الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département des Mathématiques et de l'Informatique

Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en : Informatique
Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Réseaux et Ingénierie des Données (RID)
Thème

# SYSTÈME AUTOADAPTATIF DE DÉTECTION D'INTRUSIONS INFORMATIQUES VIA APPRENTISSAGE AUTOMATIQUE.

**Présenté Par :**
1) Melle. BENNAT Ahlem
2) Melle. LAOURI Dounya

**Devant le jury composé de :**

| | | | |
|---|---|---|---|
| Mme. BOUHALOUAN Djamila | MAA | UAT.B.B (Ain Temouchent) | Présidente |
| Mme. SAIDI Samira | MAA | UAT.B.B (Ain Temouchent) | Examinatrice |
| Dr. BELGRANA Fatima Zohra | MCA | UAT.B.B (Ain Temouchent) | Encadrant |

*Année Universitaire 2021/2022*

# *Remerciement*

*Nous tenons à témoigner nos reconnaissances à DIEU tout puissant,*
*qui nous a aidés et nous a bénis par sa volonté durant toute cette période.*

*Nous vous adressons nos remerciements les plus sincères aux*
*personnes qui nous ont aidés dans la réalisation de ce mémoire.*

*Nous exprimons nos remerciements à notre promotrice*
*Mme BELGRANA.F pour l'aide qu'elle nous a apporté tout*
*au long de ce travail.*

*Nos remerciements vont également aux membres de jury qui*
*ont accepté de juger ce travail, et à tous ceux*
*qui ont contribué de près ou de loin à la réalisation de ce modes travail.*

*Nos remerciements s'adressent aussi à l'ensemble des*
*enseignants du département informatique de la Faculté de science*
*de Belhadj BOUCHAIB d'Ain-Temouchent.*

# *Dédicace*

*Je dédie ce modeste travail à :*

*A tous ceux qui, de près ou de loin, ont participé à mon éducation, m'ont aidé dans les moments difficiles à surmonter mes problèmes.*

*A ceux qui ont partagé avec moi mes bonheurs et mes Soucis.*

*A mes très chers parents pour leurs Sacrifices et leur Soutien durant toutes mes années d'étude.*

*A mon mari qui n'a pas cessé de me conseiller encourager et soutenir pendant toute la durée de ce travail.*

*A mon adorable petite sœur Fatima.*

*A mon très chère frère Mohammed.*

*Sans oublier mon binôme Dounya pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

***BENNAT Ahlem***

# *Dédicace*

*Je dédie ce modeste travail :*

*A ma précieuse mère qui est loin de moi.*

*A mon cher père qui est toujours fière de moi.*

*A tous mes sœurs et frères que j'aime tant.*

*Je dédie en particulier a GHILES pour son soutien et le courage qui il m'a procuré et qui m'a accompagné tout cette période.*

*A mes deux adorables ASMA, MERIEM pour leurs fidélités.*

*A mon cher binôme avec laquelle j'ai partagé ce travail.*

*A tout la promotion 2022.*

*A tous ceux qui me connaissent.*

***LAOUARI Dounya***

# Sommaire

# Table des figures

# Liste des tableaux

# Liste des acronymes

**AA :** Apprentissage Automatique
**AP :** Apprentissage Profond
**ACL :** Access Control List
**AR :** Apprentissage par Renforcement
**DBN :** Réseau de Croyances Profondes
**DDoS :** déni de service distribué
**DoS :** déni de service
**FP :** Faux positifs
**FN :** Faux négatifs
**IETF :** Internet Engineering Task Force
**IP :** Internet Protocol Adresse
**IPSec :** Internet Procol Security
**K-PPV :** K Plus Proches Voisins
**KDD :** Knowledge Discovery in Databases
**L2TP :** Layer 2 Tunnelling Protocol
**MCLT :** Réseaux Récurrents à Mémoire à Court et à Long Terme
**MITM :** Attaque de l'homme au milieu
**ML :** Machine Learning
**MVS :** Support Vector Machin
**NB :** Naïve bayésienne
**NSLKDD :** network security laboratory Knowledge Discovery in Databases
**OSI :** Open Systems Interconnection
**PPP :** Protocole Point à Point est un Protocole
**PPTP :** Point To Point Tunneling Protocol
**RBM :** Machines Boltzman Restreinte
**RN :** Réseaux de Neurones
**RNA :** Réseaux de Neurones Artificiels
**RNC :** Réseaux de Neurones Convolutionnels
**RNR :** Réseaux de Neurones Récurrents
**RPV :** Réseaux Privés Virtuels
**RTC :** Réseau Téléphonique Commuté
**SDI :** Systèmes de Détection d'Intrusions
**SDIH :** Systèmes de Détection d'Intrusions basés système ou Hôte
**SDIR :** Systèmes de Détection d'Intrusions basés Réseau
**SQL :** Structured Query Language
**URL :** localisateur uniforme de ressource
**VN :** Vrai négatifs
**VoIP :** Voix sur IP
**VP :** Vrai positifs

# Introduction générale

Le monde a connu au cours des dernières années un grandissent considérable des systèmes informatiques, qui sont actuellement plus complexes, prenant place dans l'ensemble des secteurs professionnels ou encore de nos vies privées. Grâce à Internet, ces systèmes informatiques sont devenus interconnectés et le nombre de points d'accès ne cesse de croître. Ceci permet de faciliter la communication et le transfert des données entre les points très éloignés. Ce développement phénoménal est malheureusement accompagné par la croissance du nombre d'utilisateurs qui ne sont pas forcément pleins de bonnes intentions. En même temps que l'informatique et Internet ont révolutionné nos gestes et nos habitudes, des menaces ont aussi fait leur apparition en parallèle, il est question alors de les connaître et apprendre à les gérer. Ceux qui causent ces menaces sont appelés pirates informatiques (hackers), ils peuvent exploiter les vulnérabilités et les failles des réseaux et des systèmes pour essayer d'accéder à des informations sensibles, les modifier, ou encore les détruire afin d'arriver à perturber le bon fonctionnement du système.
Jour après jours les attaque informatiques sont en augmentation et deviennent de plus en plus sophistiquées, motivant ainsi l'apparition puis le développement d'une discipline appelée la sécurité des systèmes informatiques. Nombreuses sont les solutions de protections issues de cette discipline aussi différentes les unes des autres. Elles doivent être bien choisies, bien placées et bien configurées suivant les caractéristiques du réseau mis en place, afin d'assurer une bonne protection.
De ce fait l'exigence en termes de la sécurité devient de plus en plus indispensable, par conséquent, le but est d'assurer la disponibilité des services, l'authentification des utilisateurs, la confidentialité et l'intégrité des données. De nombreuses solutions ont été développées, citant par exemple les pare-feux qui visent à filtrer le trafic réseau, ou encore les proxys et les réseaux privés virtuels dont le but est de créer un tunnel sécurisé entre l'utilisateur et la destination souhaitée.
Dans ce contexte, les Systèmes de Détection d'intrusion (SDI) constituent une bonne alternative pour mieux protéger les réseaux informatiques. Généralement les détections des anomalies fonctionnent en deux phases la phase d'apprentissage et phase de détection. Au cours de la phase d'apprentissage, les systèmes apprennent à reconnaître les caractéristiques normales et au cours de la phase de détection les systèmes identifient celles qui sont anormales. Les systèmes pour lesquels la phase d'apprentissage continu après le début de la phase de reconnaissance, sont adaptatifs. Hélas, la propagation rapide des attaques, leurs mises à jour permanentes, ainsi que leurs utilisations des ressources bien développées, représentent des obstacles devant ces outils-la. On ne peut pas dire qu'il existe un outil assurant une sécurisation parfaite, cependant, notre objectif est de développer plutôt un SDIR permettant de maximiser le taux de détection tout en réduisant les fausses alertes.
Tout au long de notre étude, nous essayons de réaliser un système de détection d'intrusion comportemental, où il est question de faire recours à des méthodes issues de l'apprentissage automatiques vu que les bases de données disponibles sont étiquetées.

Nous avons alors utilisé les Machines de Vecteur à Support (MVS) ainsi que l'approche Naïve bayésienne. Ces méthodes malheureusement, ont leurs limites, les résultats ne sont pas aussi performants, nous avons du par la suite appliquer une méthode issue de l'apprentissage profond qui à son tour a révolutionné le domaine de la reconnaissance des formes ces dernières années, il s'agit des Réseaux Récurrents à Mémoire à Court et à Long Terme (MCLT).Une comparaison entre ces trois approches est réalisée.

Ce mémoire est organisé en trois chapitres, précédés par une introduction générale dans laquelle nous avons définis la problématique et notre contribution à la solution. Le premier chapitre décrit quelques notions de base concernant la sécurité informatique, les types d'attaques ainsi que les divers moyens de protection existant ,les SDIs, notre domaine d'application. Quant au deuxième chapitre, nous avons introduit un état de l'art sur l'apprentissage automatique et l'apprentissage profond. Le dernier chapitre est consacré à la présentation de notre approche proposée, nos méthodes adoptées, résultats obtenus ainsi que leurs discussions. Nous avons également illustré notre application ainsi que les outils, le matériel et les logiciels que nous avons utilisé quant au développement de notre SDIR.

Et enfin, nous terminons notre mémoire par une conclusion générale et des perspectives futures pour continuer et améliorer le travail que nous avons entamé.

# Chapitre I

# Sécurité informatique

# I.1 Introduction

La sécurité informatique est une discipline qui touche généralement les réseaux, l'Internet, les points de terminaison, les Cloud, les applications, etc. Elle consiste à établir un ensemble de stratégies de sécurité qui fonctionnent pour trouver des solutions de prévention afin d'aider à protéger toutes les données numériques. Elle permet de garantir la protection de tous types de ressources contre les risques. Les entreprises recherchent actuellement des méthodes et des politiques afin de mettre en œuvre un bon programme de sécurité intégré qu'elles seront en mesure d'adapter plus facilement et plus rapidement. La sécurité informatique est se charge de surveiller et de faire des tests de façon permanente sur l'environnement et sur le réseau informatique dans une parfaite organisation dans le but d'identifier les risques potentiels. La surveillance de la sécurité informatique prend la responsabilité de rechercher les changements de pare-feu qui ne sont pas autorisés en premier, les infractions à la stratégie de protection en second lieu et en dernier les tentatives pour s'introduire dans le réseau. Ensuite, cette solution mappe les réponses aux contrôles appropriés et aux différents processus. Ainsi, le risque diminue considérablement sur le réseau [cisco, 2020a]. Dans ce chapitre, nous allons alors présenter quelques notions de base quant à cette discipline, qui est si importante pour les individus comme pour les entreprises.

# I.2 La sécurité informatique

La sécurité informatique est l'ensemble des moyens mis en œuvre afin d'assurer d'une manière satisfaisante que l'ensemble des données et des ressources matérielles et logicielles d'une entreprise ne soient utilisées que dans le cadre prévu, et de minimiser le nombre de vulnérabilités d'un système contre les menaces accidentelles [Mouzer et kheddam, 2009]. La sécurité informatique est un processus qui consiste à protéger les systèmes, les réseaux et les programmes contre des actes malveillants, ces actes visent généralement à accéder, à modifier ou à détruire des informations sensibles [cisco, 2020a].

# I.3 Les critères de la sécurité

Les protocoles de la sécurité ont pour objectif d'assurer le bon fonctionnement de certains principes de base qui représente les dimensions de la sécurité :

1. **La confidentialité :** seules les personnes habilitées doivent avoir accès aux données. Toute interception ne doit pas être en mesure d'aboutir, les données doivent être cryptées, seuls les acteurs de la transaction possèdent la clé de compréhension [Heady et al., 1990].

2. **L'intégrité :** le critère d'intégrité des ressources physiques et logiques (équipements, données, traitements, transactions et services) est relatif au fait qu'elles n'ont pas été détruites (altération totale) ou modifiées (altération partielle) à l'insu de leurs propriétaires tant de manière intentionnelle qu'accidentelle. Une fonction de sécurité appliquée à une ressource pour contribuer à préserver son intégrité, permettra de la protéger plus ou moins efficacement contre une menace de corruption ou de destruction [Ghernaouti, 2008].

3. **La disponibilité :** il faut s'assurer du bon fonctionnement du système, de l'accès à un service et aux ressources à n'importe quel moment. Les données soient effectivement à disposition des ayants droit en temps voulu. L'objectif de la disponibilité est de garantir l'accès à un service ou à des ressources en temps bien précis [Pillou et Bay, 2013].

4. **La non-répudiation :** La non-répudiation est le fait de ne pouvoir nier ou rejeter qu'un événement (action, transaction) a eu lieu [Eric, 2001a].

5. **Non-rejeu :** le non-rejeu est un mécanisme garantissant qu'un message échangé entre deux entités A et B, ne doit pas être réutilisé par une entité non autorisée C. La plupart des systèmes intègrent des compteurs et des numéros de séquence différents au niveau des messages échangés, ce qui fait qu'un message ne peut pas avoir le même numéro de séquence que ses n messages précédents (n un nombre de messages qui varie selon la politique de sécurité utilisée), sinon il sera automatiquement rejeté [Hammi, 2018].

6. **L'authentification :** La première étape afin de protéger les ressources d'un système informatique est de pouvoir vérifier l'identité des utilisateurs, cette vérification s'appelle l'authentification. L'authentification permet de vérifier l'identité annoncée et de s'assurer du non usurpation de l'identité d'une entité. C'est-à-dire qu'elle permet de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre l'accès à des ressources uniquement aux personnes autorisées [Eric, 2001a].

## I.4 Les pirates informatiques (hackers)

Le pirate informatique est une personne qui utilise des compétences informatiques, en réseau ou autres pour surmonter un problème technique. Le terme peut également faire référence à quiconque utilise ses capacités pour acquérir un accès non autorisé aux systèmes ou aux réseaux afin de commettre des crimes. Un pirate informatique peut, par exemple, voler des informations (vol d'identité par exemple) ou de faire tomber un système et, souvent dans l'objectif de collecter une rançon. Le terme pirate informatique a toujours été une division, parfois utilisé comme mandat d'admiration pour les personnes présentant un degré élevé de compétences et de créativité dans leur approche des problèmes techniques. Cependant, le terme est également couramment appliqué aux personnes qui utilisent cette compétence à des fins illégales ou contraires à l'éthique [ Filippi et Dulong, 2014].

### I.4.1 Les types de pirates

Les pirates informatiques sont classés selon leurs activités et leur puissance de pouvoir produire des failles et attaquer des victimes, voici quelques pirates les plus connues [Pillou et Maraï, 2020] :

1. **Whites hat hackers :** pirate au sens noble du terme, dont le but est de contribuer à l'amélioration des systèmes et technologies informatiques. Ils sont généralement à l'origine des principaux protocoles et outils informatiques que nous utilisons aujourd'hui. Leurs objectifs sont en règle générale l'apprentissage, l'optimisation des systèmes informatiques, la mise à l'épreuve des technologies jusqu'à leurs limites afin de tendre vers un idéal plus performant et plus sûr.

2. **Black hat hackers :** plus couramment appelés pirates informatiques, c'est-à-dire des personnes s'introduisant dans les systèmes informatiques dans un but nuisible. Les motivations des blacks hats hackers peuvent être multiples d'un intérêt financier ou encore l'intérêt politique.

3. **Les scripts kiddies :** gamins du script, parfois également surnommés crashers, lamers ou encore packet monkeys(en français : les singes des paquets réseau), ce sont de jeunes utilisateurs du réseau utilisant des programmes trouvés sur Internet, généralement de façon maladroite, pour vandaliser des systèmes informatiques afin de s'amuser.

4. **Les phreakers :** ce sont des pirates s'intéressant au réseau téléphonique commuté (RTC) afin de téléphoner gratuitement grâce à des circuits électroniques (qualifiés de box, comme la blue box, le violet box...) connectés à la ligne téléphonique dans le but d'en falsifier le

fonctionnement. On appelle ainsi phreaking le piratage de ligne téléphonique. Ce type de pirate connaît un renouveau avec l'ac- croissement de l'utilisation de la voix sur IP (VoIP) comme moyen de transport des communications téléphoniques grand public.

## I.5 Les attaques informatiques

Tout ordinateur connecté à un réseau informatique est potentiellement exposé à une attaque. Une attaque est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins généralement préjudiciables [Eric, 2001b]. Les motivations des attaques peuvent être de différentes sortes :

- Obtenir un accès au système ;
- Voler des informations, telles que des secrets industriels ou des propriétés intellectuelles ;
- Recueillir des informations personnelles sur un utilisateur ;
- Récupérer des données bancaires ;
- S'informer sur l'organisation (entreprise de l'utilisateur) ;
- Troubler le bon fonctionnement d'un service ;
- Utiliser les ressources du système de l'utilisateur, notamment lorsque le réseau sur lequel il est situé possède une bande passante élevée.

### I.5.1 Les types d'attaques

Il existe plusieurs types d'attaques qui peuvent être regroupées en trois grandes classes à savoir :

#### I.5.1.1 Les attaques directes

C'est les attaques les plus simples, le pirate attaque directement sa victime à partir de son ordinateur (voir la figure I.1), la plupart des "scripts kiddies" utilisent cette technique. Les programmes de hack qu'ils utilisent ne sont que faiblement paramétrables, et un grand nombre de ces logiciels envoient directement les paquets à la victime. L'avantage de ces attaques est qu'on peut remonter facilement à l'origine [Arnaud, 2002].


FIGURE I.1 – Les attaques directes [Arnaud, 2002].

### I.5.1.2 Les attaques indirectes par rebond

Dans ce type d'attaques, le pirate attaque la machine cible par une machine intermédiaire, cette attaque est très prisée des hackers. En effet, le rebond a deux caractéristiques :

- Masquer l'identité (l'adresse IP) du hacker.
- Éventuellement, utiliser les ressources de l'ordinateur intermédiaire, car il est plus puissant (CPU, bande passante...) pour attaquer.

Le principe en lui-même, est simple : les paquets de l'attaque sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime, d'où le terme de rebond (voir la figure I.2) [Arnaud, 2002].

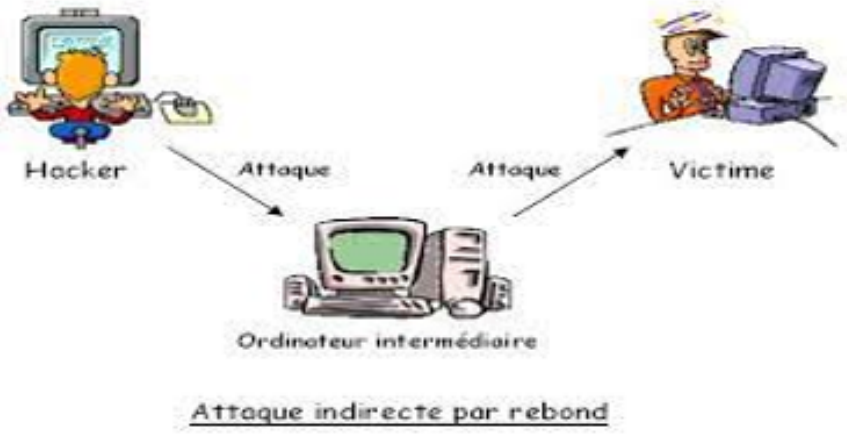


FIGURE I.2 – Les attaques indirectes [Arnaud, 2002]

### I.5.1.3 Les attaques indirectes par réponse

Cette attaque est une dérivée de l'attaque par rebond, elle offre les mêmes avantages, du point de vue du hacker. Mais au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête. Et c'est cette réponse à la requête qui va être envoyée à l'ordinateur victime (voir la figure I.3) [Arnaud, 2002].

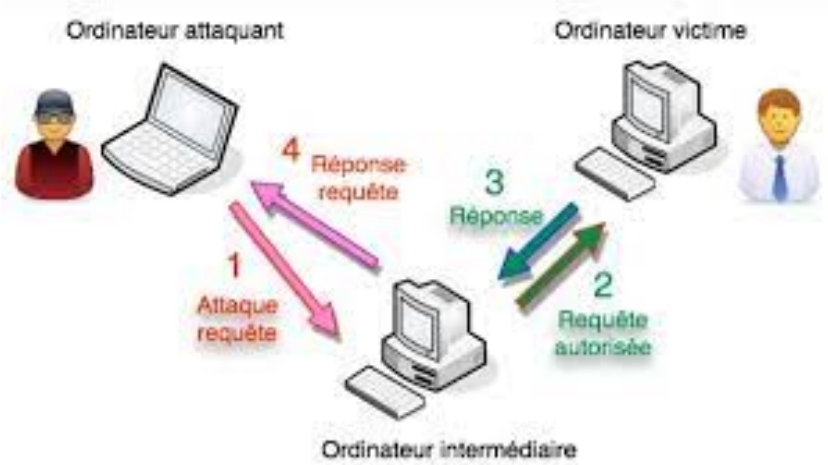


FIGURE I.3 – Les attaques indirectes par réponse [Arnaud, 2002].

## I.5.2 Les catégories d'attaques

### I.5.2.1 Les attaques passives

Elle consiste à collecter les informations relatives au système cible, mais elle n'affecte pas ses ressources. C'est une étape difficile à détecter. L'un des outils les plus utilisés est les renifleurs de paquets (en anglais sniffer), qui permet de comprendre l'architecture du réseau et de déceler les points faibles et les ressources importantes à attaquer [Eric, 2001b].

### I.5.2.2 Les attaques actives

Les attaques actives consistent à effectuer des changements non autorisés sur les données des systèmes, à s'introduire dans des équipements réseau ou à perturber leurs fonctionnements, les attaques de ce type sont bien évidemment plus dangereuses.(ex : mascarade et déni de service). Les attaques actives peuvent prendre la forme d'interruption, d'interception, de modification ou de fabrication comme le montre dans (voir la figure I.4) [Poinsot, 2019] :

- **Attaques par interruption :** c'est une attaque qui vise la disponibilité, la destruction d'une pièce matérielle (tel un disque dure), la coupure d'une ligne de communication, ou la mise hors service d'un système de gestion de fichiers (voir la figure I.4(b)).
- **Attaque par interception :** c'est une attaque portée à la confidentialité, il peut s'agir d'une personne, d'un programme ou d'un ordinateur ou une écoute téléphonique dans le but de capturer des données sur un réseau, ou la copie non autorisée de fichiers ou de programme (voir la figure I.4(c)).
- **Attaque par modification :** il s'agit d'une attaque portée à l'intégrité, telle que le changement des valeurs dans un fichier de données, altérer un programme de façon à bouleverser son comportement ou modifier le contenu des messages transmis sur un réseau (voir la figure I.1(d)).
- **Attaque par fabrication :** cette attaque vise l'authenticité, il peut s'agir de l'insertion de faux messages dans un réseau ou l'ajout d'enregistrement à un fichier (voir la figure I.4(e)).

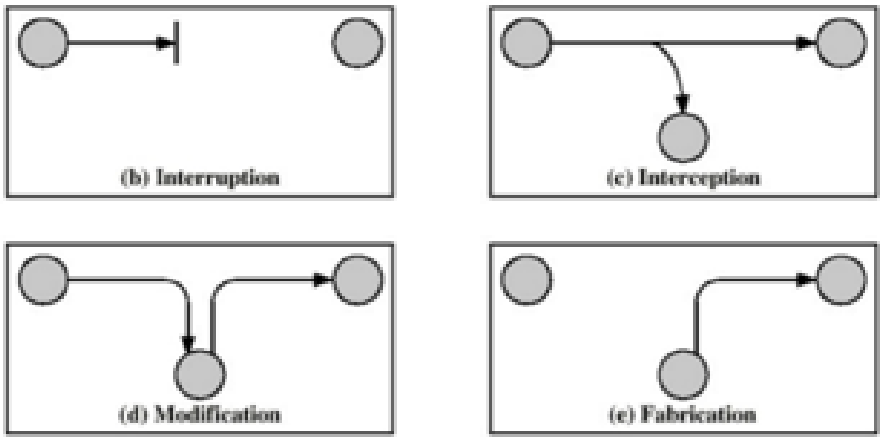


FIGURE I.4 – Les catégories des attaques [Labraoui, 2020].

### I.5.3 Les attaques les plus courantes

Tout ordinateur connecté à un réseau informatique est potentiellement vulnérable à une attaque informatique. Or, il existe plusieurs attaques courantes, nous citons :

1. **Les attaques par déni de service (DoS) et par déni de service distribué (DDoS)**
   Le déni de service (en anglais : Denial of Service, en abrégé : DoS) comme son nom l'indique est un déni, un refus, un rejet d'un service informatique. Cette attaque vise à rendre une ressource ou une information indisponible, elle peut être réalisée en inondant la machine source ou le réseau par un grand nombre de messages. Les conséquences concernant le système attaqué sont : l'instabilité, l'indisponibilité quant au DDoS, c'est un type d'attaque très évolué visant à faire planter ou à rendre muette une machine en la submergeant de trafic inutile. Plusieurs machines à la fois sont à l'origine de cette attaque (c'est une attaque distribuée) qui vise à anéantir des serveurs, des sous-réseaux, etc. D'autre part, elle reste très difficile à contrer ou à éviter. C'est pour cela que cette attaque représente une menace que beaucoup craignent [Tatarova et Giarmana, 2010].

