République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Jijel
Faculté des Sciences Exactes et Informatique
Département d'Informatique

Mémoire de fin d'études

Pour l'Obtention du Diplôme Master de Recherche en Informatique

Option : Informatique Légale et Multimédia

Thème

Application des techniques de data mining et de machine learning pour la détection d'intrusions.

Présenté par :

Mekhbi Aicha

Menouar Chahrazad

Encadré par :

Ammar Boulaiche

Promotion : 2018.

Remerciements

Je tiens à remercier en premier lieu le ***Dieu*** le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce Modeste travail.

Mes remerciements s'adressent aussi à Mr.***BOULAICHE***, qui me font le grand honneur d'accepté d'examiner mon travail.

Je tiens à remercier très sincèrement l'ensemble des membres du jury qui me font le grand honneur d'accepté de juger mon travail.

Je tiens aussi à remercier chaleureusement ma famille pour leur compréhension, leur soutien moral et leurs encouragements continus et soutenus.

Enfin, j'adresse mes plus sincères remerciements à tous mes amies, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Dédicaces

Je dédie ce mémoire à :

chers parents,*que nulle dédicace ne puisse exprimer mes sincères sentiments,pour leur patience illimitée,leur encouragement contenu,leur aide,en témoignage de mon profond amour et respect pour leurs grands sacri ces.*

Mes chers frères,*pour leur grand amour et leur soutien.*

Mes chers amis *pour leur encouragement.*

Et *à toute ma famille et à tous ceux que j'aime.*

mon binôme ***chaharzed.***

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible,je vous dis ***merci .***

Aicha

Dédicaces

Je dédie ce mémoire à :

chers parents,*que nulle dédicace ne puisse exprimer mes sincères sentiments,pour leur patience illimitée,leur encouragement contenu,leur aide,en témoignage de mon profond amour et respect pour leurs grands sacri ces.*

Mes chers frères,*pour leur grand amour et leur soutien.*

Mes chers amis *pour leur encouragement.*

Et *à toute ma famille et à tous ceux que j'aime.*

mon binôme ***Aicha .***

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible,je vous dis ***merci .***

chahrazed

Table des matières

Liste des tableaux

Table des figures

Liste des Algorithmes

Liste des abréviations

IP Internet Protocol

TCP Transmission Control Protocol

ARP Address Resolution Protocol

MAC Media Access Control

DNS Domain Name System

IDS Intrusion Detection System

SYN synchronize

NIDS Network based Intrusion Detection System

HIDS Host based Intrusion Detection System

ICMP Internet Control Message Protocol

KNN K-nearest neighbor

SVM Support Vector Machines ou machines à vecteurs de support

CAH clustering ascendante hiérarchique

K-pp K-Plus proche voisin

DARPA Defense Advanced Research Projects Agency

MIT Massachusetts Institute of Technology

DOS Denial Of Service

R2L Remote to User

U2R User to Root

KDD Knowledge Discovery in Databases

MLP MultiLayer Perceptron

Introduction générale

De nos jours, les systèmes informatiques sont devenus de plus en plus complexes et diversifiés en raison notamment de l'émergence de nouvelles technologies, de l'accroissement continu de la volumétrie des données numériques ainsi que de la multiplicité des sources de données de plus en plus hétérogènes, conjugués aux besoins pressants à exploiter ces données dans un processus d'aide à la prise de décisions.

Ce croissement a été accompagné par une augmentation phénoménale du nombre d'utilisateurs, qui ne sont pas forcément pleins de bonnes intentions vis-à-vis de ces données numériques. Ces derniers tentent, tout le temps, d'exploiter les vulnérabilités figurant dans les différents systèmes, notamment dans ceux qui contiennent des informations sensibles dans le but de les lire, les modifier et même de les détruire.

Face à ce risque croissant du piratage et de la cybercriminalité, la sécurité des systèmes informatique est devenue, de nos jours, un enjeu incontournable, tant pour les organisations que pour les individus. Ainsi, pour faire face à tous ces problèmes de sécurité informatique, différents mécanismes ont été mis en place, lors de ces dernières années, afin de prévenir toute sorte d'attaques. Parmi ceux-ci, on trouve les systèmes de cryptages, les pare-feux, les anti-virus, etc.
Cependant, ces mécanismes ont des limites et ne répondent donc pas à tous les besoins que les individus ainsi que les organismes ont en termes de besoin de sécurité et de protection contre toutes ces menaces. Malheureusement, certains types d'attaques peuvent facilement contourner ces mécanismes en nuisant la confidentialité, l'intégrité et la disponibilité des systèmes que ces derniers protègent.

Pour faire face à ce problème, le concept qui s'appelle système de détection d'intrusion (IDS) a été introduit pour renforcer la sécurité des systèmes informatiques. Ce dernier peut être défini comme tout outil, méthode et ressource qui nous aident à prévoir ou à identifier toute activité non autorisée dans un système informatique. Un IDS peut être classé en différents types selon leurs caractéristiques, leurs techniques de détection ou leurs architectures. Toutefois, la manière la plus connue dans la littérature pour classer les systèmes de détection d'intrusions est de les grouper par approche de détection. En se basant sur cette dernière, un système de détection d'intrusions peut être par anomalies (comportemental) ou par signatures. La détection d'intrusion par anomalies consiste à construire un modèle identifiant les comportements déviants du modèle de référence. Ce dernier est le résultat d'une phase d'apprentissage sur une grande base de données des comportements normaux. Alors que la détection d'intrusions par signatures, quant à elle, consiste simplement à comparer les activités suspectes du système avec les signatures des attaques connues qui sont sauvegardées dans une base appelée base des signatures d'attaques.

En fait, chacune de ces deux techniques possède des avantages et des inconvénients. La première, par exemple, a l'avantage d'être capable de détecter les nouvelles attaques, qui ne sont malheureusement pas

détectées par la deuxième technique, car les signatures de ces dernières ne sont pas encore disponibles dans la base des signatures. Ainsi, dans le cadre de ce projet de fin d'étude, nous allons nous focaliser sur la première technique et plus particulièrement sur les techniques où la génération du modèle de détection d'intrusions est basée sur les approches de data mining et de machine learning.

Problématique et objectifs du travail

Malheureusement, malgré leur utilité, notamment pour la détection des nouvelles attaques, les systèmes de détection d'intrusions comportementaux (à base d'anomalies) souffrent encore de nombreux problèmes dont les plus saillants sont le nombre important de faux positifs et de faux négatifs qu'ils génèrent. En fait, les faux positifs (c'est-à-dire les fausses alertes) sont générés lorsque l'IDS identifie des activités normales comme des intrusions, alors que les faux négatifs correspondent aux attaques ou intrusions qui ne sont pas détectées (aucune alerte n'est générée).

Pour faire face à tous ces problèmes, les techniques de détection d'intrusions à base d'anomalies sont de plus en plus orientées vers l'application des techniques de data mining et de l'intelligence artificielle. Par conséquent, des dizaines de travaux basés sur les techniques d'apprentissage automatique ont été publiés ces dernières années, des travaux qui visent à améliorer la capacité de détection des systèmes détection d'intrusion en appliquant des approches plus efficaces, telles que les arbres de décision, les algorithmes génétique les K plus proches voisins, les réseaux de neurones, etc.

Ainsi, dans le cadre de ce projet de fin d'étude, nous allons nous intéresser plus particulièrement à l'application de certaines méthodes de l'intelligence artificielle et du data mining pour réaliser des systèmes de détection d'intrusion comportementaux. L'objectif de notre travail, présenté dans ce mémoire, est donc de faire une étude comparative permettant d'identifier lesquelles des techniques de l'intelligence artificielle sont plus efficaces pour la détection d'intrusions, notamment en termes de leur taux de détection, du taux de fausses qu'elles génèrent, du taux d'attaques qui passent inaperçues, etc. Les techniques traitées dans ce travail sont : les réseaux de neurones, les supports à vaste marge ou SVM (Support Vector Machines), les arbres de décision et le K-plus Proche voisin (KNN).

Organisation du mémoire

Ce mémoire est structuré en trois chapitres encadrés par une introduction générale et une conclusion.

Le premier chapitre présente les différents concepts de la sécurité informatique, ses objectifs, les attaques que les systèmes informatiques subissent ainsi que les différents outils de défense contre ces attaques. Il s'intéresse plus particulièrement aux systèmes de détection d'intrusions, leur contexte, leur définition, leur architecture, leurs critères de classification et leurs mesures d'efficacité. Le chapitre se termine par une présentation des différentes limites de ces systèmes.

Le deuxième chapitre est réservé à la présentation de quelques concepts et définitions du Data Mining et de l'intelligence artificielle. Il présente les différentes étapes du processus de data mining, les différentes techniques de l'apprentissage automatique, leur définition, leur classification (supervisée ou non supervisée) ainsi que les algorithmes les plus utilisés dans chaque classe avec leurs avantages et inconvénients. Le chapitre se termine par une petite comparaison entre l'apprentissage supervisé et non supervisé.

Le dernier chapitre présente le travail réalisé dans ce projet de fin d'étude. Il commence par une petite description du benchmark NSL-KDD utilisé pour générer les différents modèles. Il présente ensuite les différentes techniques de prétraitement de données qui ont été appliquées pour préparer les données de la base NSL-KDD, ainsi que la démarche suivie pour générer et pour tester et comparer les différents modèles de détection d'intrusions. Le chapitre se termine par une présentation des résultats obtenus avec une analyse comparative de ces derniers. Enfin , nous finirons le mémoire par une annexe, dans laquelle nous présentons une description détaillée de l'ensemble de donnée KDD99.

Chapitre 1

Sécurité informatique et systèmes de détection d'intrusions

1.1 Introduction

La sécurité informatique est devenue une préoccupation depuis les années 80. Plus particulier, depuis l'apparition du ver *Morris*[1] qui profitait d'une vulnérabilité dans le service de messagerie *sendmail*. Ce qui était juste un *jeu innocent* pour son auteur (un étudiant) lui a couté une condamnation pénale et a déclenché la création du premier CERT (Computer Emergency Response Team). Au début, la notion de la sécurité informatique se résumait dans la mise en place d'une politique de sécurité visant à garantir la confidentialité, l'intégrité et la disponibilité de ressources informatiques sensibles.

Toutefois, avec le développement du monde informatique et des techniques de piratage et de cybercriminalité, cette notion a totalement changé. Ainsi, dans ce chapitre, nous allons présenter la notion de la sécurité informatique telle qu'elle est définie aujourd'hui, les différents types d'attaques et leur classification (selon la source, l'impact, la cible et l'objectif de l'attaque), les différents outils de sécurité, etc. Nous allons présenter ensuite, les systèmes de détection d'intrusions, leur contexte, leur architecture, leur classification, ainsi que leurs points forts et leurs points faibles.

1.2 Sécurité informatique

1.2.1 Qu'est-ce que la sécurité informatique ?

La sécurité informatique est l'ensemble des moyens matériels et logiciels mis en œuvre pour minimiser la vulnérabilité d'un système informatique contre des menaces accidentelles ou intentionnelles [2].

La sécurité informatique consiste à garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu [2].

Autrement dit, La sécurité informatique est la protection de l'information et des systèmes d'information contre l'accès, l'utilisation, la divulgation, la perturbation, la modification ou la destruction afin de garantir les objectifs de la sécurité informatique [3].

La sécurité d'un système repose sur cinq grands objectifs [4, 5] :

- **La disponibilité** : c'est la propriété que l'information sur un système est disponible si nécessaire. Elle permet de maintenir le bon fonctionnement du système d'information. Un autre aspect de la

disponibilité est d'assurer que les ressources nécessaires sont utilisables et accessibles à n'importe quel moment.

- ➢ **L'intégrité** : signifie l'état de données au moment de traitement, de conservation ou de transmission, qui ne doit pas être modifié ou effacé de façon non autorisée. L'intégrité des données comprend quatre éléments : l'intégralité, la précision, l'exactitude/authenticité et la validité.
- ➢ **La confidentialité** : signifie que seules les personnes autorisées ont accès aux informations qui leur sont destinées. Tout accès indésirable doit être empêché. Personne ne peut accéder à l'information s'il n'en a pas le droit.
- ➢ **L'authentification** : c'est la procédure qui consiste, à vérifier l'identité d'une entité (personne, ordinateur ...), afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications ...). L'authentification permet donc de valider l'authenticité de l'entité . Le service d'authentification permet évidemment d'assurer l'authenticité d'une communication.
- ➢ **La non-répudiation** : c'est le fait de ne pas pouvoir nier ou rejeter qu'un événement a eu lieu. Elle permet donc de garantir qu'une transaction ne peut être niée.

1.2.2 Attaques informatique

Les problèmes techniques actuels de sécurité informatique peuvent être classés en deux grandes catégories [6] :

- Ceux qui concernent la sécurité de l'ordinateur proprement dit, serveur ou poste de travail, de son système d'exploitation et des données qu'il abrite .
- Ceux qui découlent directement ou indirectement de l'essor des réseaux, qui multiplie la quantité et la gravité des menaces .

La deuxième catégorie augmente la probabilité d'être vulnérable aux différentes attaques.

1.2.2.1 Définition des attaques

Une attaque informatique est définie par toute tentative de détruire, exposer, modifier, désactiver, voler ou obtenir un accès non autorisé ou toute utilisation non autorisée d'une information, d'un logiciel ou d'un matériel physique comme un serveur, un service, des personnes et de leurs qualifications [3].

1.2.2.2 Classification des attaques

La classification des attaques informatiques peut être donnée selon plusieurs critères [7, 8, 9] :

1.2.2.2.1 Classification selon la source de l'attaque

- ➢ **Les attaques internes** :sont celles provenant des utilisateurs situés dans votre réseau. Elles proviennent essentiellement de maladresse des utilisateurs, de leur méconnaissance des outils qu'ils utilisent mais aussi de leur malveillance.
- ➢ **Les attaques externes** : viennent des utilisateurs situés en dehors de votre système informatique, qui essayent d'accéder à des informations ou à des ressources d'une manière illégitime et non autorisée.

1.2.2.2.2 Classification selon l'impact de l'attaque

- **Les attaques passives** :Ce sont les attaques qui visent essentiellement à capturer le contenu d'une information et l'analyse de trafic. Ce type d'attaques est généralement très difficile à détecter car elles ne causent aucune altération des données.
- **Les attaques actives** :Contrairement au premier type, les attaques actives impliquent certaines modifications du flot de données ou la création d'un nouveau flot frauduleux ; elles peuvent être subdivisées en quatre catégories : mascarade, rejeu, modification de messages et déni de service.

1.2.2.2.3 Classification selon la cible de l'attaque

- **Les attaques réseaux** :s'appuient sur des vulnérabilités liées directement aux protocoles réseaux ou à leur implémentation.
- **Les attaques applicatives** :s'appuient principalement sur des vulnérabilités spécifiques aux applications utilisées.

1.2.2.2.4 Classification selon l'objectif de l'attaque

Dans une telle classification, les attaques peuvent perturber le flux normal des paquets en se basant sur la modification, l'interception, l'interruption ou la fabrication d'une partie ou de la totalité de ces paquets, ou sur la combinaison de deux ou de plusieurs de ces dernières.

- **Interruption** :Un atout du système est détruit ou devient indisponible ou inutilisable. C'est une attaque portée à la disponibilité. Cette attaque peut avoir une cible spécifique ; Par exemple, Une forme de déni de service est la perturbation d'un réseau entier, soit en désactivant le réseau ou en le surchargeant de messages afin de dégrader les performances. La destruction d'une pièce matérielle (tel un disque dur), la coupure d'une ligne de communication, etc.
- **Modification** : La modification des messages signifie simplement qu'une partie d'un message légitime est modifiée, ou que les messages sont retardés ou réordonnés, de façon (presque) indétectable, pour produire un effet non autorisé. Il s'agit d'une attaque portée à l'intégrité. Par exemple, 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.
- **Interception** (analyse de traffic) : Une tierce partie non autorisée obtient un accès à un atout. C'est une attaque portée à la confidentialité. Il peut s'agir d'une personne, d'un programme ou d'un ordinateur. Par exemple, Une écoute téléphonique dans le but de capturer des données sur un réseau, ou la copie non autorisée de fichiers ou de programmes.
- **Fabrication**(mascarade) : Une tierce partie non autorisée insère des contrefaçons dans le système. C'est une attaque portée à l'authenticité. Il peut s'agir de l'insertion de fausses données dans un réseau, compromet la fiabilité des informations transmises, ou l'ajout d'enregistrements à un fichier. L'obtenir des privilèges en usurpant l'identité d'une entité qui a privilèges à l'autorisation.

