour le même hôte utilisent le même service
A34	dst_host_same_srv_rate	% de connexion pour le même hôte utilisent le même service

A35	dst_host_diff_srv_rate	% de connexion pour le même hôte utilisent le différent service
A36	dst_host_same_src_port_rate	% de connexions pour le même hôte ayant port src
A37	dst_host_srv_diff_host_rate	% de connexions pour le même hôte et le même service utilisent différents hôtes
A38	dst_host_serror_rate	% de connexions pour le même hôte ayant l'erreur ''SYN''
A39	dst_host_srv_serror_rate	% de connexions pour le même hôte et le meme service ayant l'erreur ''SYN''
A40	dst_host_rerror_rate	% % de connexions pour le même hôte ayant l'erreur ''REJ''
A41	dst_host_srv_rerror_rate	% de connexions pour le même hôte et le meme service ayant l'erreur ''REJ''

Tableau 3.4. Les 41 attrubites de la base de données NSL KDD.

Ces attributs peuvent être classés en trois groupes: du trafic tels que nombre d'échec de connexion et le nombre d'accès aux fichiers :

- **Les attributs de base:** ces attributs décrivent les informations de base d'une connexion, telles que la durée, les hôtes source et destination, port et flag.
- **Les attributs du trafic:** ces attributs sont basés sur des statistiques, tels que le nombre de connexions vers la même machine.
- **Les caractéristiques du contenu:** ces attributs sont construits à partir de la charge utile (Data) des paquets de contrôle.

II. Les techniques de classifications

Le choix de l'approche supervisée est motivé par le fait que les donnes d'intrusion de la base de données KDD sont étiquetées. C'est-à-dire pour chaque entrée de la base, l'attaque est bien spécifiée sous forme d'une classe (DOS, Neptune….). Nous avons pris en compte les trois algorithmes de classification, à savoir l'algorithme Knn, naïve bayes et Random forest.

II.1 K Nearest Neighbor (KNN)

II.1.1 Définition

K Nearest **N**eighbor (**PPV P**lus **P**roches **V**oisins) L'algorithme K-plus proches voisins (KNN) est un type d'algorithme de l'apprentissage automatique supervisé qui peut être utilisé à la fois pour la classification et les problèmes prédictifs de régression. Cependant, il est principalement utilisé pour les problèmes prédictifs classification dans l'industrie. Les deux propriétés suivantes définiraient bien KNN :

- **Algorithme d'apprentissage paresseux:** est un algorithme d'apprentissage paresseux car il n'a pas de phase de formation spécialisée et utilise toutes les données pour la formation lors de la classification.
- **Algorithme d'apprentissage non paramétrique:** est également un algorithme d'apprentissage non paramétrique car il ne suppose rien sur les données sous-jacentes.

II.1.2 Principe de fonctionnement

L'algorithme K-plus proches voisins (KNN) utilise la "similarité des caractéristiques" pour prédire les valeurs des nouveaux points de données, ce qui signifie en outre que le nouveau point de données se verra attribuer une valeur en fonction de sa correspondance avec les points de l'ensemble d'apprentissage. Nous pouvons comprendre son fonctionnement à l'aide des étapes suivantes :

- **Étape 1 :** Pour implémenter n'importe quel algorithme, nous avons besoin d'un ensemble de données. Ainsi, lors de la première étape de KNN, nous devons charger la formation ainsi que les données de test.
- **Étape 2 :** Ensuite, nous devons choisir la valeur de K, c'est-à-dire les points de données les plus proches. K peut être n'importe quel nombre entier.
- **Étape 3 :**
  - En fonction de la valeur de distance, triez-les par ordre croissant.
  - Ensuite, il choisira les K premières lignes du tableau trié.
  - Maintenant, il attribuera une classe au point de test en fonction de la classe la plus fréquente de ces lignes.
- **Étape 4 :** Fin.

Un exemple pour comprendre le concept de K et le fonctionnement de l'algorithme KNN. Supposons que nous ayons un ensemble de données qui peut être tracé comme suit :

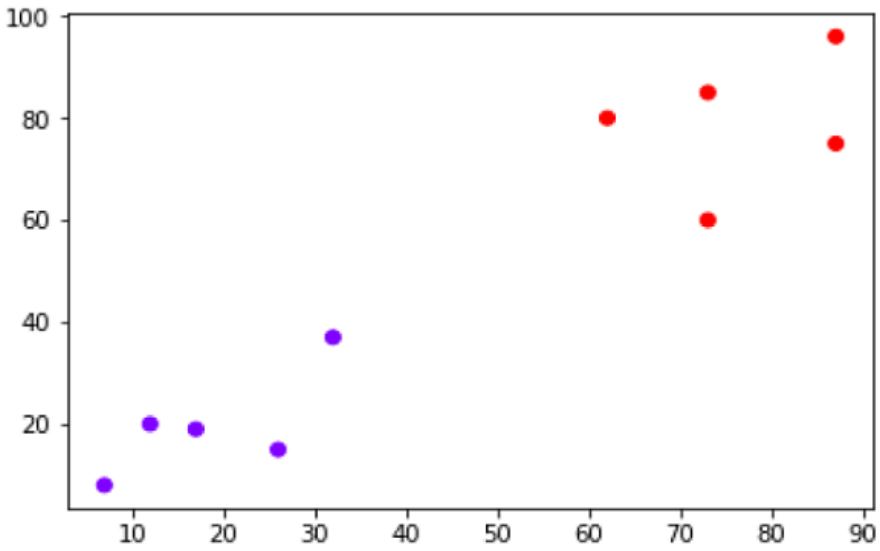

Figure 3.1. Exemple de classification Knn **[41].**

Maintenant, nous devons classer le nouveau point de données avec un point noir (au point 60,60) en classe bleue ou rouge. Nous supposons que K = 3, c'est-à-dire qu'il trouverait les trois points de données les plus proches.

Il est montré dans le schéma suivant :

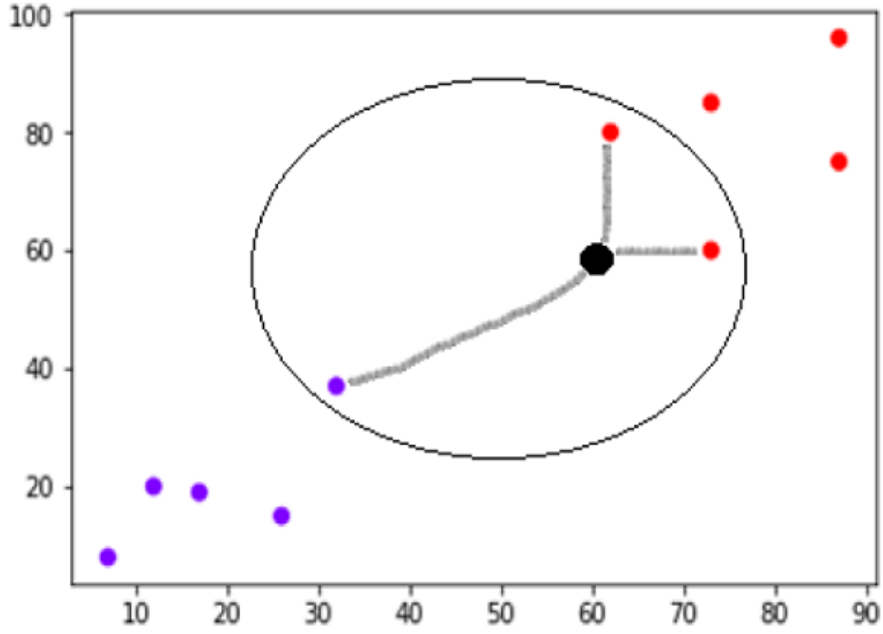

Figure 3.2. Exemple de classification Knn **[41].**

Nous pouvons voir dans le diagramme ci-dessus les trois voisins les plus proches du point de données avec un point noir. Parmi ces trois, deux d'entre eux se situent dans la classe rouge, d'où le point noir sera également attribué à la classe rouge.