2. **L'attaque par Usurpation d'IP**
   Il consiste à se faire passer pour un autre système en falsifiant son adresse IP. L'attaquant doit dé- terminer les systèmes ou les adresses IP autorisées à se connecter au système cible en récoltant le maximum de détails sur ce dernier (voir figure I.5 ). Le pirate ensuite attaque la cible en utilisant l'adresse IP falsifie [Valgasu, 2017].

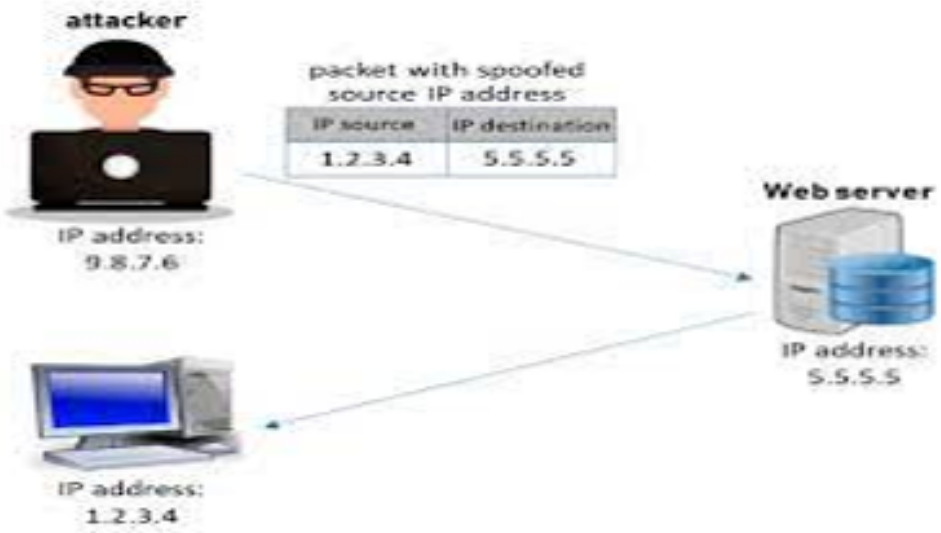


FIGURE I.5 – Usurpation d'IP [Valgasu, 2017].

3. **L'attaque par hameçonnage**
   L'hameçonnage est une attaque de type réseau où l'attaquant crée le faux d'une page Web existante pour tromper un utilisateur en ligne afin d'obtenir des informations personnelles [Gupt et al., 2017]. Dans ce type d'attaque, les attaquants envoient des URL suspectes aux victimes ciblées par le biais de spams, de SMS ou de réseaux sociaux en ligne. Un attaquant diffuse une fausse variante d'un site Web original, par e-mail, téléphone, etc. Ils ciblent probablement la victime pour inclure ses données personnelles ou très sensibles (par exemple, coordonnées bancaires, numéro d'épargne du gouvernement, etc.). Une attaque de phishing amène un attaquant à acquérir des informations de carte bancaire et des données de connexion [Basit et al.,2021].

4. **Attaque par injection SQL**
   Les injections SQL sont nommées aussi attaques applicatives, car ses dernières se basent sur des failles dans les programmes utilisés, ou encore des erreurs de configuration, les injections SQL profitent de paramètres d'entrée non vérifies. Le but des injections SQL est d'injecter du code SQL dans une requête de base de données, ainsi il est possible de récupérer des informations se trouvant dans la base ou encore de détruire des données [cisco, 2020b].

5. **Attaque de l'homme au milieu (Man in The Middle : MITM)**
   L'attaque MITM est un scénario dans lequel un pirate a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication a été compromis. Le pirate doit être capable d'observer et d'intercepter les messages d'une victime à l'autre pour qu'il puisse prendre l'identité d'un des deux [Ritambhara, 2020].

## I.6 Les différents types de protections contre les attaques

Nous avons vu précédemment les différents types d'attaque dans un réseau informatique, pour cela il est important de bien comprendre comment protéger l'intégrité d'un système informatique et avoir une expérience sur le comportement de tous types d'intrusion, donc il existe plusieurs mécanismes et dispositifs de sécurité. Voici plusieurs logiciels permettant de détecter les intrusions :

- Antivirus ;
- Pare-feu ;
- Serveur proxy ;
- Réseaux Privés Virtuels (RPV) ;
- Systèmes de Détection d'Intrusions (SDIs) ;
- La cryptographie .

### I.6.1 Antivirus

Les antivirus sont des programmes contenant une base de données de tous les codes malicieux connus (base antivirale), qui permettent donc de détecter, de supprimer et éventuellement de réparer les fichiers infectés par les virus. La désinfection se fait par les méthodes suivantes : [Michel, 2003]

- La suppression du code correspondant au virus dans le fichier infecté ;
- La suppression du fichier infecté ;
- La mise en quarantaine du fichier infecté, consistant à le déplacer dans un emplacement où il ne pourra pas être exécuté.

#### I.6.1.1 Les types d'antivirus

Il existe plusieurs types d'antivirus à savoir le scanner à la demande, le moniteur en temps réel et la base de signature de virus [Eric, 2001b] :

1. **Scanner à la demande**
   Le scanner examine (scan) l'ordinateur ou le mobile à la demande : un fichier, un dossier ou tous les fichiers du disque. En particulier un scan complet consomme beaucoup de ressources matérielles et de temps, mais il faut le faire de temps en temps.

2. **Moniteur en temps réel** Le moniteur analyse en temps réel les fichiers auxquels il est question d'y accéder au cours d'une utilisation normale, il identifie rapidement et stoppe immédiatement une exécution virale. Il est composé de plusieurs modules dont le nom change suivant les logiciels. Par exemple McAfee VirusScan en a quatre, chacun dédié à une tâche : e-mail, Web, téléchargement, système. En fonction de sa configuration et de la puissance de l'ordinateur, il ralentit plus ou moins vos applications.

3. **Base de signature virale** Une signature est une chaîne de caractères binaires identifiant un virus ou autres programmes mal- veillants, un peu comme une empreinte digitale humaine. Ainsi un fichier infecté par un programme malveillant contient sa signature. Une base de données contient les signatures de virus et autres programmes malveillants : c'est la base de signature virale.

#### I.6.1.2 Fonctionnement de l'antivirus

La plupart des antivirus fonctionnent selon trois modes :

- En mode statique : l'utilisateur active l'antivirus à la demande ;

- En mode dynamique : l'antivirus tourne de manière continuelle en analysant tous les événements et actions de la machine ;

- En mode planifié : l'antivirus se lance selon une planification établie par l'utilisateur ou l'administrateur.

Généralement les antivirus utilisent trois processus de détection qui sont très efficaces [advisio,2019] :

- **Par analogie :** ainsi, l'antivirus reconnaît un virus déjà connu à partir d'une base de signature contenant les empreintes des codes malveillants. Cependant, l'antivirus n'arrive pas toujours à détecter les mutations des virus. Le maintien à jour de la base de signatures virales est le principal inconvénient de ce mode.

- **Par analyse spectrale :** l'antivirus recherche à partir d'une liste d'instructions d'un programme des caractéristiques de virus ou de ver. L'inconvénient de cette technique est qu'elle génère plus de fausses alertes. L'avantage est qu'elle peut détecter des nouveaux virus même s'ils mettent en œuvre des techniques de polymorphisme.

- **Par heuristique :** cette technique a pour principe de reconnaître des comportements ou des actions identifiées comme malveillantes ou comme une tentative d'abus du système, tout ceci à partir de signatures de code. Le principal inconvénient de cette technique est que l'antivirus se trouve vite limité en face aux techniques d'exploitation novatrices.

#### I.6.1.3 Les limites de l'antivirus

La solution antivirus est le dernier rempart du système pour prévenir les attaques mais ils sont insuffisants pour détecter de nouvelles attaques qui ne sont pas dans la base de données, voici quelques limites [Advancia, 2021] :

- Les antivirus traditionnels basés sur la signature sont incapables de détecter les nouveaux virus qui ne sont pas inscrits dans leur base de données, un malware doit exister avant de pouvoir être classé comme étant un programme malveillant ;

- La performance de l'antivirus dépend de la capacité de son éditeur à mettre à jour la base de données pour qu'elle contienne les derniers virus ;

- Il est destiné à la détection des virus, des trojans, des vers et backdoors. L'antivirus ne détectent ni les spywares, ni les adwares et ni les rogues ;
- Neutralisation de l'antivirus (arrêt du logiciel, ou blocage des traitements) ;
- La complexité de détection des virus peut amener à une sur-détection de virus (faux positifs) ou la non-détection de virus (faux négatifs).

### I.6.2 Pare-feu

Afin d'éviter que des attaques puissent parvenir jusqu'au réseau d'une entreprise, il convient d'isoler le réseau interne (privé) de l'entreprise du réseau externe (internet). La méthode la plus connue est le firewall, donc Le pare-feu est un système aux fonctions de filtrage évoluées. Chaque paquet reçu est examiné, [Andrew, 2003], une décision de rejet ou d'acceptation est prise en fonction de nombreux critères :

- Adresse source ;
- Adresse destination ;
- Port source ;
- Port destination ;
- Le protocole transporté (ICMP, UPD. . .) ;
- La valeur de certains flags (ACK, SYN. . .).

Le firewall, placé à l'entrée du réseau, constitue ainsi un unique point d'accès par où chacun est obligé de passer (voir la figure I.6) [Andrew, 2003].

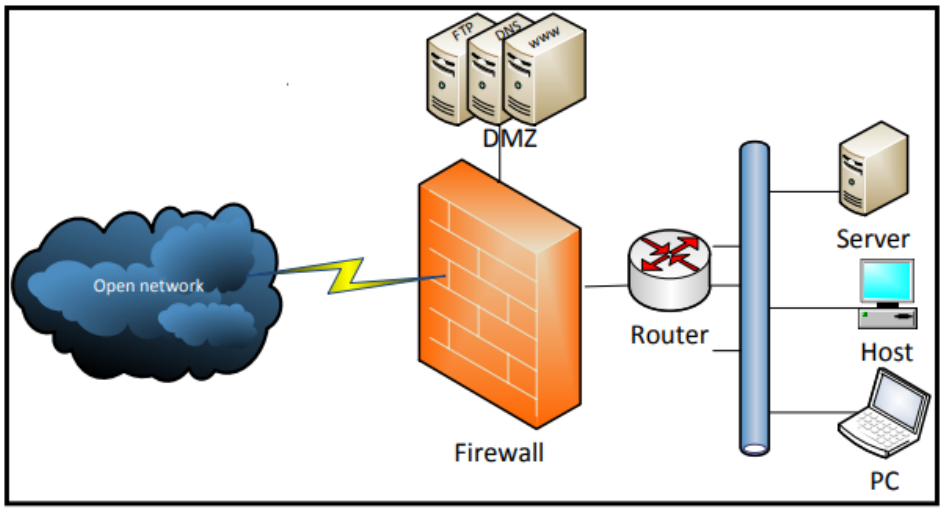


FIGURE I.6 – Architecture d'un pare-feu [Kartit, 2016].

#### I.6.2.1 Fonctionnement de pare-feu

Un système Pare-feu contient un ensemble de règles prédéfinies permettant [Pillou et Bay, 2013] :

- D'autoriser la connexion [allow] ;
- De bloquer la connexion [deny] ;

- De rejeter la demande de connexion sans avertir l'émetteur [drop].

En se basant sur un ensemble de règles qui permettent de mettre en œuvre une méthode de filtrage qui dépend de la politique de sécurité adoptée par l'entité. On distingue habituellement deux types de politiques de sécurité permettant :

- Soit d'autoriser uniquement les communications ayant été explicitement autorisées : « tout ce qui n'est pas explicitement autorisé est interdit » ;
- Soit d'empêcher les échanges qui ont été explicitement interdites.

#### I.6.2.2 La méthode de filtrage

Il existe trois méthodes de filtrage [Grim et al., 2014] :

- **Le filtrage simple de paquet (Stateless) :** c'est la méthode de filtrage la plus simple, elle opère au niveau de la couche réseau et transport du modèle OSI. La plupart des routeurs d'aujourd'hui permettent d'effectuer du filtrage simple de paquet. Cela consiste à accorder ou refuser le passage de paquets d'un réseau à un autre en se basant sur l'adresse IP source/destination, le numéro de port source/destination et bien sûr le protocole de niveaux 3 ou 4. Cela nécessite de configurer le Firewall ou le routeur par des règles de filtrage, généralement appelées des ACL (Access Control Lists).
- **Le filtrage de paquet avec état (Stateful) :** L'amélioration par rapport au filtrage simple, est la conservation de la trace des sessions et des connexions dans des tables d'états internes au Firewall. Le Firewall prend alors ses décisions en fonction des états de connexions, et peut réagir dans le cas de situations protocolaires anormales. Ce filtrage permet aussi de se protéger face à certains types d'attaques « DoS ».
- **Le filtrage applicatif (ou pare-feu de type proxy ou proxying applicatif) :** le filtrage applicatif est comme son nom l'indique, réalisé au niveau de la couche application. Pour cela, il faut bien sûr pouvoir extraire les données du protocole de niveau 7 pour les étudier. Les requêtes sont traitées par des processus dédiés, par exemple une requête de type http sera filtrée par un processus proxy http. Le pare-feu rejettera toutes les requêtes qui ne sont pas conformes aux spécifications du protocole. Cela implique que le pare-feu proxy connaisse toutes les règles protocolaires des protocoles qu'il doit filtrer.

#### I.6.2.3 Les limites du pare-feu

Parmi les limites du pare-feu, ce dernier ne peut hélas [Gérard, 2017] :

- Protéger un réseau d'un trafic qui ne passe pas par le pare-feu (exemple de modems) ;
- Protéger contre les utilisateurs internes (selon leurs droits) ;
- Protéger contre les virus ;
- Protéger contre des menaces imprévues (hors politique) ;
- Être gratuit et se configurer tout seul.

### I.6.3 Serveur proxy

En anglais, le terme « proxy » désigne un mandataire, une personne que vous avez autorisée à effectuer certaines actions en votre nom, par exemple à voter dans une réunion importante à laquelle vous ne pouvez pas assister. Un serveur proxy remplit exactement le même rôle, mais en ligne. Au lieu de communiquer directement avec le site Web qui vous intéresse, un proxy se charge de gérer cette relation à votre place. Lorsque vous naviguez sur Internet, vous vous connectez directement au site Web qui vous intéresse, un proxy établit à votre place la communication avec le site Web (voir la figure I.7) [Belcic, 2020].

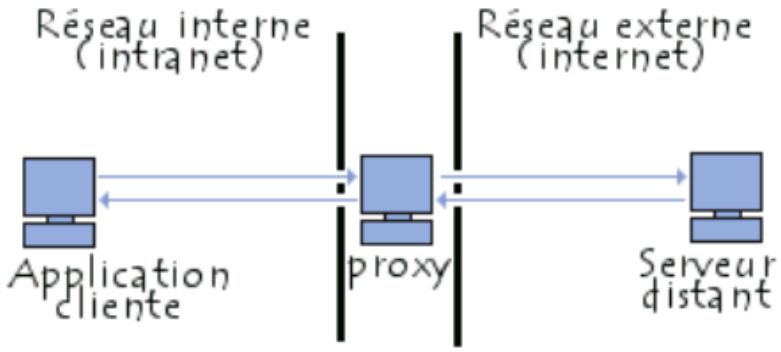


FIGURE I.7 – Architecture d'un proxy [Belcic, 2020]

#### I.6.3.1 Fonctionnement d'un serveur proxy

Sur Internet, tout appareil ou logiciel relève généralement de l'une de ces deux catégories : clients ou serveurs. Un client, par exemple, si un utilisateur souhaite d'accéder à un site web, le serveur proxy reçoit la demande du client, récupère la page web puis l'envoi au client.
En l'absence de proxy, l'ordinateur communique directement avec les serveurs Web. Un serveur proxy se place devant le client ou un réseau de clients, et gère ce trafic à sa place. Ce serveur proxy est un autre ordinateur connecté à la fois à Internet et à l'ordinateur qui dispose de sa propre adresse IP, cet ordinateur communique uniquement avec le proxy et ce dernier transfère toutes les communications vers Internet au sens large.
Lorsqu'Internet répond, le proxy transmet ces réponses à l'ordinateur. De nombreux serveurs proxy (mais pas tous les types) peuvent masquer l'adresse IP des ordinateurs, de sorte que les sites Web auxquels on y accède ne connaissent pas l'identité de l'utilisateur [Belcic, 2020].

#### I.6.3.2 Les limites de serveur de proxy

Parmi les limites de ce type de protection, nous citons[Belcic, 2020] :

- **Instabilité :** les serveurs proxy, en particulier les proxys gratuits, ne sont pas spécialement reconnus pour leurs performances à toute épreuve. Des interruptions et déconnexions impromptues sont à prévoir.

- **Lenteur :** si le proxy prévoit la mise en cache, il peut accélérer le temps de chargement des sites Web. Dans le cas contraire, un proxy peut ralentir la connexion. Il s'agit d'un effet secondaire résultant de l'acheminement du trafic via un proxy.
- **Sécurité limitée :** les serveurs proxy peuvent effectivement masquer l'adresse IP de l'utilisateur et héberger des pare-feu, mais certains ne sont pas en mesure de chiffrer votre trafic comme le ferait un VPN.

### I.6.4 Réseau Privé Virtuel (RPV)

Un réseau privé virtuel (Virtual Private Network en anglais, abrégé en VPN) est un système e qui permet d'étendre un réseau privé sur un réseau public afin de créer un lien direct entre des appareilles distants (voir la figure I.8). Un RPV isole les échanges des deux parties du reste du trafic qui se déroule sur le reste des réseaux publics. Ce réseau est dit virtuel, car il relie deux réseaux physiques par Internet. Il est dit privé, car seuls les ordinateurs des réseaux locaux de chaque côté du RPV peuvent voir les données [Pillou et Bay, 2013].

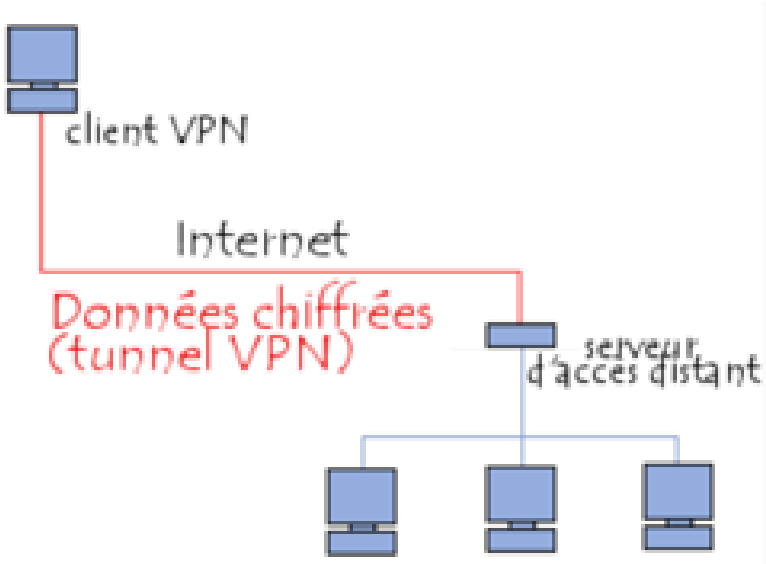


FIGURE I.8 – Architecture d'un RPV [Pillou et Bay, 2013].

#### I.6.4.1 Fonctionnement du RPV

Les données à transmettre peuvent appartenir à un protocole différent d'IP. Dans ce cas le protocole de tunneling encapsule les données en rajoutant un entête. Permettant le routage des trames dans le tunnel. Le tunneling est l'ensemble des processus d'encapsulation, transmission et de dés - encapsulation [Arkoub et Amghar, 2015] :

- **PPTP (Point To Point Tunneling Protocol) :** développé par Microsoft, il permet de créer des trames sous le protocole PPP (protocole point à point est un protocole de transmission pour Internet, il permet d'établir une connexion entre deux hôtes sur une liaison point à point. Il fait partie de la couche liaison de données (couche 2) du modèle OSI) et de les encapsuler dans un datagramme IP.
- **IPSec (Internet Procol Security) :** protocole défini par l'IETF (Internet Engineering Task Force) permettant de sécuriser les échanges au niveau de la couche réseau.

- **L2TP (Layer 2 Tunnelling Protocol) :** développé par Cisco très proche du PPTP, il encapsule des trames de protocole PPP, encapsulant elle-même d'autres protocoles.

#### I.6.4.2 Les limites d'un RPV

Étant donné que la connexion au réseau privé fonctionne en connectant le réseau de l'utilisateur à un serveur privé avant que ce dernier ne puisse réellement accéder au site web souhaité, cela peut créer un problème de performances dans le processus. Plusieurs pays considèrent les réseaux privés comme illégaux, et par conséquent les RPV aussi, ce qui est une limitation dans certaines situations. Les RPVs sont des connexions point à point et ne tendent pas à soutenir les domaines de diffusion ; par conséquent, les communications, les logiciels et la mise en réseau, qui sont basés sur la couche 2 du modèle OSI et les paquets de diffusion, tels que NetBIOS utilisé dans les réseaux Windows, peuvent ne pas être entièrement pris en charge comme sur un réseau local. Des variantes sur RPV telles que le VPLS (Virtual Private LAN Service) et les protocoles de tunneling de couche 2 sont conçues pour surmonter cette limitation [cisco, 2020c].

### I.6.5 La cryptographie

La cryptographie désigne l'ensemble des méthodes ou techniques (chiffrement, signature numérique et certificat) permettant de garantir l'intégrité, authenticité et la confidentialité des informations sensibles, or la cryptographie permet d'assurer cette dernière, et cela, grâce au chiffrement [Dumont, 2009].

#### I.6.5.1 Chiffrement et déchiffrement

Les données qui peuvent être lues et comprises sans mesures spéciales sont appelées texte clair (ou libellé). Le procédé qui consiste à dissimuler du texte clair de façon à cacher sa substance est appelé chiffrement (dans le langage courant on parle plutôt de cryptage et de ses dérivés : crypter, décrypter). Chiffrer du texte clair produit un charabia illisible appelé texte chiffré (ou cryptogramme). Vous utilisez le chiffrement pour garantir que l'information est cachée à quiconque elle n'est pas destinée, même ceux qui peuvent lire les données chiffrées. Le processus de retour du texte chiffré à son texte clair originel est appelé déchiffrement [Bloch et al., 2007].

#### I.6.5.2 La signature

La signature numérique (parfois appelée signature électronique), est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur , par analogie avec la signature manuscrite d'un document papier. Elle se différencie de la signature écrite par le fait qu'elle n'est pas visuelle, mais correspond à une suite de nombres [Bloch et al, 2007].

## I.7 Les Systèmes de détection d'intrusions

Les systèmes de détection ont été imposés en raison de la nécessité d'améliorer la capacité à auditer et à surveiller les systèmes et réseaux informatiques. James Anderson [Anderson ,1980] a été le premier à introduire le concept de systèmes de détection en 1980, mais le premier modèle a été créé par Denning Dorothy [Denning, 1987] et plusieurs prototypes ont été produits. Des budgets importants sont investis dans la recherche dans ce domaine jusqu'à ce jour [Denning, 1987].

### I.7.1 Définition

La détection d'intrusion est un procédé utilisable par le personnel de sécurité pour protéger l'entreprise contre les attaques [Nobelis, 2004]. Il s'agit d'un ensemble de composants logiciels et matériels dont la fonction principale est d'analyser et de détecter toute tentative d'effraction ou activité suspecte comme les activités de sondage (le balayage des ports), des tentatives de compromission des systèmes, des activités suspectes internes, des activités virales, etc [Mouzer et kheddam, 2009].

### I.7.2 Les caractéristiques d'un SDI

Les caractéristiques suivantes sont souhaitables dans un SDI : [Nevski ,2003]

- Il doit fonctionner de manière continue avec une présence humaine minimum ;
- Il doit être tolérant aux fautes c'est-à-dire qu'il doit être capable de retrouver son état initial de fonctionnement après un crash causé soit par une manipulation accidentelle soit par des activités émanant de personnes malintentionnées ;
- Il doit résister à la subversion, l'SDI doit être capable de se contrôler lui-même et de détecter s'il a été modifié par un attaquant ;
- Il doit être configurable d'après les politiques de sécurité du système qu'il supervise ;
- Il doit également être capable de s'adapter aux changements du système et des comportements des utilisateurs au cours du temps ;
- Il doit imposer une supervision minimale du système sur lequel il tourne afin de ne pas interférer avec ses opérations normales ;
- Il doit être capable de superviser un nombre important de stations tout en fournissant des résultats de manière rapide et précise ;
- Il doit fournir "un service minimum de crise", c'est-à-dire que si certains composants de l'SDI cessent de fonctionner, les autres composants doivent être affectés le moins possible par cet état de dégradation.

### I.7.3 L'architecture de base d'un SDI

Un SDI est constitué essentiellement des composants suivants (voir la figure I.9) [Bouras et Fethallah, 2017] :

- **Le senseur :** il est responsable de la collecte d'informations du système tel que des paquets d'un réseau, ou des données d'un logiciel.
- **L'analyseur :** il reçoit l'ensemble des informations venant des senseurs. Il est responsable de les analyser et d'indiquer si une attaque a lieu ainsi qu'éventuellement sa réponse.
- **L'interface utilisateur :** elle permet aux utilisateurs de l'SDI de visualiser et de définir le comportement système.
- **Manager :** En plus de la notification des alertes, il offre à l'administrateur la possibilité de configurer une sonde et de gérer les alertes envoyées par l'analyseur.