1.2.2.3 Quelques attaques informatiques

Il existe un grand nombre d'attaques permettant à une personne mal intentionnée de s'approprier des ressources, de les bloquer ou de les modifier. Certaines requièrent plus de compétences que d'autres, en voici quelques unes [10, 11] :

- **IP Spoofing (falsification d'adresse source IP)** : est une technique d'attaque consistant à usurper l'identité d'un autre utilisateur du réseau en utilisant son adresse IP, ce qui permet de faire croire que la connexion provient d'un compte d'utilisateur autorisé. L'autre utilisation de l'IP Spoofing permet de profiter d'une relation de confiance entre deux machines pour prendre la main sur l'une des deux.
- **ARP Spoofing « ARP cache poisoning »** : est un type d'attaque dans lequel un acteur malveillant envoie des messages ARP falsifiés sur un réseau local. Cela entraîne la liaison de l'adresse MAC d'un attaquant avec l'adresse IP d'un ordinateur ou serveur légitime sur le réseau. Une fois que l'adresse MAC de l'attaquant est connectée à une adresse IP authentique, l'attaquant commencera à recevoir toutes les données destinées à cette adresse IP. L'usurpation d'ARP peut permettre à des tiers malveillants d'intercepter, de modifier ou d'arrêter des données transmis.
- **Spoofing DNS :** le protocole DNS assure la correspondance entre le nom d'une machine et son adresse IP. Les attaques décrites ici concernent les faiblesses du protocole DNS :
  - **DNS ID Spoofing** : à pour but de d'envoyer une fausse réponse à une requête DNS avant le serveur DNS. De cette façon, le pirate peut rediriger vers lui le trafic à destination d'une machine qu'il l'intéresse. Pour implémenter cette attaque, le pirate doit connaître l'ID de requête DNS. Pour cela, il peut utiliser un sniffer s'il est sur le même réseau, soit prédire les numéros d'ID par l'envoi de plusieurs requêtes et l'analyse des réponses.
  - **DNS Cache Poisoning** : les serveurs DNS possèdent un cache permettant de garder pendant un certain temps la correspondance entre un nom de machine et son adresse IP. Le DNS Cache Poisoning consiste à corrompre ce cache avec de fausses informations. Ces fausses informations sont envoyées lors d'une réponse d'un serveur DNS contrôlé par le pirate à un autre serveur DNS, lors de la demande de l'adresse IP d'un domaine . Le cache du serveur ayant demandé les informations est alors corrompu.
- **SYN Flooding** : cette attaque exploite le mécanisme de poignée de main en trois temps (Three-ways handshake) du protocole TCP. Elle consiste en l'envoi d'un grand nombre de demandes de connexions au serveur cible (SYN) à partir de plusieurs machines et ne pas y répondre. Lors d'une demande de connexion, le serveur est en attente et bloque pendant un certain temps une partie de ses ressources pour cette nouvelle connexion. Le but est d'envoyer plus de demandes qu'il ne peut en traiter dans un temps donné. Ainsi, le serveur gaspille toutes ses ressources réseau à répondre à des requêtes qui ne mènent nulle part et il ne pourra plus subvenir aux besoins de vrais clients.
- **Smurf** :c'est une attaque qui s'appuie sur le Ping (Ping est un outil exploitant le protocole ICMP, permettant de tester les connexions sur un réseau en envoyant un paquet et en attendant la réponse) et les serveurs de diffusion (broadcast) pour paralyser le réseau (Un serveur broadcast est un serveur capable de dupliquer un message et de l'envoyer à toutes les machines présentes sur le même réseau).

On falsifie d'abord l'adresse IP source pour se faire passer pour la machine cible. On envoie alors un ping sur un serveur de broadcast. Il le fera suivre à toutes les machines qui sont connectées qui répondent par un message ICMP Echo Reply à la direction de la cible. Dans ce cas tout le réseau cible subit le déni de service, car l'énorme quantité de trafic généré par cette attaque entraîne une congestion du réseau.

1.2.3 Outils de sécurité

Les attaquants disposent de plusieurs moyens pour réussir chaque phase d'attaque. La disponibilité des outils d'attaques et la richesse des sources d'informations accentuent le risque des intrusions. Par conséquent les administrateurs sécurisent de plus en plus leurs systèmes informatiques. Ils s'appuient sur diverses solutions comme l'authentification, le contrôle d'accès aux ressources, les scanners de vulnérabilités, les anti-virus, la cryptographie, le firewall, et la détection d'intrusion. Nous détaillons dans la suite chacune de ces méthodes.

1.2.3.1 Authentification et contrôle d'accès aux ressources

C'est le fait de prouver son identité par une quelconque méthode, qu'elle soit choisie ou imposée. Cette opération intervient juste après l'identification. Elle peut utiliser plusieurs moyens, information que seuls nous connaissons (mot de passe), une autorité tierce (carte d'identité), ou encore des moyens biométriques.

1.2.3.2 Scanners de vulnérabilités

Ce sont des outils qui évaluent les faiblesses et les dangers qui pèsent sur un système d'information ou sur un réseau et qui dressent un journal de ce qu'ils trouvent. Ils sont des outils très utiles pour les administrateurs système et réseau afin de surveiller la sécurité du parc informatique dont ils ont la charge. Nous citons Quelques outils : Nessus ,Saint, whisker [13]. A contrario, cet outil est parfois utilisé par des pirates informatiques afin de déterminer les brèches d'un système. Cependant, malgré le grand nombre de vulnérabilités détectées, les scanners d'aujourd'hui sont inaptes à déterminer toutes les faiblesses possibles. De plus, la mise à jour de ces produits ne suit pas le rythme de la découverte des nouvelles vulnérabilités [12].

1.2.3.3 Anti-virus

C'est un programme ou un ensemble de programmes de sécurité qui sont installés sur l'ordinateur pour le protéger contre les logiciels malveillants tels que les virus, les vers, les chevaux de Troie, les logiciels publicitaires, etc. Les actions d'un logiciel anti-virus sont :

- Il protège en scrutant tous les fichiers qui pénètrent sur l'ordinateur .
- Il analyse périodiquement le contenu du disque dur .
- Il désinfecte en cas de contamination .
- Il supprimer les fichiers infectés .
- Etc.

1.2.3.4 Cryptographie

Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer et de déchiffrer des messages à partir d'algorithmes cryptographiques. Il sert à préserver la confidentialité , l'intégrité et l'authenticité des données ; dans un canal de communication (courrier, réseaux téléinformatiques divers, etc.). Le chiffrement consiste donc à transformer une donnée lisible (ou clair) en une donnée illisible (ou incompréhensible par un humain, un logiciel, ou un cryptogramme) .Le déchiffrement est logiquement l'opération inverse du chiffrement. Il existe deux types de cryptographie : symétrique (cryptographie à clé secrète) et asymétrique (cryptographie à clés publiques).

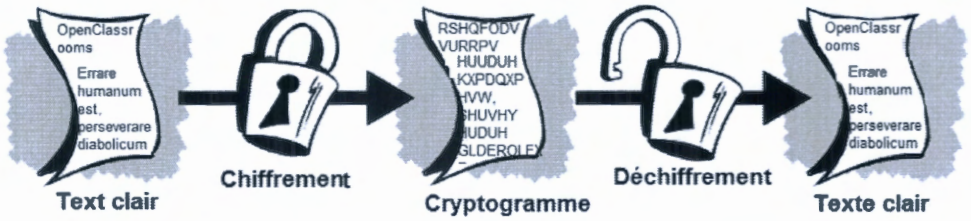

FIGURE 1.1 – Chiffrement d'un message
[14].

1.2.3.5 Firewall

Un pare-feu (firewall) est un système matériel ou logiciel qui inspecte les flux entrant et sortant du réseau. Ils font la première ligne de défense de la plupart des systèmes informatiques, ils se basent sur un ensemble de règles afin d'autoriser ou interdire le passage des paquets. Malgré leur grand intérêt, les pare-feux présentent quelques lacunes. En effet, un attaquant peut exploiter les ports laissés ouverts pour pénétrer le réseau local. De plus, l'opération supplémentaire d'encapsulation/décapsulation des données permet à l'attaquant de contourner le pare feu. Les scripts constituent aussi des sources d'intrusion que les pare feux échouent à détecter.

FIGURE 1.2 – Protection par firewall
[15].

1.2.3.6 Détection d'intrusion

La détection d'intrusion est une partie essentielle de l'infrastructure de sécurité. Elle est utilisée pour détecter, identifier et arrêter les intrus. Les administrateurs peuvent compter sur les systèmes de détection d'intrusion pour découvrir les attaques et empêcher leur utilisation. Ces derniers permettent, de façon automatisée, de collecter l'information sur le comportement des utilisateurs du système et de détecter tout comportement malicieux. Dans la deuxième partie de ce chapitre, nous présentons une analyse détaillée de ces systèmes.

1.3 Systèmes de détection d'intrusions

En effet, les anti-virus et les pare-feux représentent des solutions de protection qui s'avèrent limitées face au développement rapide des techniques de piratage, d'où vient la nécessité de mettre en place un système de détection et de prévention d'intrusion(IDS), qui permet de détecter toute violation de la politique de sécurité et de signaler les attaques portant atteinte à la sécurité du réseau informatique.

Dans la suite de ce chapitre nous allons d'abord présenter la notion de système de détection d'intrusion, l'architecture, la classification, les techniques utilisées dans les systèmes de détection d'intrusion.

1.3.1 Contexte

En 1980, le concept de détection d'intrusion a commencé avec *James d'Anderson*[16] ; Il a introduit un modèle de classification des menaces qui constitue le noyau d'un système de surveillance de sécurité basé sur la détection d'anomalies dans le comportement des utilisateurs. Après, En 1987 *Dorothy Denning* a publié le premier modèle de détection d'intrusion [17] .

Depuis lors, plusieurs mécanismes de détection d'intrusions ont été développés pour faire face aux attaques informatiques. Toutefois, le processus réactif de ces derniers n'est pas encore suffisant. En fait, il est nécessaire de détecter toute violation de la politique de sécurité, ce qui n'est pas toujours le cas pour les systèmes de détection d'intrusions actuels.

1.3.2 Définitions d'un système de détection d'intrusions

Il existe plusieurs définitions pour les systèmes de détection d'intrusions. Cependant, nous allons seulement citer quelques-unes [18, 19].

- **Définition 1**

  - L'intrusion : est une pénétration illégale au système, une tentative d'un utilisateur du système d'obtenir des privilèges non autorisés, ou bien toute tentative de compromettre les services standards de la sécurité : la confidentialité, l'intégrité ou la disponibilité des informations.

  - La détection d'intrusion (ID) :est le processus d'identification et de réponse aux activités malveillantes ciblant les ressources informatiques et réseau. Cette définition introduit la notion de détection d'intrusion en tant que processus impliquant la technologie, les personnes et les outils.

  - Un système de détection d'intrusion (IDS) : est un dispositif logiciel ou matériel, ou bien une combinaison des deux, qui est chargé de la surveillance d'un système d'information pour détecter toute effraction dans l'utilisation des ressources.

- **Définition 2**

  Un IDS est un mécanisme de contrôle dont le but est de détecter les menaces qui affectent le bon fonctionnement du système d'information. Ces menaces peuvent être internes ou externes, liées aux actions qui exploitent des failles connues ou inconnues dans le système.

- **Définition 3**

  Un système de détection d'intrusion (IDS) est un logiciel utilisé pour détecter l'activité d'un intrus, il renifle, analyse, enregistre le trafic réseau et génère des alarmes en cas d'intrusion .

1.3.3 Architecture de base d'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 : capture, analyseur et Manager [20, 21] .

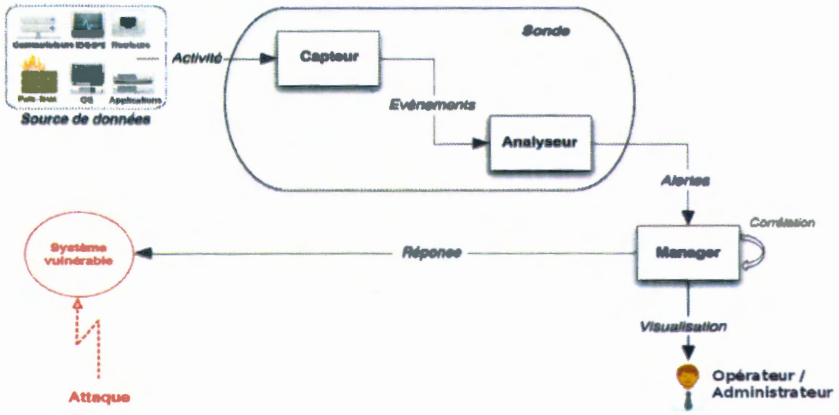

FIGURE 1.3 – Architecture d'un IDS [23].

Capteur(senseur) : c'est un module qui s'occupe de la collecte d'informations depuis une source de données (interface réseau, journaux système, etc.), et de les envoyer à l'analyseur.

Analyseur : permet d'analyser les informations collectées par le Capteur. Il est responsable de déterminer si une intrusion a eu lieu ou pas. S'il détecte une activité intrusive, l'analyseur génère des alertes qui guident l'administrateur en proposant des solutions aux problèmes rencontrés. Donc, l'analyseur peut être considéré comme le noyau de l'IDS.

Manager : il permet de collecter et de valider les alertes produites par l'analyseur et met en place des contre-mesures appropriées (réponses) qui peuvent être automatisées. L'administrateur de sécurité (composante humaine) configure la sonde et le gestionnaire d'alertes conformément à la politique de sécurité. Il est à noter que l'opérateur et l'administrateur peuvent être la même personne.

1.3.4 Classification des IDS

On peut classer les systèmes de détection d'intrusions selon plusieurs critères comme la figure 1.4 montre [22, 24] :

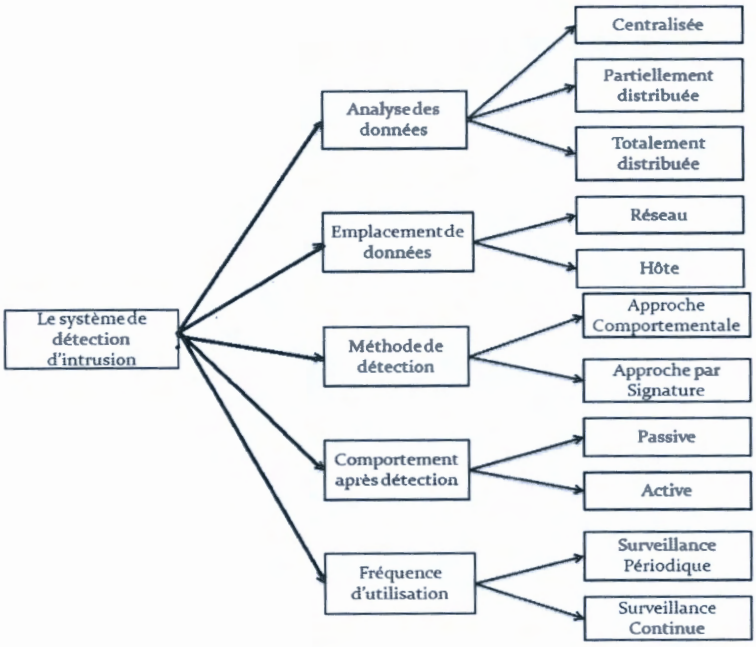

FIGURE 1.4 – Critères de classification d'IDS.

1.3.4.1 Analyse des données

La technologie des systèmes de détection d'intrusions permet d'analyser les données recueillies de trois façons [25] :

1.3.4.1.1 Analyse centralisée

Cette analyse est connue aussi par l'analyse monolithique .L'IDS possède plusieurs capteurs, il centralise les événements (ou alertes) pour les analyser sur une seule machine. Avec une stratégie de contrôle centralisée, la surveillance, la détection et le reporting sont commandés directement d'un endroit central.

1.3.4.1.2 Analyse partiellement distribuée

Cette analyse est connue aussi par l'analyse hiérarchique .Elle est caractérisée par l'existence des secteurs de contrôle hiérarchiques. Dans cette architecture, le réseau est divisé en groupes sous-réseaux (appelés clusters) où chacun possède son propre IDS local .Un chef de cluster (Cluster head) gère la communication à l'intérieur du cluster, et avec d'autres clusters. Tandis que chaque nœud (IDS local) dans le cluster effectue une détection , une surveillance et une analyse locale, et envoi les résultats au chef de cluster qui va produire les réponses possibles. Les chefs de clusters quant à eux effectuent une détection globale [26].

1.3.4.1.3 Analyse totalement distribuée

Cette analyse est connue aussi par l'analyse distribuée ou coopérative .Dans cette architecture, le réseau

est divisé en plusieurs sous-réseaux où chacun possède son propre IDS qui fait la collecte d'informations, l'analyse et la détection, de plus les alertes seront réalisées au niveau local de chaque nœud sans transmettre les messages à un autre. Toutefois, dans le cas d'information incomplète, ou bien suspicion, les nœuds peuvent déclencher des procédures de collaboration supervisées à travers des nœuds maitres.

1.3.4.2 Emplacement de données

Les IDS sont également classés en fonction de leur source d'information. Certains IDS analysent les paquets réseau en cherchant les paquets qui peuvent constituer une éventuelle attaque. D'autres IDS analysent les fichiers journaux produits par les systèmes d'exploitation et les applications qui le constituent pour détecter des signes d'intrusion [12, 27] .

1.3.4.2.1 IDS basé réseau *(NIDS :Network-based IDS)*

Un NIDS est un IDS orienté réseau. Il permet d'analyser le trafic circulant au niveau du réseau TCP/IP pour détecter, en temps réel, d'éventuelles intrusions. En d'autres termes, un NIDS est un système de détection d'intrusions qui écoute le trafic réseau, puis analyse et génère des alertes si des paquets semblent dangereux. Le NIDS offre l'avantage de sa furtivité et de n'ajouter aucune surcharge au réseau en terme de trafic. La figure 1.5 montre l'architecture d'un réseau contenant un NIDS.

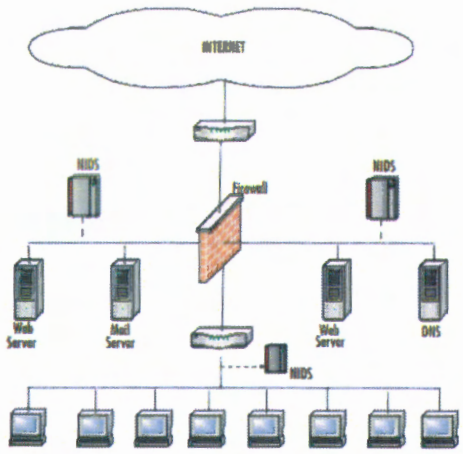

FIGURE 1.5 – Modèle d'architecture pour NIDS proposé par le groupe IDWG [28].

Avantages et inconvénients de NIDS

✓ **Avantages :**

☞ Le déploiement des NIDS a peu d'impact sur un réseau existant. Les NIDS sont généralement des dispositifs passifs qui écoutent sur un fil de réseau sans interférer avec le fonctionnement normal d'un réseau.

☞ Les NIDS peuvent surveiller un grand réseau sans affecter les performances de ce dernier.

✓ **Inconvénients :**

- ☞ Les NIDS ne peuvent pas analyser des informations chiffrées. ils peuvent uniquement analyser les parties non chiffrées d'un paquet.
- ☞ Les NIDS peuvent avoir des difficultés à posséder tous les paquets dans un réseau important ou occupé et, par conséquent, peuvent ne pas reconnaître une attaque lancée pendant une période de fort trafic.
- ☞ la probabilité de faux négatifs (attaques non détectées) est élevée et il est difficile de contrôler le réseau entier.