II.1.3 Les avantages de l'algorithme KNN

- C'est un algorithme très simple à comprendre et à interpréter.
- Il est très utile pour les données non linéaires car il n'y a aucune hypothèse sur les données dans cet algorithme.
- C'est un algorithme polyvalent car nous pouvons l'utiliser pour la classification ainsi que pour la régression.
- Il a une précision relativement élevée, mais il existe de bien meilleurs modèles d'apprentissage supervisé que KNN.

II.1.4 Les inconvénients de l'algorithme KNN

- C'est un algorithme un peu coûteux en calcul car il stocke toutes les données d'entraînement.
- Stockage de mémoire élevé requis par rapport aux autres algorithmes d'apprentissage supervisé.
- La prédiction est lente en cas de grand N.

II.1.5 Domaines d'applications de Knn

Voici quelques-uns des domaines dans lesquels KNN peut être appliqué avec succès

- **Système bancaire :** KNN peut être utilisé dans le système bancaire pour prédire si un individu est apte à l'approbation d'un prêt ? Cet individu a-t-il des caractéristiques similaires à celles des défaillants.
- **Calcul des cotes de crédit:** Les algorithmes KNN peuvent être utilisés pour trouver la cote de crédit d'un individu en comparant avec les personnes ayant des traits similaires.
- **Politique :** Avec l'aide d'algorithmes KNN, nous pouvons classer un électeur potentiel en différentes classes comme "Va voter", "Ne votera pas", "Va voter pour le parti Congrès", "Va voter pour le parti BJP".

D'autres domaines dans lesquels l'algorithme KNN peut être utilisé sont la reconnaissance vocale, la détection d'écriture manuscrite, la reconnaissance d'image et la reconnaissance vidéo.

II.2 Naïve Bayes

II.2.1 Définition

Il s'agit d'une technique de classification basée sur le théorème de Bayes avec une hypothèse d'indépendance entre les prédicteurs. En termes simples, un classificateur Naïve Bayes suppose que la présence d'une caractéristique particulière dans une classe n'est pas liée à la présence de toute autre caractéristique.

Par exemple, un fruit peut être considéré comme une pomme s'il est rouge, rond et d'environ 3 pouces de diamètre. Même si ces caractéristiques dépendent les unes des autres ou de l'existence d'autres caractéristiques, toutes ces propriétés contribuent indépendamment à la probabilité que ce fruit soit une pomme et c'est pourquoi on l'appelle "Naïf".

Le modèle Naïve Bayes est facile à construire et particulièrement utile pour les très grands ensembles de données. En plus de sa simplicité, Naïve Bayes est connu pour surpasser même les méthodes de classification les plus sophistiquées.

Le théorème de Bayes fournit un moyen de calculer la probabilité a posteriori P (c|x) à partir de P(c), P(x) et P (x|c).

Regardez l'équation ci-dessous :

$$\boldsymbol{P(c|x) = \frac{P(x|c)\, P(c)}{P(x)}}$$

$$\boldsymbol{P(c|X) = P(x_1|c) \times P(x_2|c) \times \ldots\ldots \times P(\mathbf{x}_n \,|\, c)}$$

- **P (c|x) :** est la probabilité a posteriori de la classe (c, cible) compte tenu du prédicateur (x, attributs).
- **P(c) :** est la probabilité a priori de la classe.
- **P (x|c) :** est la vraisemblance qui est la probabilité du prédicateur pour une classe donnée.
- **P(x) : e**st la probabilité a priori du prédicateur.

II.2.2 Principe de fonctionnement

Comprenons-le à l'aide d'un exemple. Ci-dessous, j'ai un ensemble de données d'entraînement sur la météo et la variable cible correspondante "Jouer" (suggérant des

possibilités de jouer). Maintenant, nous devons classer si les joueurs joueront ou non en fonction des conditions météorologiques. Suivons les étapes ci-dessous pour l'exécuter :

- **Étape 1 :** Convertir l'ensemble de données en un tableau de fréquences
- **Étape 2 :** Créez une table de probabilité en trouvant les probabilités comme la probabilité de **Overcast** = 0,29 et la probabilité de **Play** est de 0,64.

Weather	**Play**
Sunny	NO
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	**No**	**Yes**
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	**No**	**Yes**		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Figure 3.3. Exemple de fonctionnement de l'algorithme Naïve Bayes **[41].**

- **Étape 3 :** Maintenant, utilisez l'équation bayésienne naïve pour calculer la probabilité a posteriori pour chaque classe. La classe avec la probabilité a posteriori la plus élevée est le résultat de la prédiction.
    - **Problème :** Les joueurs joueront (play) si le temps est Sunny. Cette affirmation est-elle correcte ?

Nous pouvons le résoudre en utilisant la méthode de probabilité postérieure discutée ci-dessus.

P (yes | Sunny) = P (Sunny | yes) * P (yes) / P (yes)

Ici nous avons **P (Sunny| yes) = 3/9 = 0,33, P(Sunny) = 5/14 = 0,36, P (yes)= 9/14 = 0,64**

Maintenant, **P (yes | Sunny) = 0,33 * 0,64 / 0,36 = 0,60,** ce qui a une probabilité plus élevée.

Naïve Bayes utilise une méthode similaire pour prédire la probabilité de différentes classes en fonction de divers attributs. Cet algorithme est principalement utilisé dans la classification de texte et avec des problèmes ayant plusieurs classes.

II.2.3 Les Avantages de l'algorithme Naïve Bayes

- Il est facile et rapide de prédire la classe de l'ensemble de données de test.
- Il fonctionne également bien dans la prédiction multi-classes.
- Lorsque l'hypothèse d'indépendance est valable, un classificateur Naïve Bayes est plus performant que d'autres modèles comme la régression logistique et vous avez besoin de moins de données d'entraînement.
- Il fonctionne bien dans le cas de variables d'entrée catégorielles par rapport aux variables numériques.
- Pour la variable numérique, une distribution normale est supposée (courbe en cloche, qui est une hypothèse forte).

II.2.4 Les inconvénients de l'algorithme Naïve Bayes

- Si la variable catégorielle a une catégorie (dans l'ensemble de données de test), qui n'a pas été observée dans l'ensemble de données d'apprentissage, le modèle attribuera une probabilité de 0 (zéro) et ne pourra pas faire de prédiction. Ceci est souvent connu sous le nom de « fréquence zéro ».
- Pour résoudre ce problème, nous pouvons utiliser la technique de lissage.
- L'estimation de Laplace est l'une des techniques de lissage les plus simples.
- De l'autre côté, Bayes naïf est également connu comme un mauvais estimateur, donc les sorties de probabilité de predict_proba.

II.2.5 Domaines d'applications des algorithmes Naïve Bayes

- **Prédiction en temps réel :** Naïve Bayes est un classificateur avide d'apprentissage et il est certainement rapide. Ainsi, il pourrait être utilisé pour faire des prédictions en temps réel.
- **Prédiction multi-classes :** cet algorithme est également bien connu pour la fonction de prédiction multi-classes. Ici, nous pouvons prédire la probabilité de plusieurs classes de variable cible.
- **Classification de texte/filtrage de spam/analyse des sentiments :** les classificateurs Naïve Bayes principalement utilisés dans la classification de texte (en raison de meilleurs résultats dans les problèmes multi-classes et de la règle d'indépendance) ont un taux de réussite plus élevé par rapport aux autres algorithmes.
- En conséquence, il est largement utilisé dans le filtrage anti-spam (identifier les spams) et l'analyse des sentiments (dans l'analyse des médias sociaux, pour identifier les sentiments positifs et négatifs des clients).
- **Système de recommandation :** Naïve Bayes Classifier et Collaborative Filtering créent ensemble un système de recommandation qui utilise des techniques d'apprentissage automatique et d'exploration de données pour filtrer les informations invisibles et prédire si un utilisateur aimerait ou non une ressource donnée**.**

II.3 L'algorithme Random forest

II.3.1 Définition

Random forest (La forêt aléatoire) est un algorithme d'apprentissage supervisé qui est utilisé à la fois pour la classification et la régression. Mais cependant, il est principalement utilisé pour des problèmes de classification. Comme nous le savons, une forêt est composée d'arbres et plus d'arbres signifie une forêt plus robuste. De même, l'algorithme de forêt aléatoire crée des arbres de décision sur des échantillons de données, puis obtient la prédiction de chacun d'eux et sélectionne finalement la meilleure solution par vote. C'est une méthode d'ensemble qui est meilleure qu'un arbre de décision unique car elle réduit le sur-ajustement en faisant la moyenne du résultat.