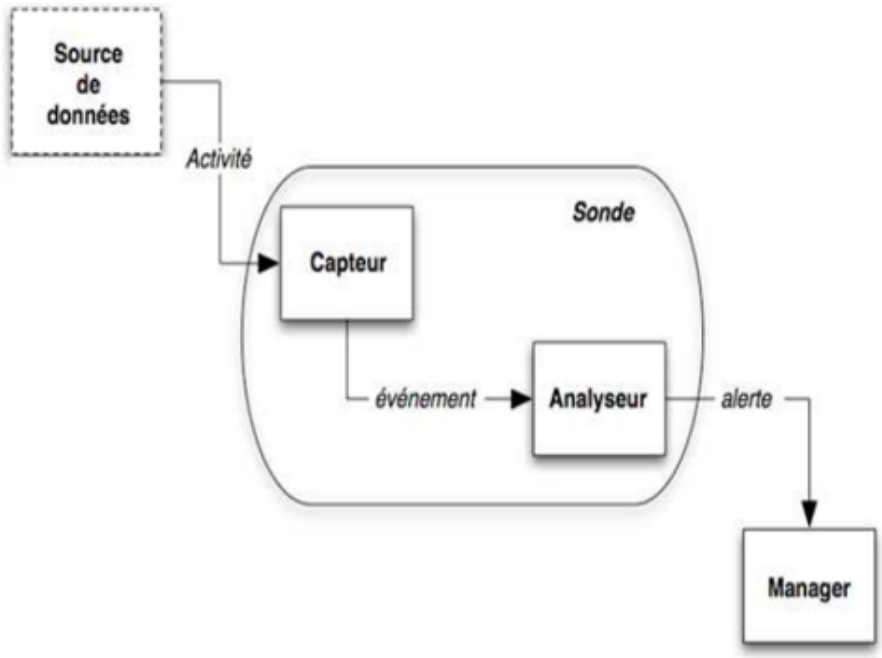


FIGURE I.9 – L'architecture de base d'un SDI [Bouras et Fethallah, 2017].

### I.7.4 Taxonomie des SDIs

Les SDIs ont été classés selon différents critères (voir la figure I.10), la méthode de détection étant le principal, deux approches dérivant de cette dernière existent aujourd'hui : l'approche comportementale et l'approche par scénarios. D'autres critères ont été cités dans la littérature à savoir :

- Le comportement après la détection ;
- La source des données ;
- La source de données à analyser ;
- La fréquence de l'analyse.

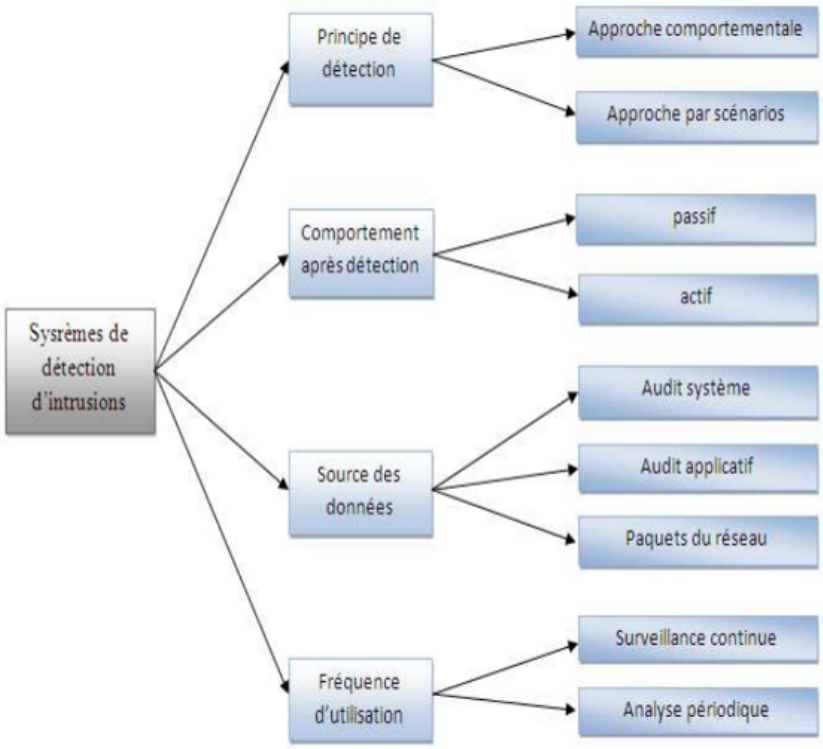


FIGURE I.10 – La classification des systèmes de détection d'intrusions informatiques [Bouras et Fethallah, 2017]

### I.7.4.1 Classification selon la méthode de détection

Deux différentes techniques existent dans cette catégorie, la première consiste à détecter des signatures d'attaques connues dans les paquets circulant sur le réseau. La seconde, consiste quant à elle, à détecter une activité suspecte dans le comportement de l'utilisateur.

1. **Approche par scénario (détecter par signature)** Cette technique s'appuie sur la connaissance des techniques utilisées par les attaquants pour déduire des scénarios typiques. Elle ne tient pas compte des actions passées de l'utilisateur et utilise des signatures d'attaques (ensemble de caractéristiques permettant d'identifier une activité intrusive : une chaîne alphanumérique, une taille de paquet inhabituelle, une trame formatée de manière suspecte,etc,[Righidel, 2006].

   - **Les avantages**
     - Reconnaissance des attaques sans générer trop de fausses alarmes (faux positifs) ;
     - Capable de diagnostiquer rapidement l'utilisation d'une technique d'attaque ou d'un outil d'attaque spécifiques ;
     - Possibilité d'aider les administrateurs système (moyennant leur niveau d'expertise) à traquer un problème de sécurité en initiant des procédures de gestions d'incidents [Xin et Yun-jie, 2010].
   - **Les inconvénients**
     - Impossibilité de détecter des attaques non connues, et donc nécessité de mettre à jour régulièrement la base de signature.
     - Une attaque n'est pas toujours identique à 100% à sa signature, une moindre différence pourra provoquer la nondétection de l'attaque [Xin et Yun-jie, 2010].

2. **Approche comportementale**
   Le but de cette approche est la prédiction du comportement. La mise en œuvre d'un SDI comportemental comprend toujours une phase d'apprentissage durant de laquelle il va découvrir le fonctionnement normal du système à surveiller, il va constituer un profil. Ainsi des attaques inconnues peuvent être détectées contrairement à l'approche par scénario. Une fois le profil établi, tout comportement qui s'éloigne trop du comportement habituel déclenche une alarme de sécurité, or, tout comportement inhabituel du système ne signifie pas forcément un comportement hostile, ce qui peut générer un nombre élevé de fausses alarmes. La création du profil peut se faire grâce aux différents paramètres tels que la bande passante, la durée de connexion et les ressources utilisées [Hay et Cid, 2008].

   - **Les avantages [Vigna et al., 2000] :**
     - La détection d'anomalies permet de détecter un comportement non usuel et ainsi offrent la possibilité de trouver des symptômes d'une attaque sans en connaître les détails ;
     - Peut permettre de produire de l'information qui peut être utilisée pour définir des signatures utilisables pour les systèmes à signatures .
   - **Les inconvenients [Vigna et al., 2000] :**
     - Produit une quantité énorme de fausses alertes à cause du caractère imprévisible des utilisateurs et des réseaux (ruptures...) ;
     - Demande une intense phase d'apprentissage pour caractériser la normalité des comportements ;
     - Si un pirate attaque pendant cette phase, ses actions seront assimilées à un profil utilisateur, et donc passeront inaperçues lorsque le système de détection sera complètement opérationnel ;
     - Un pirate peut s'introduire dans le système et modifier le fichier contenant les profils des utilisateurs, ce qui lui permettra de mettre en place son attaque sans qu'elle ne soit détectée [Vigna et al., 2000].

3. **Comparaison entre les deux approches de détection, par comportement et par signature**
   Le tableau ci-dessous représente un récapitulatif des deux approches de détection [Hay et Cid, 2008].

TABLE I.1 – Comparaison entre les deux approches de détection, par comportement et par signature [Hay et Cid, 2008].

| **Signature** | **Comportement** |
|---|---|
| Spécification complexe des scénarios | Taille des automates générés |
| Nombre de faux positifs réduit | Les faux positifs nombreux |
| Ne connait pas de nouvelles attaques | Pris en compte aux nouvelles attaques |
| Mise à jour rapide | Mise à jour délicate |
| Protection facile à contourner | Phase risque d'entraînement |

#### I.7.4.2 Classification selon le comportement après la détection

Une façon de classer les systèmes de détection d'intrusions consiste à les classer par type de réaction lorsqu'une attaque est détectée :

1. **Les systèmes passifs**
   La plupart des systèmes de détection d'intrusions n'apportent qu'une réponse passive à l'intrusion. Lorsqu'une attaque est détectée, ils génèrent une alarme et notifient l'administrateur système par e-mail, message dans une console, voir par beeper. C'est alors lui qui devra prendre les mesures qui s'imposent [Michel ,2003].

2. **Les systèmes actifs**
   D'autres systèmes peuvent, en plus de la notification de l'opérateur, prendre automatiquement des mesures pour stopper l'attaque en cours. Par exemple, ils peuvent couper les connexions suspectes ou même, pour une attaque externe, reconfigurer le firewall pour qu'il refuse tout ce qui vient du site incriminé. Des outils tels que RealSecure ou NetProwler proposent ce type de réaction. Toutefois, il apparaît que ce type de fonctionnalité automatique est potentiellement dangereux, car il peut mener à des Denis de services provoqués par le SDI. Un attaquant déterminé peut, par exemple, tromper le SDI en usurpant des adresses du réseau [Michel ,2003].

### I.7.4.3 Classification selon la source de données

La source de données à analyser est une caractéristique essentielle des SDIs et un critère important pour leur classification. La manière la plus connue pour classifier les SDIx est de les grouper par sources d'informations (sondes). Certains SDIs analysent des paquets capturés à partir du réseau, d'autres SDIs analysent des informations produites par le système d'exploitation (hôte) ou par des applications pour la recherche des signes d'intrusions.

1. **Les SDIs basés Réseau (SDIR)**
   Les SDIRs (NIDS : Network-based Intrusion Detection System) analysent le trafic réseau, ils constituent généralement une sonde qui "écoute" sur le segment de réseau à surveiller et un moteur qui réalise l'analyse du trafic afin de détecter les divergences face au modèle de référence. La plupart des SDIR sont aussi dits SDI inline car ils analysent le flux en temps réel. Pour cette raison, la question des performances est très importante. De tels SDI doivent être de plus en plus performants afin d'analyser les volumes de données pouvant transiter sur les réseaux et qui ne cessent d'accroître (voir la figure I.11) [Jacon, 2006].

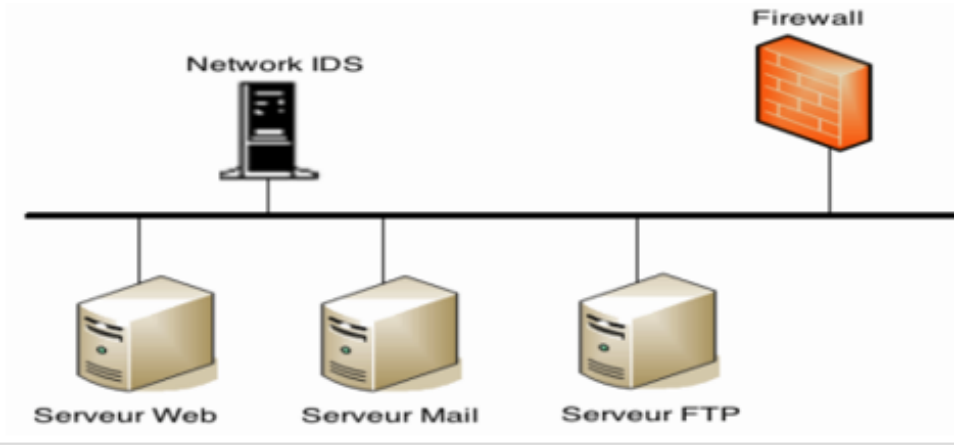


FIGURE I.11 – Système de détection d'intrusion réseau [Jacon, 2006].

2. **Les SDIs basés système ou Hôte (SDIH)**
   Les SDIs (HIDS : Host-based Intrusion Detection System) analysent quant à eux le fonctionnement ou l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. Pour cela, ils ont pour mission d'analyser les journaux systèmes, de vérifier l'accès

aux appels systèmes, de vérifier l'intégrité des systèmes de fichiers.
Ils sont très dépendants du système qu'il les héberge, il faut donc des outils spécifiques en fonction des systèmes déployés. Ces SDIs peuvent s'appuyer sur des fonctionnalités d'audit propres ou non au système d'exploitation, pour en vérifier l'intégrité, et générer des alertes. Il faut cependant noter qu'ils sont incapables de détecter les attaques exploitant les faiblesses de la pile IP du système, notamment les Dénis de service comme SYN FLOOD ou autre (voir la figure I.12) [Jacon, 2006].

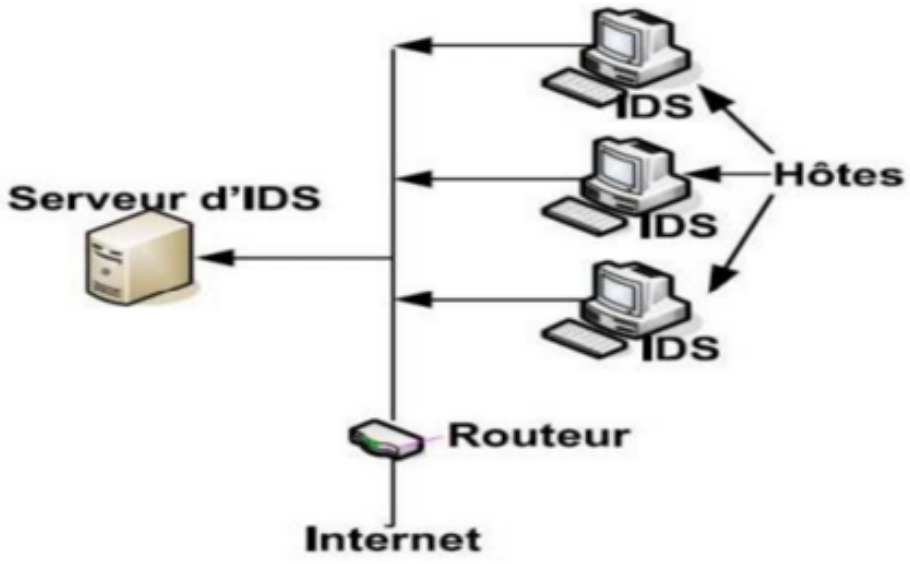


FIGURE I.12 – Architecture d'un SDIH [Jacon, 2006].

3. **Les SDIs hybrides**
Les SDIs hybrides rassemblent les caractéristiques des SDIR et SDIH. Ils permettent, en un seul outil, de surveiller le réseau et les terminaux. Les sondes sont placées en des points stratégiques, et agissent comme SDIR et / ou SDIH suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout, et agréger / lier les informations d'origines multiples. Ainsi, on comprend que les SDIs hybrides sont basés sur une architecture distribuée, où chaque composant unifie son format d'envoi (par exemple IDMEF : Intrusion Detection Message Exchange Format). Cela permet de communiquer d'extraire des alertes plus pertinentes [Jabou et al., 2009].

## I.8 Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur la sécurité informatique en décrivent les différents types et catégories d'attaques, où nous avons illustré quelques-unes telles que les virus, IP Spoofing, et DOS, qui menace hélas les systèmes informatiques. Il devient donc urgent de mettre en place des mécanismes pour satisfaire au mieux les exigences de la sécurité. Parmi les solutions existantes, nous trouvons les systèmes de détection d'intrusion qui font objet de notre projet et notamment notre second chapitre où nous verrons un état de l'art sur les SDIs ainsi que les méthodes utilisées pour mettre en œuvre ces derniers, plus particulièrement ceux issues de l'apprentissage automatique.

# Chapitre II

# État de l'art

# II.1 Introduction

L'accès par Internet à une masse croissante d'informations actualisées en temps réel accroît le risque d'avoir des activités anormales (attaques), où on ne prête hélas jamais beaucoup d'attention aux défis qui accompagnent l'utilisation de la toile mondiale. Il est par conséquent indispensable de réaliser une politique de sécurité qui répond aux besoins des personnes et des entreprises. Parmi les solutions existantes on trouve les systèmes de détection d'intrusions, qui sont devenus une partie intégrante de tout réseau dont l'objectif de le rendre plus stable. Les SDIs font généralement appel aux techniques issues de l'intelligence artificielle afin de renforcer la sécurité et d'augmenter les taux de détection des attaques, tout en réduisant les fausses alarmes. Dans ce chapitre, nous allons présenter un état de l'art sur quelques algorithmes appliquant un apprentissage automatique, ainsi que quelques travaux récents sur les SDIs.

# II.2 Les Approches utilisées lors de la réalisation des SDIs

Nous allons dans cette partie présenter les principales méthodes et approches dédiées à la réalisation des systèmes de détection d'intrusions, où nous illustrerons quelques notions de base concernant les algorithmes utilisés afin d'atteindre cet objectif. Ces algorithmes sont issus de l'Intelligence Artificielle(IA), un champ de recherche qui regroupe l'ensemble des techniques et méthodes qui tendent à comprendre et à reproduire le fonctionnement de cerveau humain, et qui a donnée naissance à l'apprentissage automatique et l'apprentissage profond qui ont révolutionné ce domaine.

## II.2.1 L'Apprentissage Automatique (AA)

L'apprentissage automatique (Machine Learning ML) est un sous-ensemble de l'IA, un vaste domaine interdisciplinaire, qui s'appuie sur des concepts informatiques sciences, statistiques, sciences cognitives,ingénierie, la théorie de l'optimisation et de nombreuses autres disciplines de mathématiques et sciences [Ghahramani, 2004].
L'AA est axé sur la création des systèmes qui apprennent et améliorent les performances, en se basant sur des données qu'ils traitent. Les algorithmes d'apprentissage automatique entrent en jeu pour optimiser, fluidifier, et sécuriser cette dernière [Clayton, 2019].
Il existe trois grandes familles d'apprentissage, l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement [Ghahramani, 2004].

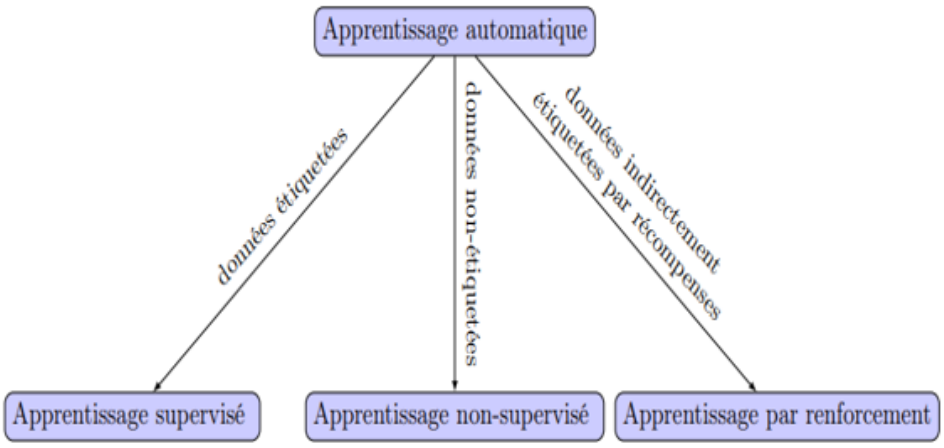


FIGURE II.1 – Les types de l'apprentissage automatique [Ghahramani, 2004].

#### II.2.1.1 Apprentissage supervisé

L'apprentissage supervisé est le type d'apprentissage automatique dans lequel les machines sont entraînées à l'aide de données d'apprentissage qui sont étiquetées. Comme entré, l'algorithme possède des données (base d'apprentissage) et le résultat attendu qui étiqueter. Une fois que l'apprentissage est terminé, le modèle obtenu, permettra de faire des prédictions [Jalam, 2003]. Il y a deux types d'apprentissage supervisé [Bensiah, 2020] :

- **Classification :** la classification consiste à trouver le lien entre une variable d'entrée (X) et une variable de sortie discrète (Y).
- **Régression :** la régression consiste à prédire une valeur continue pour la variable de sortie.

- Les arbres de discision ;
- Les Machines à vecteurs de support (Support Vector Machin MVS) ;
- Les Réseaux de Neurones RNs ;
- La méthode de classification naïve bayésienne (Naïve Bayes).

1. **Arbre de décision**
   Les arbres de décision sont des algorithmes de classification supervisés qui sont souvent utilisés pour représenter des connaissances, des informations ou encore des observations qu'on appelle aussi des exemples. Un arbre de décision est une représentation sous forme graphique (voir la figure II.2) d'un diagramme illustrant des règles de décision, il est composé de : [Haddaoui, 2015]

   - Nœuds de décision : autrement dis « nœuds internes » chaque nœud est étiqueté par un test portant généralement sur un seul et unique attribut.
   - Branches : ce sont des arcs issus des nœuds de décision correspondant à l'une des valeurs possibles des attributs sélectionnés.
   - Nœuds feuilles : comprenant des objets qui appartiennent à la même classe.

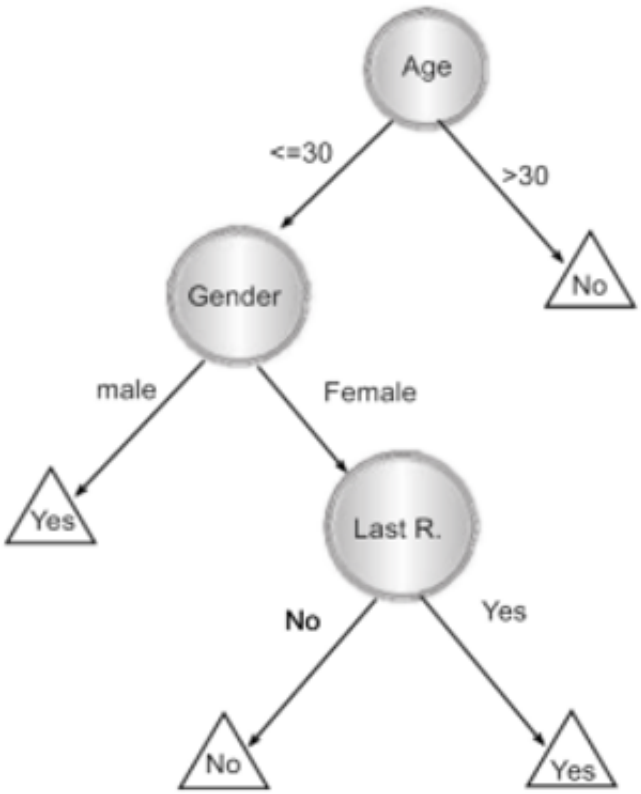


FIGURE II.2 – Exemple d'arbre de décision [Rokach et Maimon, 2005].

(a) **Algorithme de base**

**Entrée : échantillon S**
**Début**
**Initialiser l'arbre courant à l'arbre vide ; la racine est le nœud courant**
**Répéter**
**Décider si le nœud courant est terminal**
**Si le nœud est terminal alors**
**Lui affecter une classe**
**Sinon Sélectionner un test et créer autant de nouveaux nœuds fils qu'il y a de réponses possibles au test**
**FinSi**
**Passer au nœud suivant non exploré s'il en existe**
**Jusqu'à obtenir un arbre de décision**

FIGURE II.3 – Algorithme de base d'arbre de décision [Haddaoui, 2015].

(b) **Approches d'SDI proposées avec arbre de décision** Plusieurs auteurs ont fait appel aux arbres de décision dont l'objectif est de réaliser un système de détection d'intrusion tel qu'Elekar et Waghmare [Elekar et Waghmare, 2015] qui ont implémenté différents classifieurs tels que l'arbre de décision C4.5, foret aléatoire « Random Forest » et l'arbre de Hoeffding pour la détection des intrusions, ils comparent les résultats à l'aide de WEKA. Les résultats montrent que l'arbre de Hoeffding donne le meilleur résultat parmi les différents classifieurs pour détecter les attaques sur les données de test. Bajaj et Arora [Bajaj et Arora, 2013] ont discuté les différentes méthodes de sélection distinctives telles que le gain d'informations, le rapport de gain et la sélection de caractéristiques basée sur la corrélation où ils ont sélectionné 33 caractéristiques sur 41, puis ils ont classé ces caractéristiques pour comparer les résultats. Les résultats obtenus avec l'algorithme SCA (Simple CART Algorithm) sont d'une grande précision ,par

rapport à ceus de l'arbre de décision C4.5. Une des recherches similaires à la classification de certaines caractéristiques est le travail d'Alazab et al. [Alazab et al, 2012] où ils ont utilisé le gain d'informations et l'arbre de décision pour détecter à la fois les anciennes et les nouvelles attaques.

2. **Les Machines à Vecteurs de Support (MVS)** Les machines à vecteurs de support (MVS : Support vecteur machine) ou séparateurs à vaste marge ont été introduites par Vapnik en 1995 [Vapnik, 1995]. C'est un ensemble de techniques d'apprentissage destinées à résoudre des problèmes de classification (décider à quelle classe appartient un échantillon) et de régression (prédire la valeur numérique d'une variable). Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. Elles reposent sur deux notions principales : la notion de marge maximale et la notion de fonction noyau. Les MVSs ont été appliquées avec succès dans divers domaines comme la vérification et la reconnaissance, telle que l'identification de visages, la reconnaissance de caractères manuscrits et des chiffres, la vérification et reconnaissance du discours du parlant et de la rédaction [Vapnik, 1995].