1.3.4.2.2 IDS basé hôte *(HDIS :Host-Based IDS)*

Un HIDS est un système de détection d'intrusion placé sur une machine afin d'analyser, en temps réel, les actions effectuées sur cette dernière. Ces actions sont généralement enregistrées dans des fichiers journaux. Un HIDS ne protège donc que le système local de la machine sur laquelle il est installé. Il peut aussi capturer les paquets réseaux entrant/sortant de l'hôte pour y déceler des signaux d'intrusions (Déni de Services, Backdoors, chevaux de Troie, tentatives d'accès non autorisés, exécution de codes malicieux, attaques par débordement de buffers ...).

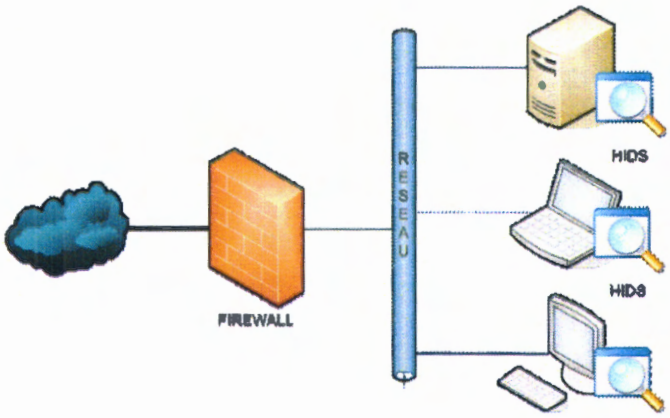

FIGURE 1.6 – Réseau HIDS
[28].

Avantages et inconvénients de HIDS

✓ **Avantages :**

- ☞ Les HIDS peuvent utiliser les évènements et les informations disponibles sur le système d'exploitation hôte ce qui leur apporte des informations claires sur la réussite ou non des attaques.
- ☞ Les HIDS peuvent détecter des attaques qui ne sont pas vues par les NIDS.
- ☞ Les NIDS peuvent fonctionner même dans des environnements cryptés.

✓ **Inconvénients :**

- ☞ Ils sont incapables de détecter les attaques de Déni de service.
- ☞ Ils sont inefficaces contre les attaques qui visent plusieurs hôtes.
- ☞ Il faut installer et configurer un HIDS par poste, ce qui constitue une tâche très couteuse en termes de coût et d'effort.

1.3.4.3 Méthode de détection

Généralement, dans les systèmes de détection d'intrusions, il existe deux types d'approches pour détecter les intrusions : l'approche par signature et l'approche comportementale[29].

1.3.4.3.1 Approche comportementale

L'approche comportementale *(appelée aussi détection d'anomalies)* est une approche de détection qui permet de classifier le comportement du trafic réseau ou de l'utilisateur du système en comportement normal ou anormal, elle permet donc de détecter toutes les activités inhabituelles comparées à celles qui sont préalablement définies.

Cette approche recouvre en fait deux problèmes distincts : la définition du comportement «normal» (souvent appelé profil) d'une part et la spécification des critères permettant d'évaluer le comportement observé par rapport à ce profil d'autre part. D'une façon générale, la détection d'anomalies est composée de deux phases [30] :

- **Phase d'apprentissage :** la détection d'anomalies est précédée d'une phase d'apprentissage qui permet de créer le profil normal de l'utilisateur ou du trafic réseau. Ceci est fait en observant le comportement normal du système pendant une certaine période de temps afin de le prendre comme référence. Par exemple, on fournit les traces collectées sur le réseau d'une banque durant une période donnée. Cette source sera prise ensuite comme une référence et devra donc refléter l'activité normale du réseau de la banque.
- **Une phase de détection :** après la création du profil normal, dans la phase d'apprentissage, le système de détection d'intrusions compare le comportement normal enregistré dans sa base de connaissance (profil normal) avec le comportement actuel du système pour le classer en comportement normal (même comportement que celui rencontré durant la phase d'apprentissage) ou malicieux (dévie du comportement normal rencontré lors de la phase d'apprentissage).

Plusieurs méthodes de modélisation ont été proposées pour établir le profil de l'entité surveillée, les plus utilisées sont : les modèles statistiques, les systèmes experts, les réseaux de neurones, les approches immunologiques[30, 31] .

Avantages et inconvénients de l'approche comportementale

L'approche comportementale possède un certain nombre d'avantages et d'inconvénients :

✓ **Avantages :**

- ☞ Elle est capable de détecter la plupart des attaques, y compris les nouvelles attaques, car elle est basée sur la signalisation de toute éventuelle déviation par rapport au profil normal du système surveillé.

- ☞ Elle permet de détecter facilement toute mauvaise utilisation des privilèges du super utilisateur.
- ☞ Elle permet de produire des informations pertinentes qui peuvent être employées pour définir des signatures pour l'analyse basée connaissance.
- ☞ Les techniques basées sur les anomalies utilisent moins de règles que les techniques basées sur les signatures, ce qui permet d'augmenter ses performances ainsi que son efficacité.

✓ **Inconvénients :**

- ☞ Elle génère souvent de nombreux faux positifs, car une déviation du comportement normal ne correspond pas toujours à l'occurrence d'une attaque.
- ☞ Les alarmes générées par cette approche ne sont pas significatives.
- ☞ Elle nécessite une phase d'apprentissage pour caractériser les profils des comportements normaux. Il faut donc s'assurer que la base d'apprentissage soit exempte d'intrusions. Dans le cas contraire, l'IDS risquerait d'apprendre des comportements intrusifs et ne serait donc pas capable de les détecter ensuite.

1.3.4.3.2 Approche par signature

L'approche par signature *(appelée aussi approche par scénario, approche basée sur la connaissance ou détection d'abus)*, est une approche dont le principe est basé sur l'utilisation d'une base de données ou de connaissances, contenant des spécifications de scénarios d'attaques (on parle de signatures d'attaques et de base de signatures). Le détecteur d'intrusions compare donc les données observées aux signatures de cette base et remonte une alerte si une de ces dernières correspond à une signature prédéfinie.

Avantages et inconvénients de l'approche par signature

Tout comme l'approche comportementale, la présente approche possède pour sa part un certain nombre d'avantages et d'inconvénients [29] :

✓ **Avantages :**

- ☞ Les alarmes générées sont significatives par rapport aux alarmes générées par l'approche comportementale.
- ☞ La précision des diagnostiques qu'elle fournit est plus avancée que celle des diagnostiques fournis par l'approche comportementale.
- ☞ Elle est très efficace pour la détection des attaques connues, avec un taux très bas de fausses alarmes.

✓ **Inconvénients :**

- ☞ Elle est inefficace contre les attaques inconnues, car elle ne permet de détecter que les attaques dont la signature est préalablement enregistrée dans sa base de connaissances. Cette dernière doit donc être constamment mise à jour en lui ajoutant les signatures de toutes les nouvelles attaques.
- ☞ Définir de façon exhaustive les signatures de toutes les attaques possibles est l'une des principales difficultés à laquelle se heurte cette approche.
- ☞ L'attaquant peut facilement influencer sur la détection après la reconnaissance des signatures.

1.3.4.4 Comportement après détection

Une autre façon de classer les systèmes de détection d'intrusions consiste à les classer par type de réponse lorsqu'une attaque est détectée [18, 19] :

1.3.4.4.1 Réponse passive

La réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par la suite par le responsable du système qui va prendre les mesures adéquates pour assurer la sécurité du système. Donc, en cas d'une attaque, l'IDS ne fait qu'informer l'administrateur du système qu'une attaque a eu lieu via un message enregistré dans un fichier log.

1.3.4.4.2 Réponse active

La réponse active d'un IDS a pour but de stopper une attaque, de manière automatique, au moment de sa détection. Pour faire cela, l'IDS peut couper les connexions suspectes ou reconfigurer le pare-feu. 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. 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.

1.3.4.5 Fréquence d'utilisation

Une autre façon de classer les systèmes de détection d'intrusions consiste à les classer selon leur fréquence d'utilisation : périodique ou continue[7, 22] .

1.3.4.5.1 Surveillance périodique

Dans ce type de surveillance, les systèmes de détection d'intrusions analysent périodiquement les fichiers d'audit en cherchant des indications d'intrusions et de mauvais usages du système surveillé.

1.3.4.5.2 Surveillance continue (en temps réel)

Les systèmes à surveillance continue collectent et analysent les informations du système surveillé en temps réel. Le processus de détection se passe ici de manière suffisamment rapide pour entraver les attaques. Les systèmes temps réel fournissent une variété d'alarmes temps réel aussi bien que des ruptures automatiques des attaques.

1.3.5 Efficacité des IDS

Les IDS sont très importants dans une stratégie de sécurité, c'est pour quoi le choix d'un l'IDS est très décisif et doit être basé sur ses caractéristiques, les tâches qu'il devra accomplir, et sur l'architecture du réseau. Selon *DEBAR* [32, 33] on peut déterminer l'efficacité d'un IDS par les mesures suivantes :

- ✓ **Exactitude :** un système de détection d'intrusion efficace ne doit pas identifier une action légale dans un environnement système comme une mauvaise action ou une anomalie. Cette caractéristique correspond généralement à un faux positif.

✓ **Performance :** la performance d'un système de détection d'intrusions est définie comme étant le taux auquel les événements du système sont traités. La haute performance de l'IDS conduit à la détection d'intrusion en temps réel.

✓ **Complétude :** c'est la capacité du système à découvrir toutes les attaques. Incomplétude survient lorsque le système ne parvient pas à détecter une attaque. Pour assurer la complétude, l'IDS doit être toujours à jour envers les nouvelles attaques.

✓ **Tolérance aux pannes :** le système de détection d'intrusion doit résister aux attaques et être capable de gérer les conséquences. surtout en cas des attaques de déni de service.

✓ **Rapidité :** le système de détection d'intrusion doit effectuer l'analyse plus rapidement, afin de permettre au responsable de sécurité de réagir avant que trop de dommages ne soient causés et d'empêcher l'attaquant de subvertir la source d'audit ou le système de détection d'intrusion lui-même .

1.3.6 Limites Des IDS

Comme tout système informatique, les IDS ont des limites. On peut en citer [34] :

➢ **Pollution/surcharge :** les IDS peuvent être pollués ou surchargés, par la génération d'un trafic important qui est très difficile et plus lourd à analyser. Une quantité importante d'attaques peut, par exemple, être envoyée afin de surcharger le système de détection d'intrusions et de perturber le responsable de sécurité en lui générant une très grande quantité d'alertes. Parmi les conséquences possibles de cette surcharge, on trouve, la saturation des ressources matérielles (disque, CPU, mémoire), la perte de paquets, le déni de service partiel ou total du système de détection d'intrusions, etc.

➢ **Consommation des ressources :** outre la taille des fichiers de logs (de l'ordre du Go), la détection d'intrusions est excessivement gourmande en ressources. En effet un système NIDS doit générer des journaux de comptes-rendus d'activités anormales ou douteuses sur le réseau.

➢ **Perte de paquets (limitation des performances)** : les vitesses de transmission sont parfois trop grandes telles qu'elles dépassent largement la vitesse d'écriture des disques durs, ou même la vitesse de traitement des processeurs. Il n'est donc pas rare que des paquets ne soient pas traités par l'IDS, et que certains d'entre eux soient néanmoins reçus par la machine destinataire.

➢ **Vulnérabilité aux dénis de service :** un attaquant peut essayer de provoquer un déni de service au niveau du système de détection d'intrusions, ou pire au niveau du système d'exploitation de la machine supportant l'IDS. Une fois l'IDS désactivé (hors service), l'attaquant peut tenter tout ce qui lui convient.

1.4 Conclusion

Au cours de ce chapitre, nous avons pris connaissance des différents aspects liés à la sécurité informatique, aux attaques qui menacent cette dernière et aux solutions mises en place pour protéger les systèmes (Anti-virus, Cryptographie, Détection d'intrusion, etc.). Nous avons concentré sur les systèmes de détection d'intrusion s, qui analysent le trafic réseau (NIDS) et les activités hôtes (HIDS),

afin d'en extraire des informations leur permettant de détecter des intrusions. Pour détecter une intrusion, ces derniers se basent généralement sur deux approches différentes, à savoir celle à base de signatures et celle à base d'anomalies. La première, comme son nom l'indique, se base sur une base de signatures des attaques connues. Alors que la deuxième consiste à décrire le comportement (profil) usuel d'un utilisateur, et ce, afin de détecter toute action anormale ou inhabituelle de cet utilisateur. Cette dernière permet donc de détecter toutes les attaques dont le comportement dévie du profil défini, y compris les attaques inconnues. C'est l'avantage qui lui a permis d'être l'approche la plus étudiée dans la littérature, notamment avec l'évolution des techniques de data mining et de l'intelligence artificielle. Ainsi, dans notre projet, nous nous somme basés sur ces derniers pour générer quelques modèles différents de détection d'intrusions. Pour cela, nous allons aborder dans le chapitre suivant le domaine du data mining et de l'intelligence artificielle en présentant les différentes étapes du processus de data mining, ainsi que les différentes techniques de classification automatique.

Chapitre 2

Data mining et Apprentissage automatique

2.1 Introduction

L'évolution technologique de ces dernières années a permis aux scientifiques d'élaborer et de perfectionner des méthodes d'extraction d'informations et d'apprentissage automatique très efficaces. L'émergence de ces nouvelles méthodes a permis de faire face au défi posé par le nombre très important de données traitées aujourd'hui par les systèmes informatique. Ces dernières connaissent, de nos jours, un succès croissant et ont prouvé leur efficacité dans plusieurs domaines, y compris le domaine de la sécurité informatique et plus particulièrement celui de la détection d'intrusions. Cette efficacité a été montrée par le nombre très élevé de travaux de recherches publiés lors de ces dernières années dans ce sens. Cette tendance nous a motivé pour faire une étude comparative de ces techniques et leur utilisation dans le domaine de la détection d'intrusions. Mais avant d'entrer aux détails de notre travail, nous devons tout d'abord aborder le domaine de data mining et de machine learning.

Dans ce chapitre, nous présentons, dans la première partie, le principe de data mining, ses définitions, ses étapes principales et ses techniques. Nous passerons ensuite, dans la deuxième partie, à présenter le principe de l'apprentissage automatique, ses définitions, sa classification, etc. et nous finirons par une présentation détaillée des différentes approches de l'apprentissage automatiques avec les avantages et les inconvénients de chacune de ces approches.

2.2 Data Mining

2.2.1 Concepts et Définitions

Le Data mining, appelé aussi forage de données, exploration de données ou fouille de données, est défini comme étant le processus d'extraction de la connaissance à partir d'une grande quantité de données. Il regroupe un ensemble de techniques et de méthodes destinées à l'exploration et à l'analyse de grandes bases de données informatiques en vue de détecter dans ces données des règles, des relations, des Associations et des structures permettant d'en extraire des informations utiles et pertinentes [3] [35]. Le data mining est aujourd'hui utilisé dans de nombreux domaines comme la finance, l'ingénierie,

la biomédecine et le cyber sécurité.

2.2.2 Etapes du processus de data mining

L'extraction de connaissances dans le processus de data mining passe généralement par trois étapes : Etape de préparation de données, étape de modélisation de données et étape d'évaluation du modèle généré. La figure 2.1 représente les étapes du la processus de Data Mining :

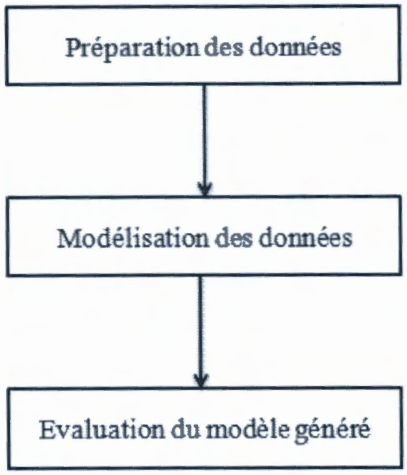

FIGURE 2.1 – Etapes de Processus de Data Mining

2.2.2.1 Etape de préparation de données

Dans cette étape, on applique les différentes techniques et méthodes de prétraitement de données en essayant de comprendre et de préparer les données sources pour qu'elles soient exploitables lors de la phase de modélisation et d'extraction de connaissance. Cette étape consiste donc à effectuer les opérations suivantes [36] :

1. Intégration de données : regrouper et combiner les données provenant des sources multiples et hétérogènes dans une seule base de données.
2. Nettoyage de données : supprimer le bruit ainsi que les données non pertinentes, telles que le code ou bien la clé primaire d'une base de données, le nom et prénom des personnes, les données corrélées (variables qui ont une dépendance fonctionnelle entre elles), etc.
3. Réparation de données : réparer les valeurs manquantes et aberrantes (valeurs extrêmes ou bien isolées), en les supprimant ou en les remplaçant par d'autres valeurs, telles que la valeur moyenne de la variable en question, une valeur prise au hasard dans la distribution des valeurs de la variable en question, etc.
4. Transformation de données : transformer les données au format exigé par le modèle utilisé dans la phase de modélisation et d'extraction de connaissance, cela inclut la numérisation des données, la normalisation des données, etc.

5. Sélection de données : réduire la dimension des données en ne prenant que les données correspondant aux meilleurs attributs.

2.2.2.2 Etape de modélisation et d'extraction de connaissances

La modélisation et l'extraction de connaissances constitue le coeur du processus de data mining. Il s'agit à ce niveau de trouver et d'extraire des règles intéressantes à partir de ces données. Le principe de cette étape consiste donc à appliquer des méthodes et des techniques de machine learning et de l'intelligence artificielle dans le but de générer un modèle qui résume les relations entres les différentes données et qui permet de comprendre les différents phénomènes cachés derrière ces données et d'émettre des prévisions pour les nouvelles données aperçues[39].