II.3.2 Principe de fonctionnement

Nous pouvons comprendre le fonctionnement de l'algorithme Random Forest à l'aide des étapes suivantes :

- **Étape 1 :** Tout d'abord, commencez par la sélection d'échantillons aléatoires à partir d'un ensemble de données donné.
- **Étape 2 :** Ensuite, cet algorithme construira un arbre de décision pour chaque échantillon. Ensuite, il obtiendra le résultat de prédiction de chaque arbre de décision.
- **Étape 3 :** Dans cette étape, le vote sera effectué pour chaque résultat prédit.
- **Étape 4 :** Enfin, sélectionnez le résultat de prédiction le plus voté comme résultat de prédiction final.

Le schéma suivant illustrera son fonctionnement :

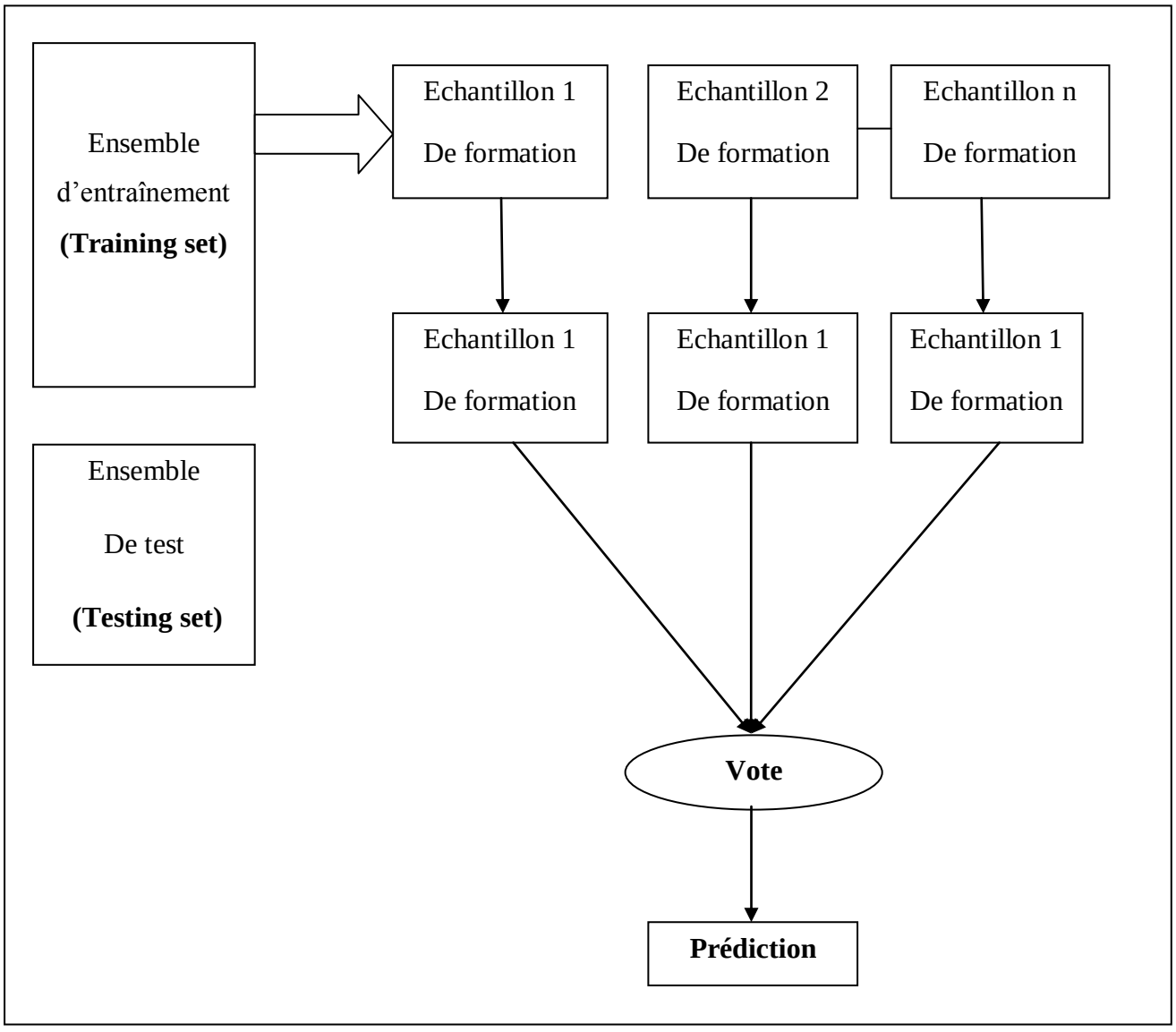

Figure 3.4. Schéma de fonctionnement de l'algorithme Random Forest.

II.3.3 Les avantages de l'algorithme Random forest

- Il surmonte le problème de sur ajustement en faisant la moyenne ou en combinant les résultats de différents arbres de décision.

- Les forêts aléatoires fonctionnent mieux pour une large gamme d'éléments de données qu'un seul arbre de décision.
- La forêt aléatoire à moins de variance que l'arbre de décision unique.
- Les forêts aléatoires sont très flexibles et possèdent une très grande précision.
- La mise à l'échelle des données ne nécessite pas d'algorithme de forêt aléatoire. Il maintient une bonne précision même après avoir fourni des données sans mise à l'échelle.
- Les algorithmes Random Forest conservent une bonne précision même si une grande partie des données est manquante.

II.3.4 Les inconvénients de l'algorithme Random forest

- La complexité est le principal inconvénient des algorithmes de forêt aléatoire.
- La construction de forêts aléatoires est beaucoup plus difficile et prend beaucoup de temps que les arbres de décision.
- Le processus de prédiction utilisant des forêts aléatoires prend beaucoup de temps par rapport à d'autres algorithmes.

III. Les mesures de performance

Pour mesurer la qualité de la performance du modèle de détection d'intrusions, le résultat de ce dernier sera comparé avec les données réelles (données marquées). Pour l'ensemble de données NSL-KDD de Test, où toutes les données ont été étiquetées, c'est-à-dire que la classe de chaque instance est connue. Chaque instance est qualifiée de normale ou d'anomalie. Le tableau 4.2, montre les résultats possibles de la nature du résultat du modèle proposé.

III.1 La matrice de confusion

		Classe **détectée (prédite)**	
		Normale	**Attaque**
Classe réelle	**Normale**	Vrai négatif **TN** (True Négative)	Faux Positif **FP** (False Positive)
	Attaque	Faux négatif **FN** (False Negative)	True Positif **TP** (True Positive)

Tableau 3.5. La matrice de confusion.

- **Vrai positif (TP) :** une attaque correctement détectée par le test.
- **Faux positif (FP) :** une activité normale détectée comme attaque par le test.
- **Vrai négatif (TN) :** une activité normale correctement détectée par le test.
- **Faux négatif (FN) :** une attaque détectée comme activité normale par le test.