   (a) **Principe des MVSs** À l'origine, la méthode des machines à vecteurs de supports était destinée à la classification binaire. D'une manière simple, soient x1, x2,. . ., xn un ensemble d'observations décrites par des attributs prédéfinis. Soient C1 et C2 deux classes possibles. Étant donné un étiquetage connu des n observations en termes de classes, la méthode MVS cherche à estimer les paramètres d'un classifieur pour étiqueter une nouvelle observation. En effet, cela revient à chercher l'hyperplan optimal, celui dont la distance aux exemples d'apprentissage est maximale. Cette distance est appelée marge entre l'hyperplan et les exemples. Les points les plus proches qui suffisent à déterminer cet hyperplan sont appelés vecteurs de support. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [Balm, 1997].
   Donc, le but de MVS est de trouver un classifieur qui va séparer les données et maximiser la distance entre ces deux classes, c'est un classifieur linéaire appelé hyperplan. Les points-clés de cette methode sont définis dans ce qui suit.

   - **Hyperplan :** on appelle hyperplan séparateur un hyperplan qui sépare les deux classes (voir la figure II.4), en particulier il sépare leurs points d'apprentissage. Comme il n'est en général pas possible d'en trouver un, on se contentera donc de chercher un hyperplan discriminant qui est une approximation au sens d'un critère à fixer (maximiser la distance entre ces deux classes) [Mohamadally et Fomani, 2006].

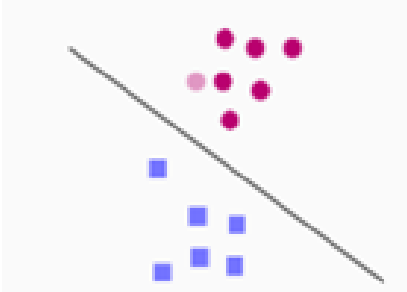

FIGURE II.4 – Exemple d'un hyperplan [Mohamadally et Fomani, 2006].

- **Les vecteurs supports :** pour déterminer l'hyperplan séparable des MVS il faudrait trouver les points les plus proches (i.e. les points de la frontière entre les deux classes des données) parmi l'ensemble total d'apprentissage, ces points sont appelés vecteurs supports (voir la figure II.5 ) [Mohamadally et Fomani, 2006].

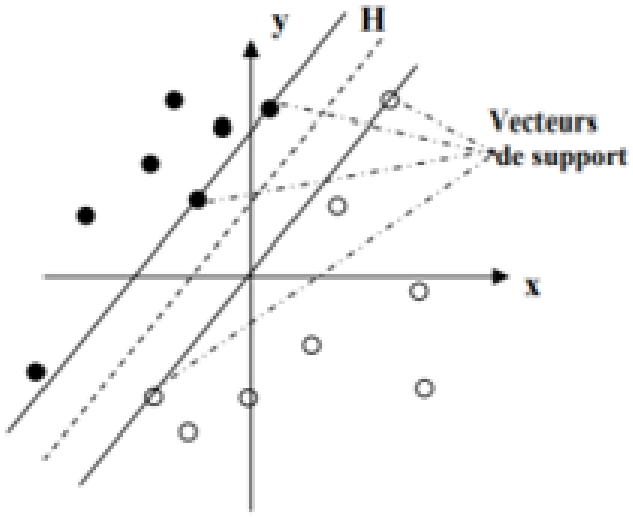


FIGURE II.5 – Exemple de vecteurs de support [Mohamadally et Fomani, 2006].

- **La marge :** il existe une infinité hyperplans capables de séparer parfaitement les deux classes d'exemples. Le principe des MVSs est de choisir celui qui va maximiser la distance minimale entre l'hyperplan et les exemples d'apprentissage ( la distance entre l'hyperplan et les vecteurs supports), cette distance est appelée la marge maximale. Le but de la marge maximale est d'avoir plus de sécurité lors de la classification d'un nouvel exemple, ceci est illustré dans (voir la figure II.6) présentée ci-dessous, où la partie droite montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge, contrairement à la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé [Mohamadally et Fomani, 2006].

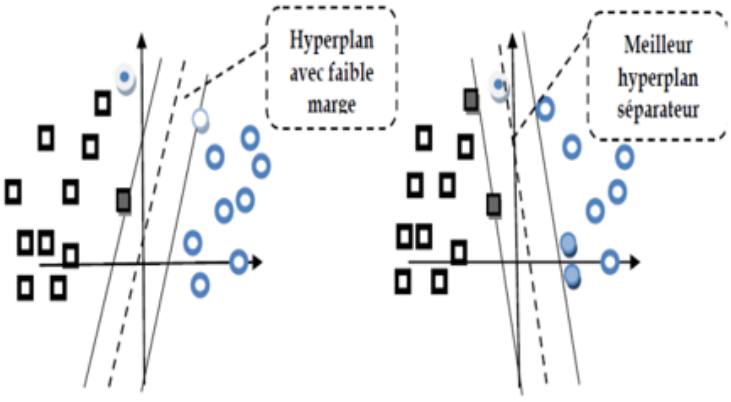


FIGURE II.6 – Marge optimale et marge faible[Mohamadally et Fomani, 2006].

(b) **Différents types d'MVS**

- **MVS linéaire :** c'est le cas où les données d'apprentissage sont linéairement séparables, ceci dit qu'il existe un hyperplan qui sépare les données (voir la figure II.7 ) [Mariette et Rahul, 2015].

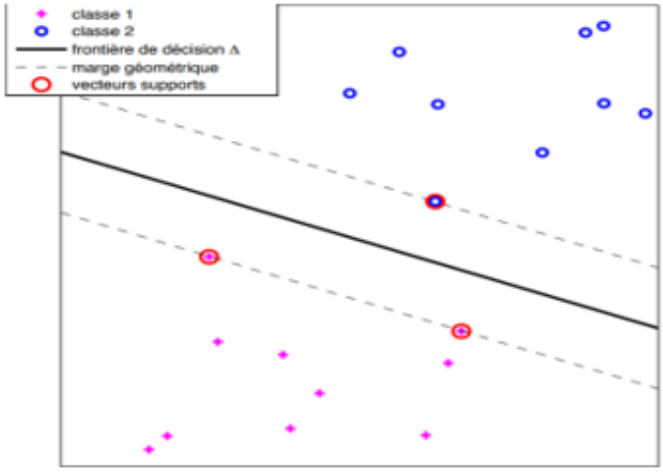


FIGURE II.7 – MVS linéaire [Mariette et Rahul, 2015].

La fonction de décision est une fonction linéaire de type :

$$Y = W.X_i + b \tag{II.1}$$

Où $W \in R^n$ dans vecteur de poids et $b \in R^n$ est le biais, les $x_i \in R^n$ sont les exemples (où chaque donnée est représentée par un vecteur).

- **MVS à marge dure :**[Mariette et Rahul, 2015] l'hyperplan séparateur est représenté par l'équation suivante :

$$H(x) = W^T.X + b \tag{II.2}$$

Puisque les deux classes sont linéairement séparables, il n'existe aucun exemple qui se situe sur l'hyperplan, c-à-d qui satisfait H(x) = 0 (voir la figue II.8). Il convient alors d'utiliser la fonction de décisions suivante :

$$classe = \begin{cases} 1 & si\ H(x) > 1 \\ -1 & si\ H(x) < -1 \end{cases} \tag{II.3}$$

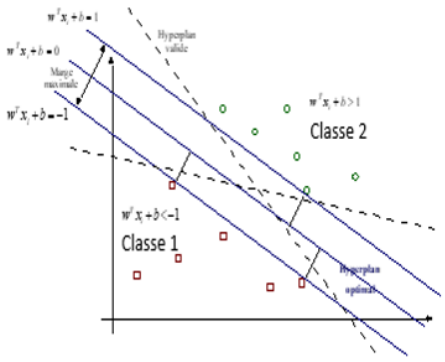


FIGURE II.8 – Illustration d'un hyperplan optimal [Cornuéjols, 2002].

- **MVS à Marge molle :** la deuxième méthode dite à marge molle propose de chercher un hyperplan qui minimise le nombre d'erreurs si les données sont difficiles à séparer ceci est illustré dans (voir la figure II.9) présentée ci-dessous [Mariette et Rahul, 2015].

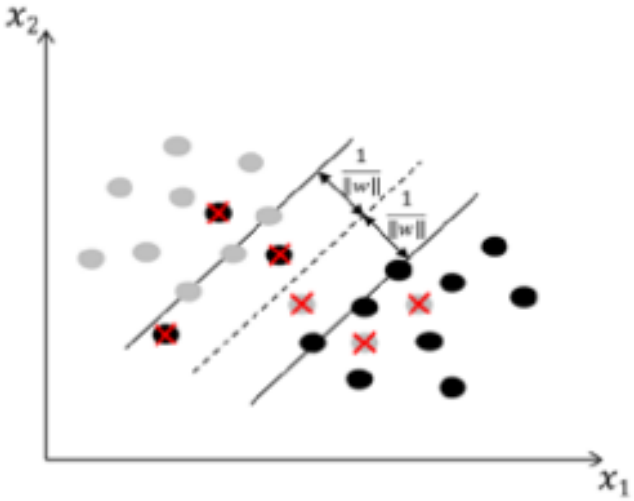


FIGURE II.9 – MVS à Marge molle [Mohamadally et Fomani, 2006]

- **MVS non linéaire :** ce type de MVS propose de transformer les données dans un espace de dimension supérieure lorsque les données sont non-séparables linéairement. Cette transformation des données peut permettre une séparation linéaire des exemples dans un nouvel espace. Cette nouvelle dimension est appelée « espace de rescription » (voir la figure II.10) [Mohamadally et Fomani, 2006].

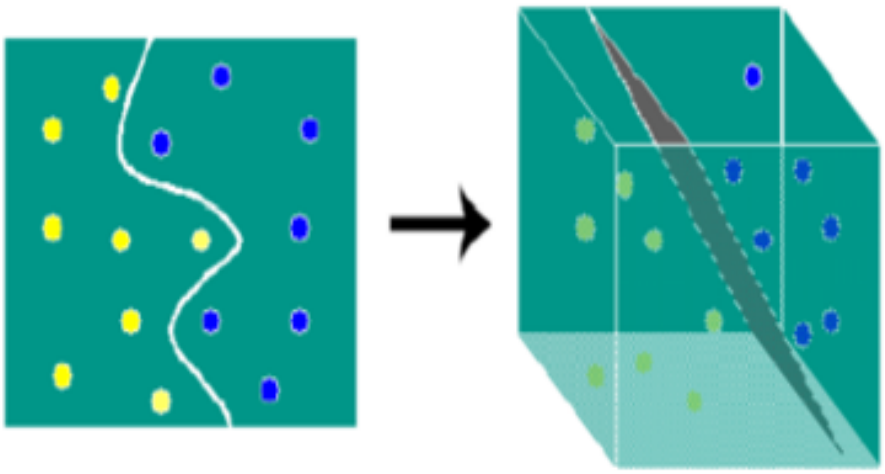

FIGURE II.10 – Passage d'une MVS linaire vers une MVS non linéaire [Mohamadally et Fomani, 2006]

(c) **Usage des MVSs dans les SDIs**
MVS est essentiellement une méthode d'apprentissage automatique supervisée conçue pour la classification binaire. L'utilisation de MVS dans le domaine des SDIs présente certaines limites. MVS étant une méthode d'apprentissage automatique supervisé, il faut des informations étiquetées pour un apprentissage efficace. MVS a la limitation structurelle intrinsèque du classificateur binaire, c'est-à-dire qu'il ne peut gérer que la classification de classe binaire alors que la détection d'intrusion nécessite une classification multi-classe [Shon et al., 2005]. Jha et Ragha quant eux [Jha et Ragha, 2013] proposent un modèle de MVS, dont l'apprentissage est fait via l'ensemble de données NSL-KDD. Les auteurs ont appliqué une réduction de dimensionnalité sur la base avant de faire la classification avec MVS en utilisant le rapport de gain . ils sélectionnent les meilleures caractéristiques via la méthode K-means .

3. **Les Réseaux de Neurones Artificiels (RNAs)** Un RNA est un modèle informatique basé sur la structure et les fonctions des réseaux de neurones biologiques. Les informations qui circulent à travers le réseau affectent la structure de l'RNA,(voir la figure II.11) car un ce dernier apprend en fonction de ces entrées (x) et sorties (s) (base d'apprentissage étiqueté) et détermine les valeurs de la matrice des poids (W). Par conséquent les RNAs sont considérées comme des outils de modélisation de données statistiques [Gershenson, 2003].

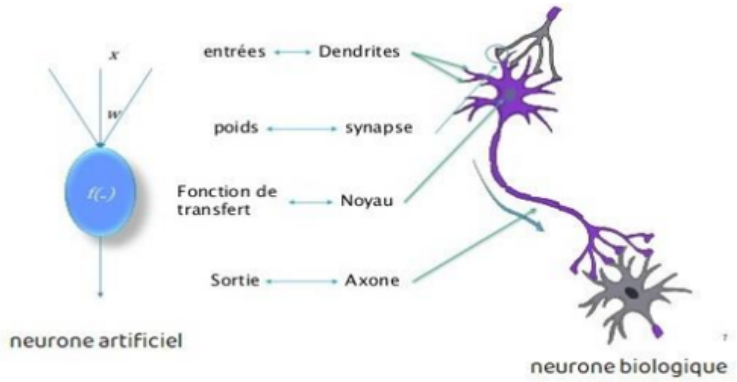


FIGURE II.11 – mémoire/Neurone biologique et neurone artificiel [Gershenson, 2003].

(a) **Avantage des RNAs**
Les RNAs ont plusieurs avantages tels que leurs capacités d'apprentissage adaptatif, ils peuvent créer leurs propres organisations ou représentation des informations qu'ils reçoivent durant l'apprentissage. De plus, les RNAs ont la capacité de fonctionner en temps réel grâce à leurs calculs qui peuvent être effectués en parallèle. Finalement, les RNAs ont une tolérance aux pannes via un codage d'informations redondant, la destruction partielle d'un réseau entraine la dégradation correspondante des performances. Cependant, certaines capacités du réseau peuvent être conservées même en cas de dommages importants du RN [Eluyode et Akomolafe, 2013].

(b) **Architecture des réseaux de neurones**
De façon générale, on définit un réseau de neurones par les cinq paramètres (voir la figure II.12)[Rfia, 2012] à savoir :

i. La nature des entrées (booléenne ou réelle). x1, x2,. . ., xn ;
ii. La fonction d'entrée totale, définissant le prétraitement effectué sur les entrées ;
iii. La fonction d'activation du neurone définissant son état interne en fonction de la somme pondérée de ses entrées ;

iv. La fonction de sortie calculant la sortie du neurone en fonction de son état d'activation ;

v. La nature des sorties du neurone.

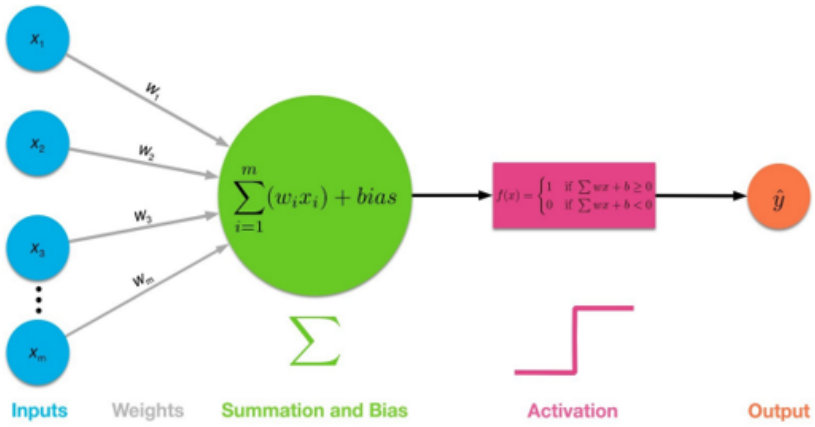


FIGURE II.12 – Présentation d'un neurone artificiel [Rfia, 2012].

(c) **Les SDIs avec les RNAs**

Les SDIs basés sur les RNAs peuvent être plus efficaces, s'ils sont bien entrainés avec des ensembles de données étiquetés. Nazir [Nazir, 2013] présente dans son article une étude comparative entre plusieurs SDIR basés sur différentes architecures de RN. Quatre catégories d'attaques ont été prises en compte dans les modèles développés. L'auteur a évalué les RN du type MLFF, CEBP et ELBP. Les résultats montrent que plus le nombre de paquets distincts pour l'apprentissage du réseau de neurones est élevé, plus la capacité de généralisation du réseau est améliorée, ce qui conduit à un meilleur taux de classification. Les résultats montrent que le réseau de neurones ELBP avec la fonction d'apprentissage "trainrp" obtient de meilleurs taux de classification que tous les autres RNs . La modification du nombre de neurones dans la couche cachée a entraîné une modification du taux de classification , lorsque ce nombre est proche de la taille du vecteur d'entrée (c'est-à-dire 41), le taux de classification obtenu est meilleur que celui obtenu avec un nombre de neurones infeireur. Pour tous les types d'attaques considérées. Raman et son équipe [Raman et al, 2017] proposent une nouvelle approche destinée à la detection d'intrusion basée sur la propriété de Helly (HP : Helly Property) d'un hypergraphe et un Réseau de Neurones Probabiliste (RNP) basé sur l'arithmétique modulaire.

4. **L'algorithme des K Plus Proches Voisins (k-PPV)**

L'algorithme des k-Plus Proches Voisins (k-PPV) est un des plus anciens mais aussi des plus simples algorithmes de classification supervisée où le but de ses algorithmes est de construire une fonction capable d'étiqueter/classer au mieux un nouvel objet à partir d'un échantillon d'objets étiquetés/classés qui ne contient pas cet objet. Contrairement à beaucoup d'autres méthodes d'apprentissage automatique telles que les réseaux de neurones artificiels l'algorithme k-PPV n'a pas de phase de détermination de paramètres d'une fonction par le biais d'une optimisation mathématique qui signifie qu'aucun apprentissage ne prend place en réalité (voir la figure II.13)[Ralaivola, 2007] [Morin, 2014].

5. **L'algorithme de classification Naïve bayésienne (NB)**

(a) **Définition**

Les algorithmes NB sont souvent utilisés dans la catégorisation et la classification de documents. Ils permettent d'estimer la probabilité de chaque classe parmi les exemples,

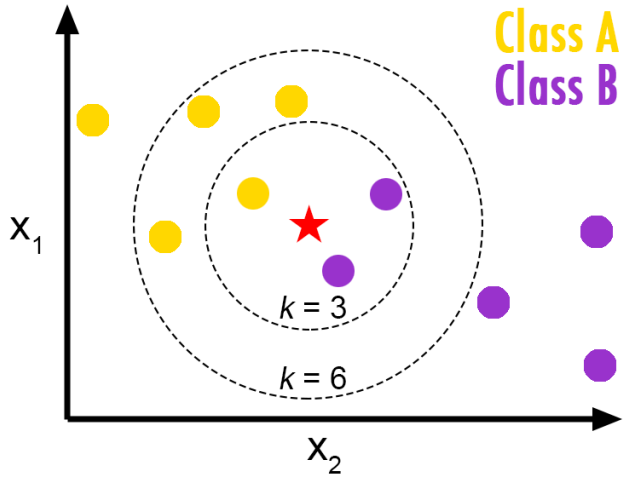


FIGURE II.13 – Principe de fonctionnement de l'algorithme pour k=3 et k=6 [Justyyuk, 2016].

étant donné un document, et affectent à ce dernier la classe la plus probable. On appel ce procédé «Prior probabilities» [Nakache, 2007] C'est une technique de classification basée sur le théorème de Bayes avec une hypothèse d'indépendance parmi les prédicateurs. En termes simples, un classificateur naïf bayésien suppose que la présence d'une fonctionnalité particulière dans une classe, n'est pas liée à la présence d'une autre fonctionnalité. Par exemple, un fruit peut être considéré comme une pomme s'il est rouge, rond et d'environ 3 pouces de diamètre. Même si ces caractéristiques dépendent les unes des autres ou de l'existence d'autres caractéristiques, toutes ces propriétés contribuent indépendamment à la probabilité que ce fruit soit une pomme et c'est pourquoi il est appelé «naïf».
Basé sur un calcul de probabilité P (A|B), qui veut dire la probabilité que l'événement A se réalise sachant que l'événement B s'est déjà réalisé. Le théorème de bayes :

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{II.4}$$

(b) **Les SDIs avec la méthode naïve bayésienne**
Mukherjeea et Harmaa [Mukherjeea et Harmaa, 2012] ont discuté l'importance de réduire les caractéristiques (champs d'un paquet) pour créer un SDI efficace. Ils ont étudié les performances trois méthodes de réduction à savoir, le gain d'information, rapport de gain et la sélection des caractéristiques basée sur la corrélation. Ils ont proposé une méthode de réduction basée sur la vitalité des champs, ou il est question d'enlever un champs de tout les paquets de la base et voir l'impacte sur les résultats Ils ont appliqué le classificateur NB sur l'ensemble des données NSL KDD. Les résultats expérimentaux ont montré que certaines caractéristiques sélectionnées offrent de meilleures performances de détection d'intrusions.

#### II.2.1.2 Apprentissage non supervisé

Contrairement à l'apprentissage supervisé, seules les informations en entrée sont fournies au système. Celui-ci doit donc déterminer ses sorties en fonction des similarités détectées entre les différentes entrées, c'est-à-dire en fonction d'une règle auto-organisatrice. Le système est appelé donc à découvrir les régularités présentes dans ces configurations qui peuvent servir à les diviser en groupes de configurations semblables [Mezaache, 2007]. Il y a deux types d'apprentissage non supervisé : [Géron, 2017]

- Regroupement(Clustering) : c'est une méthode d'analyse statistique utilisée pour organiser des données brutes en silos homogènes, à l'intérieur de chaque grappe, les données sont regroupées selon une caractéristique commune.
- Réduction de la dimension : l'objectif est de simplifier les données sans perdre trop d'informations, à titre d'exemple, fusionner plusieurs caractéristiques en un seul caractère.

Parmi les algorithmes les plus célèbres utilisés dans cette approche, nous trouvons l'algorithme des K-Moyennes, un algorithme de regroupement (Clustering). Il regroupe dans les mêmes Clusters (Groupes) les données similaires, il utilise un raffinement itératif pour produire un résultat final [Issarane, 2019].

#### II.2.1.3 Apprentissage par Renforcement (AR)

Le AR (ou RL pour Reinforcement Learning) est un type d'apprentissage issu de la théorie du contrôle optimal. Dans ce paradigme, un système, dénommé agent, apprend par essais-erreurs à contrôler un environnement par le biais d'actions choisies de façon séquentielle et d'une optimisation réalisée sur la base de récompenses indicatrices de la qualité de ses décisions [Sutton et Barto, 2015]. Comme le montre (voir la figure II.14), à chaque instant t, l'agent perçoit son environnement à travers un état, noté st, ensuite l'agent devra prendre une action at parmi un jeu d'actions possibles. Cette prise de décision aura pour conséquence la modification de l'état courant de l'environnement, qui transitera alors vers un nouvel état st+1. L'agent doit apprendre quelle action choisir dans chaque état afin de suivre une séquence d'action qui lui soit la plus favorable possible. Donc le but de l'apprentissage par renforcement est d'adapter un agent à un environnement en tenant compte de récompenses/punitions (les renforcements).
L'apprentissage par renforcement utilise une évaluation pour améliorer les performances du système. Cette forme d'apprentissage constitue une méthode d'apprentissage par essais et erreurs qui s'appellent méthode de Markov qui est un Processus de Décision de Markov (PDM). C'est un modèle stochastique qui peut être vu comme une chaîne de Markov contrôlée, c'est-à-dire à laquelle on ajoute une composante décisionnelle [Mezaache, 2007].
C'est un outil mathématique qui permet de formaliser l'apprentissage par renforcement. Un PDM

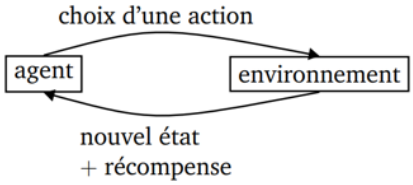


FIGURE II.14 – Apprentissage par renforcement [Richard et Perrson, 2009].

vérifie la propriété de Markov qui peut être résumée de la manière suivante : prédire le futur partir du présent est tout aussi efficace que le prédire en possédant des informations concernant le passé [Mezaache, 2007].