La génération du modèle de raisonnement dans cette étape se fait à partir de plusieurs techniques, dont certaines techniques visent à classer (classification, segmentation ou clustering, etc.), alors que d'autres visent à prédire (règles d'association, arbre de décision, etc.). Nous pouvons donc distinguer deux grandes catégories de techniques de modélisation et d'extraction de connaissances[37] :

1. **Les techniques descriptives (technique non supervisées) :** elles produisent des modèles de clustering, qui à partir des valeurs d'un ensemble de variables, ils classent l'objet en cours dans une classe (cluster), les classes sont inconnues à l'avance [37] [38].
   Dans cette catégorie, on trouve trois classes de techniques : Description, regroupement (clustering) et Association.
   - Description : regroupe les techniques qui permettent de décrire les liens entre les différentes variables du concept.
   - Regroupement (clustering) : regroupe les techniques qui permettent de créer des classes de données similaires entre elles et différentes des données d'une autre classe (c'est-à-dire, l'intersection entre les classes doit toujours être vide).
   - Association : regroupe les techniques qui permettent de décrire les liens entre les valeurs des différentes variables du concept en produisant par exemple un modèle de règles d'association.
2. **Les techniques prédictives(technique supervisées) :** elles produisent des modèles de prédiction, qui à partir des valeurs d'un ensemble de variables prédicteurs (valeurs d'entrée), ils prédissent la valeur d'une variable cible ou variable à expliquer (valeur de sortie)[38] [39].
   Tout comme les techniques descriptives, cette catégorie comporte trois classes de techniques : Estimation, Classification et Prévision.
   - Estimation : regroupe les techniques qui permettent de définir le lien entre un ensemble de variables prédicteurs et une variable cible de type numérique.
   - Classification : regroupe les techniques qui permettent de définir le lien entre un ensemble de variables prédicteurs et une variable cible de type catégorielle, le plus souvent booléenne.
   - Prévision : similaire à l'estimation et à la classification sauf que les résultats portent sur le futur.

La figure 2.2 représente les méthodes de Data Mining :

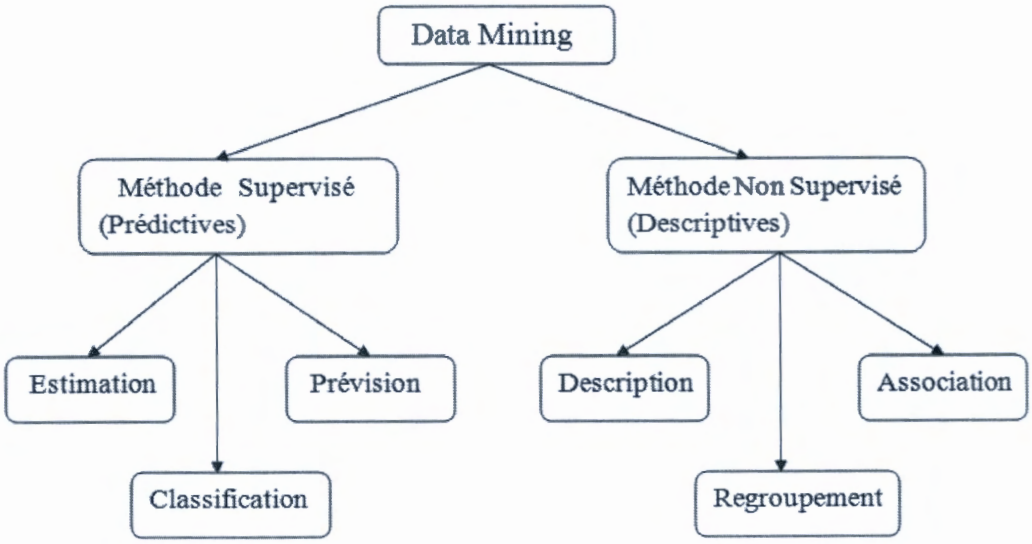

FIGURE 2.2 – Méthodes de data mining

2.2.2.3 Etape d'évaluation du modèle généré

Cette étape consiste simplement à évaluer la qualité du modèle généré en fonction d'un certain nombre de critères, tels que sa performance, sa fiabilité, sa compréhensibilité, sa rapidité de construction et d'utilisation et enfin son évolutivité. En effet, l'efficacité du modèle généré dépend essentiellement du bon choix de la méthode adéquate au problème en question. Il est parfois nécessaire de combiner plusieurs méthodes pour essayer d'avoir un modèle plus performant[40].

Dans l'étape de La modélisation et l'extraction de connaissances est appliquer les méthodes de l'apprentissage automatique , et vont être étudiés en détails dans les parties suivantes de ce chapitre.

2.3 Apprentissage Automatique

2.3.1 Conception et Définition

L'apprentissage automatique *(machine learning en anglais)* est l'un des domaines de l'intelligence artificielle, il est défini comme étant le processus de développement, d'analyse et d'implémentation de techniques et de méthodes qui permettent à une machine d'évoluer et de raisonner comme un être

humain, en essayant de résoudre des problèmes qu'il est difficile ou impossible de les résoudre par des moyens algorithmiques classiques. Il s'agit donc d'un domaine de l'intelligence artificielle dont l'objectif est de chercher des moyens permettant à une machine d'apprendre à s'adapter aux nouvelles situations sans intervention humain. Pour ce faire, l'apprentissage automatique se base généralement sur des outils et des algorithmes qui permettent d'acquérir et d'extraire des connaissances à partir de bases de données d'exemples, qui sont appelées *bases d'apprentissage* ou *jeu d'apprentissage*[60][41].

2.3.2 Domaine d'application de l'apprentissage automatique

L'apprentissage automatique s'applique à un grand nombre d'activités humaines et convient en particulier au problème de la prise de décision automatisée. Les techniques d'apprentissage automatiques sont ainsi utilisées par exemple pour :

- La reconnaissance des formes (texte, son, image, vidéo, etc).
- La mise en place d'outils d'aide à la décision (diagnostique des pannes, etc).
- La fouille de données (extraction de connaissances) .
- Contrôle automatique (déclencher un processus d'alerte en fonction de signaux reçus par des capteurs) .
- etc .

2.3.3 Types d'apprentissage automatique

Les algorithmes d'apprentissage peuvent se catégoriser selon le type d'apprentissage qu'ils emploient, si les classes sont prédéterminées et les exemples sont étiquetés, on parle alors d'apprentissage supervisé. Quand le système ou l'opérateur ne disposent que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature ne sont pas prédéterminés, on parle d'apprentissage non supervisé.

➢ **Apprentissage supervisé :**

L'apprentissage supervisé constitue la technique d'apprentissage la plus populaire et la plus utilisée. Il correspond au cas où l'objectif de l'apprentissage est déterminé explicitement via la définition d'une cible à prédire. Il s'agit donc d'une technique qui permet d'apprendre à partir d'exemples, dont les exemples sont accompagnés d'une information complémentaire relative à leur appartenance ou non au concept. L'apprentissage supervisé se base donc sur le fait qu'il existe déjà une classification de données, c'est-à-dire qu'on dispose d'un ensemble de données déjà classées qu'on appelle *ensemble d'apprentissage* et que l'on utilise comme base pour classer le reste des données. On essaie dans ce type d'apprentissage de collecter le maximum d'informations lors de la phase d'apprentissage, afin de classifier ensuite correctement les nouveaux exemples observés lors de la phase opérationnelle.

Parmi les algorithmes d'apprentissage supervisé les plus connus, on trouve :

- l'algorithme des k plus proches voisins .
- Machine à vecteurs de support (SVM).

- Réseau de neurones.
- Arbre de décision .

➢ **Apprentissage non supervisé :**

Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé (parfois dénommé clustering) correspond au cas où aucune cible n'est prédéterminée. Ainsi, l'ensemble d'apprentissage ne contient aucune information complémentaire relative à leur appartenance ou non au tel ou tel concept. Ce type d'apprentissage consiste donc à inférer des connaissances sur des données d'apprentissage sans savoir a priori à quelles classes ils appartiennent, c'est l'algorithme d'apprentissage qui doit déterminer les différentes sorties en fonction des similarités détectées entres les données d'entrées . On pourrait imaginer que l'algorithme d'apprentissage décide lui-même des classes qui existent et de la classification de chaque exemple.

Parmi les algorithmes d'apprentissage non-supervisé les plus connus, on trouve :

- l'algorithme des k-meams .
- la classification ascendante hiérarchique.

2.3.3.1 Algorithmes d'apprentissage supervisé

2.3.3.1.1 Réseaux de neurones

Définition

Un réseau de neurones est un modèle de classification dont le fonctionnement vise à simuler le fonctionnement des neurones biologiques. Il s'agit d'un des algorithmes les plus utilisés pour la classification, l'estimation, la prédiction et la segmentation des nouvelles observations (sur des variables spécifiques) à partir d'autres observations (soit les même ou d'autres variables) après avoir exécuté un processus d'apprentissage sur des données existantes (données d'apprentissage)[60].

Ce sont des réseaux connectés avec des processeurs élémentaires fonctionnant en parallèle, dont chaque processeur élémentaire calcule une valeur unique en sortie à base des informations entrant[68].

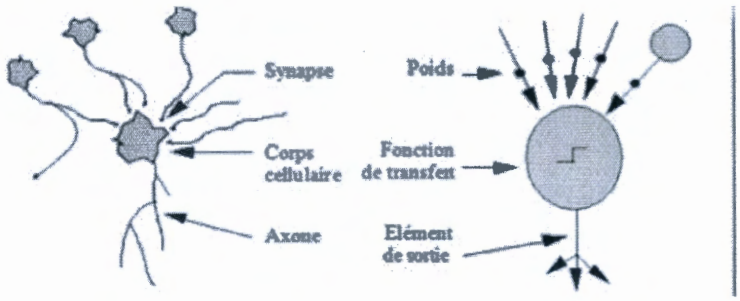

FIGURE 2.3 – Correspondance entre neurone artificiel et neurone biologique [60].

Mise en œuvre

Un réseau neuronal est une association d'un ensemble d'objets élémentaires et de neurones formels en un graphe complexe. Dans un réseau de nuerons artificiel, chaque nœud (neurone formel) reçoit des valeurs en entrée et renvoie une valeur en sortie. La valeur de sortie est calculée en fonction des valeurs d'entrée en utilisant, comme le montre la figure suivante, deux fonctions distinctes. La première fonction est la fonction de combinaison qui calcule une première valeur, appelée valeur d'entrée, à partir des nœuds en entrée et des poids des connexions. Dans les réseaux de neurones à perceptrons, il s'agit de la somme pondérée *($\Sigma nipi$)* des valeurs des nœuds en entrée dont les poids des connexions représentent les valeurs de pondération. La deuxième fonction est la fonction d'activation, appelée aussi fonction de transfert, qui est appliquée sur la valeur d'entrée, fournie par la première fonction, pour calculer la valeur de sortie du neurone[41].

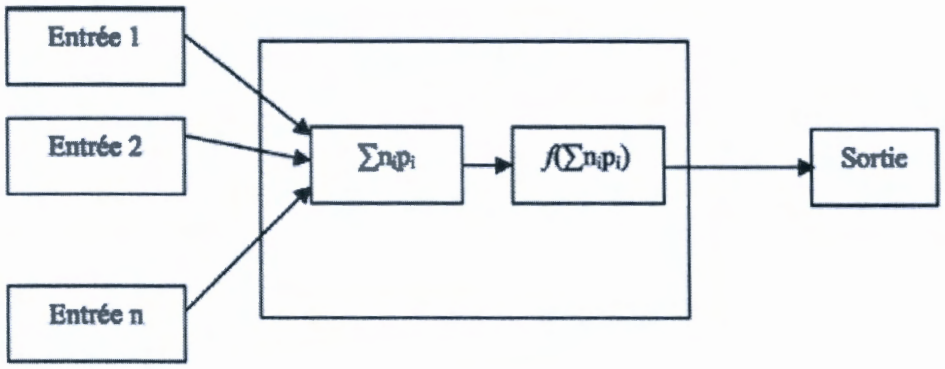

FIGURE 2.4 – Neurone artificiel

Avantages et inconvénients des réseaux de neurones

✓ **Avantages :**

- ☞ Lisibilité du résultat d'apprentissage : le résultat de l'apprentissage est un réseau constitué de cellules organisées selon une architecture bien déterminée.
- ☞ Traitement de problèmes variés et complexes.
- ☞ Bonne performance.

✓ **Inconvénients :**

- ☞ Le temps d'apprentissage peut être long .
- ☞ Effet boîte noire, algorithmes plus difficiles à expliquer [41].

2.3.3.1.2 Arbres de décision

Définition

Un arbre de décision est un outil de classification et d'aide à la décision très puissant qui est à la fois descriptif et prédictif, dont le modèle généré est représenté sous forme d'un arbre facile à comprendre

et à utiliser par un utilisateur humain. Dans l'arbre de décision, le chemin depuis la racine jusqu'à une feuille correspond à une règle de classification. Contrairement aux autres modèles de classification, les arbres de décision sont extrêmement intuitifs et fournissent une représentation graphique permettant de simplifier le processus de prédiction.

Les arbres de décisions sont aujourd'hui utilisés dans de nombreux domaines, tels que la sécurité des systèmes informatique, la fouille de données, la médecine, etc. Leur popularité actuelle est expliquée par leur lisibilité, leur rapidité d'exécution et le nombre très faible d'hypothèses q'ils nécessitent[21].

Mise en œuvre

La construction d'un arbre de décision repose essentiellement sur la partition récursive des objets de l'ensemble d'apprentissage en se basant sur des tests définis à l'aide des attributs, jusqu'à ce que l'on obtienne des feuilles ne contenant que des objets appartenant tous à la même classe. Pour diviser l'ensemble d'apprentissage, on utilise une fonction F qui minimise l'erreur empirique de cette classification locale, jusqu'à ce que l'on obtienne des sous-ensembles plus homogènes que l'ensemble de base, c'est-à-dire ne contenant que des éléments appartenant tous à une même classe [43] [44] [45]. Le pseudo code de l'algorithme est présente dans algorithme 2.1 :

Algorithme 2.1 : Algorithme D'apprentissge générique.

```
1: Entrée : Ensemble de données D
2: Début
3: Initialiser à l'arbre vide ; la racine est le nœud courant
4:         Répéter
5:         Décider si le nœud est terminal
6:                 Si le noued est terminal alors
7:                 Affecter une classe
8:                 Sinon
9:                 Sélectionner un test et créer le sous-arbre
10:                FinSi
11:        Passer au nœud suivant non exploré s'il en existe
12:        Jusqu'aobtenir un arbre de décision
13: Fin
```

Avantages et inconvénients de l'arbre de décision

✓ **Avantages :**

- ☞ Les arbres de décision sont capables de produire des règles compréhensibles .
- ☞ Les arbres de décision effectuent la classification sans exiger beaucoup de calcul.

✓ **Inconvénients :**

- ☞ Manque de performance dans le cas de classification multi-classes .
- ☞ La construction et l'élagage de l'arbre de décision sont trop couteux en termes de temps de calcul et de ressources de stockage [41].

2.3.3.1.3 K-Plus proches voisins

Définition

L'algorithme des K-plus proches voisins *(KNN ou K-nearest neighbor en anglais)* est l'un des algorithmes de classification les plus simples et les plus directes. C'est un algorithme non paramétrique très simple, dont la classification est tout simplement basée sur un simple vote des classes des observations les plus proches. Il ne nécessite donc pas de phase d'apprentissage pour générer un modèle comme c'est souvent le cas dans la plus part des algorithmes de classification. La classification est donc basée directement sur les données de la base d'apprentissage et non pas sur un modèle de classification comme c'est le cas pour les autres algorithmes[47].

Mise en œuvre

Le principe de l'algorithme de K-plus proche voisin est très simple, il se base sur un ensemble de données d'apprentissage, une fonction de distance et une fonction de vote de la classe la plus proche. Ainsi, pour tout nouveau point observé, pour lequel on doit prendre une décision, l'algorithme recherche dans l'ensemble de données d'apprentissage les points les plus proches au point observé, et lui attribue la classe la plus fréquente dans ces voisins. Dans le cas général, prendre plusieurs plus proches voisins, plutôt qu'un unique plus proche voisin permet une certaine robustesse face aux erreurs d'étiquetage[47].

Le principe de l'approche des K-plus proches voisin est résumé dans l'algorithme suivant 2.2 [48] :

Algorithme 2.2 : Algorithme K-PPV

```
1: Déclaration
2: -M : nombre de classes d'apprentissage C =c1,...,cM ;
3: -N : nombre d'exemples d'entrainement E =e1,...,eN ;
4: -Ent = ei , ck : ensemble d'apprentissage formé par les couples ei,ck ;
5: -ex = exemple test ;
6: Début
7:          < On cherche à classer ex ? > ;
8:    Pour chaque exemple ei , w ϵ Ent Faire
9:          < calculer la distance D ei,ex entre ei et ex > ;
10:   FPour
11:   < Trier les échantillons ei par ordre croissant des distances > ;
12:   Pour les K plus proches ei de ex (les K premières - ayant les plus petites-D ei , ex ) Faire
13:         < Compter le nombre d'occurences de chaque classe > ;
14:   FPour
15:    < Attribuer à ex la classe Cj la plus frèquente > ;
16: Fin
```

Avantages et inconvénients de K-Plus proches voisins

✓ **Avantages :**

- ☞ La qualité de la méthode s'améliore en introduisant de nouvelles données d'apprentissage sans nécessiter la reconstruction d'un modèle .
- ☞ La clarté des résultats : la classe attribuée à un objet peut être expliquée en exhibant les plus proches voisins qui ont amené à ce choix.

✓ **Inconvénients :**

- ☞ Temps de classification : la méthode ne nécessite pas d'apprentissage ce qui implique que tous les calculs sont effectués lors de la classification .
- ☞ Méthode donnera de mauvais résultats si le nombre d'attributs pertinents est faible relativement au nombre total d'attributs [41].

2.3.3.1.4 Séparateurs à vast Marge

Definition

Les Séparateurs à Vastes Marges (SVM) *(Support Victor Machines ou machines à vecteurs de support)* sont considérés actuellement parmi les algorithmes de classification les plus efficaces. C'est un algorithme dont la classification est basée sur la construction d'un hyperplan qui sépare les éléments de différentes classes de sorte que les éléments séparés soient le plus loin possibles de l'hyperplan construit. Il est essentiellement utilisé dans le cas où l'espace de représentation des données est linéairement séparable[50].