III.2 Les mesures d'évaluation

- **La précision :** cette métrique, également relative à chaque catégorie, renseigne sur la probabilité qu'une prédiction d'une catégorie donnée soit correcte.

$$\textbf{précision} = \left(\frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}}\right) \times \textbf{100}\%$$

- **Le taux de détection (Rappel / Recall) :** C'est le rapport entre le nombre d'intrusions correctement détectées et le nombre total d'intrusions.
- Et décrit par la formule

$$\textbf{Rappel} = \left(\frac{\textbf{TP}}{\textbf{FN} + \textbf{TP}}\right) \times \textbf{100}\%$$

- **Le taux de faux positif (FP) (Le taux des fausses alertes) :** est calculé comme le rapport entre les nombres de trafic normal qui sont incorrectement classés comme intrusions et le nombre total de trafic normal.

$$\mathbf{FP} = \left(\frac{\mathbf{FP}}{\mathbf{TN+FP}}\right) \times \mathbf{100\%}$$

- **le taux de réussite *(Accuracy)* :** Indique la façon dont la technique de détection est correcte. C'est une métrique qui traduit également le rapport entre les détections correctes et les détections totales obtenues.

$$\mathbf{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{FP+FN+TP+TN}} \times \mathbf{100\%}$$

IV. Protocol expérimental

Comme nous avons indique plus haute, nous expérimentons trois algorithmes de classification supervisée sur les donnes de detection d'intrusion, issues de la base de donnés spécialisée, KDD dans sa version NSL KDD.

Pour cela et comme tous les modèles de classification, l'élaboration de notre modèle respecte des phases suivantes: le chargement des données (NSL KDDTest+, NSL KDDTrain_20%), l'apprentissage, la phase de test et les résultats obtenus pour chaque algorithme.

Comme l'indique le diagramme ci-dessous :

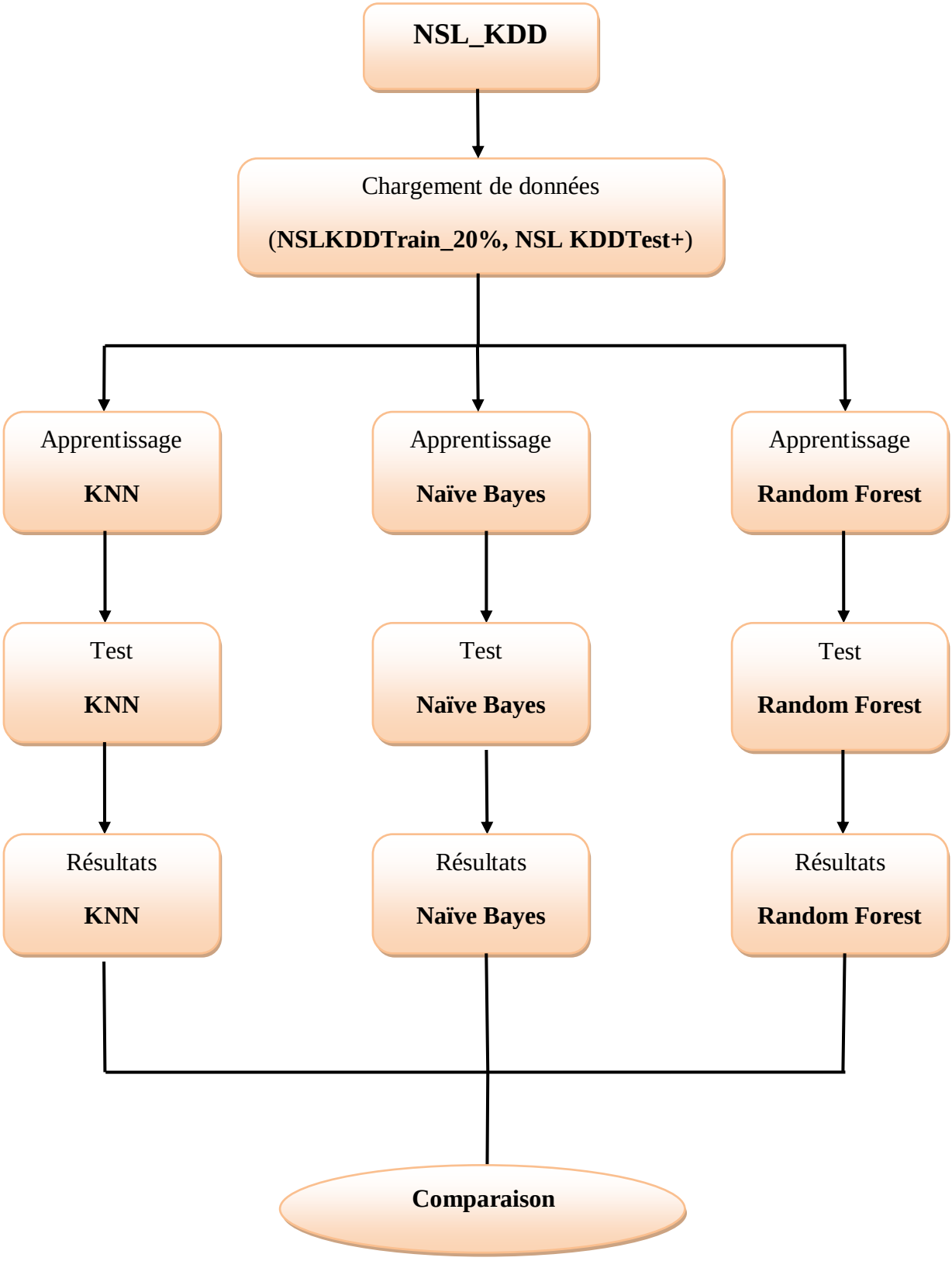

Figure 3.5. Diagramme du Protocol expérimental.

Conclusion

Nous avons abordé dans ce chapitre la base de données NSL-KDD qui a été utilisée pour l'apprentissage et test du modèle de détection d'intrusions. Nous avons considéré trois algorithmes de classifications, à savoir l'algorithme KNN, l'algorithme Naïve Bayes et l'algorithme Random forest. Dans une étude plus détaillée, il serait préférable de tester plus de classifieur afin de décider quel est le classifieur le plus approprié pour les données de la NSL KDD. Au chapitre suivant nous présentons les aspects pratique de l'expérimentation, à savoir la présentation de l'environnement matérielle et logicielle, et nous présentons également quelques résultats de test.

Chapitre 04 : Implémentation et résultats

Introduction

Après avoir présenté au chapitre précédent notre démarche de test des algorithmes de classification pour la base de données NSL-KDD. Nous présentons dans ce chapitre, l'expérimentation et les résultats obtenus. Nous commençons par la présentation de l'environnement d'implémentation à savoir l'environnement weka puis nous passons aux résultats expérimentaux et à leur comparaison.

I. Outils et Langage Utilisée

I.1 Environnement de réalisation

Dans cette partie, on va présenter :

- L'environnement matériel.
- L'environnement logiciel.

I.1.1 L'environnement matériel

Pendant la phase de documentation, de spécification des besoins, de conception et de développement, on a utilisé un PC ayant les caractéristiques suivantes :

- Processeur Intel® Pentium® i3 CPU.
- 4 Gb MB de mémoire vive.
- Disque dur de capacité 500 Go.
- Système d'exploitation Microsoft Windows 8 Professionnel.

I.1.2 L'environnement logiciel

I.1.2.1 NetBeans 8.2

Pour la réalisation de notre application JAVA en utilisant l'environnement de développement NetBeans 8.2 est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Développement and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refacturions, éditeur graphique d'interfaces et de pages web) **[38]**.

I.1.2.2 Le langage Java

Java est un langage de programmation informatique orienté objet, la particularité et l'objectif central de java est que les logiciels écrit dans écangage doivent être très facilement portables sur plusieurs systèmes d'exploitation. Java a donné naissance à un système d'exploitation (Java OS), a des environnements de développements (JDK), des machines virtuelles(MSJVM, JRE) applications multiplateformes (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes(Oracle SQL…) ,des technologies web(applets, servlets) et une déclinaison pour l'entreprise(La J2EE) .Le point essentiel en java c'est la portabilité de ses applications ,elles peuvent être exécuté sur n'importe quelle machine **[39].**

I.1.2.3 le jeu de donnée Nsl Kdd : (déjà définir en détail dans le troisième chapitre).

I.1.2.4 L'environnement Weka

Figure 4.1. Logo de l'environnement Weka **[40]**.