### II.2.2 Apprentissage Profond : AP (Deep Learning DL)

La notion d'apprentissage profond est tout d'abord une traduction directe du terme anglais « Deep Learning » , c'est un type d'apprentissage automatique(Machine Learning ML) (Voir figure II.15) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées. L'AP a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composées de

divers transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minime. Par exemple pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles, etc. des couches supérieures identifient des formes, des combinaisons [Maina et Moumen, 2021 ]. Les caractéristiques essentielles du traitement ne seront plus identifiées par un traitement humain dans un algorithme préalable, mais directement par l'algorithme d'apprentissage profond [Nuageo, 2017]. Cette approche permet, à partir de données d'entrée, d'en extraire une représentation plus riche. Les réseaux profonds sont composés de plusieurs couches de neurones. Chaque couche est une étape qui représente les données de façon un peu plus complexe (abstraite) en se basant sur ce qui a été appris dans la couche précédente. Cette approche est à l'image de l'apprentissage humain qui commence par apprendre des concepts simples, comme l'addition et la soustraction en mathématiques, pour ensuite se baser sur ces concepts afin d'en apprendre des plus complexes, comme la multiplication et le principe de fonction [Stanislas ,2016]. L'apprentissage en profondeur investit progressivement notre quotidien, il est utilisé dans plusieurs domaines tels que la reconnaissance vocale, la synthèse vocale avancée, la segmentation sémantiques d'images, la détection d'intrusions informatiques, etc.

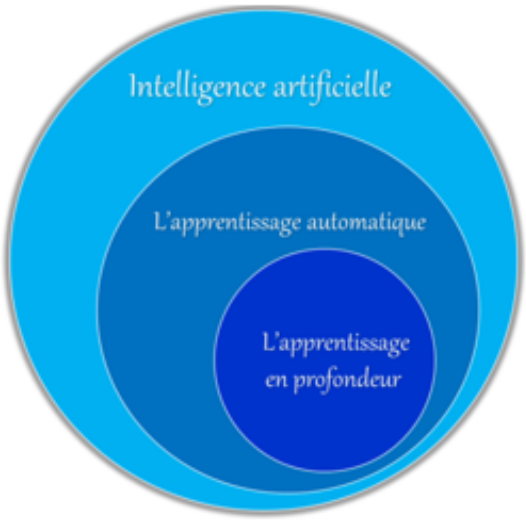


FIGURE II.15 – L'intelligence artificiel et ses sous branches [Moussi et Lounis, 2020].

#### II.2.2.1 Quelques architectures issues de l'apprentissage profond

Il existe un grand nombre de variantes d'architectures profondes. La plupart d'entre elles sont dérivées de certaines architectures parentales originales. Il n'est pas toujours possible de comparer les performances de toutes les architectures, car elles ne sont pas toutes évaluées sur les mêmes ensembles de données. L'AP est un domaine à croissance rapide, et de nouvelles architectures, variantes ou algorithmes apparaissent toutes les semaines [Moualek ,2017].

1. **Les Réseaux de Neurones Convolutionnels (RNC)** Le RNC (CNN pour convolutionnel neural network) est un type de réseau de neurones artificiel. Les RNCs sont des structures de réseaux de neurones particulières puisque l'opération élémentaire n'est plus une multiplication matricielle mais une convolution. Ces réseaux ont été inventés afin de tirer parti de données possédant une structure (spatiale, temporelle) telles que des images, des vidéos, ou encore des signaux temporels. [Valentin, 2015].
   Les trois principaux types de couches pour construire ce type de réseau à savoir, la couche

convolutive, couche de pooling et couche entièrement connectée [Teghlil et Laouira, 2021].

- **Couche convolutive :** c'est la couche la plus importante et le cœur des éléments constitutifs du réseau convolutif, et c'est aussi elle qui effectue le plus de calculs lourds ;
- **Couche de pooling :** Il est courant d'insérer périodiquement une couche Pooling dans ce type d'architecture. Sa fonction est de réduire progressivement la taille spatiale de la représentation pour réduire le nombre de paramètres et de calculs dans le réseau, et donc de contrôler également l'overfitting.
- **Couche entièrement connectée :** les neurones d'une couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente.

Il y a eu quelques études utilisant les RNCs dans le domaine de la détection d'intrusion ; leur plus grand avantage est qu'ils partagent les mêmes noyaux convolutionnels, ce qui permet une réduction du nombre de paramètres et de la quantité de calculs durant l'apprentissage. Vinayakumar et al [Vinayakumar et al., 2017] ont analysé l'efficacité des réseaux neuronaux convolutionnels ( RNC ) en vue d'une réalisation d'un SDI en modélisant les événements du trafic du réseau (paquets TCP/IP) dans des périodes de temps prédéfinies (une série chronologique de données).
Les auteurs ont proposé différentes architectures du RNC contient une couche d'entrée, une couche cachée et une couche de sortie, ou la couche cachée contient une ou plusieurs couches RNC .

2. **Réseaux de Neurones Récurrents (RNRs)**
Les RNRs (RNN pour Recurrent Neural Network) ont été introduit afin de pouvoir traiter des données séquentielles. L'objectif est de modéliser ces données ainsi que les éventuelles dépendances temporelles qui y apparaissent. Les RNNs conservent un état caché qui résume leur historique. En effet, un RNR à l'instant t calcule les sorties en fonction de l'entrée x (x0 . . . xm)t mais aussi de l'état de la couche cachée au temps précédent. Ainsi, il fera évoluer un état interne qui fait office de mémoire à court terme (short-term memory) et qui permet de prendre en compte les dépendances temporelles que manifestent les entrées. Les RNRs ont été développés afin de garder un contexte temporel pour chaque événement en entrée. Ils ont été particulièrement utilisés pour de l'analyse de séries temporelles, de données audio, ou de textes dans lesquelles le contexte est important afin d'analyser chaque nouvelle entrée. L'idée consiste à garder des informations au cours du temps à l'intérieur des couches de neurones afin de donner un contexte aux données analysées. La sortie du RNR, à un instant t, va dépendre non seulement de l'entrée à l'instant t mais également de l'état du RNR calculé à l'instant t – 1 (voir la figure II.16) [Corentin ,2019].

3. **Réseaux Récurrents à Mémoire à Court et à Long Terme (MCLT)**

   (a) **Définition**
   Le MCLT (LSTM pour Long Short Term Memory) est un type spécial de RNR. Les Réseaux neuronaux récurrents présentés dans la section précédente sont capables d'apprendre des règles de mise à jour de séquence arbitraire en théorie. Dans la pratique, cependant, ces modèles oublient rapide- ment le passé. C'est ce qu'on appelle le problème de la disparition de gradient et c'est pourquoi ils ont inventé le MCLT. La cellule MCLT est une adaptation de la couche récurrente qui permet aux signaux plus anciens des couches profondes de se déplacer vers la cellule du présent. La (figure 2.21) ci-dessous illustre une chaîne de trois cellules MCLT [Oinkina et Hakyll, 2015]. Althubiti

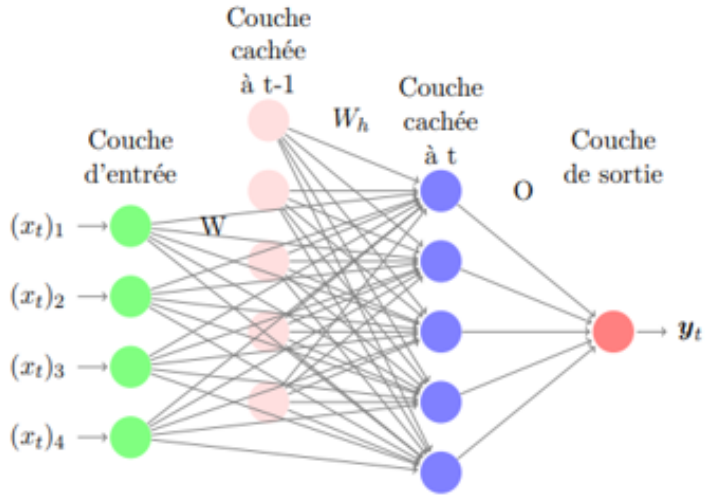


FIGURE II.16 – Exemple d'un réseau RNN [Corentin ,2019].

et son équipe [Althubiti et al, 2018] ont appliqué le MCLT pour la détection des intrusions réseau, ils ont obtenu une précision optimale ou ils ont évalué leur modèle sur l'ensemble de données CIDDS-001. Ils ont également révélé que MCLT fonctionnait mieux que les algorithmes d'apprentissage automatique . Boukhalfa et al [Boukhalfa et al, 2020] ont également proposé une méthode basée sur MCLT pour la classification binaire et multiclasse, la méthode proposée à une plus grande capacité à mémoriser et discriminer le trafic normal du malveillant.

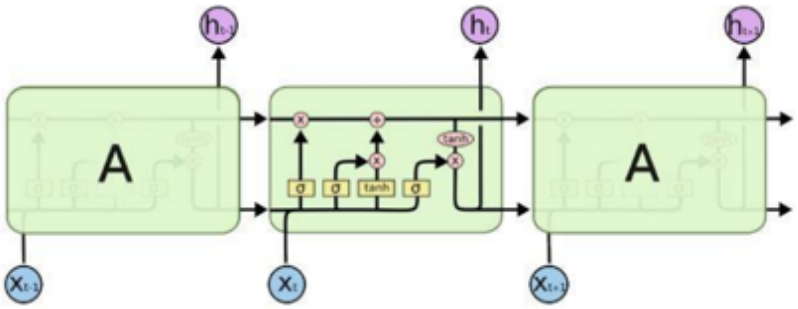


FIGURE II.17 – Fonctionnement de MCLT [Oinkina et Hakyll, 2015].

### II.2.3 D'autres travaux de détection d'intrusions via l'apprentissage automatique

Les constructeurs d'SDI font généralement recoure à l'apprentissage automatique en optant notamment à des hybridations entre plusieurs algorithmes afin d'aboutir à de meilleures performances de détection. Sharma et Yadav [Sharma et Yadav, 2021] proposent un SDI en commençant tout d'abord par une réduction de la dimensionnalité de la base de données KDD CUP 99, via méthode d'élimination récursive en arrière. Par la suite les auteurs réalisent la détection en faisant appel aux arbres de discision connus pour leur interprétation solide des valeurs prédites et leur taux d'adaptabilité élevé pour travailler avec divers ensembles de données ( données catégorielles ou numériques), l'algorithme de la forêt aléatoire (Random Forest Classifier) dans lequel une structure semblable à une forêt est formée à l'aide des arbres de décision multiples , enfin ils font aussi usage des MVSs (voir la figure II.18). Après avoir comparé les trois techniques de classification, la forêt aléatoire s'est avérée plus performante que la méthode MVS avant la sélection des caractéristiques. Mais après avoir implémenté la sélection, MVS a donnée de meilleurs résultats que la forêt aléatoire et l'arbre de décision.


FIGURE II.18 – Diagramme de flux de données du système proposé dans[Sharma et Yadav, 2021].

Quant à Ye ainsi que Song et son équipe [Ye, 2000] [Song et al, 2009] se sont intéressées à l'utilisation des chaines de Markov dans la détection d'anomalies. Ils ont utilisé des modèles de chaîne de Markov pour représenter le profil temporel du comportement normal du système. Les données historiques de ce dernier sont utilisées pour construire le modèle de chaîne de Markov du profil normal. Le comportement du système observé est ensuite analysé par rapport au modèle de chaîne de Markov du profil normal pour vérifier si le comportement observé est conforme ou non au profil normal. Une faible valeur de conformité indique un comportement anormal. Le principal inconvénient du modèle de détection d'anomalies basé sur la chaîne de Markov est que leur complexité et leur temps de calcul augmentent rapidement avec l'augmentation du nombre d'états de transition. Salama et al. [Salama et al, 2011] ont présenté une approche hybride combinant le réseau de croyances profondes (DBN pour Deep Beliefs Network ) et MVS afin de classifier les intrusions du réseau en deux catégories, normale ou attaque. Le réseau DBN est composé de multiples couches de machines Boltzman restreinte (RBM pour Restricted Boltzman machines) qui sont utilisées comme une méthode de réduction de dimension pour obtenir de meilleures caractéristiques d'apprentissage suivies par un classifieur MVS. Les performances de l'approche DBN-MVS proposées sont testées sur NSL-KDD data-set. Le nombre de caractéristiques a été réduit grâce au réseau DBN qui ont été ensuite transmises à un classeur MVS pour effectuer une classification binaire (normale/attaque). Les résultats ont montré que le réseau DBN autant qu'une méthode de réduction des caractéristiques est plus performant par rapport à d'autres méthodes d'analyse de données comme l'ACP, Gain Ratio et à Chi-Square. Une autre approche proposée dans [Mushtaq et al, anneé], c'est une entre le MCLT et auto-encodeur, ce dernier est utilisé pour extraire le sousensemble de caractéristqiue robuste et optimal. Trente caractéristiques sont extraites de la couche de goulot d'étranglement de l'auto-encodeur et introduites dans le modèle MCLT pour la classe finale, Les performances du modéle proposé montrent une amélioration de la précision et du rappel.

## II.3 Conclusion

L'intelligence artificielle est omniprésente dans tous les domaines et notamment celui de la sécurité, donnant naissance à l'apprentissage automatique et l'apprentissage profond qui ont à leur tour révolutionné la reconnaissance des formes et la détection. Les SDIs issus de la sécurité informatique sont répondu dans les organisations d'aujourd'hui, ils sont aussi importants pour les personnes afin de les protéger contre toute atteinte à la vie privée. Il est question alors d'améliorer leurs performances de détection en augmentant le taux de vraies positives et négatives tout en réduisant les fausses alertes. Dans ce chapitre, nous avons tout d'abord présenté les différentes approches de détections d'intrusion quelques soit le critère de classification utilisé, le type de réponse, la source, ou même la méthode de détection. Notant toutefois que nous nous sommes situés tout au long de notre travail dans une approche SDIR basée signatures. Nous avons parcouru par la suite quelques méthodes appliquant un apprentissage qui soit supervisé, ou non, par renforcement ou profond, dont pas mal d'auteurs ont opté pour la réalisation des SDIs. Le chapitre suivant sera consacré à notre approche proposée, aux méthodes que nous avons adoptées, ainsi qu'à l'implémentation, l'application et les résultats obtenus.

# Chapitre III

# Approche proposée et implémentation

# III.1 Introduction

Lors du chapitre précédent nous avons présenté un état de l'art sur les SDIs, où nous avons quelques travaux dans ce contexte faisant appel à l'apprentissage automatique. Tout au long de notre travail, nous nous somme situés dans une approche de SDIR basée signature. Nous avons opté dans un premier temps pour un l'algorithme de classification supervisé , il s'agit de la méthode Naïve Bayésienne(NB), et dans un deuxième temps pour un algorithme basé sur l'apprentissage profond en utilisant les Réseaux Récurrents à Mémoire à Court et à Long (MCLT), puis finalement une hybridation entre ces deux méthodes .L'approche que nous avons proposée et développée repose sur une phase d'apprentissage afin de définir le profil du système, or nous avons utilisé la Base De Données( BDD) NSL-KDD [NSL-KDD, 2009],
Ce chapitre est consacré à la présentation de notre approche proposée, ainsi qu'à la phase d'implémentation. Nous abordons l'aspect pratique de notre application, nous présentons son architecture conceptuelle en illustrant son principe et ses techniques.
Notre SDIR a pour but de sécuriser les systèmes informatiques dans un réseau, tout en essayant de satisfaire le maximum des caractéristiques souhaitées d'un SDI. Il est structuré en deux grandes phases :

- Phase d'apprentissage : modélise le profil normal de fonctionnement du réseau.

- Phase de test : permet de tester le système et détecter les attaques.

Nous allons également présenter les résultats obtenus et quelques comparaisons en matière de précision, rappel, taux de réussite, taux d'erreur, Mesure-F, et taux de classification.

# III.2 Partie 1 : approche proposée

## III.2.1 Principe générale de l'approche proposée

Nous avons créé un SDIR en utilisent des méthodes de classifications qui dépendent fortement de la quantité et la qualité des données d'apprentissage, plus de donnés sont de qualité, plus les résultats sont meilleurs. Dans notre cas, nous avons une masse de données importante, et qui nécessite de faire un prétraitement.
Apres l'étape de prétraitement des données, nous avons adopté deux méthodes qui se basent respectivement sur l'apprentissage automatique et l'apprentissage profond, et qui sont toutes les deux supervisées. Premièrement, nous avons utilisé la méthode Naïve bayes en proposant deux versions, une permettant une classification en 5 classe(NB5C) et l'autre permettant une classification en 2 classe(NB2C) .Deuxièmement, nous avons opté pour un apprentissage profond via MCLT, une méthode inspiré à partir des réseaux de neurones afin de définir un modèle de détection .En fin, nous avons fait une hybridation entre les deux méthodes NB2C et MCLT.
L'objectif est de déterminer si un paquet reçu est de type attaque ou normal, puis déterminer notamment la catégorie de l'attaque si elle existe et faire une comparaison entre ces approches afin de déterminer le meilleur algorithme. Pour cela, nous avons utilisé des estimateurs basés sur la matrice de confusion.
Les étapes d'apprentissage ainsi que la validation de notre SDI sont réalisées via la base de données NSL-KDD [NSL-KDD, 2009].qui regroupe deux catégories, base d'apprentissage et base de test.
Apres la réalisation de notre SDI via ces trois approches, nous avons fait des simulations pour valider ses performances, ces simulations sont faites via une architecture client –serveur. L'architecture de notre SDIR est présentée par la figure III.1 suivante.

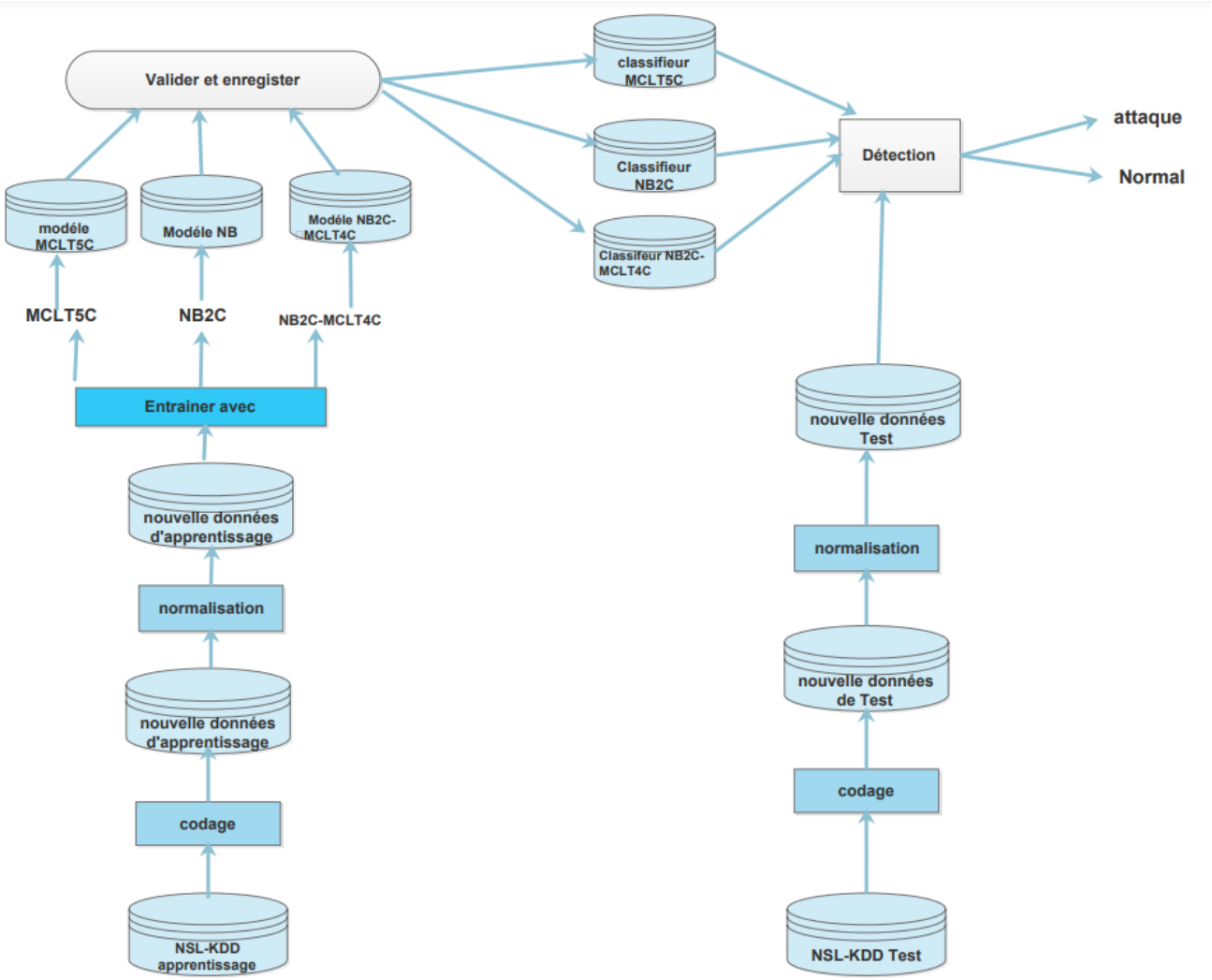


FIGURE III.1 – Architecture de nos SDIRs implémentés : SDIR-NB5C, SDIR-MLCT5C, et approche hybride proposée SDIR-NB2C –MLCT4C

#### III.2.1.1 Méthodes adoptées

A) **L'approche naïve bayésienne**

Ce modèle, il est facile à construire et particulièrement utile pour les ensembles de données assez grands. Outre sa simplicité, Un de ces points fort est le besoin d'une faible quantité d'informations pour la phase d'apprentissage. Il est alors réputé de surpasser même les méthodes de classification les plus sophistiquées [Douid, 2019].

Ce classificateur est basé sur le théorème de Bayes élémentaires. Il peut atteindre relativement une bonne performance sur les tâches de classification. Il Simplifie grandement l'apprentissage en supposant que les caractéristiques sont indépendantes. Il est notamment très rapide, en effet les calculs de probabilités ne sont pas très coûteux et la classification est possible avec un petit jeu de données.

Nous avons dans un premier temps appliqué la méthode NB pour classer les paquets en cinq classes : normal, Dos, R2L, U2r, Probe. Nous avons calculé la Probabilité de chaque classe (Pro (classe)) via la formule suivante [Faour et al., 2005] :

$$Pro(classe) = \frac{F}{T} \tag{III.1}$$

Où
F : Fréquence de la classe dans l'ensemble de données d'apprentissage.
T : taille totale de l'ensemble de données d'apprentissage.
Les probabilités Pro (classe) calculées pour les cinq classes normal, U2R, Dos, probe et R2l sont présentés dans la table III.1 suivante :

TABLE III.1 – Probabilité de chaque classe

| Classe | Normal | U2R | Dos | Probe | R2l |
|---|---|---|---|---|---|
| **Pro(classe)** | 67949/125973 =0.53 | 52/125973 =0.0004 | 45927/125973 =0.36 | 11656/125973 =0.09 | 995/125973 =0.007 |

Par la suite nous avons calculé la probabilité de valeur dans la classe (Pro (classe/ entité), sachant que l'entité dans notre cas est un paquet. Elle est calculée via la formule ivant [Faour et al, 2005] :

$$Pro(classe/entité) = \frac{FA}{FB} \tag{III.2}$$

Où
FA : Fréquence de la valeur de l'attribut qui apparaît dans la classe
FB : Fréquence de la classe dans l'ensemble de données d'apprentissage
Un exemple de calcul de la Pro(classe / entité) concernant un paquet pour les cinq classes normal, U2R , Dos, pro et r2l sont présentés dans la table III.2 suivante :

TABLE III.2 – Un exemple de calcul de la Pro (classe / entité) pour un paquet

| Classe | Normal | U2R | Dos | Probe | R2l |
|---|---|---|---|---|---|
| **Pro (classe/ entité exemple)** | 0.3 | 0.07 | 0.39 | 0.2 | 0.04 |

Exemple de calcule de probabilité d'un paquet pour la classe dos :

$$Pro(dos|0.0, 0.5, ..., 1.0) = \frac{Pro(0.0|dos) * Pro(0.5|dos) * ... * Pro(1.0|dos)}{Pro(0.0) * Pro(0.5) * .... * Pro(1.0)} \tag{III.3}$$

Afin de classer n'importe quel nouveau paquet caractérisé par ses valeurs d'attributs, sa classes choisies seront celles dont la probabilité est la plus grande, il attribue souvent une probabilité maximale à la bonne classe ce qui signifie la sortie est une catégorique Par la suite et dans un deuxième temps, afin de procéder à l'hybridation, nous avons classé les paquet deux classe : normal et intrusion, sans spécifier le type. Les résultats de la classification via NB 2 classes sont meilleurs que ceux de la méthode NB 5 classes.

B) **Notre Réseau MCLT**
MCLT est un réseau de neurone appartenant aux algorithmes avec apprentissage profond, il est capables d'apprendre des dépendances à long terme, ce réseau sont distingués par la possibilité de suppression ou de rajout d'informations à l'état de la cellule (voir chapitre 2 section 3) .Ce sont des réseaux auto-adaptatives, il utilise trois portes à savoir :

- **Porte d'oubli** (Forget Gate) : traduit la capacité d'oublier une information.
- **Porte d'entrée** (Imput Gate) : permet de rajouter des informations et mettre à jour l'état de la cellule

- **Porte de sortie** (Output Gate) : représente le résultat de la classification (réponse du modèle), il est ici question multiplier la sortie sigmoïde par la sortie tanh.