Mise en œuvre

Le principe des séparateurs à vaste marge est de chercher un hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Pour ce faire, on applique au vecteur d'entrée x une transformation non-linéaire pour décrire les données dans un autre espace (appelé espace de redescription). Puis on cherche alors l'hyperplan séparateur optimal dans l'espace de redescription obtenu [49] [48] .

La figure 2.5 représente un exemple de Séparation de deux ensemble de points par un séparateur linéaire [59].

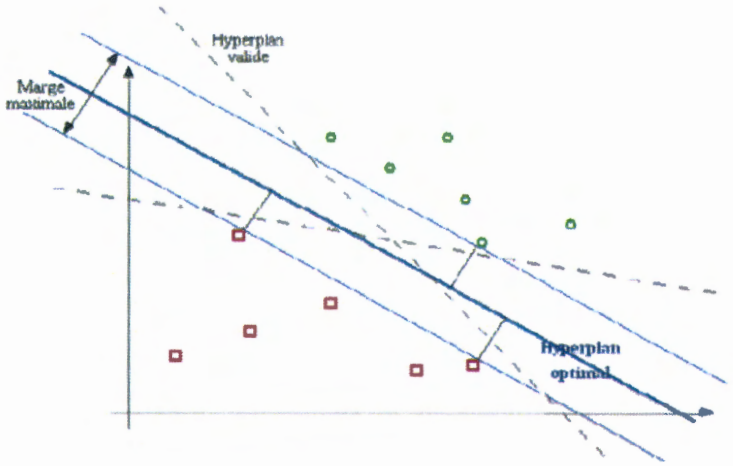

FIGURE 2.5 – Séparation de deux ensemble de points par un séparateur linéaire.

Avantages et inconvénients de les Séparateurs à vast Marge

✓ **Avantages :**

☞ Les points observés lors de la phase de classification sont comparés uniquement avec les supports vecteur et non pas avec tous les exemples d'apprentissage.

☞ Classification rapide. Il suffit de calculer la position du nouveau point observé par rapport au séparateur du modèle.

✓ **Inconvénients :**

☞ Temps de calcul élevé lors d'une régularisation des paramètres de la fonction noyau.

☞ Grande quantité d'exemples en entrées implique un calcul matriciel important [41].

2.3.3.2 Apprentissage non-supervisée

2.3.3.2.1 k-means

L'algorithme k-means, appelé aussi algorithme des centres mobiles , mis au point par McQueen en 1967 [62], est l'un des algorithmes d'apprentissage non supervisé les plus simples. Il s'agit d'un outil de partitionnement des données non hiérarchique qui permet de répartir les données en k clusters homogènes. Son principe de partitionnement consiste à diviser l'ensemble des données en un nombre prédéfini de clusters, appelé k. Pour ce faire, k points de l'ensemble de données sont initialement sélectionnés de façon semi-aléatoirement pour construire K clusters dont les points sélectionnés représentent les centres de gravité (centroides ou centroid en anglais) de ces clusters. Le reste des éléments de l'ensemble de données sont ensuite répartis sur les clusters dont le centre (centroïde) est le plus proche. On recalcule ensuite les centres de gravité des k clusters pour qu'ils se retrouvent au centre des éléments affectés à ces clusters. On répète cette étape jusqu'à ce que l'algorithme converge et que les centres ne se déplacent plus (abouti à un état stationnaire), ici l'algorithme est arrêté [63]. La figure 2.6 représente un exemple de clustering à base de K-Means :

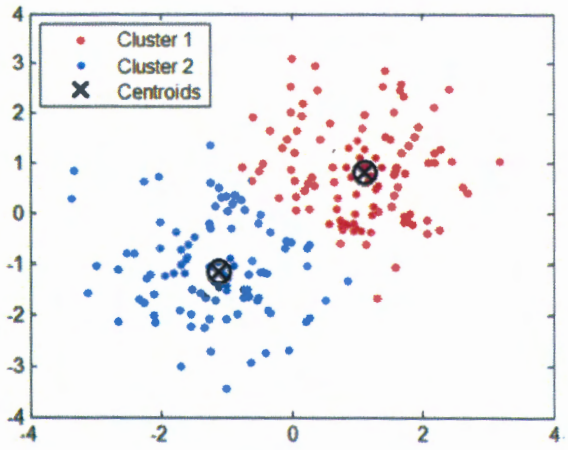

FIGURE 2.6 – Classification à base de K-Means

Les principaux problèmes de l'algorithme k-means, comme tous les algorithmes de clustering par partitionnement des données, sont l'influence de la partition initiale (qui est souvent choisie de façon aléatoire), et le choix du paramètre K qui n'est pas toujours évident.
Le pseudo code de l'algorithme est présente dans algorithme 2.3[64] :

Algorithme 2.3 : Algorithme K-means .

Entrée
    Ensemble de N données, noté par x ;
    Nombre de groupes souhaité, noté par k ;
Sortie
    Une partition de K groupes$\{C_1, C_2 ... C_k\}$
Début
    1) Initialisation aléatoire des centres C_k ;
    Répéter
    2) Affectation : générer une nouvelle partition en assignant chaque objet au groupe dont le centre est le plus proche ;
    $xi \in C_k \ Si \ \forall j \ |xi - \mu_k| = min|xi - \mu_k| .....(1)$
    Avec μ_k le centre de la classes K ;
    3) Représentation : Calculer les centres associe à la nouvelle partition ;
    $\mu_k = \frac{1}{N} \Sigma_{x_i \in C_k} \quad x_i .....(2)$
    Jusqu'à convergence de l'algorithme vers une partition stable ;
Fin

Avantages et inconvénients de K-means

✓ **Avantages :** Les principaux avantages de l'algorithme k-means sont :

- ☞ Est un algorithme très populaire, du fait qu'il est très facile à comprendre et à mettre en œuvre .
- ☞ Il résolve une tâche non supervisée, donc il ne nécessite aucune information sur les données .
- ☞ Il est applicable à tout type de données (mêmes textuelles), en choisissant une bonne notion de distance.

✓ **Inconvénients :**

- ☞ La partition finale dépend de la partition initiale. Le calcul des centroïdes, après chaque affectation d'un individu, influence le résultat de la partition finale. En effet, ce résultat dépend de l'ordre d'affectation des données .
- ☞ Le nombre de classes doit être fixé au départ .
- ☞ Le nombre de classes est un paramètre de l'algorithme. Un bon choix du nombre k est nécessaire, car un mauvais choix de k produira de mauvais résultats.

2.3.3.2.2 Clustering Ascendants Hiérarchique (CAH)

La Classification Ascendante Hiérarchique (CAH) est un algorithme classique de clustering hiérarchique. Elle permet de construire une hiérarchie entière des objets observés sous la forme d'un "arbre" ascendant. Dans cette approche, la classification commence par considérer chaque individu comme une classe avant d'essayer de fusionner les classes appropriées (selon une similarité) deux à deux pour former de nouvelles classes. Le processus s'arrête quand on arrive à une seule classe. Cette

dernière représente la classe racine de la structure arborescente de cet algorithme. Cette classification génère un arbre que l'on peut couper à différents niveaux pour obtenir un nombre de classes plus ou moins grand.
La figure 2.7 représente un exemple simple de cet algorithme.

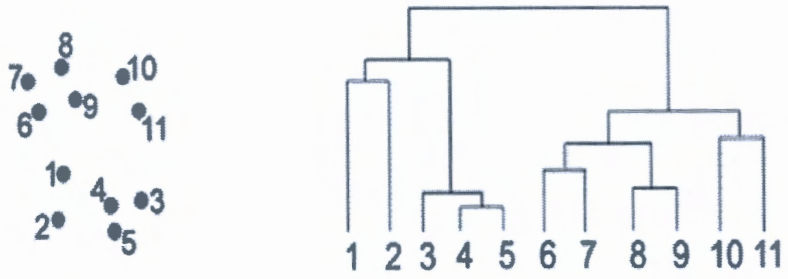

FIGURE 2.7 – Exemple d'application de l'algorithme CHA.
[54]

Comme le montre la figure, il s'agit de regrouper itérativement les individus, en commençant par le bas (les deux plus proches) et en construisant progressivement un arbre, ou dendrogramme, regroupant finalement tous les individus en une seule classe à la racine. Ceci suppose de savoir calculer, à chaque étape ou regroupement, la distance entre un individu et un groupe ainsi que celle entre deux groupes.
Cette procédure est basée sur 2 choix :

1. La détermination d'un critère de ressemblance entre les individus : distance euclidienne, distance de Manhattan, etc .
2. La détermination d'une dissimilarité entre classes (clusters) : procédé appelé généralement *critère d'agrégation*. Il y en a 4 selon la méthode d'agglomération qu'on souhaite appliquer :
   - Le saut minimum : retient le minimum des distances entre individus de C_I et C_J (simple, mais a tendance à tout agglomérer de proche en proche).
   - Le saut maximum : retient le maximum des distances entre individus de C_I et C_J .
   - Le lien moyen : consiste à calculer la moyenne des distances entre les individus de C_I et C_J .
   - La distance de Ward : vise à maximiser l'inertie interclasse (produit des clusters plus compacts)[65].

Le pseudo code de CHA est présente dans algorithme 2.4 :

Algorithme 2.4 : Algorithme de Clustering Ascendante Hiérarchique CHA

1: **Initialisation :** Chaque individu est assimilé à un cluster,
On calcule la matrice de ressemblance M entre chaque couple de clusters.
2: **Répéter**
- On sélectionne dans M les deux clusters les plus proches C_I et C_J selon l'ultra métrique choisie ,
- On fusionne C_I et C_J pour former un cluster C_G, les points de C_I et C_J sont alors assimilés à leur centre ,
- On met à jour M en calculant la ressemblance entre C_G et les clusters existants ,

Répéter Jusqu'à la fusion des 2 derniers clusters .
3: **Fin**

Avantages et inconvénients de Clustering Ascendants Hiérarchique

✓ **Avantages** :

La classification ascendante hiérarchique (CAH) est une méthode de classification qui présente les avantages suivants :

☞ On travaille à partir des dissimilarités entre les objets que l'on veut regrouper. On peut donc choisir un type de dissimilarité adapté au sujet étudié et à la nature des données .

☞ L'un des résultats est le dendrogramme, qui permet de visualiser le regroupement progressif des données. On peut alors se faire une idée d'un nombre adéquat de classes dans lesquelles les données peuvent être regroupées .

☞ Il dépend de la distance de dissimilarité entre des individus et la distance de dissimilarité entre classes choisie.

✓ **Inconvénients** :

Les méthodes HAC représentent certaines faiblesses, comme par exemple :

☞ Le temps énorme consacré à la classification .

☞ L'utilisation de plusieurs types de métriques, pour mesurer la distance entre deux clusters, peut générer des résultats différents .

☞ Il est difficile de savoir à quelle hauteur couper le dendrogramme pour déterminer les clusters.

Enfin, nous terminons cette partie par une petite comparaison entre l'apprentissage supervisé et l'apprentissage non supervisé , qui ont résumée dans le tableau suivant :

Apprentissage Supervisé	Apprentissage Non Supervisé
-L'extraction des données est prédictive.	-L'extraction des données est descriptive.
-Les données d'apprentissages sont accompagnés par les labels indiquant leurs classes (observation).	-Les éléments données ne sont pas étiquettes.
-Les nouvelles données sont classifiées en se basant sur le training set.	-Il sont utilisé pour comprendre et explorée les données.
- Le nombre de classes est connu au préalable.	-Le nombre de classe est inconnu au préalable.

TABLE 2.1 – Différence entre l'apprentissage Supervisé et l'apprentissage non Supervisé.

2.4 Conclusion

Dans ce chapitre, nous avons présenté le principe de data mining et d'apprentissage automatique, en spécifiant les différents types d'apprentissage. Nous nous sommes focalisés sur les deux modes d'apprentissage automatique les plus répandus (supervisé et non supervisé). Nous avons décrit les classifieurs les plus utilisés dans ces deux modes, tels que les réseaux de nuerons, les arbres de décision, les K-Plus Proche voisin et les Séparateur à Vaste Marge pour l'apprentissage supervisé ; Et le K-means et la classification hiérarchique ascendante pour l'apprentissage non supervisé.

Chapitre 3

Expérimentations et résultats

3.1 Introduction

Dans ce chapitre, nous présentons l'implémentation de notre travail, en commençant par une description de la base NSL-KDD, une présentation du processus de génération des modèles de classification, les étapes de prétraitement que nous avons fait afin de préparer les données de la base NSL-KDD, etc. Nous discutons ensuite les résultats obtenus en comparant les modèles générés (KNN, arbres décision, réseaux de neurones et SVM), dont le fonctionnement est basé sur l'analyse du comportement des connexions TCP/IP de la base NSL-KDD, un fonctionnement qui permet de classifier ces connexions en deux classes (attaque et normale). Cette comparaison est effectuée en fonction des différentes métriques de performance, telles que le taux de réussite, le taux de détection, la précision et le taux de fausses alarmes.

3.2 Présentation de la base NSL-KDD

3.2.1 Historique

L'ensemble de données KDD'99 est une base de données contient des connexions TCP/IP et extraites de l'ensemble de données d'évaluation de systèmes de détection d'intrusions DARPA'98 (qui a provenu de laboratoire "MIT's Lincoln Lab").

Les chercheurs ont procédé à une analyse statistique sur KDD'99 et ont trouvé des problèmes importants qui affectent fortement les performances des systèmes évalués, et les résultats. Pour résoudre ces problèmes et à une suite des efforts de recherche et des améliorations sur KDD'99 dataset, ils ont proposé un nouvel ensemble de données, NSL-KDD, qui a été dérivée en 2010 du KDD'99 [3] .

L'ensemble de données NSL-KDD présente les avantages suivants par rapport à l'ensemble original KDD'99 [66] :

- Il n'inclut pas les enregistrements redondants dans les données d'apprentissage, pour que les classifieurs soient plus performants.
- Il n'y a pas des enregistrements redondants dans les ensembles de données de test proposées, ce qui conduit à obtenir des bons résultats.

- Le nombre d'enregistrements des données d'apprentissage et de test est raisonnable, ce qui rend abordable d'exécuter les expérimentations sur l'intégralité des données sans la nécessité de choisir au hasard une petite. Par conséquent, les résultats de l'évaluation des différents travaux de recherche seront cohérents et comparables.

3.2.2 Description de la base NSL-KDD

NSL-KDD contient 4 échantillons de données qui sont : KDDTrain+, KDDTrain+_20Percent, KDDTest+, KDDTest-21[66].

- **KDDTrain+** :représente toutes les donnes d'apprentissage du NSL-KDD.
- **KDDTrain+_20Percent** :représente 20% des données d'apprentissage du NSL-KDD.
- **KDDTest+** :représente toutes les données du test du NSL-KDD.
- **KDDTest-21** :représente toutes les données du test du NSL-KDD qui ne contient pas les enregistrements avec le niveau de difficulté de 21 sur 21.

Ces échantillons contiennent des enregistrements de connexion TCP/IP, dont chaque enregistrement est constitué de 41 attributs caractérisant la connexion, et un attribut étiquetant la nature de la connexion, ce dernier prend le nom du type d'attaque en question s'il s'agit d'une attaque ou la valeur «normal» s'il ne s'agit pas d'une attaque.

3.2.2.1 Classes d'attaques

Les principales classes d'attaques de la base NSL- KDD99 sont [45] :

3.2.2.1.1 Attaques de Dénis de Services

Le DOS est un type d'attaque visant à rendre indisponible pendant un temps indéterminé les services d'un système. Il vise à occuper par exemple des ressources, telles que la mémoire, par des fausses requêtes. Il existe plusieurs types de déni de services, d'une part ceux qui exploitent les bugs d'une application et d'autre part ceux qui exploitent une mauvaise implémentation d'un protocole ou des faiblesses de celui-ci.

3.2.2.1.2 Attaques de type 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.

3.2.2.1.3 Attaques User to Root

L'objectif de cette classe d'attaques est d'obtenir les privilèges de l'administrateur système (Root) à partir d'un simple compte utilisateur. Pour avoir ces privilèges, l'attaquant essaye d'exploiter des vulnérabilités présentes dans le système, telles que les débordements des Buffers (buffer overflow).

3.2.2.1.4 Probing (Sondage)

Dans cette classe d'attaque, l'attaquant essaye de récolter des informations sur la machine cible, telles que le système d'exploitation installé, les services offerts (les ports ouverts), la topologie du réseau sur lequel la machine est installée, etc. Pour collecter ces informations, l'attaquant utilise plusieurs techniques, telles que le scan des ports de la machine cible, le scan des machines actives dans le réseau cible, le sniffing du trafic du système cible, etc.

La base NSL-KDD99 comporte 38 types d'attaque différents, le regroupement de ces types d'attaques sur les quatre classes d'attaques est donné dans le tableau suivant :

La classe	Types D'attaques
Dos	Back , Land ,Neptune , Pod , Smurf , Teardrop , Apache2 , Udpstorm , Processtable , Worm .
Probe	Satan , Ipsweep , Nmap , Portsweep , Mscan , Saint .
R2L	Guess_Password , Ftp_write , Imap , Phf , Multihop , Warezmaster , Warezclient , Spy , Xlock , Xsnoop , Snmpguess , Snmpgetattack , Httptunnel , Sendmail , Named .
U2R	Buffer_overflow , Loadmodule , Rootkit , Perl , Sqlattack , Xterm , Ps .

TABLE 3.1 – Types d'attaques dans la base NSL- KDD.

La distribution des connexions réseau du trafic normal et des quatre classes d'attaques dans la base KDDTest+, et KDDTrain_20%, utilisées dans notre projet, est donnée dans le tableau suivant [67] :

Catégorie	Nombre d'enregistement en KDDTrain 20%		Nombre d'enregistement 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 totale des enregistrements	25191		22544	

TABLE 3.2 – Distribution des connexions réseau de KDDTest+, et KDDTrain_20%.