I.1.2.4.1 Définitions

Weka (**W**aikato **E**nvironment for **K**nowledge **A**nalysis) est un ensemble d'outils permettant de manipuler et d'analyser des fichiers de données, implémentant la plupart des algorithmes d'intelligence artificielle, entre autres, les arbres de décision et les réseaux de neurones. Les algorithmes peuvent être appliqués directement à un ensemble de données ou appelés à partir d'un code Java. Il est également bien adapté au développement de nouveaux modèles de Machine Learning. Weka est utilisé dans divers domaines comme dans la recherche, l'éducation et même les entreprises **[40].**

Weka est un logiciel open source publié sous licence publique générale GNU. Il a été développé à l'Université de Waikato en Nouvelle-Zélande.

Weka est extensible et est devenu une collection d'algorithmes d'apprentissage automatique pour résoudre les problèmes d'exploration de données dans le monde réel. Il est écrit en Java et fonctionne sur presque toutes les plateformes. Il est facile à utiliser et à appliquer sur plusieurs niveaux différents. Il est possible d'accéder à la bibliothèque de classes Weka à partir d'un programme Java et implémenter de nouveaux algorithmes d'apprentissage automatique.

I.1.2.4.2 Composants de l'environnement Weka

- **Il se compose principalement :**
  - De classes Java permettant de charger et de manipuler les données.
  - De classes pour les principaux algorithmes de classification supervisée ou non supervisée.
  - D'outils de sélection d'attributs, de statistiques sur ces attributs.
  - De classes permettant de visualiser les résultats.
- **On peut l'utiliser à trois niveaux :**
  - Via l'interface graphique, pour charger un fichier de données, lui appliquer un algorithme, vérifier son efficacité.
  - Invoquer un algorithme sur la ligne de commande.
  - Utiliser les classes définies dans ses propres programmes pour créer d'autres méthodes, implémenter d'autres algorithmes, comparer ou combiner plusieurs méthodes.

II. Description de l'application

Notre application n'est pas basée sur l'aspect design et interfaces, mais plutôt sur l'aspect expérimental. Pour cela, nous avons créé une interface simple illustrée dans la figure suivante :

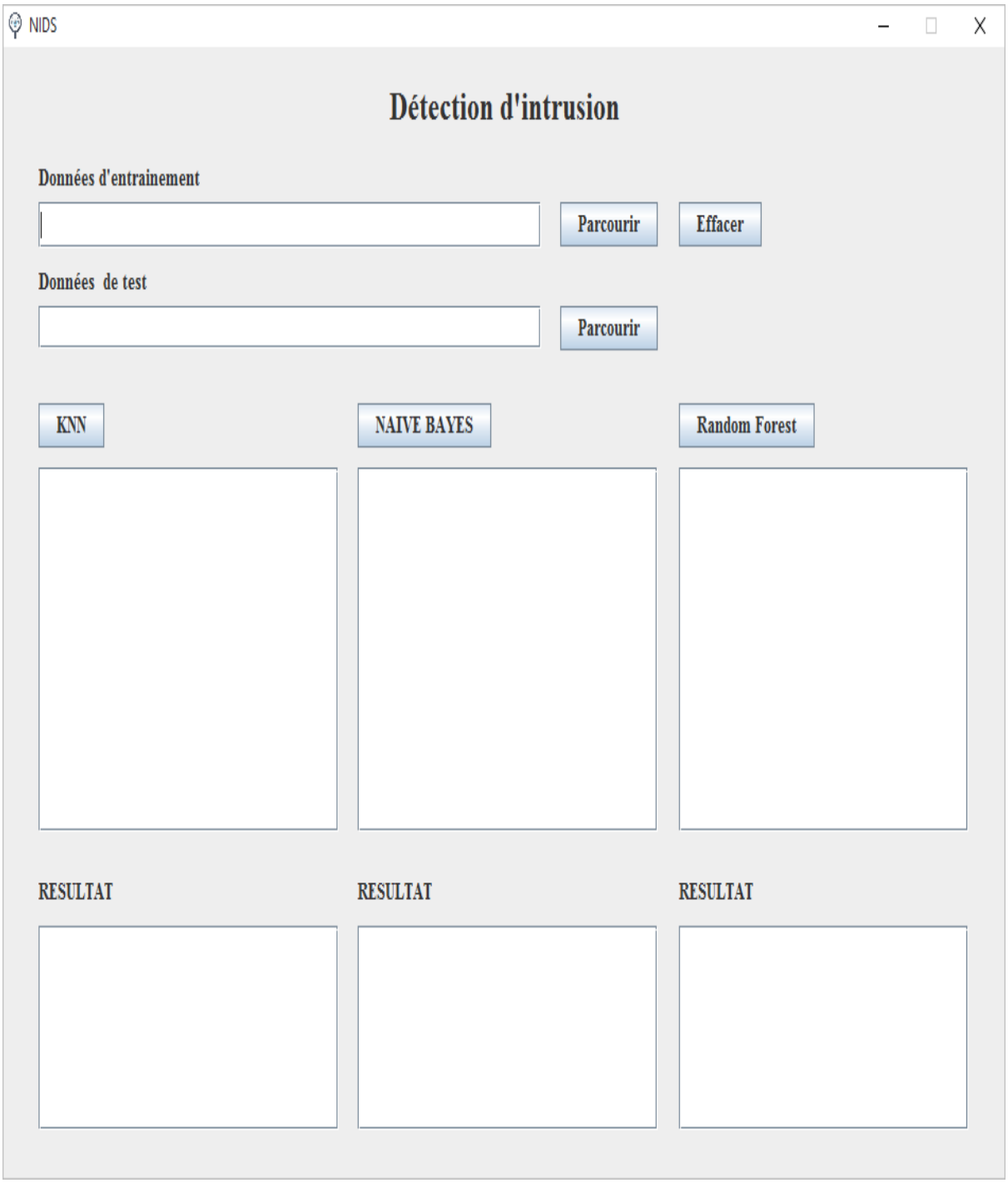

Figure 4.2. L'interface de l'application

III. Expérimentations

Comme nous avons mentionné au chapitre trois, nous avons travaillé avec la version NSL KDD de la base de données d'intrusion KDD. Et nous avons utilisé la plateforme Weka comme bibliothèque en java pour instancier les algorithmes de classification.

III.1 Le chargement de données

III.1.1 Chargement les données d'apprentissage (NSL-KDDTrain_20%)

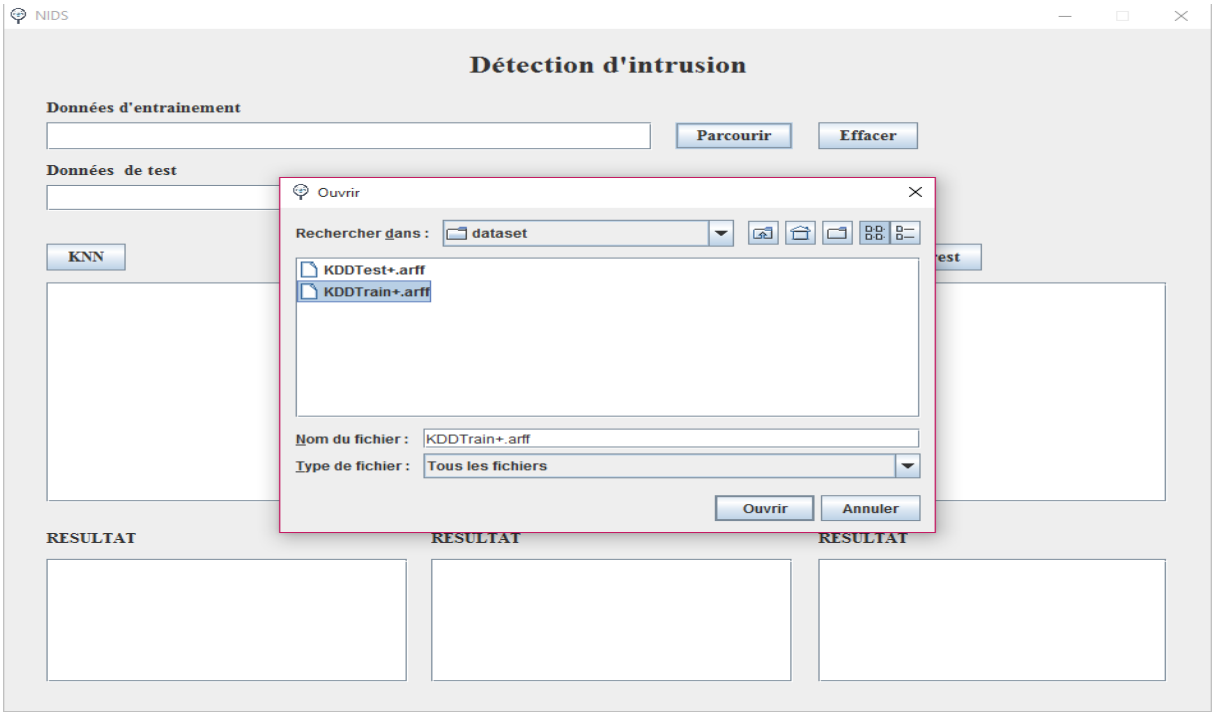

Figure 4.3. Chargement de NSL-KDDTrain_20%.