Quant à l'état de la nouvelle cellule, il est obtenu en ajoutant la sortie de la porte d'entrée

- **La couche d'entrée :** nous avons comme entré 5 vecteurs de taille 45 c'est-à-dire nous avons calculé la moyenne de chaque colonne pour chaque classe (40 pour les attributs et nous avons combiné un vecteur de taille 5).
- **Fonction d'activation :** nous avons utilisé deux types de fonctions d'activation, la première est la fonction Sigmoid avec un intervalle de sortie est [0,1], et qui est définit ainsi [Parizeau, 2004] :

$$Sigmoid(logistic) = f(x) = \frac{1}{1 + e^{-x}} \tag{III.4}$$

  La deuxième fonction est la fonction TanH, qui est utilisée par le MCLT pour des données en continue et dont l'intervalle de sortie est [-1,1], elle est définit comme suit [Parizeau, 2004] :

$$\tanh = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}} \tag{III.5}$$

- **La couche récurrente :** l'opération principale consiste en des cellules et des portes cette couche permet d'exécuter le principe de MCLT.
- **La couche de sorties :** contient les 5 vecteurs où chaque vecteur est dédié à une classe.

Notre MCLT prend 3 vecteurs en entré (xt, ht-1, ct-1) La figure III.2 représente l'architecture détaillée d'un réseau MCLT dont :

- xt : représente la donnée de l'apprentissage
- Ct-1 : mémoire de précédente MCLT ;
- ht-1 : la sortie de MCLT précédent
- S : la fonction d'activation Sigmoïde ;
- T : la fonction d'activation tangente hyperbolique ;
- X : Opérateur de multiplication ;
- + : Operateur d'addition ;
- b : valeur bayes ;
- Ct : la nouvelle mémoire à l'instant t ;
- ht : la sortie de MCLT.

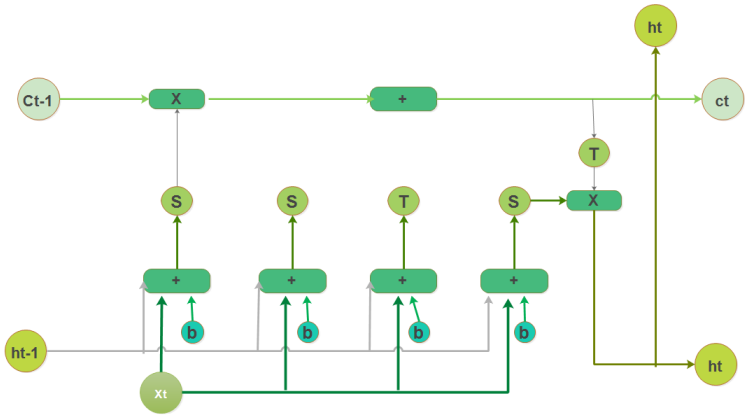


FIGURE III.2 – Architecture de MCLT

### III.2.1.2 L'approche Hybride proposée NB2C-MCLT4C

Dans cette approche, nous avons fait une hybridation entre les deux méthodes décrites précédemment à savoir la méthode naïve bayésienne et le réseau récurrent MCLT (voir la figure III.3). Nous avons introduit les résultats issus l'algorithme NB comme entrée au réseau MCLT. Nous avons appliqué la méthode NB2C afin filtrer les paquets en paquets normaux et paquets avec intrusion. Le MCLT recevra seulement les paquets intrusifs afin de déterminer la catégorie de l'intrusion.

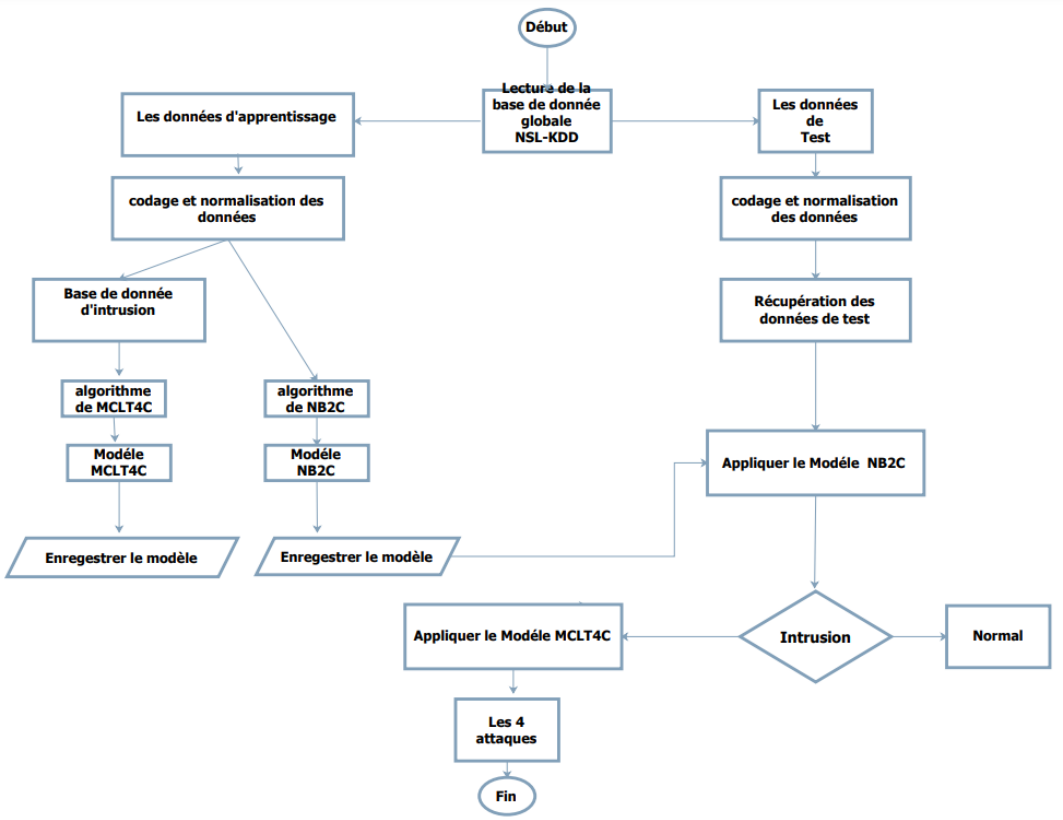


FIGURE III.3 – Algorigramme de l'approche proposée SDIR-NB2C-MCLT

## III.2.2 La base de données utilisée : NSl-KDD

### III.2.2.1 Définitions

- **KDD (knowledge Discovery in Database) :** c'est un processus d'extraction des connaissances à partir des données, il permet le stockage, la préparation l'analyse des données en utilisant de nombreuses techniques afin d'extraire les connaissances et les évaluer [Fayyad et al., 1996] .

- **NSL-KDD :** nous utilisons dans ce travail la base NSL-KDD qui repose sur un autre BDD populaire, il s'agit du KDD Cup 99 [NSL-KDD, 2009]. Cette dernière créée en 1999 pour une compétition d'apprentissage automatique. Le but de cette compétition était de classifier correctement des connexions réseau en 5 catégories : normal, déni de service (DoS), sonde réseau (probe), distant à local (R2L – Remote to Local), utilisateur à root (U2R – User to Root). Chaque connexion possède 41 caractéristiques(attributs) qui permettent au classificateur de prédire correctement sa classe. Ces caractéristiques sont des informations ou des statistiques calculées à partir de l'écoute d'un réseau local : durées de la connexion, type du protocole, pourcentage de connexions au même service, etc [Mukkamala et al., 2005].

### III.2.2.2 Description de la base NSL-KDD

NSL-KDD a été créé en 2009 pour résoudre certains problèmes inhérents à KDD Cup 99. Elle reprend ainsi les mêmes données que cette dernière, mais apportant des corrections importantes. Ainsi, les connexions redondantes ou dupliquées, qui composaient de 75% à 78% de la BDD, ont été supprimées.Le nombre total de données a été largement réduit, en passant de 805050 connexions(paquet) pour KDD Cup [Tavallaee et al., 2009].

- **Contenu :** la base de données NSL-KDD contient une collection de fichiers téléchargeables à savoir [Dhanabal et Shantharajah, 2015] :

  - KDDTrain + .ARFF ;
  - KDDTrain + 20Percent.ARFF ;
  - KDDTrain + .arffKDDTrain +20Percent.TXT ;
  - KDDTrain + .txtKDDTest + .ARFF : KDDTest + .TXT ;
  - CSV KDDTest-21.ARFF
  - KDDTest-21.TXT.

  Nous nous somme intéressés que par les deux fichiers KDDTrain et KDDT et qui sont présentés dans la table III.3 illustrée ci-dessous.

TABLE III.3 – Description des fichiers Train et Test [Dhanabal et Shantharajah, 2015].

| Nom du fichier | Description | Normal | Dos | Probe | R2l | U2R |
|---|---|---|---|---|---|---|
| KDDTrain+ | Ensemble complet de trains NSL-KDD incluant les étiquettes d'attaque et le niveau de difficulté au format csv | 67343 (53.46%) | 45927 (36.46%) | 11656 (9.25%) | 955 (0.79%) | 52 (0.04%) |
| KDDTest+ | Ensemble de test complet NSL-KDD incluant les étiquette d'attaque et le niveau de difficulté au format csv | 9711 (43.08%) | 7460 (33.09%) | 2421 (10.74%) | 2885 (12.80%) | 67 (0.30%) |

- **Les types d'attaques de NSL-KDD**
  La base donnée NSL-KDD recense 23 types d'attaques possibles [Berlin et al, 2016] (listées dans le tableau III.4 suivant) qui peuvent être regroupées en quatre catégories :

TABLE III.4 – Les types d'attaques [Berlin et al., 2016].

| Attaque | Échantillons | Catégorie |
|---|---|---|
| smurf | 280790 | Dos |
| neptune | 107201 | Dos |
| back | 2203 | Dos |
| teardrop | 979 | Dos |
| pod | 264 | Dos |
| land | 21 | Dos |
| normal | 97277 | Normal |
| satan | 1589 | Probe |
| ipsweep | 1247 | Probe |
| portsweep | 1040 | Probe |
| nmap | 231 | Probe |
| warezclient | 1020 | r2l |
| guess_passwd | 53 | r2l |
| warezmaster | 20 | r2l |
| imap | 12 | r2l |
| ftp_write | 8 | r2l |
| multihop | 7 | r2l |
| phf | 4 | r2l |
| spy | 2 | r2l |
| buffer_overflow | 30 | u2r |
| rootkit | 10 | u2r |
| loadmodule | 9 | u2r |
| perl | 3 | u2r |

- **Les catégories des attaques**
  Les attaques les plus courants dans les systèmes de detection d'intrusion sont les suivants [Sharma et Yadev, 2021] :

  1. **Déni de service (DOS) :**
     attaque permet de rend le système réseau refuser toute sorte de service à l'utilisateur. Il bloque tous les services disponibles dans le réseau.
  2. **Sonder(Probe) :**
     L'attaquant identifie les vulnérabilités d'un réseau qui peut être mal utilisé pour une attaque.
  3. **U2R (utilisateur à Root) :**
     Chaque fois qu'un attaquant a accès aux services utilisateurs mais tente ensuite d'accéder aux fichiers d'administration ou de racines, une telle attaque est appelé U2R.
  4. **R2L (Distant à Local) :** Chaque fois qu'un attaquant gagne non autorisé l'accès au service de l'utilisateur et l'attaque R2L a lieu.

#### III.2.2.3 Prétraitement de la base NSL-KDD

La phase de prétraitement des données constitue une phase importante dans le processus global de data mining. Elle vise à rechercher une représentation des données afin d'améliorer la qualité des résultats de la fouille. Typiquement cette phase comporte des opérations de nettoyage (suppression de bruit, de valeurs aberrantes, etc.) [Lahbib, 2012].Dans notre travail, nous avons besoin de faire

des calculs sur des valeurs numériques, or nous avons procédés dans un premier prétraitement à un codage des valeur alphanumérique, et suite aux variation des intervalles de valeurs entre les s différentes caractéristiques de chaque paquet (champs) nous avons effectué une normalisation .

a) **Numérisation ou codage (conversion des données symboliques vers numériques)** Cette étapes consiste à convertir les attributs symboliques en numériques, dans NSL-KDD il y a que trois attributs qui ne sont pas numériques : protocole-type (3 symboles différents), service (70 symboles différents) et flag (11 symboles différents) , nous avons alors procédé à une transformation en des valeurs numériques comme illustré dans les tables suivantes :

- **L'attribut Protocol_type**

TABLE III.5 – Numérisation de l'attribut Protocol_type

| Etat normal | codification |
|---|---|
| ICMP | 0 |
| TCP | 1 |
| UDP | 2 |

- **L'attribut flag**

TABLE III.6 – Numérisation de l'attribut flag

| Etat normal | codification | Etat normal | codification |
|---|---|---|---|
| OTH | 0 | S0 | 5 |
| REJ | 1 | S1 | 6 |
| RSTO | 2 | S2 | 7 |
| RSTOS0 | 3 | S3 | 8 |
| RSTR | 4 | SF | 9 |
| SH | 10 | | |

- **L'attribut Service**

TABLE III.7 – Numérisation de l'attribut Service

| Etat normal | codification | Etat normal | codification | Etat normal | codification |
|---|---|---|---|---|---|
| aol | 0 | http_443 | 23 | printer | 46 |
| auth | 1 | http_8001 | 24 | private | 47 |
| bgp | 2 | imap4 | 25 | red_i | 48 |
| courier | 3 | IRC | 26 | remote_job | 49 |
| csnet_ns | 4 | iso_tsap | 27 | rje | 50 |
| ctf | 5 | klogin | 28 | shell | 51 |
| daytime | 6 | kshell | 29 | smtp | 52 |
| discard | 7 | ldap | 30 | sql_net | 53 |
| domain | 8 | link | 31 | ssh | 54 |
| domain_u | 9 | login | 32 | sunrpc | 55 |
| echo | 10 | mtp | 33 | supdup | 56 |
| eco_i | 11 | name | 34 | Systat | 57 |
| ecr_i | 12 | netbios_dgm | 35 | Telnet | 58 |
| efs | 13 | netbios_ns | 36 | Tftp_u | 59 |
| exec | 14 | netbios_ssn | 37 | Tim_i | 60 |
| finger | 15 | netstat | 38 | Time | 61 |
| ftp | 16 | nnsp | 39 | urh_i | 62 |
| ftp_data | 17 | nntp | 40 | urp_i | 63 |
| gopher | 18 | ntp_u | 41 | uucp | 64 |
| harvest | 19 | other | 42 | uucp_path | 65 |
| hostnames | 20 | pm_dump | 43 | vmnet | 66 |
| http | 21 | pop_2 | 44 | whois | 67 |
| http_2784 | 22 | pop_3 | 45 | X11 | 68 |
| Z39_50 | 69 | | | | |

b) **Normalisation**
Les valeurs obtenues après l'opération de la numérisation sont très variées, certains attributs prennent de grandes valeurs (src_bytes, dst_bytes, etc.), alors que d'autres ne prennent que des petites valeurs (serror_rate, same_srvrate, etc.), et cela peut nuire à la rentabilité du modèle de détection d'intrusions. Afin d'éviter ce problème et garantir l'efficacité du modèle généré, les valeurs de la base données doivent être ajustées ou normalisées. Dans notre cas les données de la base NSL-KDD sont normalisés dans l'intervalle [0, 1]. Nous avons utilisé la fonction décrite par la formule suivante [Mokhtari et Moulkhaloua, 2018]

$$valnouv = \frac{valanc - Minanc}{Maxanc - Minanc} * (Maxnouv - Minnouv) + Minnouv \quad \text{(III.6)}$$

sachant que :

- valanc : est la valeur à normaliser.
- valnouv : est la valeur après la normalisation.
- Minanc : est la limite inférieure de l'intervalle à que valanc appartient
- Maxanc : est la limite supérieure de l'intervalle à que valanc appartient.
- Minnouv : est la limite inférieure de l'intervalle à que valnouv va appartenir.
- Maxnouv : est la limite supérieure de l'intervalle à que valnouv va appartenir.

Après le prétraitement de la base de données, nous avons appliqué notre phase d'apprentissage avec les trois SDIRs que nous avons réalisés.

## III.2.3 Résultats et discussions

Un fois la phase d'apprentissage fini, les modèles de chaque SDIR que nous avons implémenté sont enregistrés. Nous avons alors effectué une phase de test, il est question ici de réaliser une prédiction pour voir l'effet de chaque modèle, on parle de performance de détection ou de classification. Pour cela nous avons utilisé quelques estimateurs basés sur la matrice de confusion. Dans cette phase de teste, nous utilisons la base de test d'NSL-KDD étiqueté en paquets normal et paquet intrusif avec comme spécification la catégorie de l'attaque. Chaque paquet est constitué de 41 attributs ainsi qu'un attribut étiquète.

### III.2.3.1 Les estimateurs de performances

- **La matrice de confusion** La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle. La matrice de confusion est une matrice 2X2 qui rapporte le nombre de Vrais Positifs (VP), de vrais Négatifs (VN), de Faux Positifs (FP) et de Faux Négatifs (FN) comme le montre la table [Caelen, 2017].

TABLE III.8 – La matrice de confusion

| | | **Classe détectée** | |
|---|---|---|---|
| | | **Normal** | **Attaque** |
| **Classe réelle** | **Normal** | Vrai négatif VN | Faux positif FP |
| | **Attaque** | Faux négatif FN | Vrai positif VP |

Sachant que :

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

- **Les mesures d'évaluation :**

  - **La précision(taux de réussite) :** Appelé aussi justesse, cette mesure permet de calculer le taux de précision globale de la classification de l'ensemble de données de tests. Elle traduit le rapport entre les détections correctes et les détections totales obtenues. [Bourouh et Kanoun, 2018]

$$Précision = (\frac{VP + VN}{VP + VN + FP + FN}) * 100\% \tag{III.7}$$

  - **Taux du vrai positif (TVP) :** La probabilité qu'une prédiction d'une catégorie donnée soit correcte. Elle est définie comme suit [Pietraszek, 2004] :

$$TVP = (\frac{VP}{VP + FP}) * 100\% \tag{III.8}$$

  - **Le taux de détection (Rappel) :** C'est le rapport entre le nombre d'intrusions correctement détectées et le nombre total d'intrusions. Elle est décrit par la formule [Mokhtari et Moulkhaloua, 2018] :

$$Rappel = (\frac{VP}{FN + VP}) * 100\% \tag{III.9}$$

  - **Mesure-F (Moyenne harmonique) :** la moyenne harmonique F combine le rappel et la précision en un nombre compris entre 0 et 1, elle est calculé via la formule suivante [Mokhtari et Moulkhaloua, 2018] :

$$mesure - f = (\frac{2}{\frac{1}{Rappel} + \frac{1}{Precision}}) * 100\% \tag{III.10}$$

#### III.2.3.2 Comparaison des résultats

Dans ce qui suit nous allons présenter l'évaluation des méthodes que nous avons implémentés ainsi que l'approche que nous avons proposée en utilisant les estimateurs décrits dans la section 2.4.1 suite a des tests en utilisant l'ensemble complet des données de test.

- **Comparaison entre les algorithmes NB2C et NB5C**

TABLE III.9 – Comparaison entre NB2C et NB5C

| Les algorithmes | Précision | Rappel | TVP | Mesure-F |
|---|---|---|---|---|
| **NB2C** | **83,11%** | **73,17%** | **97,1%** | **83,45%** |
| **NB5C** | **2 %** | **3,22** | **4.08%** | **2,33** |

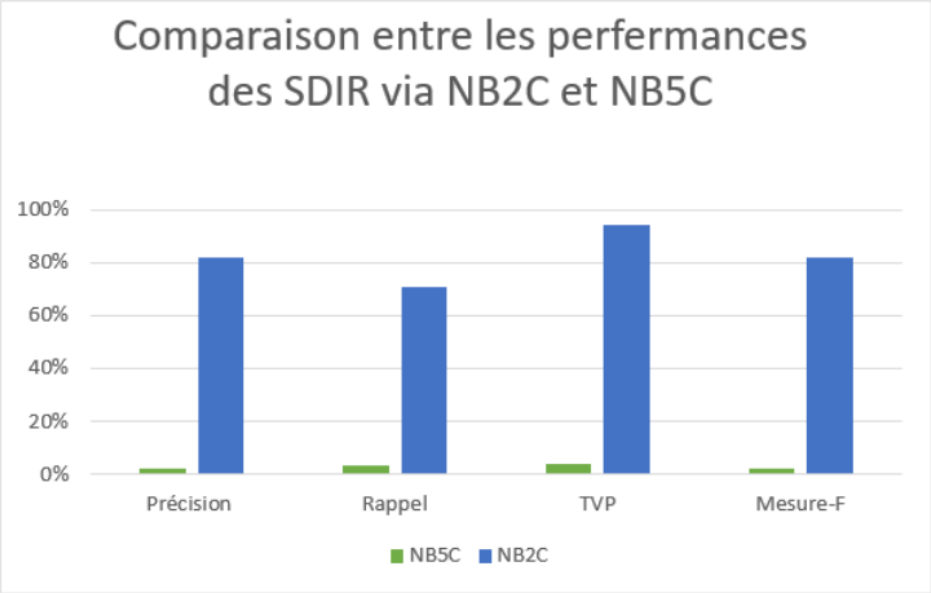


FIGURE III.4 – Comparaison des résultats obtenus via SDIR-NB2C et SDIR-NB5C en matière de rappel, précision et F-mesure

Les résultats présentés dans la table III.9 montrent que les performances de notre modèle de détection d'intrusion avec NB2C sont meilleures que ceux de NB5C avec un taux de précision de 83.11% pour le NB2C contre une valeur de 2% pour le NB5C. Pour le rappel et la mesure-F ont atteint respectivement les valeurs de 73.17% et 83.45% pour le NB2C contre des valeurs de 3.22% et 2.33% pour le NB5C.Le taux des vrais positive de NB2C est élevé jusqu'à 97,1% contre une valeur de pour NB5C 4.08%.
Suite à ces résultats, nous avons réalisé l'hybridation entre les deux approches NB2C et MCLT.

- **MCLT en variant du nombre de couches**
  Nous avons effectué plusieurs tests, en variant le nombre de couches et le nombre d'itérations plusieurs fois, les résultats obtenus sont exprimés dans la table III.10 présentée ci-dessous. À travers la table III.10, nous remarquant les taux de détection en matière de réussite, de

TABLE III.10 – Résultats de détection via le MCLT en variant différentes couches

| **Algorithme MCLT5C** | **Précision** | **Rappel** | **Mesure-F** |
|---|---|---|---|
| **2 couches** | 63,5 | 96,7 | 59,4 |
| **5 couches** | 70% | 98% | 74,5 |
| **10 couches** | 86.58% | 90.5% | 88.6% |

F-Mesure et de rappel sont obtenu avec le MCLT5C.

- **Comparaison des résultats obtenus via SDIR-NB2C, SDIR-MCLT et SDIR-NB2C-MCLT**
  Nous avons effectué une deuxième comparaison entre les algorithmes SDIR-NBC, SDIR-MCLT et SDIR-NB2C-MCLT, qui est illustrée dans la table III.11 présentée ci-dessous.

TABLE III.11 – Comparaison des résultats obtenus via SDIR-NBC, SDIR-MCLT et SDIR-NB2C-MCLT

| Les algorithmes | Précision | Rappel | Mesure-F | TVP |
|---|---|---|---|---|
| **NB2C** | **83.11%** | **73.17%** | **83.45%** | **97.1%** |
| **MCLT5C** | **86.58%** | **90.5%** | **88.6%** | **86.6%** |
| **NB2C-MCLT4C** | **68.29%** | **62.2%** | **71.1%** | **83.1%** |

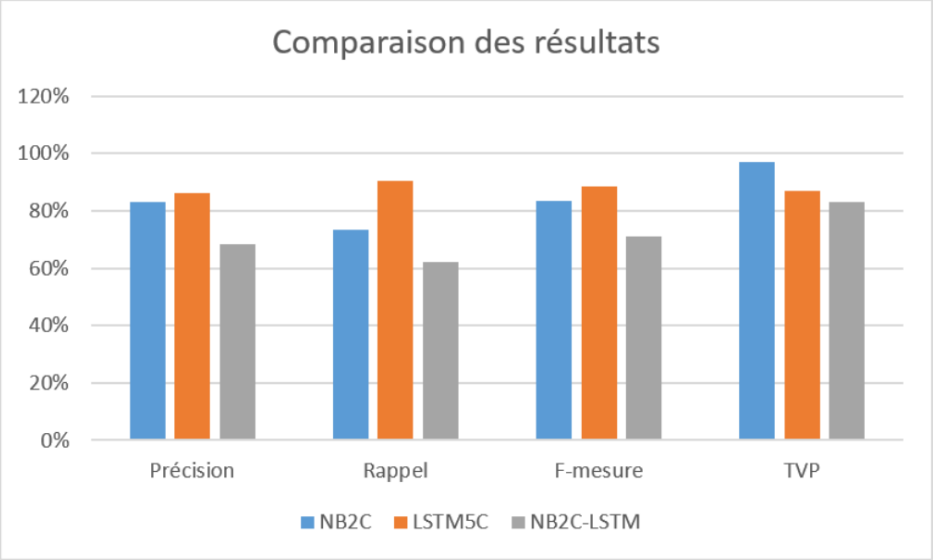


FIGURE III.5 – comparaison des résultats obtenus via MCLT SDIR-5C , SDIR-MCLT et SDIR-NB2C-MCLT en matière de rappel, précision et F-mesure

Afin de réaliser notre approche proposée, nous avons implémenté un modèle hybride entre les approches NB2C et MCLT4C. Dans un premier temps nous avons implémenté l'algorithme de naïve bayes effectuant une classification en deux classes normale et attaque, ensuite nous avons pris le résultat de détection et plus précisément les paquets intrusifs afin de les introduire au MCLT qui à son tour les classifie en 4 classes.
Les résultats illustrés dans la table III.11 montrent une hausse en matière de réussite, de F-Mesure et de rappel sont obtenu avec le NB2C, MCLT5C et l'hybridation entre ces deux dernières.
Après toutes ces études,nos résultats montrent que MCLT est adaptative à notre système de detection d'intrusion.