3.2.2.2 Attributs

Chaque enregistrement de la base NSL-KDD est constitué de 41 attributs.Ces attributs peuvent être classés en trois groupes :

- **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 du trafic tels que nombre d'échec de connexion et le nombre d'accès aux fichiers de contrôle.

Le tableau (1) de l'annexe récapitule les 41 attributs de la base NSL-KDD.

3.3 Processus de génération des modèles de classification

Pour toute modèle de classification des connexions TCP/IP pour la détection d'intrusion , doit passer par trois phases principales : la phase de prétraitement, la phase d'apprentissage et la phase de test. Ces dernières peuvent être résumées dans le graphe 3.1[45] :

3.3.1 Prétraitement des données

Le prétraitement des données est la phase la plus importante dans le processus de génération des modèles de classification. En effet, la qualité des modèles qui nous avons implémenter sont fortement liée aux opérations effectuées dans cette phase. Dans ce travail, quatre opérations de prétraitement ont été réalisées sur la base NSLKDD : le nettoyage des données, la numérisation des données, la normalisation des données et la sélection des meilleurs attributs.

3.3.1.1 Nettoyage des données

Dans cette opération, nous avons supprimé tous les attributs constants qui ont une seule valeur constante pour tous les enregistrements de la base, car ces derniers n'ont aucune valeur ajoutée au les modèles . Pour la base NSL-KDD, il y a un seul attribut constant « num_outbound_cmds ».

3.3.1.2 Numérisation des données

En effet, la base NSL-KDD est constituée d'un grand nombre d'enregistrement, dont chaque enregistrement est formé de 41 attributs, les valeurs de ces derniers sont de différentes natures, certains prennent des valeurs numériques alors que d'autres prennent des valeurs alphabétiques ou symboliques, il est donc très important de convertir les attributs symboliques en numériques, car les modèles des qui nous avons implémenter n'acceptent que des attributs numériques. Dans cette opération, chaque valeur symbolique est remplacée par son entier équivalent, par exemple : les valeurs des attributs symboliques « protocol_type » (3 valeurs symboliques différentes), « services » (70 valeurs symboliques différentes) et « flag » (11 valeurs symboliques différentes) sont remplacées par des valeurs entières de 0 à N-1 où N est égal au nombre de valeurs symboles de l'attribut.

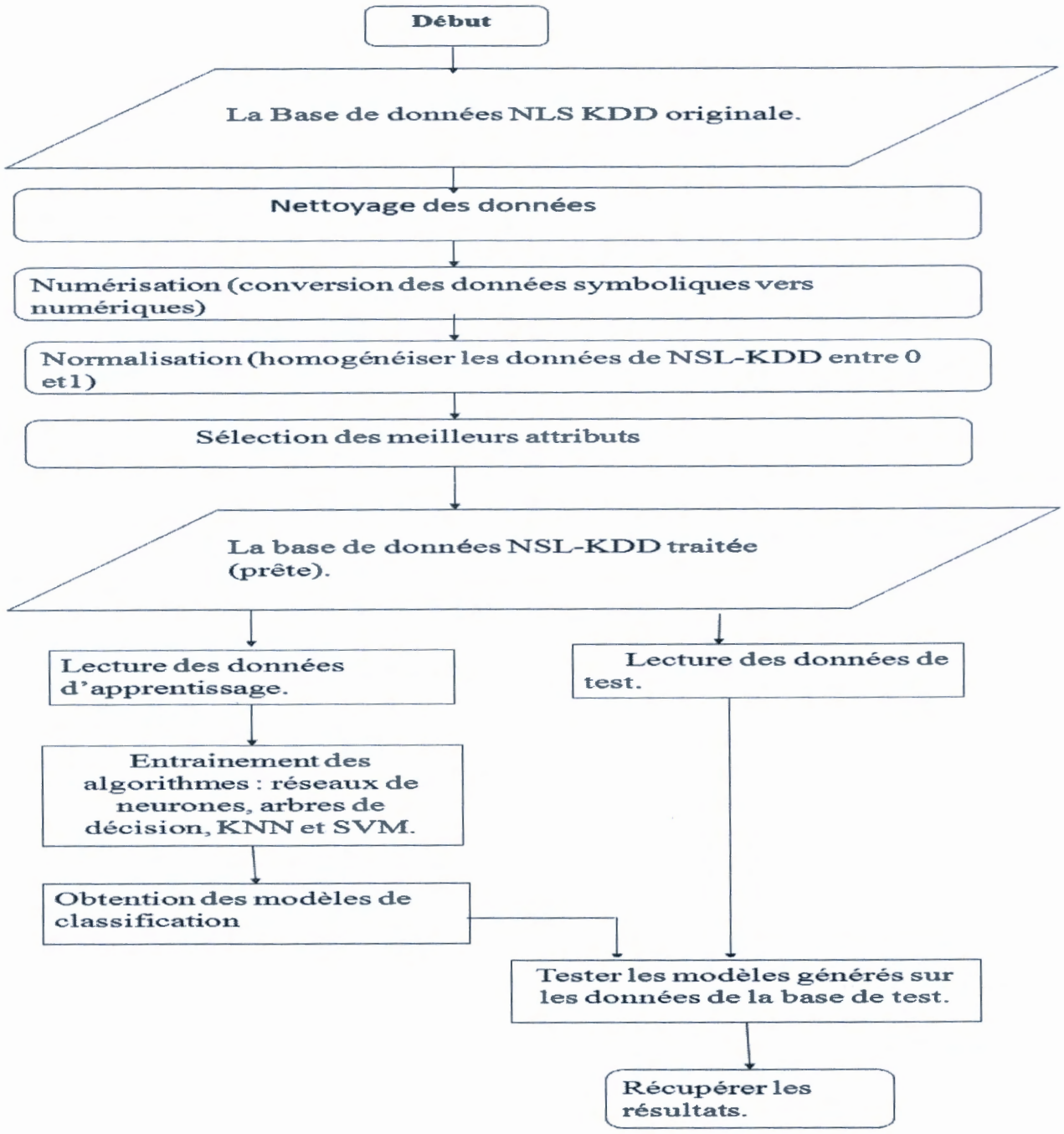

FIGURE 3.1 – Organigramme de fonctionnement de modèle de détection d'intrusion

3.3.1.3 Normalisation des données

En fait, les valeurs numériques des attributs de la base NSL-KDD sont très variées. Par exemple, certains attributs prennent des grandes valeurs (src_bytes, dst_bytes, etc.), alors que d'autres ne prennent que des petites valeurs (serror_rate, same_srvrate, etc.). L'utilisation de ces valeurs, telles qu'elles sont, peut affecter considérablement les performances des modèles. Il est donc très important d'ajuster toutes les valeurs de la base pour que les modèles soit cohérent. La normalisation des données est généralement effectuée suivant une fonction de transformation. Dans notre cas, la fonction utilisée est la fonction Min-Max qui est donnée par la formule suivante :

$$val_{nouv} = \frac{val_{anc}+Min_{anc}}{Max_{anc}-Min_{anc}} *(Max_{nouv} - Min_{nouv}) + Min_{nouv}$$

Où :

- val_{anc} : est la valeur à normaliser.
- val_{nouv} : est la valeur après la normalisation.
- Min_{anc} : est la limite inférieure de l'intervalle à qui val_{anc} appartient.
- Max_{anc} : est la limite supérieure de l'intervalle à que val_{anc} appartient.
- Min_{nouv} : est la limite inférieure de l'intervalle à que val_{nouv} va appartenir.
- Max_{nouv} : est la limite supérieure de l'intervalle à que val_{nouv} va appartenir.

Les données de notre base sont normalisées entre 0.0 et 1.0.

3.3.1.4 Sélection des meilleurs attributs

Pour les données de grandes dimensions, telles que celles de la base NSL-KDD (41 attributs), l'utilisation de la totalité des attributs pour générer le modèle de classification peut affecter considérablement les performances de ce dernier, il faut donc choisir les meilleurs attributs pour que le modèle généré soit préformant. La sélection des attributs constitue donc une étape importante dans le prétraitement des données de grandes dimensions, c'est un processus qui consiste à chercher dans l'ensemble des attributs de la base un sous-ensemble optimal des attributs les plus importants au système en question. Plusieurs techniques et approches ont été proposées pour réaliser cette opération. Dans notre travail, nous avons poursuit une approche qui consiste à choisir les attributs avec les meilleurs gains. Cette approche est résumée dans les quatre étapes suivantes :

Étape 1 : Calcul de l'entropie pour chaque attribut.

$$H(x_i) = -\sum_{j=1}^{n} p(x_j|c_1)\log_2 p(x_j|c_1) + p(x_j|c_2)\log_2 p(x_j|c_2)$$

Où :

- c_1,c_2 : dénotent les deux classes de classification (normale, attaque).
- x_i : représente un attribut.

— x_j : représente une valeur particulière de l'attribut .

— n : dénote le nombre de valeurs de l'attribut .

— p : la probabilité.

Étape 2 : Calcul de gain pour chaque attribut.

$$Gain = Entropie_E - H(x_i)$$

Où :

$$Entropie_E = \frac{nb_{normale}}{nb_{connexion}} \log_2(\frac{nb_{normale}}{nb_{connexion}}) + \frac{nb_{attack}}{nb_{connexion}} \log_2(\frac{nb_{attack}}{nb_{connexion}})$$

— $nb_{normale}$: présente le nombre des connexions qui sont classifiées comme normal .

— nb_{attack} : présente le nombre des connexions qui sont classifiées comme une tentative d'attaque.

— $nb_{connexion}$: présente le nombre total des connexions dans la base d'apprentissage .

Étape 3 :Élimination des attributs ayant un gain inférieur à un seuil donné.

Étape 4 : Utilisation du sous ensemble d'attributs restant (les attributs ayant un gain supérieur ou égal au seuil donné) pour générer le système de classification.

3.3.2 Test et évaluation du modèle généré

Lors de la phase de test, les modèles de détection d'intrusion générés sont testé en utilisant, dans notre cas, une base de test contenant 22544 enregistrements. Lors de cette phase, quelques métriques d'évaluation sont calculées, ces dernières sont empruntées du domaine de l'évaluation des classificateurs. Il s'agit notamment de la matrice de confusion, la précision, le rappel, le taux de réussite, le taux des fausses alertes, et le F-mesure. Nous avons retenu ces métriques qui sont nécessaires à l'analyse et à l'explication des résultats obtenus lors des différentes expérimentations sur NSL-KDD. Ces métriques sont définies comme suit :

- **La matrice de confusion** :est un tableau bidimensionnel où les lignes représentent le nombre d'occurrence des classes réelles de la base, tandis que les colonnes représentent le nombre d'occurrence des classes prédites par le système de détection [48].

		Classe dètectèe (prèdite)	
		Normale	*Attaque*
Classe rèelle	*Normale*	Vrai Nègatif TN (True Negative)	Faux positif FP (False Positive)
	Attaque	Faux Nègatif FN (False Negative)	Vrai positif FP (True Positive)

TABLE 3.3 – Matrice de confusion.

De cette matrice de confusion, il ressort que :

— Un vrai négatif TN (True Negative) : est une activité normale considérée en tant que telle.

- Un faux positif FP (False Positive) : est une activité normale considérée comme une attaque.
- Un faux négatif FN (False Negative) : est une attaque ratée, c'est-à-dire, une attaque considérée comme une activité normale.
- Un vrai positif TP (True Positive) : est une attaque correctement détectée.

- **L'exactitude (Accuracy) ou le taux de réussite** :c'est le rapport entre le nombre d'enregistrements de test correctement classés et le nombre total d'enregistrements de test.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100$$

- **Le rappel** :c'est une métrique qui traduit le taux des intrusions correctement détectées par rapport au nombre total d'intrusions existantes (TP+FN).

$$Rappel = \frac{TP}{(TP+FN)} * 100$$

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

$$Précision = \frac{TP}{(TP+FP))} * 100$$

- **Le taux des fausses alertes (taux des faux positifs)** : correspond au nombre de faux positifs par rapports au nombre total de connexions normales.

$$Taux\ des\ faux\ positifs = \frac{FP}{(FP+TN))} * 100$$

- **F-mesure (F-measure ou F-score en anglais)** : c'est une mesure populaire qui combine la précision et le rappel, elle donne une évaluation de synthèse de la classification.

$$F - Mesure = \frac{(1+\beta^2)*(Précision*Rappel)}{\beta^2*(Précision+Rappel)}$$

Où β_2 est généralement mis à 1.

3.4 Implémentation et analyse des résultats

3.4.1 Environnement de programmation

Pour implémenter les modèles d'apprentissage à comparer, nous avons utilisé le langage Java (l'environnement NetBeans 7.2.1) pour réaliser les différents prétraitements nécessaires pour préparer les données de la base NSL-KDD, et le langage Matlab (Matleb 2009) pour générer, tester et comparer ces modèles. Le choix du langage matlab a été guidé par les avantages que ce dernier offre aux programmeurs, il fournit tout un ensemble d'algorithmes d'apprentissage sous formes de fonctions prédéfinies. Il est très utilisé dans le domaine de l'apprentissage automatique, facile à manipuler et compatible avec le format de données que nous avons travaillé avec. Le choix de Java et de Netbeans est fondamental puisqu'il est standard, de bibliothèques de classes très riches comprenant la gestion des exceptions, la variété des types, les collections d'objets (de taille fixe comme tableaux, de taille variable comme

arraylist), les accès aux fichiers, etc. De plus cet IDE est extensible et disponible gratuitement.

Les implémentations ainsi que les expérimentations sont effectuées sur une machine dont les caractéristiques sont résumées dans le tableau suivant :

Composants	Valeurs
Processeur	AMD A8-4500M APU
Vitesse	1.90 GHz
Mémoire	4.00 Go
Système d'exploitation	Windows 732 bits

TABLE 3.4 – Spécifications techniques de l'ordinateur utilisé pour les expérimentations.

3.4.2 Démarche suivie pour réaliser nos expérimentations

Dans cette section, nous nous intéressons à la démarche suivie pour tester, analyser et comparer les différents algorithmes de classification supervisé, tels que : les K-plus proches voisins (kNN) , les arbres de décision , les réseaux neurones (MPL) et les séparateurs à vaste marge(SVM).

L'objectif est donc de comparer leurs performances sur la base NSL-KDD en fonction de certains métriques, telles que l'exactitude, la précision, etc. Les résultats obtenus sont ensuite analysés et interprétés.

Dans notre étude expérimentale et avant de lancer le processus d'apprentissage des différents algorithmes de classification, nous avons tout d'abord lancé l'algorithme de sélection des meilleurs attributs, dans lequel la sélection est répétée plusieurs fois en modifiant à chaque fois la valeur du seuil du gain d'information, ce qui nous a permis d'avoir plusieurs ensembles de meilleurs attributs en fonction des valeurs de seuil introduites. Les valeurs de seuil introduites dans nos expérimentations sont : 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 et 0.9.

Après avoir sélectionné les meilleurs attributs pour les différentes valeurs de seuil, nous avons lancé le processus d'apprentissage et celui de test pour chacun des algorithmes de classification (KNN, DTree, MLP et SVM). Ces processus sont répétés autant de fois que le nombre d'ensembles de meilleurs attributs.

3.4.2.1 Résultats obtenus et discussion

3.4.2.1.1 Résultats obtenus en fonction d'exactitude (Accuracy)

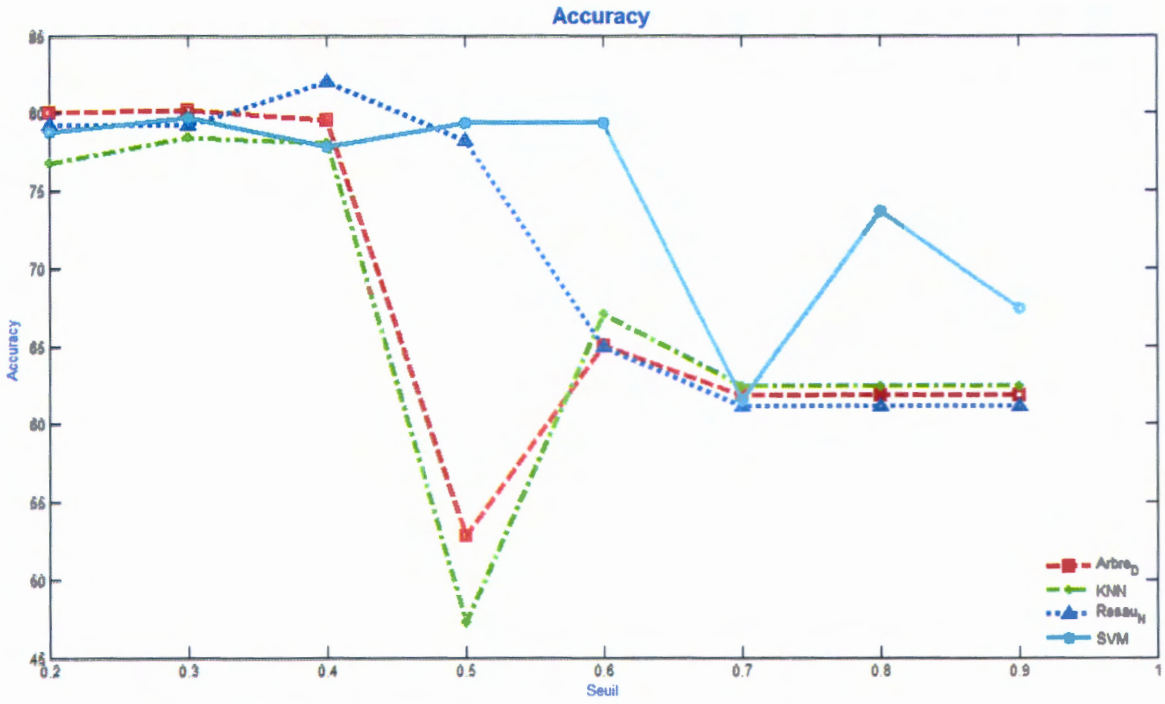

FIGURE 3.2 – Accuracy.