III.1.2 Chargement les données de test (NSL-KDDTest+)

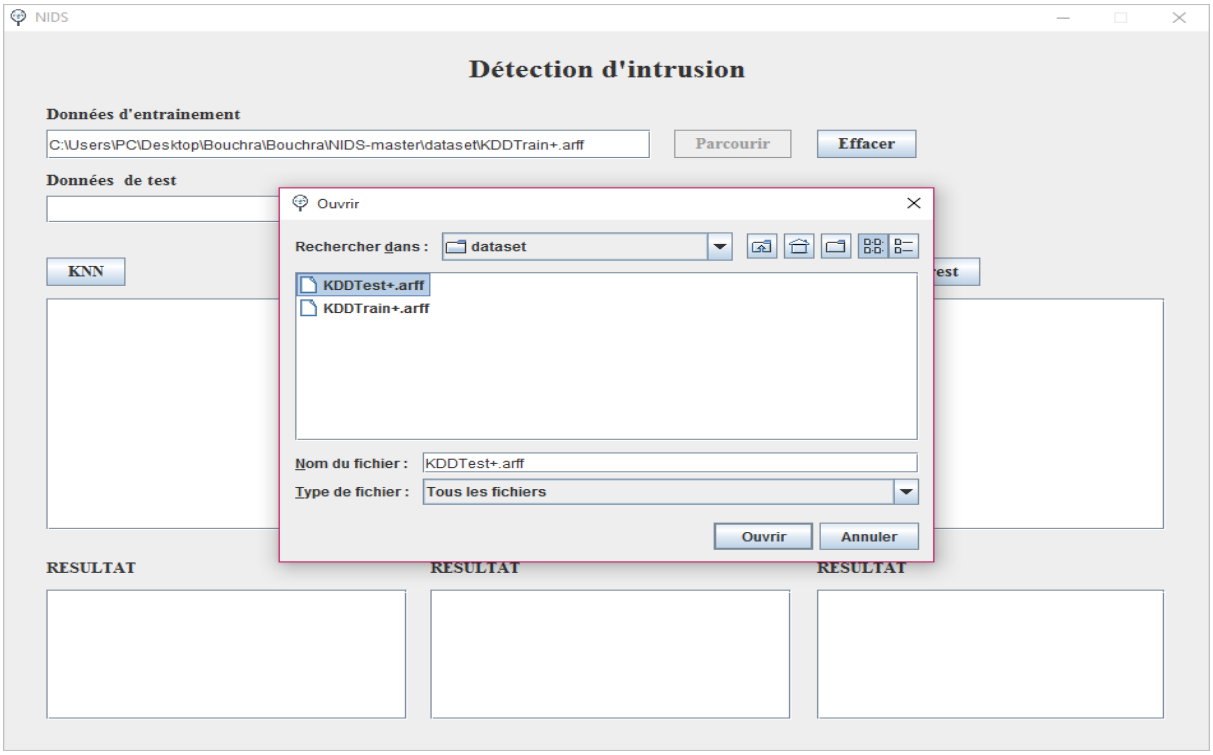

Figure 4.4. Chargement de NSL-KDDTest+.

III.2 Résultats de test avec KNN

Les figures 4.5 et 4.6 montre l'affichage des résultats de classification.

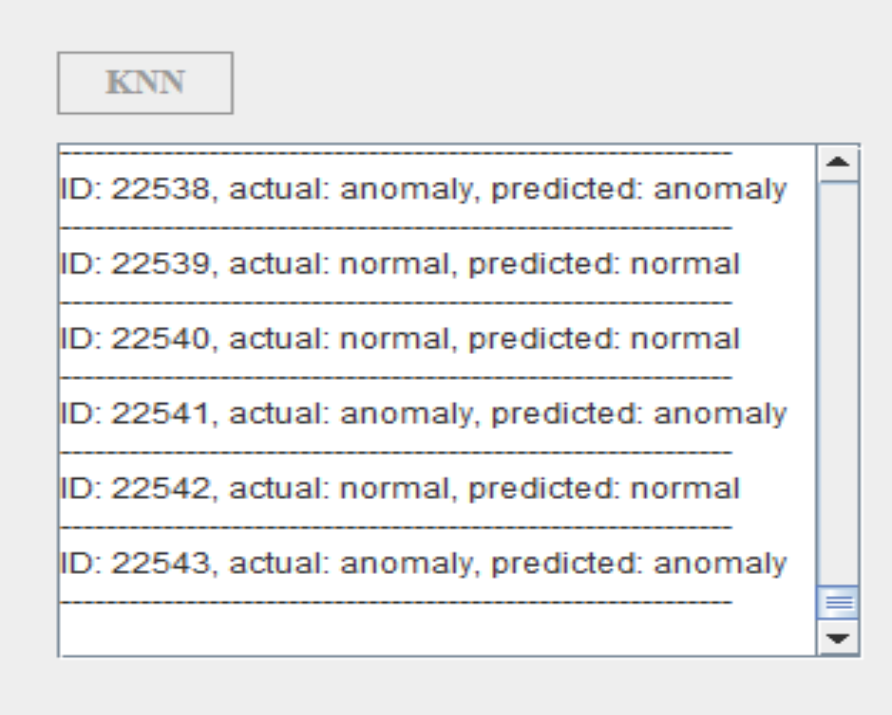

Figure 4.5. Résultats de test avec KNN**.**

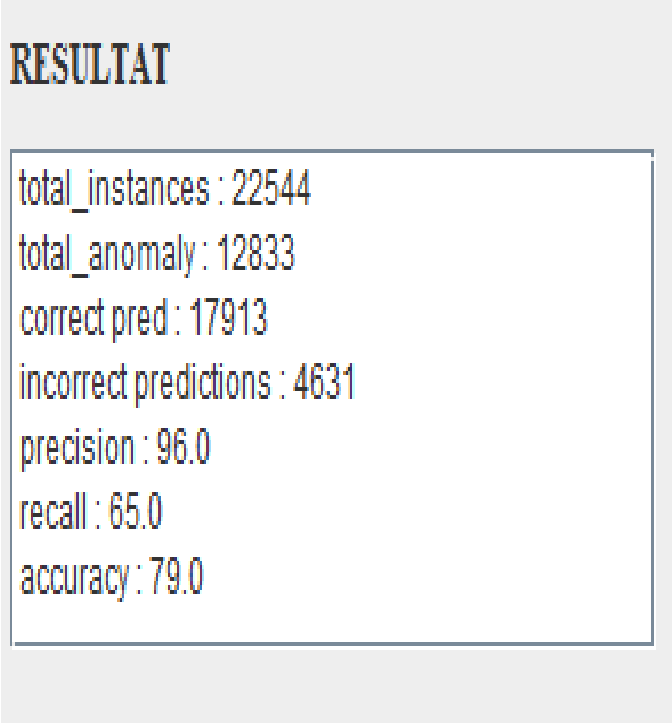

Figure 4.6. Résultats de test avec KNN.