## III.3 Partie 2 : implémentation et présentation de l'application

### III.3.1 Caractéristiques techniques du matériel utilisé

Pour réaliser notre application, nous avons utilisé les machines avec leurs caractéristiques illustrées dans la table III.12 présentée ci-dessous

### III.3.2 Outils logiciels utilisés et langage

- **Environnement De Développement**
  Le langage de programmation utilisé pour l'implémentation est le langage Java, l'environnement utilisé est Netbeans version 12.6, ce dernier est populaire parmi les

TABLE III.12 – Caractéristiques techniques de machines utilisées

| | Matériels | Caractéristique technique |
|---|---|---|
| Machine1 | PC | Asus. |
| | Processeur | intel CORE i5-8250U |
| | Ram | 8G |
| | Vitesse de processeur | 1.60GHz 1.80 GHz |
| | Système d'exploitation | Windows 11 64bits |
| Machine2 | PC | HP. |
| | Processeur | Intel(R) Celeron(R) CPU N3060 |
| | Ram | 4Go |
| | Vitesse de processeur | 1.60GHz 1.60 GHz |
| | Système d'exploitation | Windows 10 64bits |

programmeurs en java. Cet outil comporte des bibliothèques de classes très riches comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphismes), la gestion des exceptions, la variété des types, la gestion des fichiers, l'ajout automatique des déclarations d'importation, etc. Tout cela nous a été utile dans la conception de l'ensemble de nos SDIRs.
Puisque que nous nous situons dans une approche d'SDI réseau, l'apprentissage s'effectue au niveau du serveur, et c'est lui qui détecte les anomalies qui viennent de l'extérieur qui sont expédié par des attaquants (client). Par conséquent, nous avons simulé l'ensemble via une architecture client serveur où la communication entre les machines à été faite via les sockets, raison pour la quel nous avons choisi Java comme langage, en plus du faite qu'il assure une compatibilité avec tous les appareilles.

- **PostgreSQL**
  PostgreSQL est un système de gestion de bases de données relationnelles objet, il supporte une grande partie de données. Vu le nombre de vecteurs présents dans la base de données NSL-KDD, l'insertion coutera du temps en utilisant les requêtes classiques (INSERT INTO table Values (vecteur-1), Values (vecteur-2) . . . Values (vecteur-n)). Donc pour cela nous avons utilisé la« copy » de PostgreSQL qui permet de copier un fichier.csv contenant l'ensemble des vecteurs en utilisant une seule requête.

- **L'architecture Client-Serveur**
  Dans l'architecture client /serveur, une machine parmi l'ensemble est considérée comme un serveur, elle est généralement très puissante et c'est elle qui analyse les paquets qui viens de l'extérieurs. Nous avons utilisé les sockets pour assurer la communication.

- **Les sockets** Un socket est un point de communication par lequel un processus peut émettre ou recevoir des informations à partir d'un autre processus se trouvant sur la même machine ou sur une machine distante. Pour obtenir une communication entre deux machines, il faut un numéro de port et l'IP(voir la figure III.6).


FIGURE III.6 – Architecture Client-Serveur de nos SDIRs

### III.3.3 Présentation de notre application

#### III.3.3.1 Machine Serveur

- **La fenêtre d'accueil**
  Cette fenêtre s'affiche au début de l'exécution de l'interface avant le lancement du serveur. Elle est composée de deux boutons, un pour entrer à l'interface principale et une pour quitter.


FIGURE III.7 – La fenêtre d'accueil de notre application

- **La fenêtre principale de notre application se situant au niveau du serveur**
  Cette interface contient tous les principaux paramètres de la simulation, les algorithmes d'apprentissage, plus un menu de test. Elle contient aussi un menu Fichier pour lancer le chargement des fichiers de la base de données avant et après le prétraitement (dataset, dataset_codé, dataset_codé_normalisé). Ces fichiers sont utilisés lors de l'apprentissage et le test, en spécifiant le pourcentage choisi et en affichant le nombre de chaque catégorie.

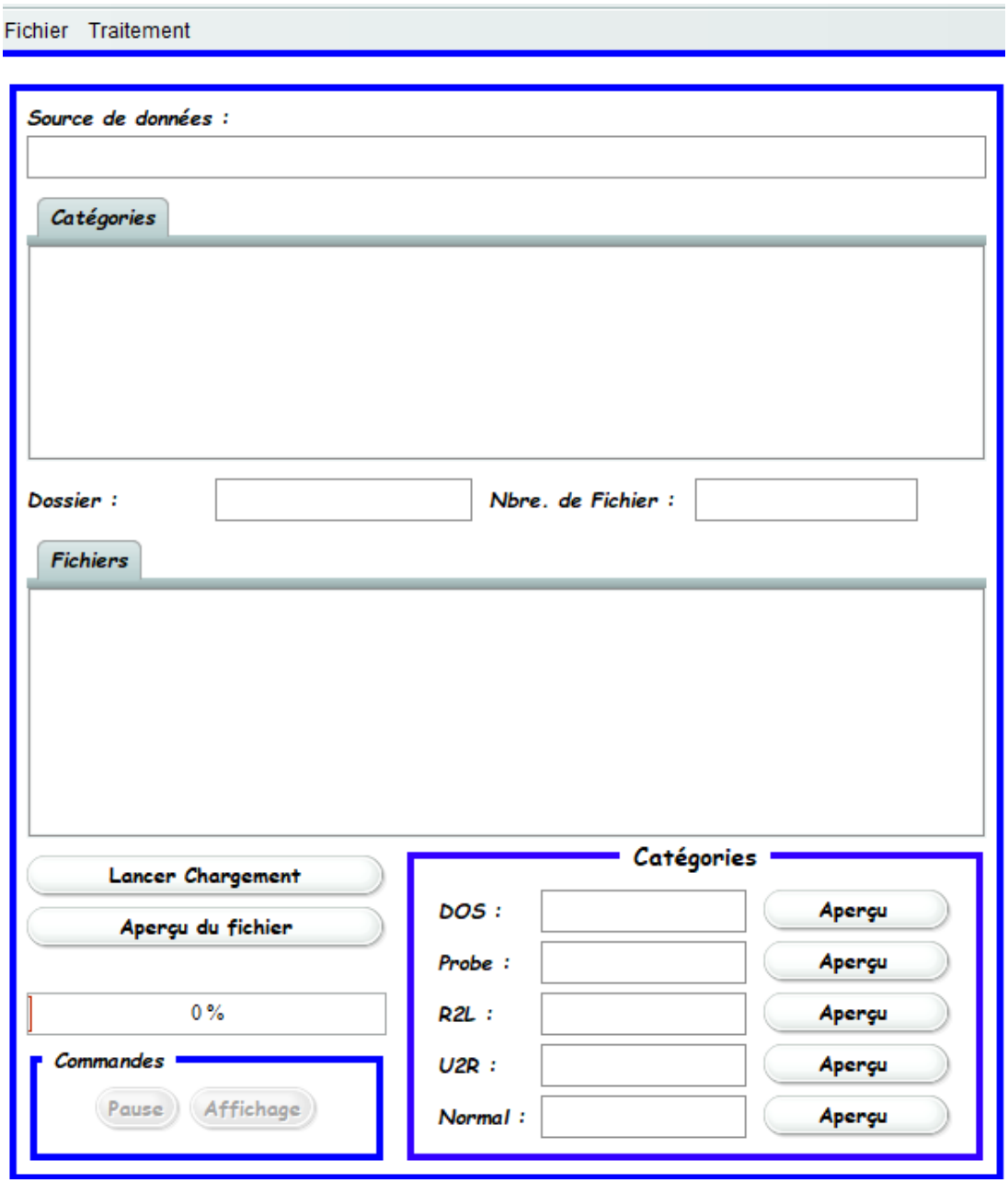


FIGURE III.8 – Fenêtre principale de l'application sur le serveur

- **Fenêtre de prétraitement**
  Cette fenêtre consiste à traiter les données brutes de la base de données NSL-KDD en passent par l'étape de codage et de normalisation afin qu'elle soit exploitable par suite.

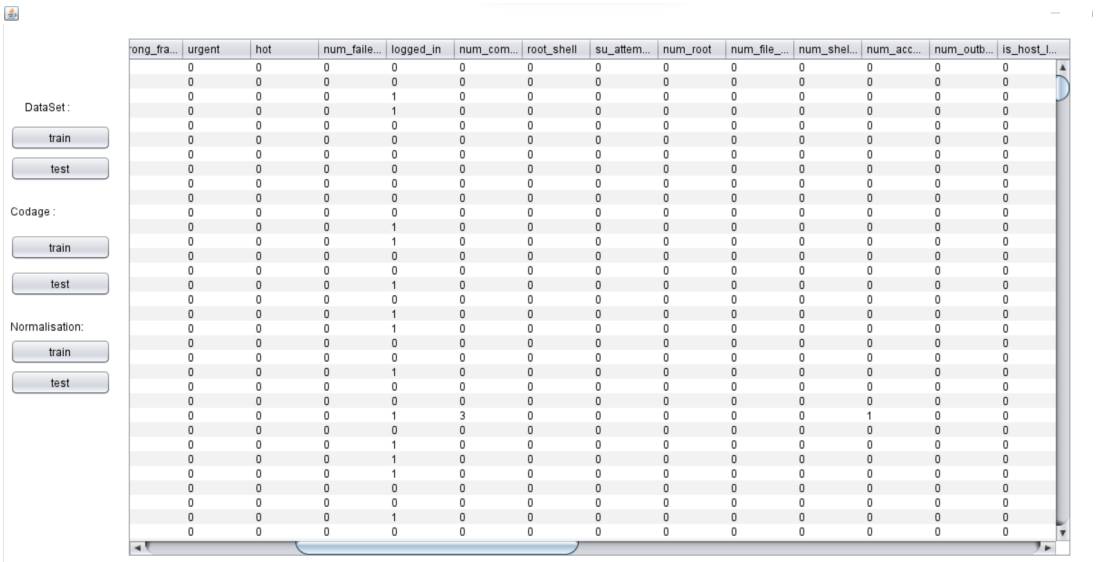


FIGURE III.9 – Fenêtre de Prétraitement

- **Fenêtre de chargement**
  une tâche consiste à charger les fichiers par catégorie en utilisant des threads.


FIGURE III.10 – Fenêtre de chargement

- **La fenêtre d'option détection**

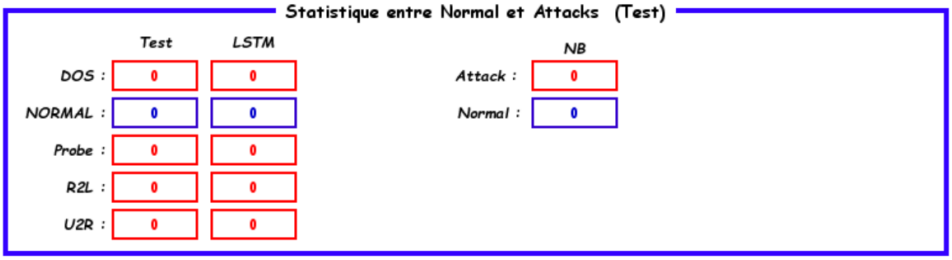


FIGURE III.11 – La fenêtre d'option détection

- **La fenêtre de l'apprentissage par MCLT**
  cette interface permet de lancer l'algorithme de MCLT (LSTM).

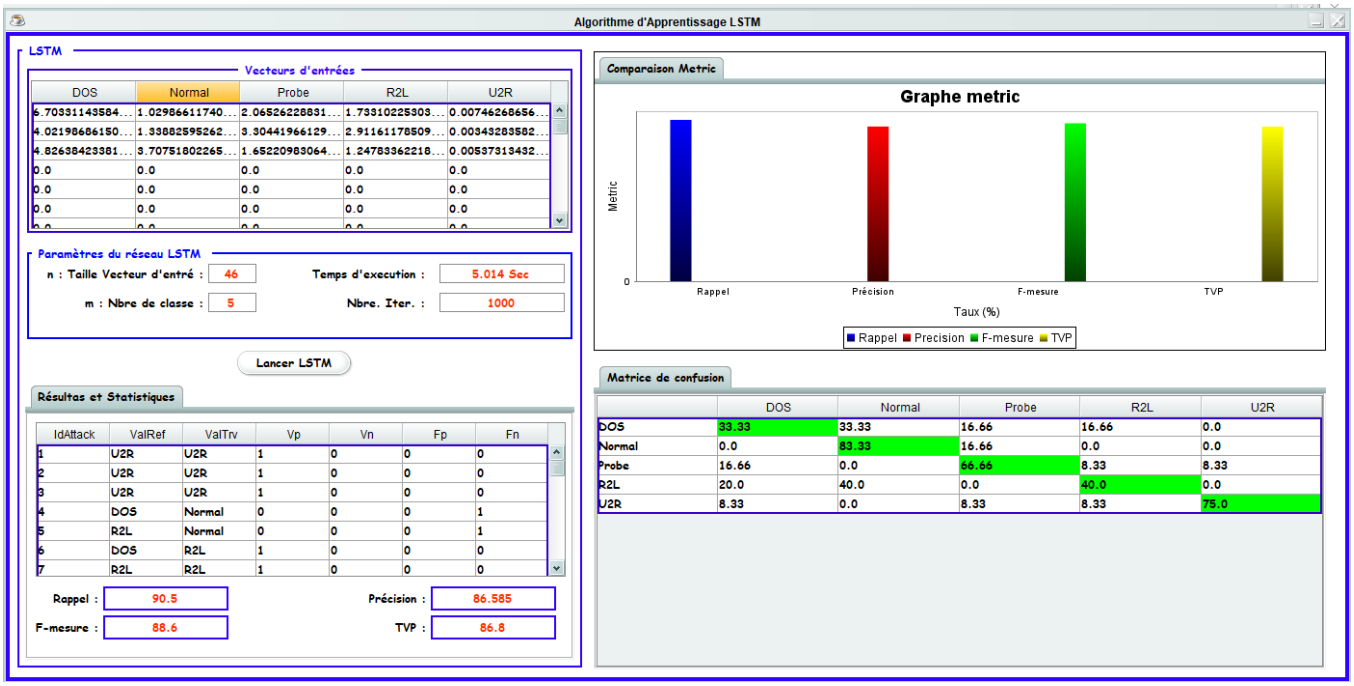


FIGURE III.12 – La fenêtre de l'apprentissage par MCLT

- **La fenêtre de NB**
  cette interface permet de lancer l'algorithme de NB.

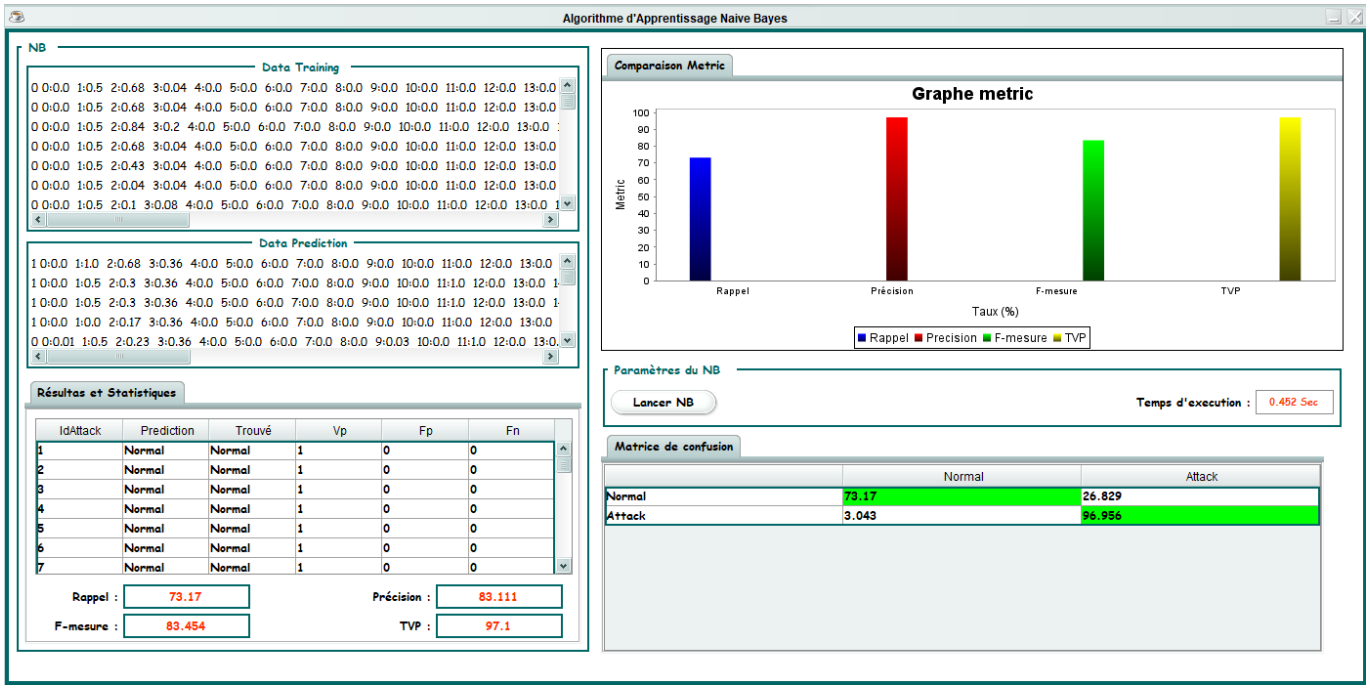


FIGURE III.13 – La fenêtre de NB

- **La fenêtre d'hybridation**
  cette interface permet de lancer l'hybridation NB2C-MCLT4C.

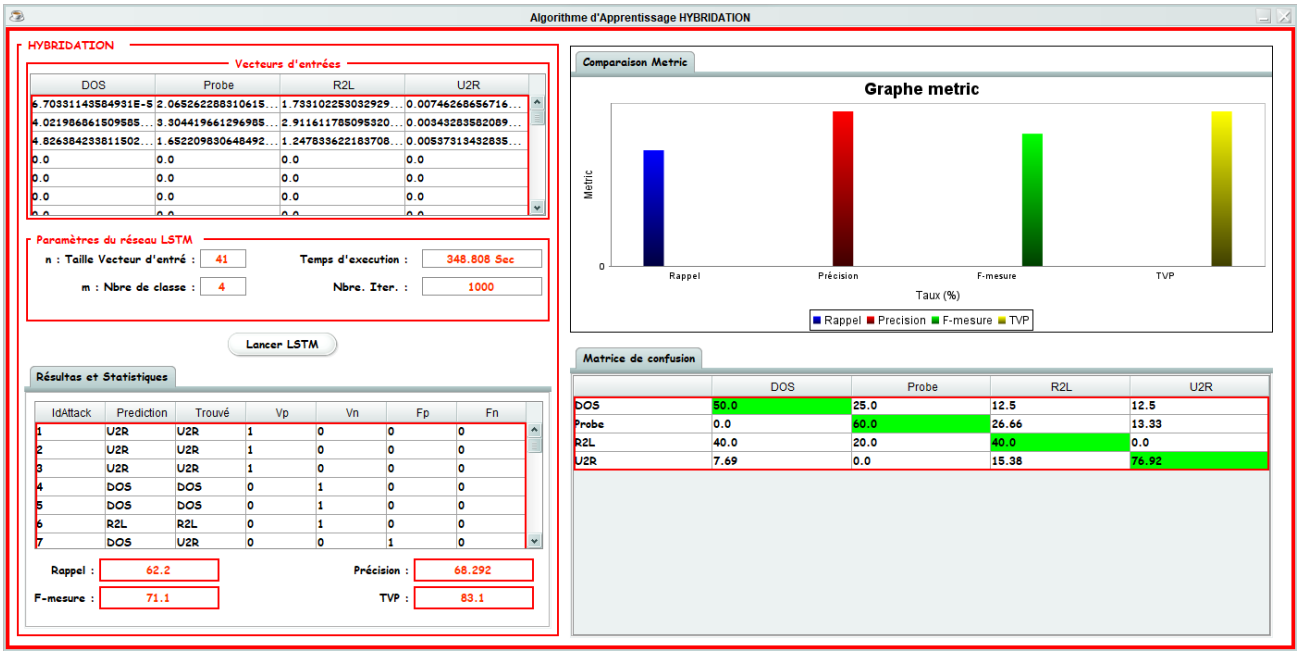


FIGURE III.14 – La fenêtre d'hybridation

### III.3.3.2 L'interface de l'attaquant

La fenêtre de l'attaquant permet d'envoyer les paquets de la base de test vers le serveur qui se trouve sur le même réseau local, afin de simuler des attaques.

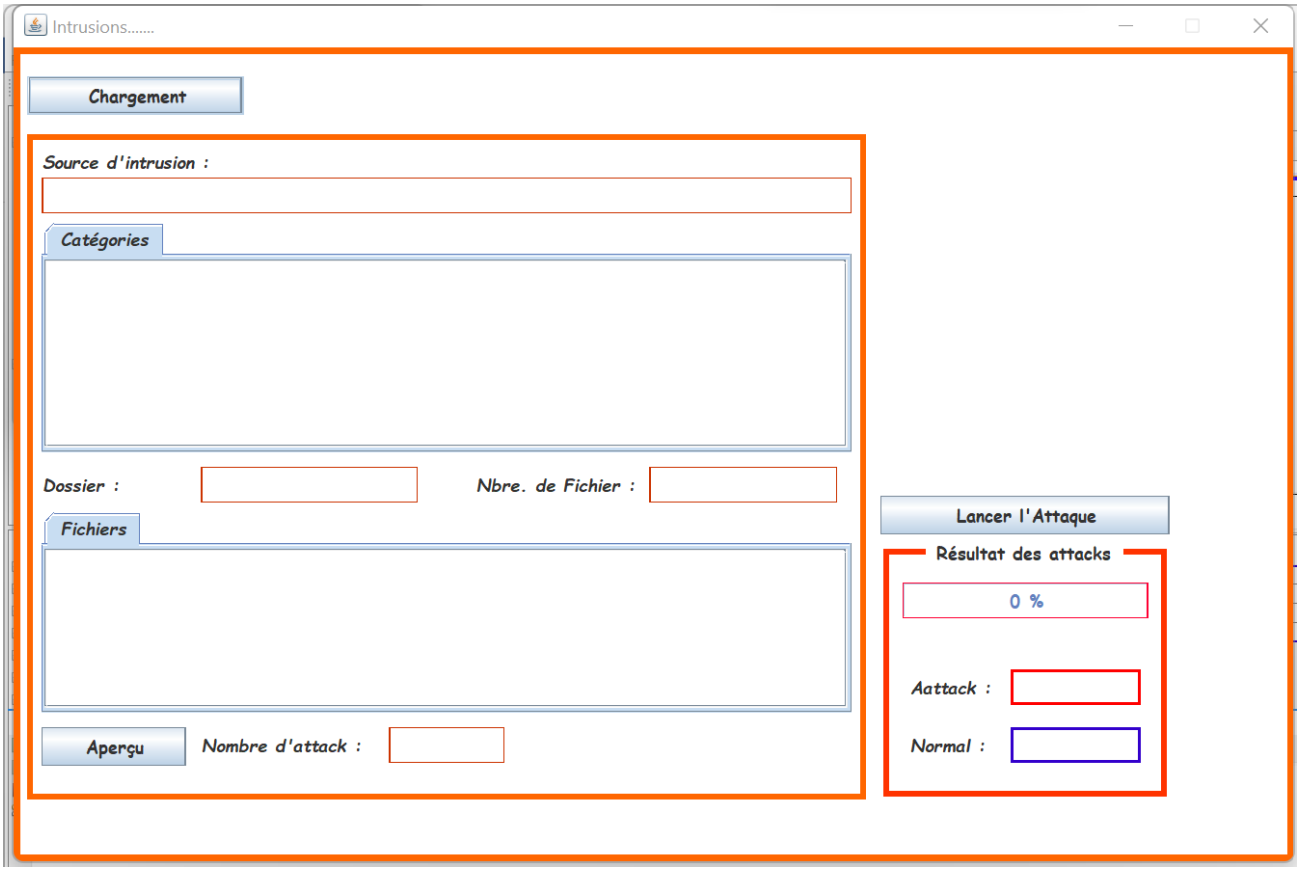


FIGURE III.15 – L'interface de l'attaquant

## III.4 Conclusion

Dans ce chapitre nous avons présenté les outils matériels et logiciels que nous avons utilisés ainsi que les méthodes que nous avons adoptées, à savoir la méthode Naïve Bayésienne issue de l'apprentissage automatique et les réseaux récurrents MLCT appliquant un apprentissage profond. Vu qu'il s'agit d'approches supervisées, nous avons entraîné et valider nos SDIRs implémentés en utilisant la base de données NSL-KDD que nous avons d'abord prétraitée (codage et normalisation).Les performances de classification de nos SDIR sont estimé via la précision, le rappel, et le F-Mesure.
Finalement nous avons illustré l'application que nous avons réalisée en présentant les différentes fenêtres de l'interface ainsi que leurs principales fonctionnalités Nous avons réalisé 3 SDIR, le premier appliquant l'algorithme NB, le deuxième appliquant l'algorithme MCLT, et le troisième est une hybridation entre la NB et le MCLT. Les résultats obtenus avec l'algorithme de MCTl sont les meilleurs.
Nous estimons avoir réalisé un système qui répond à l'objectif que nous sommes fixés à savoir la détection des attaques, et la spécification de leur catégorie, mais on espère toujours au future une amélioration en matière de résultats de classification.