Les résultats présentés ci-dessus montrent clairement que la performance des modèles implémentés, en terme d'exactitude, est acceptable. Ils détectent les intrusions avec un taux de réussite (Accuracy) varie entre 45% et 85%. Comme le montre la figure, les meilleurs résultats sont donnés lorsque le seuil de gain d'information est choisi entre 0.2 et 0.4. Dans cet intervalle, le taux de réussite de tous les algorithmes est supérieur à 75%. Le meilleur résultat est donné par l'algorithme MLP (réseaux de neurones) avec un taux de réussite de 82.13%, suivi par les algorithmes DTree (arbres de décisions) avec un taux de réussite de 80.28%, SVM avec un taux de 79.81% et K-NN avec un taux de 78.48%.

Pour les autres valeurs des seuils (entre 0.5 et 0.9), et à partir du seuil 0.5, les résultats commencent à se dégrader pour les quatre algorithmes de classification. Ceci est dû au fait que l'augmentation du seuil est accompagnée par une diminution du nombre d'attributs sélectionné, ce qui conduit à une dégradation des performances de ces derniers.

3.4.2.1.2 Résultats obtenus en fonction de précision

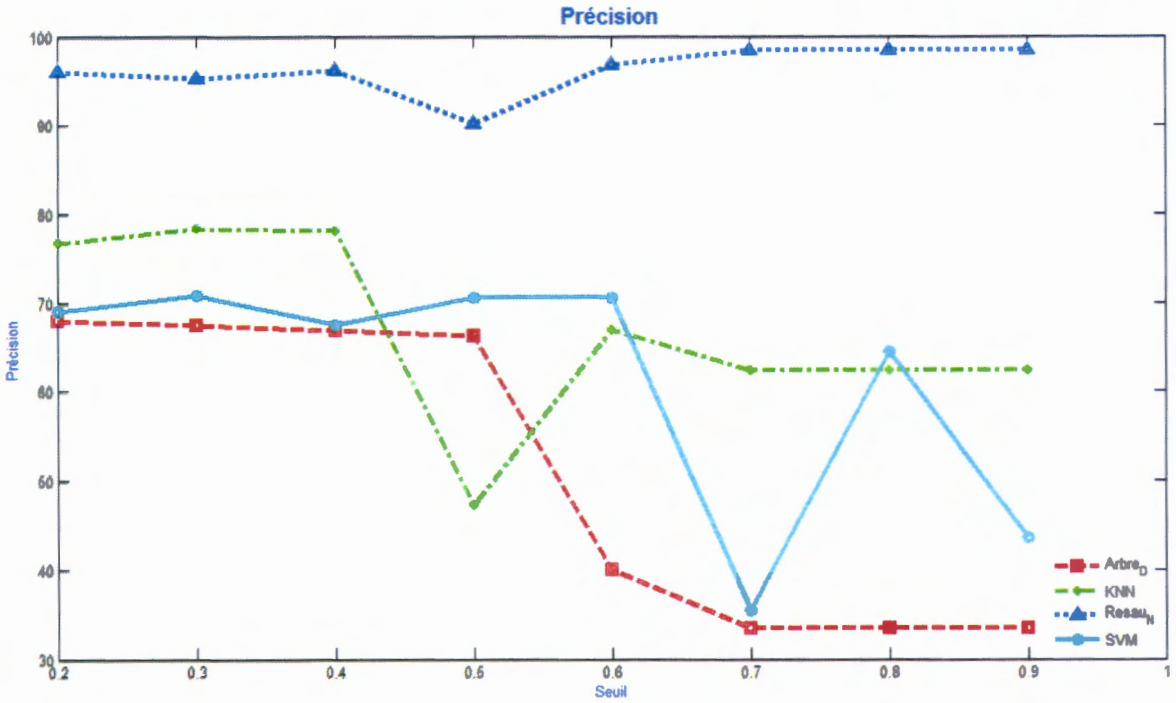

FIGURE 3.3 – Précision.

Les résultats de cette figure montrent clairement la supériorité des réseaux de neurones sur les autres algorithmes en terme de précision, où on voit que la précision de cet algorithme est toujours supérieure à 90% quelque soit la valeur du seuil de gain d'information, alors que la précision des autres algorithmes est toujours inférieure à 80% quelque soit la valeur du seuil.

La même remarque pour les valeurs des seuils qui donnent de meilleurs résultats. Nous remarquons que les valeurs entre 0.2 et 0.4 sont les valeurs qui donnent les meilleures précisions pour tous les algorithmes. Dans cet intervalle, la précision est presque stable pour les quatre algorithmes. Cependant à partir du seuil 0.5, la précision commence à se dégrader, surtout pour les algorithmes SVM, DTree et KNN.

3.4.2.1.3 Résultats obtenus en fonction de rappel

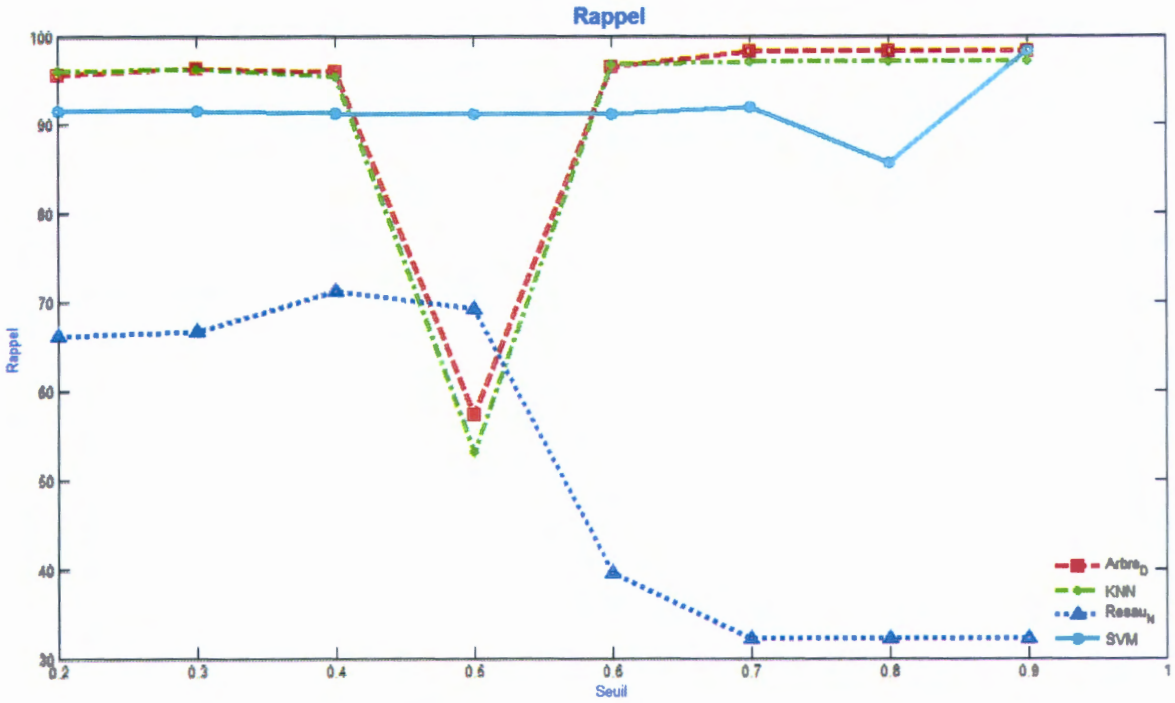

FIGURE 3.4 – Rappel.

Contrairement aux résultats précédents qui nous ont montré la supériorité des réseaux de neurones sur les autres algorithmes, les résultats de cette figure montre une infériorité de ces derniers sur les autres algorithmes. Nous voyons que le rappel des réseaux de neurones est toujours inférieurs à 75% quelque soit la valeur du seuil, contrairement aux autres algorithmes où le rappel est toujours supérieur à 90% quelque soit la valeur du seuil.

3.4.2.1.4 Résultats obtenus en fonction du taux de fausses alarmes

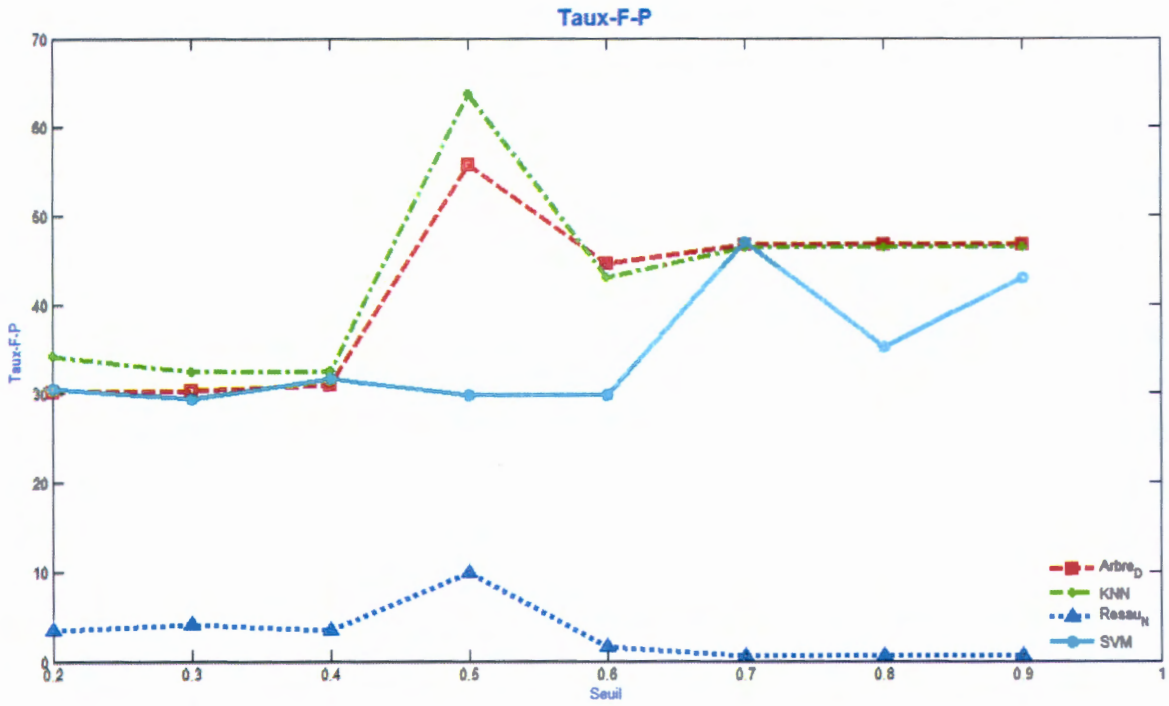

FIGURE 3.5 – Taux de faux positif.

Comme le montre cette figure, les réseaux de neurones surpassent largement leurs homologues en terme du taux de fausses alarmes qu'ils génèrent. Le taux de fausses alarmes générées par les réseaux de neurones est toujours inférieur à 10% quelques soit la valeur du seuil, il est même inférieur à 5% pour la majorité des valeurs des seuils. Par contre, le taux de fausses alarmes générées par les autres algorithmes est toujours supérieur à 30% quelque soit la valeur du seuil, ce qui rend ces derniers beaucoup plus moins performants que les réseaux de neurones.

3.4.2.1.5 Résultats obtenus en fonction de F-mesure

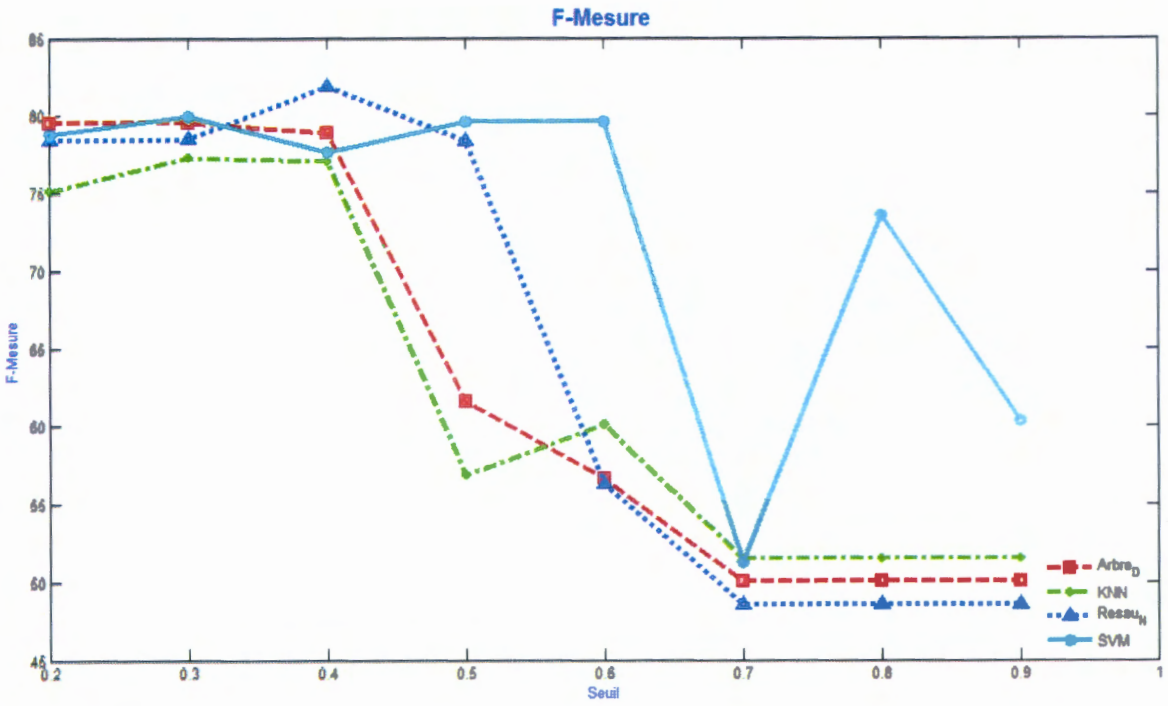

FIGURE 3.6 – F-Mesure.

En effet, la comparaison des quatre algorithmes en utilisant uniquement les métriques précédentes n'est pas toujours évidente. Car, comme nous avons observé, les réseaux de neurones surpassent leurs homologues en termes de précision et de taux de fausses alarmes, alors que ces derniers surpassent largement les premiers en terme de rappel. Dans une telle situation, c'est le F-Score qui va montrer lequel de ces algorithmes est meilleur que les autres. Car il regroupe les métriques (précision et rappel) dans une seule métrique.

Comme le montre cette figure, l'algorithme qui donne de meilleurs résultats est celui des réseaux de neurones avec un F-Score de 81.96% dans le seuil 0.4, suivi par l'algorithme SVM avec un F-Score de 80% dans le seuil 0.2, DTree (arbres de décisions) avec un F-Score de 79% dans le seuil 0.2 et KNN avec un F-Score de 77% dans le seuil 0.2.

3.5 Conclusion

Nous avons abordé dans ce dernier chapitre nos études expérimentales qui ont été basées sur les données de la base NSL-KDD. Les algorithmes de classification étudiés dans ce chapitre sont : les réseaux de neurones, les K-plus proches voisins (KNN), les arbres de décision et les supports à vaste marge (SVM). Nous avons également présenté les différents prétraitements réalisés pour préparer les données de la base NSL-KDD, la démarche suivie pour réaliser nos expérimentations, ainsi que les différentes métriques utilisées pour mesurer la performance des modèles générés.

Enfin, et à travers les résultats obtenus, nous avons conclu que les réseaux de neurones donnent de meilleurs résultats par rapports aux autres approches de classifications étudiées. Nous avons également montré à travers cette étude expérimentale l'importance des algorithmes de sélection d'attributs et leur influence sur la performance globale des différentes approches de classification.

Conclusion générale

Le travail réalisé dans ce mémoire rentre dans le cadre des travaux qui s'intéressent à la détection d'intrusions à base des techniques de data mining et de machine learing (apprentissage automatique).

Nous avons essayé de présenter d'une façon succincte les notions de base qui sont nécessaires pour comprendre le travail réalisé dans ce mémoire, que ce soit les notions relatives aux systèmes de détection d'intrusions ou celles relatives aux techniques de data mining et de machine learning. Pour atteindre cet objectif, nous avons organisé le mémoire en trois chapitres, chacun traite une partie de ce sujet.

Le premier chapitre résume les différentes solutions proposées dans le domaine de la sécurité informatique, telles que la cryptographie, les pare-feux, les scanners de vulnérabilité et plus particulièrement les systèmes de détection d'intrusions, qui sont généralement classifiés en deux classes différentes : comportementaux et par signatures, selon la technique utilisée pour détecter les intrusions. Dans ce travail, nous nous sommes basés sur les systèmes de la première classe, où nous avons utilisé les différentes techniques de data mining et d'apprentissage automatique. Ces dernières ont été présentées en détail dans le deuxième chapitre de ce mémoire.

Dans le dernier chapitre de ce mémoire, nous avons effectué une étude expérimentale comparative entre les différents modèles de classification, à savoir les réseaux de neurones, les arbres de décision, les k-plus proches voisin et les supports à vaste marge. Dans cette étude expérimentale et afin de montre l'importance de la sélection des meilleurs attributs ainsi que son influence sur la performance globale des modèles générés, nous avons implémenté un algorithme de sélection d'attributs, dont le fonctionnement est basé sur le principe de gain d'information. Lors de nos expérimentations, nous avons répété cet algorithme plusieurs fois en modifiant à chaque fois la valeur du seuil du gain d'information, ce qui nous a permis d'avoir plusieurs ensembles de meilleurs attributs en fonction des valeurs de seuil introduites. Les tests effectués ont donc été réalisés sur tous les ensembles des meilleurs attributs qu'on a obtenu, ce qui nous a permis de choisir, lors de la phase de test, lequel de ces ensembles donne de meilleurs résultats.