III.3 Résultats de test avec Naïve Bayes

Les figures 4.7 et 4.8 montre l'affichage des résultats de classification

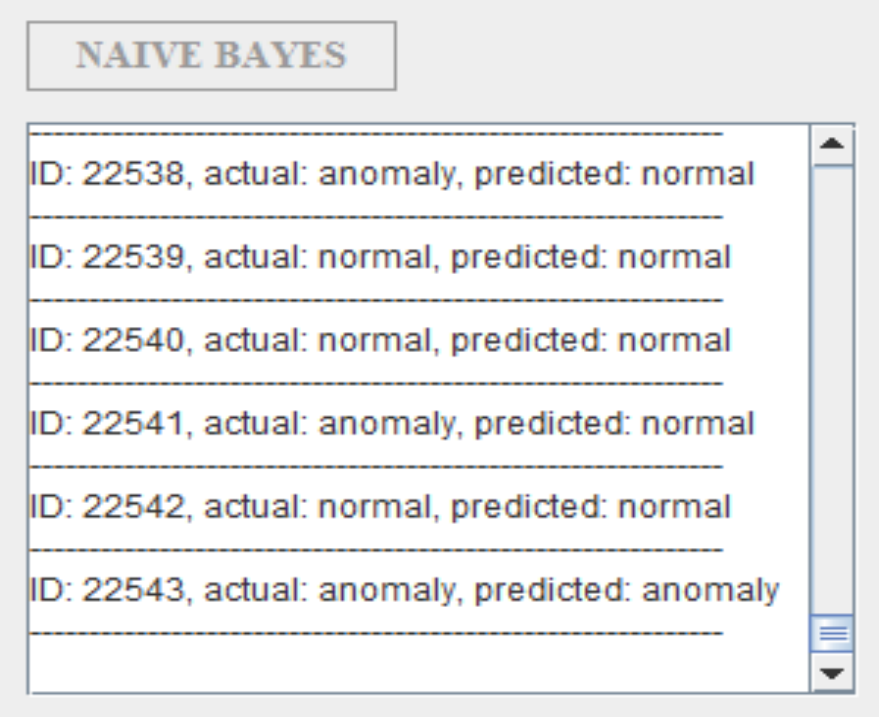

Figure 4.7. Résultats de test avec Naïve Bayes.

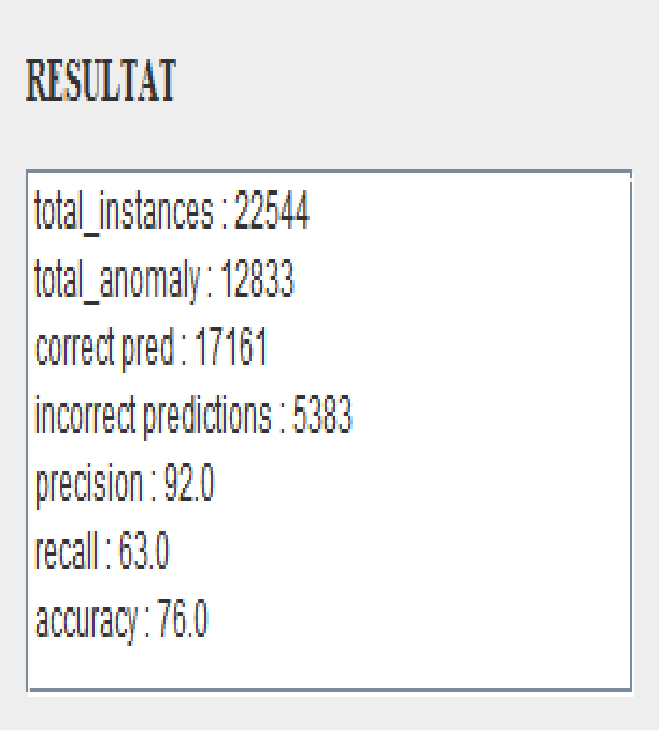

Figure 4.8. Résultats de test avec Naïve Bayes.

III.4 Résultats de test avec Random Forest

Les figures 4.9 et 4.10 montre l'affichage des résultats de classification :

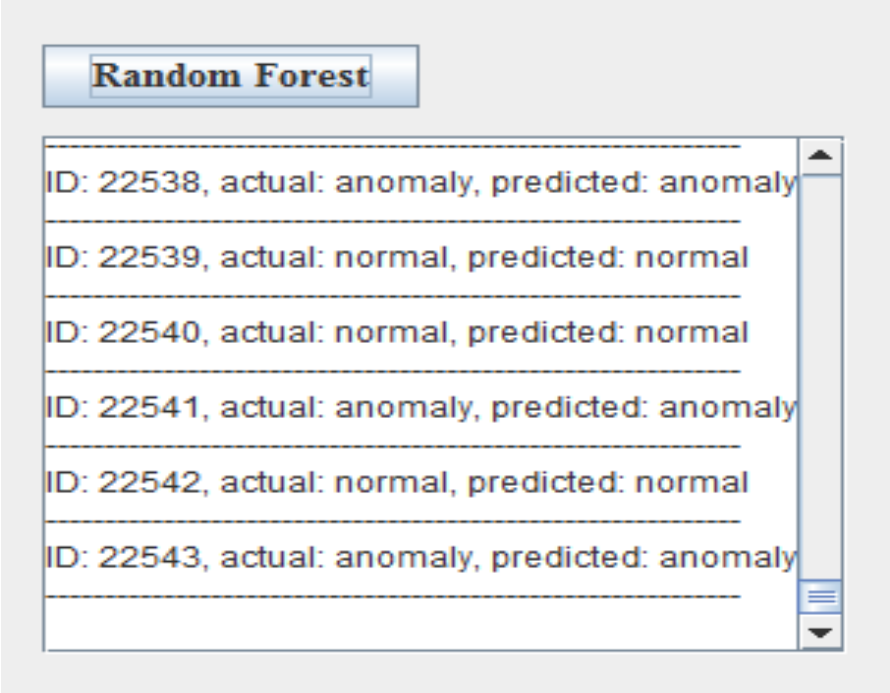

Figure 4.9. Résultats de test avec Random Forest.

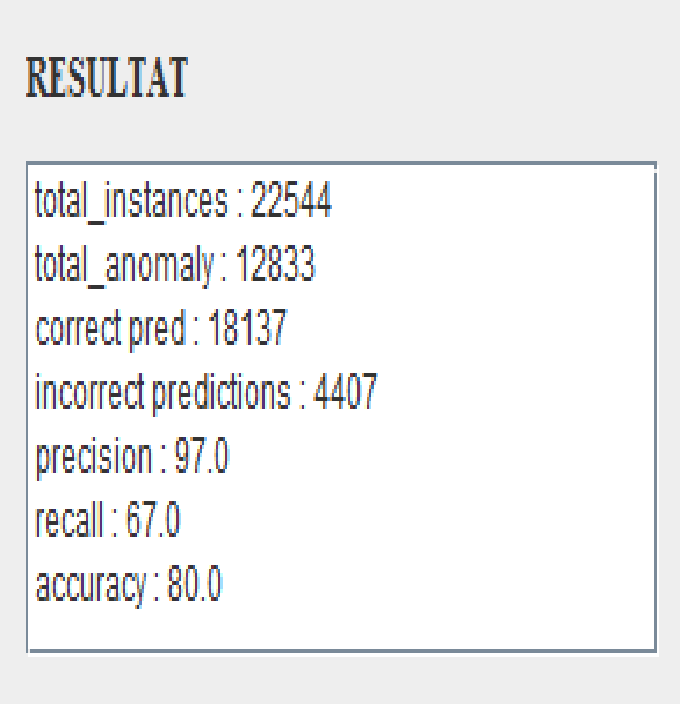

Figure 4.10. Résultats de test avec Random Forest.

IV. Analyse et comparaison des résultats

Le tableau suivant qui montre les résultats des prédictions:

	Totale instances : 22544	
	Correct prédictions	**Incorrect prédictions**
KNN	**17 913**	**4631**
Naïve bayes	**17 161**	**5383**
Random Forest	**18 137**	**4407**

Tableau 4.1. Résultats des prédications pour chaque algorithme.