# Conclusion générale

L'usage général du web et des services en ligne ont contraint l'organisation à ouvrir leurs systèmes d'information à Internet, en partant de ce moment, le nombre d'attaques sur ces systèmes à connu une hausse considérable. On peut bien les éviter en adaptant des bonnes configurations des matériels et des systèmes. Pour cela la sécurité informatique est clairement une assurance pour que les données ne soient pas altérées, et pour éviter tout d'accès non autorisé.
Nous avons présenté dans notre mémoire, dans le premier chapitre les déférents types et catégories d'attaques informatiques ainsi que les diverses méthodes de protection existantes. Parmi, nous avons les systèmes de détection d'intrusion Informatiques, qui représentent notre domaine d'étude nous nous sommes focalisés sur les SDIs, qui tendent à devenir un élément inévitable d'une architecture de sécurité informatique. Par conséquent, dans le second chapitre, Nous avons alors illustré leur taxonomie, les différentes approches existantes et finalement l'état de l'art présentant quelques travaux de la littérature faisant recours à l'AP et l'AA.
Dans le dernier chapitre, nous avons proposé un modèle de système de détection d'intrusion dans un réseau par signature qui est basé sur les techniques de l'apprentissage et la classification. Notre SDIR a pour l'objectif de pouvoir détecter les intrusions circulant sur le réseau en temps réel. Nous avons dans un premier temps, testé la classification Naïve bayésienne en deux Classes (NB2C), normale et anormale et en cinq Classes (NB5C), normale et 4 catégories d'attaques. Les résultats du premier algorithme était meilleurs, mais pas assez suffisants. Or, nous avons appliqué le réseau de neurones récurrent à Mémoire à Court Long Terme (MCLT) seul, et en l'hybridant avec le NB2C, ou il était question d'introduire les résultats obtenus de ce denier au réseau MCLT.
Afin d'évaluer ces différents algorithmes, nous avons utilisé plusieurs métriques, à savoir la précision, le rappel, la mesure-F. Pour aboutir à nos fins, il était nécessaire de passer par deux phases l'apprentissage et le teste, nous avons cependant utilisé la base NSL-KDD pour tester et évaluer notre SDIR.
Les résultats de la détection d'intrusion ainsi obtenus sur les quartes SDIRs que nous avons implémentés, montrent que l'algorithme MCLT seul donne les meilleures performances, qui restent acceptables par rapport aux autres algorithmes qui sont NB2C, NB5C, NB2C-MCLT.
Finalement, nous avons illustré l'interface de notre application, avec toutes ces fenêtres, sur les deux machines de l'attaquant (client mal intentionné) , et du serveur (qui fait l'apprentissage).
Le travail présenté nous a permis d'approfondir nos connaissances théoriques et pratiques en rapport avec l'apprentissage automatique et profond, les bases de données, les simulations réseaux via l'architecture client-serveur, plus particulièrement en phase d'implémentation.
Cependant plusieurs perspectives et améliorations peuvent être apportées, en matière des résultats de performance de détection, via d'autres hybridations possibles. Cependant plusieurs perspectives en vue d'amélioration des performances peuvent être réalisées au futur telles que : le choix des champs les plus représentatifs dans les paquets (réduction de dimensionnalité), Réalisation d'autres hybridations, Utilisation d'autres algorithmes issus de l'apprentissage profond.

# Bibliographie

# Bibliographie


[Arkoub et Amghar, 2015] Arkoub et Amghar , "conception et la réalisation d'un système de détection intrusion",thèse de doctorat, université Mouloud Mammeri,2015.

[Andrew, 2003] Andrew.T, "Systèmes d'exploitation", Pearson Education, 2003.

[Anderson, 1980] Anderson.J, "Computer security threat monitoring and surveillance",Contract 79F296400. February 26, 1980. Revised : April 15, 1980.

[Althubiti et al., 2018] Althubiti.S, Eric.M, Roy.K, "Lstm for anomaly-based network intrusion detection", i28th International Telecommunication Networks and Applications Conference, ITNAC, 2018. [Alazab et al, 2012] Alazab.A, Hobbs.M, Abawajy.J et Alazab.M, " Using Feature Selection for Intrusion Detection System", International Symposium on Communications and Information Technologies, 2012.

[Bensiah, 2020] Bensiah.O, "La proposition d'une nouvelle approche basée Deep Learning", Mémoire fin d'étude Master,Université L'arbi Ben M'hidi Oum El Bouaghi, p.13, 14, 2020.

[Balm 1997] Balm.G.J, "An introduction to optical character reader considerations", thèse de doctorat, Université Helsinki de Technologie. 1997.

[Boukhalfa et al, 2020] Boukhalfa.A, Abdellaoui.A, Hmina.N, Chaoui.H, "Méthode d'apprentissage profond LSTM pour système de détection d'intrusion réseau", International. J. Electr. Calcul, Ing, (IJECE) 10 (3) 3316–3322,2020.

[Basit et al., 2021]Basit.A, Zafar.M, Liu.X , Javed.A, Jalil.Z et Kifayat.K, "A comprehensive survey of AI-enabled phishing attacks detection techniques", Telecommunication Systems, 76(1), 139-154,2021.

[Bloch et al., 2007] Bloch.L, Wolfhugel.C, Queinnec.C, Schauer.H, Henry.F, Makarévitch.N, "Sécurité informatique", Edition EYROLLES, 2007.

[Bouras et Fethallah, 2017] Bouras.I et Fethallah.K, "Un Système de Détection D'Intrusion pour les Smart Grids", Doctoral dissertation, Université Larbi Tébessa, 2017.

[Bajaj et Arora, 2013] Bajaj.K et Arora.A, "Improving the Intrusion Detection using Discriminative Machine Learning Approach and Improve the Time Complexity by Data Mining Feature Selection Methods". International Journal of Computer Science, vol. 76, Aug, 2013.

[Bourouh et Kanoun, 2017] Bourouh.M, Kanoun.Z, « Detection d'intrusion à basé des réseaux de neurones et algorithmes génétique », mémoire fin d'étude, Université Abou Bakr Belkaid-Tlemcen, p.56, 2017.

[Berlin et al., 2016] Berlin.H, Djionang.L, Gilbert.T, "Vers une nouvelle architecture de detection d'intrusion réseaux neurones",article de hal open science 2016.

[Bourouh et Kanoun, 2018] Bourouh, M., et Kanoun, Z. Detection d'intrusions à base des réseaux de neurones et algorithmes génétiques. 2017. Thèse de doctorat. 14-01-2018.

[Cornuejols, 2002] Cornuejols.A, "Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à vaste marge", Université de Paris-Sud, Orsay, France, vol, 51, p. 14-23, Juin 2002.

[Caelen, 2017] Caelen .O , « A Bayesian interpretation of the confusion matrix. Annals of Mathematics and Artificial Intelligence », 81(3-4), vol. 81, no 3-4, p. 429-450, 2017.

[Corentin ,2019] Corentin.H,"Contribution au développement de l'apprentissage profond dans les systèmes distribués", Intelligence artificielle,Université Rennes 1, 2019. Français.

[Dumont, 2009] Dumont , "Cryptographie et sécurité informatique". Eyrolles, 2009.

[Denning, 1987] Denning.D, "An intrusion detection model", in IEEE Transactions on software engineering, SE-13 :222-232, 1987.

[De Filippi et Dulong, 2014] De Filippi.P et Dulong.M ," Le pirate informatique", un explorateur des courants juridiques du réseau ,Tracés, (26) 43–65, 2014.

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

[Eric, 2001a] Eric.C ," Hackers Beware", first edition, New Raiders Publishing,2001.

[Eric, 2001b] Eric.M ,"Sécurité des réseaux". Campuspress, edition 2001.

[Eluyode et Akomolafe, 2013] Eluyode.O.S et Akomolafe.D.T, "Comparative study of biological and artificial neural networks". European Journal of Applied Engineering and Scientific Research 2.1, 2013.

[Elekar et Waghmare, 2015] Elekar.K.S et Waghmare.M.M, "Comparison of Tree base Data Mining Algorithms for Network Intrusion Detection",International Journal of Engineering, Education and Technology, vol. 2, no 10, p. 3253-325, 2015,

[Emilion, 2021]Emilion.M,« Matrice de confusion », jedha.com, 2021 consulté de 1 jeun 2022.

[Faour et al., 2005] Faour.A., Leray.P et Foll.C, « Réseaux bayésiens pour le filtrage d'alarmes dans les systèmes de détection d'intrusions ». In EGC (Ateliers) pp. 69-72, 2005.

[ Gershenson, 2003] Gershenson.C, "Artificiel Neural Networks for Beginners" , Universidac Nacional Autónoma de México, September 2003.

[Ghernaouti, 2008] Ghernaouti ,"Sécurité informatique et réseaux". Sciences sup.Dunod, Paris, 2e éd edition, 2008.

[Grim et al., 2014] Grim.H, Ghezzaz.Y, and Abdelhak.N , conception et realisation d'un systeme de detection d'intrusion. thèse de doctorat, Université Mouloud Mammeri,p40, 2014.

[Ghahramani, 2004] Ghahramani.Z, "Unsupervised learning, in Advanced lectures on machine learning", Ed : Springer, pp. 72- 112, 2004.

[Géron, 2017] Géron.A, "Hands On Machine Learning with Scikit-learn and trensflow", USA : O'Reilly Media, 2017.

[Gupta et al., 2016] Gupta.S, Singhal.A,Kapoor.A,"A literature survey on social engineering attacks : Phishing attack. In 2016 international conference on computing", communication and automation (ICCCA) (pp. 537-540). IEEE, 2016.

[Hammi, 2018] Hammi.M . "Sécurisation de l'Internet des objets". thesis de doctor, Université Paris Saclay, 2018.

[Heady et al., 1990] Heady, R., Luger, G., Maccabe, A., and Servilla, M. (1990). "The architecture of a network level intrusion detection system". Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States) ; New Mexico.

[Hay et Cid, 2008] Hay.A et Cid.D, "OSSEC HIDS Host-Based intrusion Detection System", Syngress edition, 2008.

[Haddaoui, 2015] Haddaoui, "conception d'un systeme de detection d'intrusion base sur un arbre de decision", Mémoire master fin d'étude, p.43, 44, 2015.

[Jacon, 2006] Jacon.Z, "Vers une detection d'intrusion à fiabilité et pertinence prouvable", thèse de doctorat, Université de technologie, Australie, 2006.

[Jabou et al, 2009] Jabou.C, Schillings.M et Hantach.A, "detection d'anomalies sur le réseau", mémoire fin d'étude université de Paris Descartes, 2009.

[Jalam ,2003] Jalam.R, "Apprentissage automatique et catégorisation de textes multilingues", thèse de doctorat, Université Lumiere Lyon 2,2003.

[Jha et Ragha, 2013] Jha.J et Ragha.L, "Intrusion detection system using support vector machine", International Journal of Applied Information Systems (IJAIS) 3, 2013.

[Kartit,2016] Kartit.Z, " Contribution à la sécurité du Cloud Computing", thèse de doctorat science et ingénieur Faculté des Sciences, 4 Avenue Ibn Battouta B.P, p ;50,2016.

[Labraoui, 2020] Labraoui.N ,"sécurite informatique",article IEEE,Université Abou Bakr Belkaid, 2020.

[Labonne et al., 2018] Labonne.M, Olivereau.A. et Zeghlache.D, « Automatisation du processus d'entraînement d'un ensemble d'algorithmes de machine learning optimisés pour la détection d'intrusion », labonne2018automatisation, 2018.

[Maina et Moumen, 2021 ] Maina.I et Moumen.I, "Attaques Adverses de l'Apprentissage Profond", mémoire fin d'étude, université Larbi Tébessi – Tébessa, p.17, 2021

[Mouzer et kheddam ,2009] Mouzer et kheddam, Mise en place d'un système de détection et de prévention, thèse de fin d'étude Université Mouloud Mammeri, 2009.

[Mohamadally et Fomani, 2006] Mohamadally.H et Fomani.B, "SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges", Versailles St Quentin, France. janvier 2006.

[Mariette et Rahul, 2015] Mariette.A., et Rahul.K, "Support Vector Machines for Classification. In : Efficient Learning Machines", Apress, Berkeley, CA. p. 39-66, 2015.

[Mukherjeea et harmaa, 2012] Mukherjeea.S et harmaa.N,"Détection d'intrusion à l'aide du classificateur Naive Bayes avec réduction de fonctionnalités", Elsevier Ltd, 2012.

[Mukkamala et al., 2005] Mukkamala.S et all "Intrusion detection using an ensemble of intelligent paradigms", Journal Network and Computer Applications 28, 167-182, 2005.

[Mezaache, 2007]Mezaache.H, "Les réseaux de Neurones formels et les systèmes Neuro-Flous pour l'apprentissage par renforcement ",thèse de doctorat,Université Hadj Lakhdar, Batna-Algérie, 2007.

[Michel, 2003] Michel.C, "Langage de description d'attaques pour la detection d'intrusion" ,Thèse de doctorat , Université de Rennes, 16 septembre 2003.

[Moussi et Lounis, 2020] Moussi.D et Lounis.k, "La Classification d'images d'insectes ravageurs en utilisant le Deep Learning", Mémoire fin d'étude master, Université Mouloud Mammeri, p.17, 2020.

[Moualek ,2017] Moualek.Y, "Deep Learning pour la classification des images", Mémoire de master, Université de Tlemcen, Algérie, 2017.

[Morin, 2014] Morin.V ,"Etude comparative d'algorithmes de data mining dans le contexté du jeu vidéo, Diss, Université du Québec à Chicoutimi, 2014.

[Mushtaq et al., 2022] Mushtaq.E, Zameer.A , Umer.M. et Abbasi.AA, "Un système de détection d'intrusion en deux étapes avec auto-encodeur et LSTM, Informatique douce appliquée", p.121, 108768, 2022.

[Mokhtari et Moulkhaloua, 2018] Mokhtari.S.M, Moulkhaloua.A , « Système DE Détection D'intrusions Informatiques par Système Multi Agents »,Mémoire fin d'étude de université Ain Temouchent, 2018.

[Nobelis, 2004] Nobelis.N,"Un modèle de case-based Reasoning pour la détection d'intrusion", Rapport de stage DEA, Septembre 2004.

[Nevski ,2003] Nevski.A, "Modélisation et la simulation d'un système de détection et de réponse aux intrusions", Laboratoire Telecom du Centre Universitaire Informatique de Genève ,2003.

[Nazir, 2013] Nazir.A, "A Comparative Study of Different Artificial Neural Networks Based Intrusion Detection Systems", Journal international des publications scientifiques et de recherche, volume 3, numéro 7, 15 ISSN 2250-3153, juillet 2013.

[Nuageo ,2017] Nuageo, "Deep Learning : définition, concept et usages potentiels", from eurocloud : .eurocloud.fr/deep-learning-definition-concept-usagespotentiels, 2017.

[Nakache, 2007] Nakache, "Extraction automatique des diagnostics à partir des comptes rendus médicaux textuels", Laboratoire CEDRIC - équipe ISID. Paris, Conservatoire National des Arts et Métiers : 219,2007.

[Pillou et Maraï, 2020] Pillou.J.F, et Maraï.R ," Tout sur la sécurité informatique", Imprimerie chirat - 42540, Saint-Just-la-Pendue , 2020.

[Pillou et Bay, 2013] Pillou.J.F et Bay.J.P , "Tout sur la sécurité informatique". Comment CaMarche.net . Dunod, Paris, 3e éd edition,2013.

[Poinsot, 2019] Poinsot.L ,"Introduction à la sécurité informatique". Support de cours, Université Paris, 2019.

[Parizeau, 2004] Parizeau.M « Réseaux de neurones » GIF-21140 et GIF-64326, 124,2004.

[Pietraszek, 2004] Pietraszek, T. Using adaptive alert classification to reduce false positives in intrusion detection. In : International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, Heidelberg, 2004. p. 102-124.

[Rokach et Maimon, 2005] Rokach.L et Maimon.O, "Decision trees",Data mining and knowledge discovery handbook, Springer, Boston, MA, 2005.

[Righidel, 2006] Righidel .R, "Sécurité Informatique et Réseaux", livre Solange Ghernaouti-Hélie,DUNOD,89-97,2006.

[Raman et al., 2017] Raman.M, Somu.G, Kirthivasan.N et Sriram.K,"A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems".Neural Networks, vol. 92, p. 89-97,2017.

[Ralaivola, 2007] Ralaivola.L, "Algorithme des k-plus-proches-voisin", Projet informatique, 2007.

[Shon et al., 2005] Shon.T, Kim.Y, Lee.C et Moon.J, "A Machine Learning Framework for Network Anomaly", Detection using SVM and GA, Actes de l'IEEE 2005.

[Sharma et Yadev, 2021] Sharma.N.V et Yadav.N.S, "An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers", article, 13 Juin 2021.

[Salama et al., 2011] Salama.M, Eid.H, Ramadan.R, Darwish.A, et Hassanien.E , "Hybrid intelligent intrusion detection scheme", In Soft computing in industrial applications, p.293–303. Springer, 2011.

[Song et al., 2009] Song.Y, Keromytis.A.D, et Stolfo.S.J, "Spectrogram : A Mixtureof-Markov-Chains Model for Anomaly Detection in Web Traffic". In : Proceedings of the Network and Distributed System Security Sympo-sium, NDSS,p.121-135, 2009.

[Sutton et Barto, 2015] Sutton.R and Barto.A, "Reinforcement Learning :An Introduction",Second edition, in progress,2015.

[Stanislas, 2016] Stanislas.L, "Exploration des réseaux de neurones à base d'autoencodeur dans le cadre de la Modélisation des données textuelles", thèse de doctorat, université de Sherbrooke, 2016.

[Tavallaee et al., 2009] Tavallaee .M, Bagheri.E, Lu.W, et Ghorbani.A, ''A detailed analysis of the KDD CUP 99 data set '', in Computational Intelligence for Security and Defense Applications, 2009. CISDA. IEEE Symposium on, 2009.

[Tatarova et Giarmana, 2010] Tatarova.R and Giarmana.G , "Détection des attaques de Déni de Service dans les réseaux IP", memoire fin d'etude, 2010.

[Teghlil et Laouira, 2021] Teghlil.F et Laouira.A, "Scanner à flux de fuite magnétique pour l'inspection des tôles ferromagnétiques", mémoire fin d'étude, Université Jijel, 2021

[Vigna et al., 2000] Vigna.G, Eckmann .S et Kemmerer.R, "Attack Languages", in Proceedings of the IEEE Information Survivability Work-shop, USA, pp. 163-166, 2000.

[Vapnik ,1995] Vapnik.N, "The Nature of Statistical Learning Theory", Springer -Verlag New York, 1995.

[Valentin, 2015] Valentin.B, "Approches Deep Learning Appliquées aux signaux audio : parole et musique", Rapport de Projet de Fin d'Étude, Institut de Recherche en Informatique de Toulouse (IRIT), 2015.

[Vinayakumar et al., 2017] Vinayakumar.R, Soman.K.P et Poornachandran.P, "Applying convolutional neural network for network intrusion detection". International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 1222–1228. IEEE, 2017.

[Xin et Yun-jie, 2010] Xin.G et Yun-jie.L,"A new Intrusion Prevention Attack System Model based on Immune Principle", International Conference on e-Business and Information System Security (EBISS), in IEEE, p.1-4, 2010.

[Ye, 2000] Ye,N. "A Markov Chain Model of Temporal Behavior for AnomalyDetection". In : Proceedings of the IEEE Workshop on InformationAssurance and Security, p.169, 2000.

# Webographie


[Advancia, 2021] Advancia IT. Pourquoi l'antivirus ne sont pas suffisant , advancia-itsystem.com , 2021, Consulté le.12 Avril 2022.

[Advisio, 2019] Advisio . Antivirus : A quoi servent-ils ?, Advisio-Com, 2019,consulté le 5 jan 2022.

[Belcic, 2020] Belcic, "proxy" , avast.com ,2020, consulté le 17 mai 2022.

[Cisco, 2020a]Cisco, "What is cybersecurity ?",cisco.com,2020, consulté le 2 Mars 2022.

[Cisco, 2020b] Cisco, "What Are the Most Common Cyber Attacks ?". Cisco.com,2020, consulté le 8 Mars 2022.

[cisco, 2020c] Cisco , "What is a VPN ? ",cisco.com, 2020,consulté le 9 Mai 2022.

[Clayton ,2019] Clayton.R, "Qu'est-ce que le machine learning ?", oracle.com,2019,consulté le 24 Mai 2022.

[Gérard, 2017] Gérard.F ,"cours de sécurité Firewalls", bestcours.com, 2017,consulté le 18 mars 2021.

[Hacker Combat, 2019] Hacker Combat. Virtual Private Network (VPN) : What VPN Is And How It Works. hackercombat.com, 2019, consulté le 5 mai 2022.

[Issarane ,2019] Issarane.H, "Apprentissage Non Supervisé", DataScientist,le-datascientist.fr,9 Mars 2019, consulté le 2 mai 2022.

[Arnaud, 2002] Arnaud.J ,"Types d'attaques réseaux Entreprise française de cyber sécurité", securiteinfo.com, 13 Janvier 2002.

[Justyyuk, 2016] Justyyuk, "A Short Introduction to K-Nearest Neighbors Algorithm", helloacm.com , 2016, consulté le 2 mai 2022.

[Richard et Perrson, 2009] Richard.N et Perrson.J, "Apprentissage par renforcement", slideshare/apprentissage par renforcement.com, 23 Novembre 2009, consulté le 28 Mai 2022.

[Oinkina et Hakyll, 2015 ]Oinkina et Hakyll,"understanding lstm networks", colah.github.io, 27 août 2015, consulte le 7 arvil 2022.

[Rfia, 2012] Rfia, "cours les réseaux de neurones RNA , rfia2012.files.wordpress.com, 2012, consulté le 24 Mai 2022.

[Ritambhara, 2020] Ritambhara, " Man-in-the-Middle Attack on Data Being Transfered over Network". Ritambhara Technologies, 8 Février 2020

[Valgasu, 2017] Valgasu,"IP Spoofing. Sécurité Informatique : Audits Et Conseils En Cybers sécurité", securiteinfo.com, 20 Mai 2017, consulté le 8 Février 2022.

[Yende ,2018] R.Yende, SUPPORT DE COURS DE SÉCURITÉ INFORMATIQUE ET CRYPTO, hal.archives-ouvertes.fr, 2018, consulté le 17 Mai 2022.

# Résumé

De nos jours, les systèmes informatiques et les réseaux sont exposés à des attaques informatiques, pour cela, les systèmes de détection d'intrusion SDI jouent le rôle important dans la cybersécurité.
Le principal objectif de cette étude est de réaliser un SDI afin de sécuriser un Réseau informatique (SDIR). Nous nous sommes situés dans une approche basée signature. Pour ce faire, nous avons fait recours à l'apprentissage automatique en adoptant la méthode Naïve Bayésienne (NB). Les résultats obtenus avec une classification en deux Classes (NB2C), normale et anormale sont meilleurs que ceux de la classification en cinq Classes (NB5C), normale et 4 catégories d'attaques. Or, nous avons opté pour le premier algorithme et nous l'avons combiné avec une méthode issue de l'apprentissage profond qui est le réseau de neurone récurrent à Mémoire à Court Long Terme (MCLT). Afin d'entraîner et de tester notre SDIR, nous avons utilisé la base de données NSL-KDD. Nous avons effectué une comparaison entre les algorithmes appliqués NB5C, NB2C-MCLT ET le MCLT, en calculant la précision, la Mesure-F et le rappel. Finalement, les résultats obtenus via le LSTM seul sont meilleurs que ceux des autres algorithmes, prouvant encore une fois l'efficacité de l'apprentissage profond.

**Mots clés :** Système de Détection d'Intrusions Réseaux(SDIR), Apprentissage Automatique (AA), Apprentissage Profond(AP), méthode Naïve Bayésienne (NB), réseau de neurone récurrent à Mémoire à Court Long Terme (MCLT).

# Abstract

These days, computer systems and networks are exposed to computer attacks, for this, intrusion detection systems IDS play the important role in cyber security. The main objective of this study is to realize an IDS in order to secure a Computer Network (SDIR). We are located in a signature-based approach. To do this, we used machine learning by adopting the Naïve Bayesian (NB) method. The results obtained with a classification in two Classes (NB2C), normal and abnormal are better than those of the classification in five Classes (NB5C), normal and 4 attack categories. However, we opted for the first algorithm and we combined it with a method from deep learning which is the recurrent neural network with Short Term Memory (STM). In order to train and test our SDIR, we used the NSL-KDD database. We performed a comparison between the applied algorithms NB5C, NB2C-MCLT AND the MCLT, calculating precision, F-Measure and recall. Finally, the results obtained via MCLT alone are better than those of the other algorithms, proving once again the effectiveness of deep learning.

**Keywords :** Network Intrusion Detection System (NIDS), Machine Learning (ML), Deep Learning (DL), Naive Bayesian (NB), Short Term Memory Recurrent Neural Network (STMN).