Enfin, et à travers les résultats de nos expérimentations, nous avons remarqué que parmi les quatre algorithmes de classification testés, les réseaux de neurones étaient le meilleur algorithme, que ce soit en termes de son taux de détection et de son exactitude ou en terme de fausses alarmes qu'il génère. Nous avons également conclu, à travers cette étude expérimentale, que le meilleur seuil du gain d'informations pour la sélection d'attribut est celui qui est choisi entre 0.2 et 0.4. Grace à tous ces conclusions, nous

pouvons dire que la majorité des objectifs de ce travail ont été atteints. Toutefois, il reste d'autres tests qui peuvent encore être réalisés dans le futur, tels que la comparaison de ces algorithmes en utilisant d'autres algorithmes de sélection d'attributs, comme par exemple les algorithmes génétiques, la comparaison de ces algorithmes lorsque les modèles à générer son des modèles multi-classes (normal, DOS, R2L, U2R et Probe), etc.

Bibliographie

[1] José-Marcio Martins da Cruz. Détection d'anomalies réseau par apprentissage non supervisé.

[2] Soda Marieme Fall. Assurance qualite d'une politique de securite informariqeu :cas de la societe energetique star oil, Mémoire de fin de formation ,Centre Africain d'Etudes Supérieures en Gestion , October 2013.

[3] Ahmed Ahmim. Système de détection d'intrusion adaptatif et distribué, Thèse de doctorat , Université Badji Mokhtar de Annaba, 2014.

[4] 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.

[5] Ali Kartit. Une nouvelle approche de détection d'intrusions et étude des problèmes liés au déploiement de politiques de sécurité dans les réseaux informatiques, Thèse de doctorat , Université Mohammed V-Agdal , 2011.

[6] Laurent Bloch, Christoph Wolfhugel, Christian Queinnec, Hervé Schauer, Florence Henry, and Nat Makarévitch. Sécurité informatique , principes et méthodes. Eyrolles, 276p, 2007.

[7] Ines Labed. Proposition d'un système immunitaire artificiel pour la détection d'intrusions, Mémoire de magistère , Université Mentouri de Constantine, 2005.

[8] Eric Detoisien. Ataques externos `https://linuxfocus.org>[consulta:20-08-2006]`.

[9] William Stallings. Cryptography and network security : principles and practices , pearson education india, 2006.

[10] Sadaoui Idir. Les attaques par déni de service distribué dans les systèmes informatiques, Mémoire de Master , Université Abderrahmane Mira de Bejaia, 2017.

[11] David Burgermeister and Jonathan Krier. Les systèmes de détection d'intrusions, Juillet, 2006.

[12] Ahmad Faour. Une architecture semi-supervisée et adaptative pour le filtrage d'alarmes dans les systèmes de détection d'intrusions sur les réseaux, Thèse de doctorat , INSA de Rouen, 2007.

[13] Tarek Abbes. Classification du trafic et optimisation des régles de filtrage pour la détection d'intrusions, Thèse de doctorat , Université Henri Poincaré-Nancy, 2004.

[14] `https://openclassrooms.com/courses/protegez-l-ensemble-de-vos-donnees-sur-votre-ordinateur-1/introduction-a-la-cryptographie`.

[15] `http://www.commentcamarche.com/faq/22304-le-pare-feu-ou-firewall`.

[16] James P Anderson. Computer security threat monitoring and surveillance. Technical Report, James P. Anderson Company, 1980.

[17] Michael Meier. Intrusion detection effektiv ! : Modellierung und analyse von angriffsmustern, Springer-Verlag, 2007.

[18] Abdelhalim Zaidi. Recherche et détection des patterns d'attaques dans les réseaux ip à hauts débits, Thèse de doctorat , Université d'Evry-Val d'Essonne, 2011.

[19] Asmaa Boughrara and Soulimane Mammar. Implementation of a snort's output plug-in in reaction to arp spoofing's attack. In â, editor, *Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on Sciences of Electronics*, 643–647, 2012.

[20] Fatiha Benali. Modélisation et classification automatique des informations de sécurité, Thèse de doctorat , Doc'INSA-INSA de Lyon, 2009.

[21] Julien Iguchi-Cartigny. Scénarios d'attaques et détection d'intrusions, Fin d'Etudes Master 2 , Université de Limoges, 2013.

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

[23] Alain Bernard. La détection d'intrusion : Une approche globale `https://connect.ed-diamond.com/MISC/MISC-072/La-detection-d-intrusion-une-approche-globale`. 2014.

[24] Huwaida Tagelsir Elshoush and Izzeldin Mohamed Osman. Alert correlation in collaborative intelligent intrusion detection systems-a survey. Applied Soft Computing , vol. 11, no 7, p. 4349-4365. Elsevier, 2011.

[25] Amina Djennane and Asma Chikh. Sécurité d'une application web à l'aide d'un système de détection d'intrusions comportementale, Mémoire de fin d'études ,Université Abou Bakr Belkaid-Tlemcen, 2012.

[26] Abdelaziz Amara Korba. Détection d'intrusion et sécurisation du routage dans les réseaux ad hoc , thèse de doctorat,université lumière lyon 2, 2016.

[27] David Pierrot, Nouria Harbi, and Jérôme Darmont. Détection des intrusions, du monitoring des systèmes d'information au graph mining , in : 4e atelier international sur l'innovation et nouvelle tendances dans les systèmes d'information (intis 2014). 2014 , Rabat, Maroc.

[28] Brian Caswell and Jay Beale. Snort 2.1 intrusion detection , syngress, 2004.

[29] Ahmed Patel, Mona Taghavi, Kaveh Bakhtiyari, and Joaquim Celestino JúNior. An intrusion detection and prevention system in cloud computing : A systematic review , journal of network and computer applications , vol. 36, no 1, p. 25-41. elsevier, 2013.

[30] Frédéric Majorczyk. Détection d'intrusions comportementale par diversification de cots : application au cas des serveurs web, Thèse de doctorat , Université Rennes 1, 2008.

[31] Batouche Sonia Saci Souhila. Etude et mise en place d'un système de détection d'intrusion sous linux, Mémoire de Master ,Université Abderrahmane Mira de Béjaïa , 2015.

[32] Hervé Debar, Marc Dacier, and Andreas Wespi. A revised taxonomy for intrusion-detection systems , annales des télécommunications ,vol. 55. no. 7-8 , pp.361–378. Springer-Verlag,2000.

[33] Hervé Debar, Marc Dacier, and Andreas Wespi. Towards a taxonomy of intrusion-detection systems , computer networks , vol.31 .no.8 , pp.805–822. Elsevier,1999.

[34] Nathalie Dagorn. Détection et prévention d'intrusion : présentation et limites ,université de nancy1 ,laboratoire lorrain de recherche en informatique et ses applications (loria) ,2006,.

[35] Adeline Abbé. Analyse de données textuelles d'un forum médical pour évaluer le ressenti exprimé par les internautes au sujet des antidépresseurs et des anxiolytiques, Thèse de doctorat ,Université Paris-Saclay ,2016.

[36] Louardi Bradji. Adaptation des techniques de l'extraction des connaissances à partir des données (ecd) pour prendre en charge la qualité des données, Thèse de doctorat , Université Mentouri Constantine, 2012.

[37] J. Fürnkranz et al. Foundations of rule learning , cognitive technologies. Springer-Verlag Berlin Heidelberg, 2012.

[38] Sumeet Dua and Xian Du. Mining and machine learning in cybersecurity, 2011.

[39] Bertrand Liaudet. Data mining modelisation presentation generale.

[40] Lamiche Chaabane. Fusion et fouille de donnees guidees par les connaissances : Application a l'analyse d'image, Thèse de doctorat ,Université Mohamed Khider-Biskra, 2013.

[41] N. Venkatesan S. Prabhu. Data mining and warehousing, new age international (p) ltd, Publishers, New Delhi, 2007.

[42] Richard Alligier. Apprentissage artificiel applique a la prevision de trajectoire d'avion, Thèse de doctorat ,Université De Toulouse, 2014.

[43] Taleb Zouggar Souad. Contribution à l'apprentissage automatique symbolique par automates d'arbre et mesures de sélection, Thèse de doctorat ,Université d'Oran, 2014.

[44] Moez Baccouche. Apprentissage neuronal de caractéristiques spatio-temporelles pour la classification automatique de séquences vidéo, Thèse de doctorat ,école Doctorale Informatique et Mathématiques de Lyon, 2013.

[45] Aissaoui Siham. Application : Un system de tetection d'intrusion basé sur les séparateurs à vaste marge(svm), Mémoire, Université d'OranEs-senia, 2007.

[46] Liran Lerman. Les systèmes de détection d'intrusion basés sur du machine learning.

[47] Pascal Vincent. Modèles à noyaux à structure locale, Université de Montréal,2003.

[48] Mokhtar Taffar. Initiation à l'apprentissage automatique, Thèse de doctorat ,Université de Jijel.

[49] Hugo Larochelle. Étude de techniques d'apprentissage non-supervisé pour l'amélioration de l'entraînement supervisé de modèles connexionnistes, Thèse de doctorat ,Université de Montréal, 2008.

[50] Hamza Cherif. Classifieurs svm et réseaux de neurones, Thèse de doctorat ,Université de Tlemcen, 2011.

[51] Sébastien Guérif. Réduction de dimension en apprentissage numérique non supervisé, Thèse de doctorat ,Université Paris13, 2006.

[52] Nicolas Nicoloyannis Gaudin. Apprentissage non supervissé de séries temporelles à l'aide des k-means et d'une nouvelle méthode d'agrégation de séries. Article, Laboratoire Eric 3038 Université Lumi*µere* – *Lyon*2.

[53] Laurent Candillier. Contextualisation,visualisation et évaluation en apprentissage non supervisé, Thèse de doctorat, Université Charles de Gaulle Lille 3, 2006.

[54] Fabien Moutarde. Apprentissage non-supervisé, Centre de Robotique (Caor), 2017.

[55] Ziani Soheyb Tabet aoul Walid Houcine. Clustering hiérarchique de données à base de ward, Mémoire de fin d'études , Université Abou Bakr Belkaid-tlemcen, 2012.

[56] Alaoui Abdiya. Application des techniques des métaheuristiques pour l'optimisation de la tâche de la classification de la fouille de données, Mémoire, Université D'Oran Mohamed Boudiaf, 2011.

[57] Marref Nadia. Apprentissage incrémental et machines à vecteurs supports, Mémoire, Université Hadj Lakhdar,2013.

[58] Sébastien Mustière. Apprentissage supervisé pour la généralisation cartographique, Thèse de doctorat, Université de Paris VI,2001.

[59] El Boujnouni Mohamed. Contribution à l'optimisation de la machine d'apprentissage svdd application à la détection de spams et de virus informatiques, Thèse de doctorat, Université Mohammed V,2015.

[60] Chami Djazia. Une plate forme orientée agent pour le data mining, Mémoire , Université Hadj Lakhdar-Batna, 2009.

[61] Fernando Santos Osório. Un systeme hybride neuro-symbolique pour l'apprentissage automatique constructif, Thèse, 2004.

[62] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML,vol.1,pp.577–584*, 2001.

[63] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining,knowledge and information systems ,springer. vol.14,no.A,pp.1–37,2008.

[64] Z Guellil and Lynda Zaoui. Proposition d'une solution au problème d'initialisation cas du k-means,ciia,2009.

[65] Mounzer Boubou. *Contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégation d'opinions.* PhD thesis, Université Claude Bernard-Lyon I, 2007.

[66] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defense Applications,pp.1–6*, 2009.

[67] L Dhanabal and SP 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,no.6,pp.446–452,2015.

[68] Claude Touzet. Les réseaux de neurones artificiels, introduction au connexionnisme. EC2,1992.

Chapitre A

annexe1

Les attributs de NSL KDD

Les détails des attributs sont répertoriés dans les tableaux suivants [67] :

N°	Nom de l'attribut	Description	Type
1	duration	La durée de connexion	Numérique
2	protocol_type	Protocole utilisé dans la connexion (tcp, udp, icmp)	Nominal
3	service	Service réseau de destination,(http,telnet, ftp_data, etc.)	Nominal
4	flag	Statut de la connexion"Normal ou Erreur (SF, REJ, S0, S1, etc.)	Nominal
5	src_bytes	Nombre d'octets de données transférés de la source à la destination (491, etc.)	Numérique
6	dst_bytest	Nombre d'octets de données transférés de destination à la source (0, etc.)	Numérique
7	land	**Si** l'adresse IP de source et destination et le nombre de port sont les mêmes **alors**, *land=1* **sinon** *land=0*	Binaire
8	wrong_fragment	Nombre total de fragments erronés dans cette connexion	Numérique
9	urgent	Nombre de paquets urgents	Numérique
10	Hot	Nombre d'indicateurs « Hot»	Numérique
11	num_failed_logins	Nombre de tentatives de connexion échouées	Numérique
12	logged_in	**Si** connecté avec succès **alors** *logged_in=1* **sinon** *logged_in=0*	Binaire
13	num_compromised	Nombre de conditions compromises	Numérique
14	root_shell	1 si le root shell est obtenu, 0 autrement	Binaire
15	su_attempted	1 **si** la commande "su root" a été tentée ou utilisée, **sinon** 0	Binaire
16	num_root	Nombre d'accès " root " ou nombre d'opérations effectuées comme racine dans la connexion	Numérique
17	num_file_creations	Nombre d'opérations de création de fichiers	Numérique
18	num_shells	Nombre d'invites du shell	Numérique
19	num_access_files	Nombre d'opérations sur les fichiers de contrôle d'accès	Numérique
20	num_outbound_cmds	Nombre de commandes sortantes dans une session FTP	Numérique

21	is_host_login	1 **si** la connexion appartient à la liste du « Hot» (root ou admin) ; **sinon 0**	Binaire
22	is_guest_login	1 **si** le login est un login « guest» ; **sinon 0**	Binaire
23	count	Nombre de connexions vers le même hôte de destination que la connexion en cours dans les deux dernières secondes	Numérique
24	srv_count	Nombre de connexions vers le même service (N° Port) que la connexion en cours dans les deux dernières secondes	Numérique
25	serror_rate	Le pourcentage de connexions qui ont activé le *flag* s0, s1, s2 ou s3, parmi les connexions agrégées dans *count*	Numérique
26	srv_serror_rate	Le pourcentage de connexions qui ont activé le *flag* s0, s1, s2 ou s3, parmi les connexions agrégées dans *srv_count*	Numérique
27	rerror_rate	Le pourcentage de connexions qui ont activé le *flag* REJ, parmi les connexions agrégées dans *count*	Numérique
28	srv_rerror_rate	Le pourcentage de connexions qui ont activé le *flag* REJ, parmi les connexions agrégées dans *srv_count*	Numérique
29	same_srv_rate	Le pourcentage de connexions qui sont au même service, parmi les connexions agrégées dans *count*	Numérique
30	diff_srv_rate	Le pourcentage de connexions qui sont aux différents services, parmi les connexions agrégées dans *count*	Numérique
31	srv_diff_host_rate	Le pourcentage de connexions qui sont Ã différentes machines de destination, parmi les connexions agrégées dans *srv_count*	Numérique
32	dst_host_count	Nombre de connexions ayant la même adresse IP de l'hôte de destination	Numérique
33	dst_host_srv_count	Nombre de connexions ayant la même numéro de port	Numérique
34	dst_host_same_srv_rate	Le pourcentage de connexions qui sont au même service, parmi les connexions agrégées dans *dst_host_count*	Numérique
35	dst_host_diff_srv_rate	Le pourcentage de connexions qui sont aux différents services, parmi les connexions agrégées dans *dst_host_count*	Numériquee
36	dst_host_same_src_port_rate	Le pourcentage de connexions qui sont au même port de source, parmi les connexions agrégées dans *dst_host_srv_count*	Numérique
37	dst_host_srv_diff_host_rate	Le pourcentage de connexions qui sont à différentes machines de destination, parmi les connexions agrégées dans *dst_host_srv_count*	Numérique

38	dst_host_serror_rate	Le pourcentage de connexions qui ont activé le *flag* s0, s1, s2 ou s3, parmi les connexions agrégées dans *dst_host_count*	Numérique
39	dst_host_srv_serror_rate	Le pourcentage de connexions qui ont activé le *flag* s0, s1, s2 ou s3, parmi les connexions agrégées dan *dst_host_srv_count*	Numérique
40	dst_host_rerror_rate	Le pourcentage de connexions qui ont activé le *flag* REJ, parmi les connexions agrégées dans *dst_host_count*	Numérique
41	dst_host_srv_rerror_rate	Le pourcentage de connexions qui ont activé le *flag* REJ, parmi les connexions agrégées dans *dst_host_srv_count*	Numérique

RÉSUMÉ

Aujourd'hui, la technologie de l'information et de la communication s'impose de plus en plus dans de nombreux domaines. Toutefois, cette évolution s'est malheureusement accompagnée d'une augmentation constante du piratage et de la cybercriminalité. Devant cette situation, et afin d'assurer la sécurité des systèmes informatiques, plusieurs outils ont été développés, parmi lesquels on trouve les systèmes de détection d'intrusions (IDS). Un IDS représente tout outil, méthode ou approche qui nous aide à prévoir ou à identifier toute activité non autorisée dans un réseau ou dans une machine. Dans le cadre de ce projet de fin d'étude, nous visons à étudier et à expérimenter l'application de certaines techniques de l'apprentissage automatique et du data mining dans le domaine de la détection d'intrusion. Les résultats obtenus, sur les données du benchmark NSL-KDD, nous ont permis de déduire laquelle de ces techniques est plus adaptée à la détection d'intrusions.

Mots clés : *sécurité informatique, détection d'intrusions, apprentissage automatique, data mining, classification supervisée, classification non supervisée, NSL-KDD.*

ABSTRACT

Today, information and communication technology is increasingly imposed in many areas. However, this development was unfortunately accompanied by a steady increase in computer hacking and cybercrime. In response to this, and to ensure the security of computer systems, several tools have been developed, such as intrusion detection systems (IDS). An IDS is any tool, method or approach that helps to predict or identify any hacking related activity in a network or a machine. Through this end of study project, we aim to study and experiment the application of some machine learning and data mining techniques in the intrusion detection area. The result of this work on the NSL-KDD benchmark data allowed us to deduce which of these techniques is more adapted for intrusion detection.

Key words : *computer security, intrusion detection, machine learning, data mining, supervised classification, unsupervised classification, NSL-KDD.*