	Précision	**Recall**	**Accuracy**
KNN	**96%**	**65%**	**79%**
Naïve Bayes	**92%**	**63%**	**76%**
Random Forest	**97%**	**67%**	**80%**

Tableau 4.2. Résultats de performance selon le test pour chaque algorithme**.**

Les résultats présentés ci-dessus ont montré que les performances des trois algorithmes de classification sur lesquels nous avons mené l'étude expérimentale sont acceptables, ils détectent les anomalies, avec un taux de réussite de plus de 75%, mais les résultats de performance de l'algorithme de forêt aléatoire étaient meilleurs par rapport à l'algorithme KNN et Naïve bayes.

Conclusion

Dans ce dernier chapitre, nous avons présenté la partie expérimentale de notre travail, qui consiste à tester les données d'intrusions de la base de données NSL-KDD.

Nous avons testé les trois algorithmes de classifications KNN, Naïve Bayes et Random Forest en présentant différents métriques de performances dont le taux de réussite et le taux de précision.

L'algorithme de classification Random forest peut obtenir un taux de précision élevé lors de la comparaison avec d'autres algorithmes de Classification, l'instance correctement classée de ce l'algorithme est 80 % à partir de ce taux de précision nous pouvons décider que l'algorithme de classification Random Forest est le meilleur algorithme de classification des jeux de données NSL-KDD.

Conclusion générale

Les attaques informatiques ont fortement augmenté ces dernières années et représentent aujourd'hui une véritable menace pour les réseaux informatiques, les applications et les systèmes d'information des entreprises.

Les attaques nous ont incités à tenter dans ce travail de développer un modèle de sécurité capable de détecter toute tentative malveillante, connue . Pour atteindre cet objectif, nous avons mené une étude comparative de différents types d'algorithmes de classification, afin de classer les connexions en deux catégories : normales ou anomalie sur la base du jeu de données NSL-KDD. Nous avons utilisé la bibliothèque Weka pour instancier les algorithmes de classification utilisés. Ce travail avait pour but, de comparer les précisions et le taux de réussite de détection d'intrusion du chaque algorithme (KNN, Naïve Bayes, Random forest). Les expérimentations que nous avons menée et les résultats que nous avons obtenu ont montré que les résultats du notre algorithmes sont proches, mais le Random forest attient des résultats plus performante que les autres. Pour cette raison, nous avons choisi la forêt aléatoire ou bien Random forest comme meilleur algorithme de classification de la base de données NSL-KDD.

Pour conclure, 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 réalisé dans le future, telles que :

- La réalisation d'un modèle de classification multi-classes (Normal, DOS, R2L, U2R et Probe) au lieu de la classification binaire (Normal, Attaque) réalisée dans ce travail.
- il est possible de tester d'autres classifieurs, supervisés et non supervisés, en utilisant la même base de données

Bibliographie

[1] https://www.redhat.com/fr/topics/security

[2] Rodrigue Mpyana Mise en place d'un système de sécurité basé sur l'authentification dans un réseau IP. Cas de Mecelco, 2011.

[3] William Stallings, network security essentials: applications and standards fourth edition, 2011.

[4] Liran LERMAN, Les systèmes de détection d'intrusion basés sur du machine Learning, UNIVERSITÉ LIBRE DEBRUXELLES.

[5] Laurent Bloch-Christophe Wolfhugel. Sécurité informatique .EYROLLES, 2eme édition. 2005.

[6] Le grand livre de la sécurité informatique. Sécurité Info, Editions du 6 novembre 2006.

[7] Laurent Poinsot «Introduction à la sécurité informatique», support de cours, Université Paris 13.

[8] https://www.sophos.com/fr/medialibrary/PDFs/case%20

[9] https://www.verizon.com/info/definitions/antivirus/

[10] https://www.futura-sciences.com/tech/definitions/internet-firewall-474/

[11] Mme. BOUKHLOUF Djemaa, Une approche à base d'agents mobiles pour la sécurité des systèmes d'informations sur le web, Thèse de Doctorat, UNIVERSITE MOHAMED KHIDERBISKRA, 2016.

[12] https://www.guill.net

[13] https://fr.wikipedia.org/wiki/ Système_de_détection_d'intrusion.html

[14] https://www.juniper.net/fr/fr/products-services/what-is/ids-ips/

[15] Abdelhalim Zaidi. Recherche et détection des patterns d'attaques dans les réseaux IP _a hauts débits. Réseaux et télécommunications [cs.NI]. Université d'Evry-Val d'Essonne, 2011.

[16] Nathalie Dagorne. Détection et prévention d'intrusion : présentation et limites. [Rapport de recherche], Université de Nancy1, France, 2006.

[17] S. AGGOUN, S. BELKACEM, Mise en œuvre d'une solution de sécurité basée sur les IDS Cas d'étude : entreprise Cevital, Mémoire de Master en Informatique, Université ABDERAHMANE MIRA de Bejaia, 2013. [18] SLIMANI Ahmed, Application des systèmes immunitaires artificiels à la détection d'intrusion, USTO-MB : 2011

[19] David Powell ET Robert Stroud: Conceptual Model and Architecture of MAFTIA. Technical Report Series-University of Newcastle Upon Tyne Computing Science, 2003.

[20] Cédric Michel, Langage de description d'attaques pour la détection d'intrusions par corrélation d'événements ou d'alertes en environnement réseau hétérogène, Université de Rennes 1,2003.

[21] Yousef Farhaoui, «Evaluation des systèmes de détection et de prévention des intrusions et la conception d'un BIDS », thèse de doctorat, Université Ibn Zohr, 2012.

[22] Lehmann Guillaum, cours de sécurité informatique 2003-04-13.

[23]- Microsoft Expériences, Tout savoir sur l'Intelligence Artificielle, (consulté le 09/02/2019), disponiblesur:https://experiences.microsoft.fr/business/intelligenceartificielle-ia-business/comprendreutiliser-intelligence-artificielle/

[24] Cours de Yann Le Cun, directeur scientifique du FAIR (Facebook Artificial Intelligence Research) sur l'intelligence artificielle au Collège de France [25] M^{elle} MAHDJANE Karim. Détection d'anomalies sue des données biologiques par SVM. Université Mouloud Mammeri de Tizi Ouzou, 14 octobre 2012.

[26] http : //www.groupe-hli.com/machine-learning-dans-industrie, Consulté le 27/03/2021.

[27] Nicolas La Roux. Avancées théoriques sur la /représentation et l'optimisation des réseaux de neurones, Université de Montréal, Mars, 2008.

[28] S. Russell et P. Norvig. Intelligence artificielle : Avec plus de 500 exercices. Pearson Education, 2010.

[29] Vasilev, D. Slater, G. Spacagna, P. Roelants, and V. Zocca, Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and Tensor-Flow. Packt Publishing Ltd, 2019.

[30] L.-P. Chen, "Mehryar mohri, afshin rostamizadeh, and ameet talwalkar: Foundations of machine learning," 2019**.**

[31] S. Balech et C. Benavent. Les techniques du NLP pour la recherche en sciences de gestion.2019.

[32] P. Laskov & P. Düssel & C. Schäfer & K. Rieck (2005), _Learning intrusion detection: Supervised or unsupervised? _, Fraunhofer FIRST.IDA, Berlin, Germany.

[33] Marcus A. Maloof (2005), _Machine Learning and Data Mining for Computer Security_ ,Springer London Ltd, ISBN-10 184628029X ; ISBN-13 978-1846280290.

[34] A.Öksüz (2007), _Unsupervised Intrusion Detection System_, Technical University of Denmark, Informatics and Mathematical Modelling, Denmark.

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

[36] Pierre Borneet all, Les réseaux de neurones présentation et applications… Edition TECHNIP, France 2007.

[37] L.Dhanabal& Dr. S.P. Shantharajah, «A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms», International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015.

[38] http://www.softfonic.com /s/NetBeans –ide-8.2.

[39] http://www.jmDoudoux.com/java.

[40] https://geek.mg/fr/tutoriel/presentation-de-weka/

[41] https://www.tutorialspoint.com/machine_learning_with_python